Clustering Sentiment Phrases in Product Reviews by Constrained Co-clustering

Yujie Cao(✉), Minlie Huang, and Xiaoyan Zhu

State Key Laboratory of Intelligent Technology and Systems,
National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University,
Beijing 100084, People's Republic of China
caoyujieboy@163.com, {aihuang,zxy-dcs}@tsinghua.edu.cn

Abstract. Clustering sentiment phrases in product reviews is convenient for us to get the most important information about one product directly through thousands of reviews. There are mainly two components in a sentiment phrase, the aspect word and the opinion word. We need to cluster these two parts simultaneously. Although several methods have been proposed to cluster words or phrases, limited work has been done on clustering two-dimensional sentiment phrases. In this paper, we apply a two-sided hidden Markov random field (HMRF) model on this task. We use the approach of constrained co-clustering with some priori knowledge, in a semi-supervised setting. Experimental results on sentiment phrases extracted from about 0.7 million mobile phone reviews show that this method is promising for this task and our method outperforms baselines remarkably.

Keywords: Sentiment analysis · Sentiment phrase clustering · Constrained co-clustering · Sentiment extraction

1 Introduction

The product reviews on the Internet can give both the sellers and the buyers very useful information. However, it's not convenient for us to go through a vast number of reviews to get the information we want. One of the solutions is to cluster sentiment phrases in product reviews. A sentiment phrase is a short phrase that consists of an aspect (feature) word and an opinion word. Nowadays many famous shopping websites or product review websites all provide the clustered sentiment phrases, as exemplified by Fig 1.

Fig. 1. Clustered Sentiment Phrase (from www.taobao.com)

J. Li et al. (Eds.): NLPCC 2015, LNAI 9362, pp. 79–89, 2015.
DOI: 10.1007/978-3-319-25207-0_7

Clustering sentiment phrases in product reviews is different from ordinary text clustering in two ways. Firstly, clustering objects are different. Sentiment phrases in product reviews are not only too short to contain as much information as documents, but also in a fixed format with an aspect (feature) word followed by an opinion word. Secondly, it needs to be clustered in two dimensions simultaneously. We cluster the phrases that are similar both in "aspect" dimension and "opinion" dimension.

To the best of our knowledge, there is limited prior work on clustering two-dimensional sentiment phrases. Typical words or phrases clustering is one-way clustering [1, 2]. And there are some works about clustering aspect-related phrases [3, 4]. Inspired by Li et al. [5] and Song et al. [6, 7], we adopt the idea of using word co-occurrence frequency and applying constrained co-clustering to our task. Word co-occurrence frequency is the frequency of the co-occurrence of an aspect word together with an opinion word. By using this we can find that usually similar aspect words have similar frequency distributions along the opinion words, and vice versa.

Constrained co-clustering is based on Information-theoretic co-clustering (ITCC), and incorporates constraints by a two-sided HMRF regularization [6, 7]. Dhillon et al. pointed out that the ideal co-clustering is one that can minimize the mutual information loss between the original random variables and the clustered ones [8]. In ITCC, the clustering of one dimension enhances that of the other dimension. This method is proper to cluster the two parts of a sentiment phrase simultaneously. However, ITCC cannot solve the problems such as that many dissimilar words of one dimension have similar distribution along the other dimension. And another problem it cannot cope with is that the opinion words opposite in sentiment share similar distribution along the aspect words. So it's preferable to take prior knowledge about clusters into consideration.

Constrained co-clustering leverages the constraints from human-labeled data or the constraints derived from the unlabeled data automatically. We use it to incorporate pairwise constraints into the ITCC, which produces better results as presented in Section 3.

We target Chinese sentiment phrases in this paper. Our experiment was conducted based on a vast number of sentiment phrases extracted from 0.7 million product reviews on mobile phones.

To sum up, the main contributions of this paper are:

- We study the problem of clustering sentiment phrases in product reviews by constrained co-clustering.
- We adopt additional constraints to help the clustering results, and most of the constraints are derived from the unlabeled data automatically.
- Experiment results shows that our method outperforms the ITCC. Our constraints produce better results.

