Clustering Product Aspects Using Two Effective Aspect Relations for Opinion Mining

Yanyan Zhao[1], Bing Qin[2], and Ting Liu[2]

[1] Department of Media Technology and Art, Harbin Institute of Technology
[2] Department of Computer Science and Technology, Harbin Institute of Technology
{yyzhao,bqin,tliu}@ir.hit.edu.cn

Abstract. Aspect recognition and clustering is important for many sentiment analysis tasks. To date, many algorithms for recognizing product aspects have been explored, however, limited work have been done for clustering the product aspects. In this paper, we focus on the problem of product aspect clustering. Two effective aspect relations: relevant aspect relation and irrelevant aspect relation are proposed to describe the relationships between two aspects. According to these two relations, we can explore many relevant and irrelevant aspects into two different sets as background knowledge to describe each product aspect. Then, a hierarchical clustering algorithm is designed to cluster these aspects into different groups, in which aspect similarity computation is conducted with the relevant aspect set and irrelevant aspect set of each product aspect. Experimental results on camera domain demonstrate that the proposed method performs better than the baseline without using the two aspect relations, and meanwhile proves that the two aspect relations are effective.

Keywords: Sentiment analysis, Product aspect clustering, Social media

1 Introduction

Social media holds a considerable amount of user-generated content describing the opinions of customers on products and services through reviews, blog, tweets, etc. These reviews are valuable for customers to make purchasing decisions and for companies to guide the business activities. Consequently, the advent of social media has stirred much excitement and provided abundant opportunities for opinion mining and sentiment analysis [12, 6, 23].

For many opinion mining applications, for example, opinion summarization [19, 18] or recommender systems [7, 14], recognizing product aspects from the product reviews is usually treated as the first step. Afterwards, we collect relevant sentences and analyze opinions for each product aspect. Thus product aspect recognition is a critical task for opinion mining [13, 5, 9, 10]. There are two sub-tasks for this task, one is aspect extraction, and the other one is aspect clustering. Aspect extraction aims to extract the entities, which the users write comments on. For example, it can extract the "图像" ("picture" in English) as

product aspect from the review "图像很漂亮" ("the picture is great"). On the other hand, aspect clustering aims to cluster the aspects that have the similar meaning into the same groups. For example, the word "图像" ("picture" in English) and "照片" ("photo" in English) express the same meaning, we need to group them.

To date, most of the aspect relevant research work is focusing on the first sub-task including many kinds of methods, such as rule-based [1, 3, 22, 13], supervised [20, 4, 8], and topic model-based [10, 2, 15] methods. Unfortunately, just a few work is done on the second sub-task. Due to the importance of the second sub-task, we need to pay more attention to it and this paper is mainly focusing on this sub-task. In the previous work, some researchers used topic model based methods [17, 10, 16] to cluster domain-specific aspects. However, topic models always jointly modeled topics and sentiment words. Mukherjee et al. [11] used a switch variable trained with Maximum-Entropy to separate topic and sentiment words. Some researchers consider this task as a traditional clustering task, the key part of which is similarity computation. Zhai et al. [21] modeled this task as a semi-supervised learning problem using lexical similarity. However, it needed some manually selected seeds as the input, which are random and accordingly hard to handle or reproduce in the experiments.

In this paper, we propose a simple and effective unsupervised method, which applies two effective aspect relations. Specifically, we treat this task as a typical clustering problem, which mainly emphasizes on the similarity computation. However, the common similarity measures are usually based on literal meaning of two aspects, which is far from enough. To address this issue, we find two interesting phenomena. One is summarized as **relevant aspect phenomenon**. That is to say, *in one sentence, if one aspect contains the other one, the two aspects are relevant and can be grouped into a same cluster*. For example, in the sentence "我今天买了个[人像镜头]，这个[镜头]的分辨率不错啊" ("I bought a [portrait lens], the resolution of the [lens] is perfect"), the phrase "人像镜头" ("portrait lens" in English) and "镜头" ("lens" in English) shows relevant aspect phenomenon. Thus the two aspects can be classified into a same group. The relationship between them is called **relevant aspect relation**.

