Clustering Chinese Product Features with Multilevel Similarity

Yu He, Jiaying Song, Yuzhuang Nan, Guohong Fu

School of Computer Science and Technology, Heilongjiang University
Harbin 150080, China
heyucs@yahoo.com,jy_song@outlook.com, yuzhuangnan@gmail.com,
ghfu@hotmail.com

Abstract. This paper presents an unsupervised hierarchical clustering approach for grouping co-referred features in Chinese product reviews. To handle different levels of connections between co-referred product features, we consider three similarity measures, namely the literal similarity, the word embedding-based semantic similarity and the explanatory evaluation based contextual similarity. We apply our approach to two corpora of product reviews in car and mobilephone domains. We demonstrate that combining multilevel similarity is of great value to feature normalization.

Keywords: Opinion mining, product reviews, aspect normalization, clustering.

1 Introduction

Feature normalization, also referred to as feature grouping or feature co-reference resolution, aims to recognize co-referred feature expressions in product reviews and normalize them with a standard name. Obviously, feature normalization benefits many opinion mining applications, such as opinion summarization and aggregation (Liu, 2010; Zhai et al., 2010; Zhai et al., 2011).

While much work has been done to date on feature normalization, studies are still needed to explore more informative clues for feature normalization. Firstly, most current study has focused on exploiting different semantic similarity measurements such as the WordNet similarity(Carenini et al., 2005) or the topic model(Guo et al.,

2009) for feature grouping, few studies use word embeddings from big data. Unlike traditional semantic representations, word embeddings employ low-dimensional and real valued vectors to preserve the semantic relationship between words(Mikolov et al., 2013c). As such, we believe that word embeddings would provide a more convenient and effective way for exploring potential semantic clues for product feature normalization. More importantly, word embeddings can be learned from large corpora in an unsupervised manner. Secondly, the evaluation expressions collocated with product features has proven to be of great value to feature grouping(Yang et al., 2012). However, previous studies usually ignore evaluation information or do not distinguish explanatory evaluations from non-explanatory evaluations(Yang et al., 2012). Actually, non-explanatory evaluations like 好 'good' provide less-informative indicators than explanatory evaluations for normalizing the features they are modifying(Kim et al., 2013).

In this work we present an unsupervised clustering method for normalizing features in online Chinese product reviews. To approach this, we explore three levels of similarities, namely the literal similarity, the semantic similarity based on word embeddings and the contextual similarity based on explanatory evaluations, to handle different connections between co-referred product features in product reviews. These similarity measures are further combined with a linear interpolation strategy under a framework of hierarchical clustering to determine whether a given set of feature expressions should be clustered into a suitable feature group. We apply our approach to two corpora of product reviews in car and mobilephone domains. We demonstrate that combining multilevel similarity is of great value to feature clustering.

2 Feature Expressions in Chinese

To investigate how product features are expressed in Chinese product reviews, we built two corpora of product reviews in car and mobilephone domains, respectively. The two corpora are manually annotated with multiple linguistic information, including word segmentation, part-of-speech tags and opinion elements (viz. opinion objects, product features, evaluations and sentiment polarity). In addition, all co-referred feature expressions within the corpora are manually recognized and paired with their corresponding explanatory evaluations.

As can be seen from Table 1, a product feature can be expressed in many ways. For example, there are more than 20 different expressions on average for a feature in car

reviews. This shows that co-referred feature expressions are very common in product reviews, illustrating in a sense the importance of feature normalization to review mining.

Table 1. Distributions of features expressions

	Car[1]	**Mobilephone**[2]
#Reviews	2340	5670
#Feature expressions	408	169
# feature groups	20	15
# co-referred expressions per feature	20.40	11.27

3 Our Method

Let S={<f_1,e_1>,<f_2,e_2>,…, <f_n,e_n>} be a set of collocated feature-evaluation pairs from product reviews, our goal is to discover all co-referred feature expressions within *S* and further cluster them into a suitable group.