2 Methodology

2.1 Information-Theoretic Co-clustering

Information-theoretic co-clustering (ITCC) is proposed by Dhillon et al. [8]. Different from ordinary one way clustering, information-theoretic co-clustering can solve the simultaneously clustering of two-dimensional data. In the process of ITCC, to determine the row cluster (column) prototype, we have to make use of the information of column (row) clustering. In other words, the clustering of two dimensions can enhance each other.

The ideal co-clustering is one with minimum mutual information loss [8]. The loss in mutual information can be written in the form of the KL divergence of the joint distribution of X and Y, which is p(X, Y), and its approximation q(X, Y). $\hat{X}$ and $\hat{Y}$ are the resulting cluster sets.

$$I(X;Y) - I(\hat{X};\hat{Y}) = KL(p(X,Y) \,||\, q(X,Y)) \tag{1}$$

where q is an approximation of p:

$$q(x,y) = p(\hat{x},\hat{y})\,p(x \mid \hat{x})\,p(y \mid \hat{y}), \text{ where } \; x \in \hat{x}, y \in \hat{y} \tag{2}$$

The objective function for loss in mutual information can be written as follows:

$$KL(p(X,Y) \,||\, q(X,Y)) = \sum_{\hat{x}} \sum_{x \in \hat{x}} p(x) KL(p(Y \mid x) \,||\, q(Y \mid \hat{x})). \tag{3}$$

The algorithm of the information-theoretic co-clustering [8] is to minimize the above function.

Here, p(X, Y) is the words co-occurrence frequency table, where each row represents an aspect word and each column represents an opinion word. Input the joint distribution of X and Y, p(X, Y), and the target row-cluster and column-cluster numbers r and c, we can get a clustering result that simultaneously clusters rows and columns.

	清晰	清楚	高	低	耐用	持久
屏幕	0.2	0.2	0	0	0	0
价钱	0	0	0.2	0.2	0	0
电池	0	0	0	0	0.1	0.1

⇩

	清晰	清楚	高	低	耐用	持久
屏幕	0.2	0.2	0	0	0	0
价钱	0	0	0.2	0.2	0	0
电池	0	0	0	0	0.1	0.1

Fig. 2. An example of co-clustering

From the above example we can see that ITCC is proper to deal with our clustering sentiment phrases, satisfying the demand of clustering two-dimensional simultaneously

according to the word co-occurrence frequency. However, it cannot cope with the following two problems. Firstly, some different aspect words have similar distributions over opinion words such that they cannot be separated by the algorithm. Secondly, some opinion words which are opposite in sentiment have similar distributions over aspect words. For instance, “quick” and “slow” always share almost the same distribution over their common aspect words. So the two words cannot be separated apart.

To address these problems, we propose further to incorporate constraints into this framework.

2.2 Constrained Co-clustering

Constrained co-clustering can add the benefits of constrained clustering to information-theoretic co-clustering, which is proposed by Song et al. [6, 7]. By taking into consideration some prior knowledge about the clusters, the problem can be addressed in a semi-supervised manner. This prior knowledge can guide the clustering process with better results.

The constrained co-clustering is a two-sided HMRF regularized ITCC model [6, 7]. The constraints are formulated by using HMRF for both dimensions. In the HMRF, some pairwise constraints are added to both dimensions.

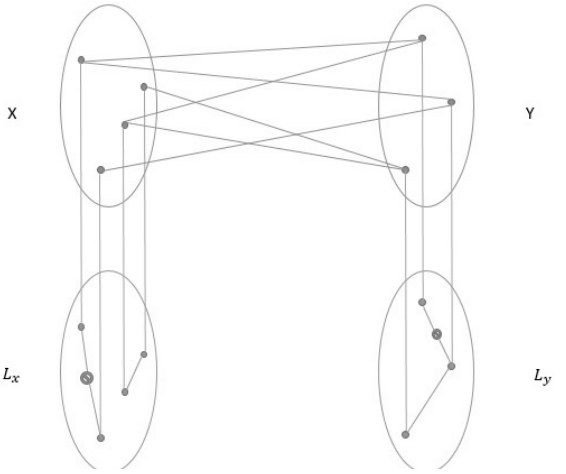

Fig. 3. The two-sided HMRF regularized ITCC model

In Fig 3 the two-sided HMRF regularized ITCC model is exemplified. L_x is the latent label set for X and L_y is the latent label set for Y in the HMRF. The latent labels actually denote the clusters index, and the lines between latent labels are the pairwise constraints.