The other phenomenon is **irrelevant aspect phenomenon**. That is to say, *in one sentence the product aspect is always used in one form instead of using different forms, even though this aspect can be expressed in other forms*. Based on this phenomenon, the aspects appearing in the same sentence can be considered as different aspects if they do not contain each other. Take the following two sentences as an example:

- Sentence 1: 我在电脑上浏览佳能600D的[照片]，感觉[照片]挺不错的，<分辨率>挺高。
  (I browsed the [*pictures*] in the computer, and found the [*pictures*] were perfect and the $< resolution >$ was high.)
- Sentence 2: 我在电脑上浏览佳能600D的[照片]，感觉[图像]挺不错的，<分辨率>挺高。

(I browsed the [*pictures*] in the computer, and found the [*photos*] were perfect and the $<$ *resolution* $>$ was high.)

In most cases, if one word must be mentioned multiple times in a sentence, people always use the same word. Thus, the word "照片" ("picture" in English) used in Sentence 1 shows the phenomenon. However, we seldom use different word form to express a same meaning in a same sentence, such as the word "照片" ("picture" in English) and "图像" ("photo" in English) used in Sentence 2. Therefore, in Sentence 1, since the aspect "照片" and aspect "分辨率" do not contain each other, they can be considered as different aspects. That is to say, they belong to different groups. Thus, the relationship between them is called **irrelevant aspect relation**.

According to these two phenomena, for each product aspect, we can collect a relevant aspect set and an irrelevant aspect set from a large corpora respectively. That is to say, we provide two kinds of background knowledge for each aspect. On one hand, the relevant aspect set is to help the given aspect to get the domain synonyms more accurately. On the other hand, the irrelevant aspect set is to help separate this aspect from other irrelevant aspects that do not refer to the same aspect.

Since aspect clustering is a typical clustering problem, a hierarchical clustering method is applied to classify the aspects into different groups based on their relevant aspect sets and irrelevant aspect sets. Several similarity computation methods are used to compute the similarity between two aspects.

We evaluate our framework on the corpus of camera domain as a case study. Experimental results show that the both two kinds of aspect relations achieve significant performance that gains over the baseline clustering method without using these two relations.

The remainder of this paper is organized as follows. Section 2 introduces the two kinds of aspect relations, and constructs the relevant and irrelevant aspect set for each product aspect. Section 3 shows the hierarchical clustering algorithm based on the two aspect relations. Section 4 presents the experiments and results. Finally we conclude this paper in Section 5.

2 Two effective aspect relations

For each aspect, two aspect sets, irrelevant aspect set and relevant aspect set can be built. Figure 1 shows an example consisting of a Chinese review, which is tagged with all the appearing product aspects.

As shown in Figure 1, four kinds of product aspects can be found. Take the aspect "镜头" ("lens" in English) as an example, since "镜头" is the suffix of "光变镜头", "光变镜头" is the relevant aspect of the given aspect "镜头" and can be concluded into the relevant set. On the other hand, "光圈" and "成像效果" are totally different from "镜头" literally, thus they are concluded into the irrelevant set.

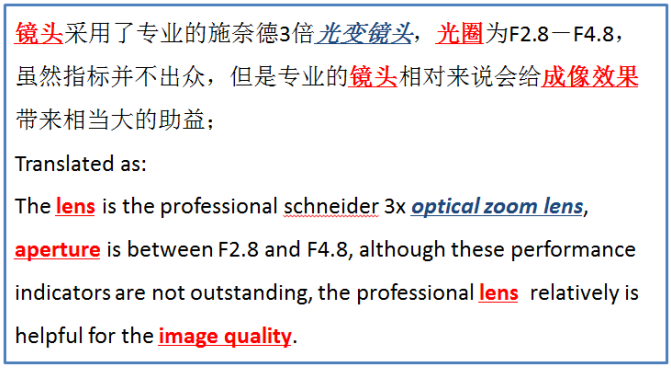

Fig. 1. An example consisting of a Chinese review, which is tagged with all the appearing product aspects with different colors.