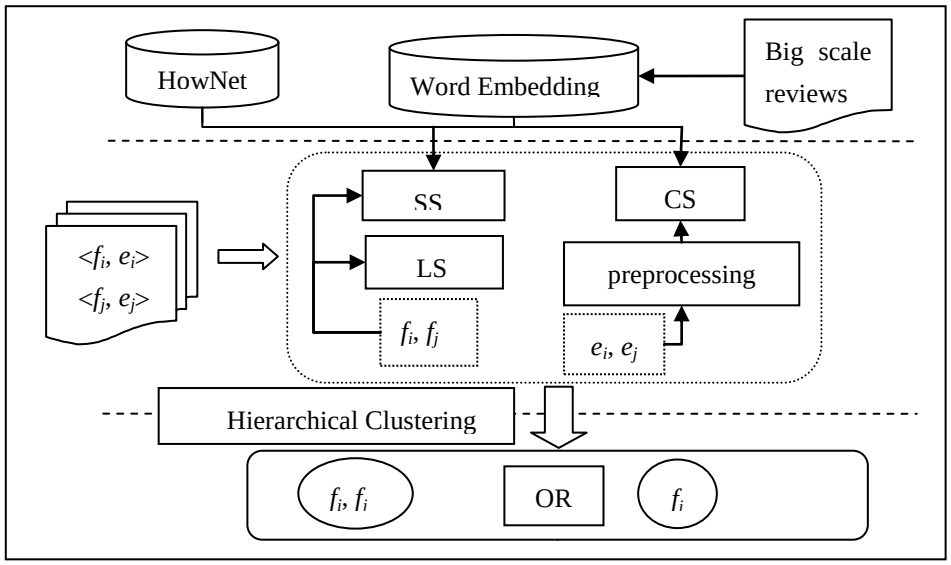

Fig. 1. Overview of our method

[1] http://www.autohome.com.cn/

[2] http://www.jd.com/

As shown in Figure 1, in order to handle the above-mentioned three levels of similarities, viz. literal similarity (*LS*), semantic similarity (*SS*) and contextual similarity (*CS*), between co-referred feature expressions, we explore multilevel similarity criteria and further combine them in a hierarchical clustering algorithm for feature grouping.

3.1 Literal Similarity

Co-referred feature expressions in Chinese usually contain identical characters or words. Take two pairs of feature expressions (外表, 外形) and (油耗, 耗油) as examples. 外表 and 外形 contain the same Chinese character 外 'outer' and thus have the same meaning 'appearance', while 油耗 and 耗油 are formed with identical Chinese characters in different orders, and therefore share the same meaning 'fuel consumption'.

Considering that edit distance cannot objectively reflect the real similarity for some co-referred feature expressions like油耗 and 耗油, for the reason that their edit distance similarity is only 0.3 and it is inconsistent with the actual similarity. We exploit Jaccard coefficient to calculate the literal similarity of two feature expressions f_1 and f_2. As shown in Equation (1), Jaccard coefficient measures the similarity of feature expression pairs by counting the number of common characters within them, ignoring the influence of character location.

$$LS_{JC}(f_i,f_j)=\frac{|\mathrm{SET}(f_i)\cap\mathrm{SET}(f_j)|}{|\mathrm{SET}(f_i)\cup\mathrm{SET}(f_j)|} \tag{1}$$

Where, SET(*fi*) denotes the set of characters within *fi* (i=1,2).

3.2 Semantic Similarity

Literal similarity relies on literal matching and works only for feature expressions with explicit literal connections, which does not always exist in co-referred feature expressions like 像素 'pixel' and 分辨率 'resolution'. Thus, we introduce semantic similarity based word embeddings.

Given two feature expressions f_1 and f_2, let $Vec(f_1)$ and $Vec(f_2)$ be their respective word embeddings, then their similarity based on word embeddings, namely $SS_{WE}(f_1, f_2)$, can be defined by Equation (2).

$$SS_{WE}(f_i, f_j) = \frac{Vec(f_i) \bullet Vec(f_j)}{|Vec(f_i)| \times |Vec(f_j)|} \tag{2}$$

3.3 Contextual Similarity

In contrast to non-explanatory evaluations, explanatory evaluations are feature-specific indicators for determining whether a set of feature expressions belongs to a feature group. The explanatory evaluation we call here is the explanatory parts in product reviews that explain why users express their opinions on product features, such Sentence B in Example 1. Sentence A is non-explanatory for it does not explain why the user likes the screen while Sentence B is explanatory because it illustrates the reason (viz. "screen resolution is very high").