Formally, we describe each aspect a as a tuple $< set\_R, set\_IR >$ as follows, where set_R is a set that stores items relevant to a and set_IR is a set that stores items irrelevant to a.

$$a : set\_R[r_1, r_2, ..., r_i, ..., r_n] set\_IR[ir_1, ir_2, ..., ir_j, ..., ir_m]$$

Here, for aspect a, there are two important evidence sets, that is, n relevant aspects and m irrelevant aspects to generate the final aspect clustering.

- **relevant aspect** r_i: aspect a and r_i is relevant, if a and r_i are appearing in the same sentence and contain inclusion relations, e.g., a is the suffix of r_i, and vice versa.
- **irrelevant aspect** ir_j: aspect a and ir_j is irrelevant, if a and ir_j are appearing in the same sentence and do not contain inclusion relation with each other.

As a result, "镜头" in Figure 1 can be expressed as:

镜头: set_R[光变镜头]set_IR[光圈, 成像效果]
(lens: set_R[optical zoom lens]set_IR[aperture, image quality] in English).

Here, "光变镜头" is its relevant aspect, "光圈" and "成像效果" are its irrelevant aspects. Since "镜头" can appear in lots of review sentences, we can accordingly acquire lots of relevant aspect sets and irrelevant aspect sets from a domain-specific corpus. Then a final set_R andset_IR with more aspect elements can be further built.

For example, for the aspect "镜头", 71 relevant aspects and 149 irrelevant aspects can be found from 138 reviews. Based on these background knowledge, we can design new hierarchical clustering algorithms to classify the domain aspects into different groups.

3 Hierarchical clustering based on two aspect relations

Since aspect clustering is a typical clustering problem, we can use many kinds of clustering algorithms. In this paper, we take hierarchical clustering algorithm as a case of study. During the process, similarity computation between aspects is the main part. Traditional similarity measures are using the thesaurus dictionaries or just computing the similarity between two aspect literally. However, they are far from sufficient due to a few reasons. First, many aspects are domain words or phrases, which are not included in the traditional thesaurus dictionaries. For example, the aspect "光变镜头" ("optical zoom lens" in English) is not appearing in any dictionaries. Secondly, many aspects are not synonyms in a dictionary, but indicating the same aspect under the given domain.

To alleviate these problems, Section 2 introduces the background knowledge for each aspect, including the relevant aspect sets and the irrelevant aspect sets. That is to say, we can use more knowledge to compute the similarity between two aspects besides their similarity literally.

Accordingly, the similarity computation between two aspects a_i and a_j composes three parts.

- Literal Similarity (LS): Similarity between a_i and a_j literally, which is recorded as $s_1(a_i, a_j)$. In this part, two factors are considered. One is to explore whether these two aspects are synonyms according to a dictionary. The other one is to compute the similarity between a_i and a_j literally. That is to say, we treat each character as an element, and then an aspect can be considered as a vector with characters. Many similarity methods can be used. In this paper we just try the Cosine similarity measurement. Based on these, we conclude this kind of similarity as follows:

$$s_1(a_i, a_j) = \begin{cases} 1 & \text{if } a_i \text{ and } a_j \text{ are synonyms,} \\ \cos(a_i, a_j) & \text{if } a_i \text{ and } a_j \text{ are not synonyms.} \end{cases} \tag{1}$$

- Relevant Set Similarity (RSS): Similarity between the relevant aspect sets of a_i and a_j, which is recorded as $s_2(a_i, a_j)$. This idea is based on such a hypothesis: the relevant aspect sets of two similar aspects that show the background knowledge are similar. Since the relevant background knowledge for each aspect can be considered as a vector. Then this kind of similarity can be converted to compute the similarity between two vectors. The computation procedure is shown as follows.

$$s_2(a_i, a_j) = sim(rel\_vector_i, rel\_vector_j) = \frac{rel\_vector_i \cdot rel\_vector_j}{\parallel rel\_vector_i \parallel \parallel rel\_vector_j \parallel} \tag{2}$$