Example 1:

A. 这部手机太棒了！我最喜欢它的屏幕！(This phone is so great! And its screen is my favorite!)

B. 这个手机的屏幕分辨率很高，我非常喜欢！(The screen resolution of this mobilephone is very high. I like it very much!)

Let e_1 and e_2 be the respective explanatory evaluations for two product features f_1 and f_2, we take the following three steps to compute their contextual similarity (denoted by $CS_{EE}(f_1, f_2)$).

(1) **Explanatory keyword extraction**. For an explanatory evaluation e with n words $\{w_1,w_2,...,w_n\}$, only parts of them are actually helpful cues for feature grouping. We refer these cue words for feature clustering to as explanatory keywords. Thus, we employ the *tf-idf* technique to extract a set of explanatory keywords (denoted by $S_{EK}(e)$).

(2) **Explanatory synset generation.** In this step, we employ the semantic paraphrasing method (Bhagat and Hovy, 2013) to generate a synset (denoted by $Set(e)$ for explanatory keywords in $S_{EK}(e)$.

(3) **Contextual similarity computing.** Let $Set(e_1)$ and $Set(e_2)$ be the explanatory synsets generated in the second step for e_1 and e_2, respectively, we can then

employ Jaccard coefficient in Equation (1) to compute their explanatory evaluation based contextual similarity, namely $CS_{EE}(f_1, f_2)=|Set(e_1)\cap Set(e_2)|/|Set(e_1)\cup Set(e_2)|$.

It should be noted that in this work we also exploit the Jaccard coefficient between common evaluation pairs to compute the contextual similarity between the relevant feature pairs, denoted by CS_{CE}, which is used as the baseline in our experiment to examine the effectiveness of the explanatory evaluation similarity (viz. CS_{EE}).

4 The Clustering Algorithm

Considering the fact that some flat clustering algorithms like k-means do not satisfy the consistency constraint for different granularity clusters (Pavlopoulos *et al.*, 2014), in this work we employ the hierarchical clustering algorithm to perform feature grouping.

Furthermore, different co-referred feature expressions may involve multiple connections. As such, each separate similarity measurement may have its own shortcomings while offering its advantages in dealing with various connections between different feature expressions. To compensate for this, we employ the linear interpolation strategy to combine the above three levels of similarity measures, as shown in Equation (3).

$$Sim(f_i, f_j) = \alpha * LS + \beta * SS + \gamma * CS \tag{3}$$

Where, α, β and γ denote the relevant interpolation coefficients and $\alpha + \beta + \gamma = 1$.

Figure 2 presents the hierarchical clustering algorithm with multilevel similarities for feature normalization. Where, θ is for the threshold for feature clustering. ClusterSim(c_i, c_j) is the average similarity between each pair of features (f_i, f_j) from the two clusters c_i and c_j, as shown in Equation (4).

$$ClusterSim(c_i, c_j) = \frac{\sum_{f_i \in c_i} \sum_{f_j \in c_j} Sim(f_i, f_j)}{|c_i| \times |c_j|} \tag{4}$$

Input: The input set of features $F=\{f_1, f_2, \ldots, f_n\}$ for normalization, and their explanatory evaluations

Output: A set of feature groups $G=\{c_1, c_2, \ldots, c_k\}$.

(1) Initialization: Let each feature $f_i \in F$ be a cluster c_i ($1 \le i \le n$), then $G=\{c_1,c_2,\ldots,c_n\}$
(2) For each $c_i \in \{c_1,c_2,\ldots,c_n\}$,
(3) if $\exists c_j$ that makes $ClusterSim(c_i, c_j)$ be the maximum, and $ClusterSim(c_i, c_j) > \theta$,
(4) then merge clusters c_i and c_j, and update G.
(5) Repeat 2-4 until the number of the groups in G remains unchanged.
(6) Output G as the feature clusters.