- IRrelevant Set Similarity (IRSS): Similarity between the irrelevant aspect sets of a_i and a_j, which is recorded as $s_3(a_i, a_j)$. This similarity is computed based on such a hypothesis: if a_i is similar to a_j, it cannot appear in the

irrelevant aspect set of a_j. We describe the similarity between a_i and a_j as follows:

$$s_3(a_i, a_j) = \begin{cases} 1 & \text{if } a_i \text{ appears in the irrelevant aspect set of } a_j, \\ 1 & \text{if } a_j \text{ appears in the irrelevant aspect set of } a_i, \\ 0 & \text{else.} \end{cases} \quad (3)$$

More formally, the final similarity between aspects a_i and a_j can be concluded as follows:

$$S_a(a_i, a_j) = \alpha * s_1(a_i, a_j) + \beta * s_2(a_i, a_j) - \gamma * s_3(a_i, a_j), \quad (4)$$

where similarity s_2 reflects the relevant aspect phenomenon and s_3 reflects the irrelevant aspect phenomenon respectively.

Based on this similarity, the hierarchical clustering algorithm is described in Figure 2 in detail.

Hierarchical Clustering Algorithm based on New Similarities

Input: Set of aspects *A*, $A = \{a_1, a_2, ..., a_n\}$,
each aspect is described with *R* and *IR*

Output: Aspect clusters *AC*

1. Suppose each aspect as a cluster, noted as $c_1,...,c_i,..., c_n$
2. Compute the similarity between each pair of clusters
   If the similarity between c_i and c_j is maximum, and greater than Θ:
   Merge c_i and c_j into a new cluster
3. Repeat 2 until the amount of the clusters does not change
4. The final clusters are *AC*

Fig. 2. Hierarchical Clustering Algorithm based on New Similarities.

Here, in Step 2, the similarity between two clusters $c_i = \{a_1^i, ..., a_p^i, ..., a_n^i\}$ and $c_j = \{a_1^j, ..., a_q^j, ..., a_m^j\}$ is computed as follows.

$$S_c(c_i, c_j) = \frac{\sum_{p=1}^{n} \sum_{q=1}^{m} S_a(a_p^i, a_q^j)}{n \times m} \quad (5)$$

4 Experiments

4.1 Experimental Setup

Corpus We conducted the experiments on a Chinese corpus of digital camera domain, which came from the corpora of the Chinese Opinion Analysis Evaluation (COAE). Table 1 describes the corpus in detail.

From 138 reviews, 4,039 aspects are manually found and annotated before reduplication removing, and 1,189 aspects are left after reduplication removing. Besides, each aspect averagely appears about 3.4 times. Therefore, for each aspect, we can collect its two effective aspect relation sets from many sources, because this aspect may appear multiple times in the corpus.

Statistics	
# reviews	138
# aspects (before reduplication removing)	4,039
# aspects (after reduplication removing)	1,189
# single aspects	867
# multiple aspects	322
average # per aspect	4,039 ÷ 1,189 ≈ 3.4

Table 1. Corpus statistics of Digital camera domain.

Evaluation We follow the evaluation metrics of Zhai et al. [21] to evaluate the clusters in this study. The evaluation metrics include two parts: *Entropy* and *Purity*. Given a data set DS, its gold partition is $G = g_1, ..., g_j, ..., g_k$, where k is the given number of clusters. Suppose our method can group DS into k disjoint subsets, that is, $DS = DS_1, ..., DS_i, ..., DS_k$, *Entropy* and *Purity* can be defined as follows.

Entropy: For each resulting cluster DS_i, we can measure its entropy using Equation (6), where $P_i(g_j)$ is the proportion of g_j data points in DS_i. The total entropy of the whole clustering (which considers all clusters) is calculated by Equation (7).

$$entropy(DS_i) = -\sum_{j=1}^{k} P_i(g_j) \log_2 P_i(g_j) \tag{6}$$

$$entropy_{total} = \sum_{i=1}^{k} \frac{|DS_i|}{|DS|} entropy(DS_i) \tag{7}$$

Purity: Purity measures the extent that a cluster contains only data from one gold-partition. The cluster purity is computed with Equation (8). The total purity of the whole clustering (all clusters) is computed with Equation (9).

$$purity(DS_i) = \max_j P_i(g_j) \tag{8}$$

$$purity_{total} = \sum_{i=1}^{k} \frac{|DS_i|}{|DS|} purity(DS_i) \tag{9}$$

Comparative systems Similarity computation between aspects is the main part during the clustering procedure. According to the three similarity computation measures between aspects, we designed four comparative systems to show the performance of each similarity measure when clustering.

- Literal Similarity (LS): We compute the similarity between two aspects a_i and a_j literally.
- Relevant Set Similarity (RSS) + LS: We compute the similarity between two aspects a_i and a_j using their relevant aspect sets, on the foundation of the literal similarity.
- IRrelevant Set Similarity (IRSS) + LS: We compute the similarity between two aspects a_i and a_j using their irrelevant aspect sets, on the foundation of the literal similarity.
- RSS + IRSS +LS: We combine the three kinds of similarities between two aspects a_i and a_j.

4.2 Results

Method	Entropy	Purity
LS (Baseline)	1.53	0.94
LS + RSS	1.39	0.95
LS + IRSS	1.40	0.95
LS + RSS + IRSS	**1.37**	**0.96**

Table 2. Comparative results on product aspect clustering.

Table 2 shows the experimental results of the four comparative systems on product aspect clustering task. Here, **LS** (Literal Similarity) is the baseline system, which is computed without any background knowledge. All the other three systems are based on the baseline system, and computed with different kinds of background knowledge.

Since we can acquire a relevant aspect set and an irrelevant aspect set to describe each aspect, two kinds of background knowledge can be expanded accordingly.

Compared with the baseline system **LS**, the system **LS+RSS** that adds the relevant aspect set as the new background knowledge can yield better results, with the *Entropy* of 1.39 and the *Purity* of 0.95. This can illustrate that the relevant aspect set is effective in aspect clustering. Specifically, for an aspect a_i, besides the knowledge of a_i's literal meaning, its relevant aspect set expanded from multiple sentence contexts is another good dimension to measure the similarity between two aspects.

Moreover, the system **LS+IRSS** that adds the irrelevant aspect set as the new background knowledge can also yield better results, with the *Entropy* of 1.40

and the *Purity* of 0.95, compared with **LS**. This proves that the irrelevant aspect set can also be treated as another important evidence for aspect clustering. Obviously, if the aspect a_i appears in the irrelevant aspect set of the aspect a_j, a_i and a_j cannot be grouped together. This background knowledge can naturally avoid a part of the situation that a_i and a_j are literally similar, but in fact they do not belong to a same group.

Based on the above, the dimension of relevant aspect similarity (**RSS**) can be considered as a supplement of the literal similarity (**LS**), and the dimension of irrelevant aspect similarity (**IRSS**) can be considered as a filter to reduce some wrong cases. Therefore, the two aspect relations **RSS** and **IRSS** are complementary to each other, we combine them into a new system **LS+RSS+IRSS** based on the baseline **LS**. Table 2 shows that **LS+RSS+IRSS** performs best among all the comparative systems, with the *Entropy* of 1.37 and the *Purity* of 0.96.

5 Conclusion and Future Work

Aspect extraction and aspect clustering are both critical for the applications of sentiment analysis and opinion mining. However, the research on the aspect clustering task is far from enough. In this paper, we propose an easy and effective unsupervised method based on two effective aspect relations, namely, relevant aspect relation and irrelevant aspect relation. These two kinds of relations can expand the background knowledge of each aspect, and improve the performance of the similarity computation between two aspects. Experimental results on camera domain show that our method achieves better performance than the baseline without using the aspect relations, which proves that the two proposed relations are useful. As the future work, in order to capture more background knowledge for each aspect, we will expand them from the Web.