Fig. 2. The algorithm for feature clustering

5 Experiments

5.1 Experimental Setup

In our experiments, the two corpora in Table 1 are used as the test sets. To learn word embeddings for feature grouping, two larger corpora of car reviews and mobilephone reviews are also collected from the Web. Furthermore, the Google open source tool[3], viz. word2vec, is used here to learn word embeddings.

To evaluate feature clustering performance, we employ entropy and purity (Zhai *et al.*, 2010). In fact, entropy measures the average uncertainty after feature clustering. Generally, feature clustering with smaller entropy has less uncertainty, indicating the result is better. On the contrary, the purity metric is for describing average purity after clustering. Higher purity indicates that a good clustering result is achieved.

As baselines, we consider methods below.

- **k-means**. k-means is a classical clustering method based on distributional similarity.
- **The latent Dirichlet allocation (LDA).** LDA is a kind of topic model and is widely used in text classification and clustering. Here we use explanatory evaluations as documents for learning LDA in that there are more representative terms.
- **SC-EM.** SC-EM is a state-of-the-art semi-supervised method for grouping product features in English product reviews(Zhai et al., 2010).

[3] http://code.google.com/p/word2vec/

- **SC-EM+WNS.** SC-EM+WNS is modified version of the SC-EM algorithm by considering both product features and their evaluation information(Yang et al., 2012).

Table 2 presents the results of baselines over the test data.

Table 2. Feature grouping results for baselines

Methods	Car		Mobilephone	
	Purity	Entropy	Purity	Entropy
k-means	0.352	2.837	0.545	1.980
LDA	0.352	2.768	0.373	2.357
SC-EM	0.572	1.909	0.672	1.634
SC-EM+WNS	0.585	1.959	0.700	1.589

5.2 Experimental Results

(1) Results for separate similarity.

Our first experiment is conducted to test the effectiveness of separate similarity measurements. The results are summarized in Table 3. It should be notated that with a consideration to the great differences between different similarity measures, we use different threshold θ for clustering. In addition, our guidelines are trying to achieve a set of clusters that is identical to the pre-defined feature groups in number.

Table 3. Results for separate similarity

Methods	Car		Mobilephone	
	P	**E**	**P**	**E**
LS_{ED}	0.690	1.557	0.727	1.137
LS_{JC}	**0.750**	**1.218**	**0.763**	**0.945**
SS_{HN}	0.245	3.137	0.302	2.626
SS_{WE}	0.310	2.777	0.547	1.164
CS_{CE}	0.301	2.674	0.359	2.244
CS_{EE}	0.335	2.661	0.388	2.220

LS_{ED} is the literal similarity of edit distance and SS_{HN} is the semantic similarity based on HowNet. From Table 3, we have a number of observations. First, among the six measurements, Jaccard coefficient yields the best performance in terms of purity

and entropy. This might be due to the fact that most co-referred feature expressions have common parts and are similar with regard to word forms. But on the contrary, some co-referred feature expressions are completely dissimilar with regard to their forms. Second, the cluster number produced by word embeddings is closest to the real number, but the relevant purity and entropy are not satisfactory. This may result from the characteristic of word embeddings. Word embeddings prove to be good tool for discovering implicit linguistic regularities. Actually, each pair of words has a certain similarity when mapping them into a vector space. However, it is very difficult to highlight the interaction between two words with the same semantics. Such interference exits in the case of explanatory evaluations with dynamic polar words. For example, the polar word 高 'high' can modify both 像素 'resolution' and 价格 'price'. Finally, the HowNet similarity produces the worst results. This may be due to the poor coverage of the HowNet lexicon to domain-specific features in online product reviews. In addition, the results for different similarity measurements also provide us with a good suggestion for choosing suitable interpolation coefficients.

(2) Results for multilevel similarity.

Our second experiment is to examine the effectiveness of multilevel similarity for feature clustering. In particular, we consider four kinds of similarity combinations, namely LS_{JC}+ SS_{WE}, LS_{JC} + CS_{EE}, SS_{WE} + CS_{EE} and LS_{JC} + SS_{WE} + CS_{EE}.