Acknowledgments

We thank the anonymous reviewers for their helpful comments. This work was supported by National Natural Science Foundation of China (NSFC) via grant 61300113, 61133012 and 61273321, and the Ministry of Education Research of Social Sciences Youth funded projects via grant 12YJCZH304.

References

1. Bloom, K., Garg, N., Argamon, S.: Extracting appraisal expressions. In: HLT-NAACL 2007. pp. 308–315 (2007)
2. Branavan, S., Chen, H., Eisenstein, J., Barzilay, R.: Learning document-level semantic properties from free-text annotations. In: Proceedings of ACL-08: HLT. pp. 263–271 (2008)
3. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of KDD-2004. pp. 168–177 (2004)

4. Jakob, N., Gurevych, I.: Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. pp. 1035–1045. EMNLP '10, Association for Computational Linguistics, Stroudsburg, PA, USA (2010)
5. Li, S., Wang, R., Zhou, G.: Opinion target extraction using a shallow semantic parsing framework. In: AAAI. pp. 1671–1677 (2012)
6. Liu, B.: Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers (2012)
7. Liu, B., Hu, M., Cheng, J.: Opinion observer: analyzing and comparing opinions on the web. In: Proceedings of WWW-2005. pp. 342–351 (2005)
8. Liu, K., Xu, L., Zhao, J.: Opinion target extraction using word-based translation model. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 1346–1356. EMNLP-CoNLL '12, Association for Computational Linguistics, Stroudsburg, PA, USA (2012)
9. Liu, K., Xu, L., Zhao, J.: Syntactic patterns versus word alignment: Extracting opinion targets from online reviews. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1754–1763. Association for Computational Linguistics, Sofia, Bulgaria (August 2013)
10. Mukherjee, A., Liu, B.: Aspect extraction through semi-supervised modeling. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 339–348. Association for Computational Linguistics, Jeju Island, Korea (July 2012)
11. Mukherjee, A., Liu, B.: Modeling review comments. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 320–329. Association for Computational Linguistics, Jeju Island, Korea (July 2012), http://www.aclweb.org/anthology/P12-1034
12. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retr. 2(1-2), 1–135 (Jan 2008)
13. Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. Computational Linguistics 37(1), 9–27 (2011)
14. Reschke, K., Vogel, A., Jurafsky, D.: Generating recommendation dialogs by extracting information from user reviews. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 499–504. Association for Computational Linguistics, Sofia, Bulgaria (August 2013), http://www.aclweb.org/anthology/P13-2089
15. Sauper, C., Haghighi, A., Barzilay, R.: Content models with attitude. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1. pp. 350–358. HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011)
16. Sauper, C., Haghighi, A., Barzilay, R.: Content models with attitude. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 350–358. Association for Computational Linguistics, Portland, Oregon, USA (June 2011), http://www.aclweb.org/anthology/P11-1036
17. Titov, I., McDonald, R.: A joint model of text and aspect ratings for sentiment summarization. In: Proceedings of ACL-08: HLT. pp. 308–316. Association for Computational Linguistics, Columbus, Ohio (June 2008)
18. Wei, W., Gulla, J.A.: Sentiment learning on product reviews via sentiment ontology tree. In: ACL. pp. 404–413 (2010)

19. Woodsend, K., Lapata, M.: Multiple aspect summarization using integer linear programming. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 233–243. Association for Computational Linguistics, Jeju Island, Korea (July 2012), http://www.aclweb.org/anthology/D12-1022
20. Yu, J., Zha, Z.J., Wang, M., Chua, T.S.: Aspect ranking: Identifying important product aspects from online consumer reviews. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1. pp. 1496–1505. HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011)
21. Zhai, Z., Liu, B., Xu, H., Jia, P.: Clustering product features for opinion mining. In: Proceedings of WSDM. pp. 347–354 (2011)
22. Zhao, Y., Qin, B., Hu, S., Liu, T.: Generalizing syntactic structures for product attribute candidate extraction. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 377–380. Association for Computational Linguistics, Los Angeles, California (June 2010)
23. Zhao, Y., Qin, B., Liu, T.: Sentiment analysis. Journal of Software 21(8), 1834–1848 (2010)