It should be noted that multilevel similarity measurements are combined via linear interpolation. Considering the effectiveness of literal similarity in feature clustering and the normalization of probability distribution as well, we employ the following two heuristic rules to determine the interpolation coefficients: (1) $\alpha+\beta+\gamma=1$; and (2) $\alpha\geq\beta\geq\gamma$. Here, we explored Hill-Climbing algorithm (Skalak, 1994) to search the optimal coefficient iteratively. Thus, we set α, β and γ to be 0.62, 0.24 and 0.14, respectively, after iterate optimization.

Tale 4 presents the results for multilevel similarities. It can be seen that the multi-similarities fusion technique with all three layers of measurements yield the best results. By comparing Tables 2 and 3, our system consistently outperforms all the baseline systems in two product reviews. The result proved that the fusion of multilevel similarity can access semantic information of features roundly and indicated in a sense the effectiveness of our approach.

Table 4. Results for separate similarity

Methods	Car		Mobilephone	
	P	E	P	E
$LS_{JC} + SS_{WE}$	0.66	1.555	0.723	0.981
$LS_{JC} + CS_{EE}$	0.655	1.453	0.719	1.086
$SS_{WE} + CS_{EE}$	0.325	2.742	0.475	1.835
$LS_{JC} + SS_{WE} + CS_{EE}$	**0.675**	**1.373**	**0.788**	**0.786**

6 Conclusions and Future Work

In this paper we have explored three layers of clues, namely literal connections, semantic similarity and explanatory evaluations, and further combine them in a hierarchical clustering algorithm for product feature expressions in online product reviews. Our experimental results demonstrate that combining different levels of clues is beneficial to feature clustering.

The encouraging results of the present study suggest several possibilities for future research. First, word embeddings has shown their great value in handling semantic similarity. However, we only employed log-linear models by Mikolov et al. (2013) to learn word embeddings. Future research might usefully extend the present method to explore systematically word embedding learning techniques to achieve more precise semantic similarity measures. Second, explanatory evaluations have proven to be informative clues for feature clustering. However, explanatory evaluation recognition is still at its earlier stage. So further exploration is still needed on explanatory evaluation recognition to acquire more desirable explanatory clues for feature grouping.

Acknowledgments

This study was supported by National Natural Science Foundation of China under Grant No. 61170148 and the Returned Scholar Foundation of Heilongjiang Province.

Reference

1. Rahul Bhagat, and Eduard Hovy. 2013. What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

2. Giuseppe Carenini, Raymond T. Ng, and Ed Zwart. 2005. Extracting knowledge from evaluative text. In *Proceedings of the 3rd International Conference on Knowledge Capture*, pages 11–18.
3. Honglei Guo, Huijia Zhu, Zhili Guo, XiaoXun Zhang, and Zhong Su. 2009. Product feature categorization with multilevel latent semantic association. In *Proceedings of CIKM'09*, pages 1087–1096.
4. Hyun Duk Kim, Malu G. Castellanos, Meichun Hsu, ChengXiang Zhai, Umeshwar Dayal, and Riddhiman Ghosh. 2013. Ranking explanatory sentences for opinion summarization. In *Proceedings of SIGIR'13*, pages 1069–1072.
5. Bing Liu. 2010. Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2:627–666.
6. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
7. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.
8. Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of HLT-NAACL'13*, pages 746–751.
9. John Pavlopoulos and Ion Androutsopoulos. 2014. Multi-granular aspect aggregation in aspect-based sentiment analysis. In *Proceedings of EACL'14*, pages 78–87.
10. Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. 2010. Grouping product features using semi-supervised learning with soft-constraints. In *Proceedings of COLING'10*, pages 1272–1280.
11. Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. 2011. Clustering product features for opinion mining. *In Proceedings of WSDM'11*, pages 347–354.
12. Yang, Y., Ma, Y., and Lin, H. 2012. Clustering product features in opinion mining. *Journal of Chinese Information Processing*. 26(3): 104-108.
13. Yu He, Da Pan, and Guohong Fu. 2015. Chinese explanatory opinionated sentence recognition based on auto-encoding features. *Acta Scientiarum Naturalium Universitatis Pekinensis*, 1-7.
14. Skalak, D. 1994. Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithm. In *Proceedings of ICML '94*, pages 293-301.