Compared to the objective function of ITCC, the objective function is added the penalty for violating pairwise constraints between points.

For a latent label l_{x_1} for x_1, the must-link set is denoted as M_{x_1}, and the cannot-link set is denoted as C_{x_1}.The penalty for violating a must-link is:

$$V(x_1, x_2 \in M_{x_1}) = a_{1,2} D(p(Y \mid x_1) \,||\, p(Y \mid x_2)) \cdot I(l_{x_1} \neq l_{x_2}) \tag{4}$$

And the penalty for violating a cannot-link is:

$$V(x_1, x_2 \in C_{x_1}) = \overline{a}_{1,2} (D_{\max} - D(p(Y \mid x_1) \,||\, p(Y \mid x_2))) \cdot I(l_{x_1} = l_{x_2}) \tag{5}$$

where $\mathrm{I}(*)$ here is an indicator function, and $\mathrm{I}(\mathrm{true}) = 1$, $\mathrm{I}(\mathrm{false}) = 0$. $a_{1,2}$ and $\overline{a}_{1,2}$ are tradeoff parameters. D here measures the distance between points. D_{max} is the maximum value for all Ds.

With constraints, we can guide the clustering results with the prior knowledge. We can solve the two problems in ITCC by adding pairwise links to a certain degree.

2.3 Constraints

Manually labeling data is expensive. So we try to generate the constraints automatically without asking the user to label a large scale of data. However, the number of aspect words in sentiment phrases are usually limited. So manually adding pairwise constraints among aspect words will result in a cheap promotion in the clustering result. We put a must-link among the aspect words which describe the same feature (e.g.: "price", "cost" and "worth"), and put a cannot-link among those describing different features (e.g.: "price" and "screen").

The number of opinion words are usually much more than the number of aspect words. We use two ways to generate constraints among these huge number of opinion words. One way is to put cannot-links between opinion words which are opposite in sentiment polarity by making full use of the sentiment information of the phrases and the words. We can get the sentiment polarity ("+1" for "positive" and "-1" for negative) of a sentiment phrase by putting it into a SVM classifier, which has been trained by a vast number of semi-structured product reviews with sentiment polarity tags from the Internet. We suppose that the opinion word contained by the sentiment phrase share the same sentiment polarity. By adding up the polarity of all the sentiment phrases that contains one particular opinion word, we can get the sentiment polarity of this opinion word. If the summed-up polarity is above 0, then this opinion word will be regarded as a positive one, and otherwise a negative one. With the information of the polarity of all the opinion words, cannot-links can be added between every two opinion words with opposite sentiment polarity.

The second way of generating constraints automatically is to take advantage of the words in common. Opinion words having common adjective words are likely to be put into one group, such as "clear" and "very clear". Here we should also take negative words into consideration. If two opinion words share common adjective but one of them contains a negative word, such as "no" or "not" in it, then they may not be put in a common group. Otherwise, if they share common adjective and both of them contain or neither of them contains negative words, then they may be put into a common group.

Note that these constraints may not be all correct. But together with the constrained co-clustering algorithm and mostly correct constraints, the clustering will have a remarkable promotion as soon shown later.

2.4 Sentiment Phrase Extraction

Sentiment Phrases are extracted from product reviews. With a product review, first of all, we split it with some punctuations such as "……，。；？！„,;?!" into short sub-sentences. Next, word segmentation will be conducted in each sub-sentence.

With the sentences after word segmentation, we will check that whether they contain any one of the aspect words obtained in advance. If so, we look up for the first adjective word after that aspect word. Once we find the first adjective word, we get a phrase between the aspect word and this adjective word, and we will continue looking up in the sentence until the end of this short sentence. If the next word is an adjective word, then it will be added up to the phrase we get and the looking up process continues. If the next word is a noun and it is also at the end of the sentence, then it will be added to the phrase. If the next word is a verb, then it will be added in and the looking up process stops. When we get the phrase, we only preserve those which contain less than 5 words.

3 Experiment Results

3.1 Data Preparation

The details of our review corpus are given in Table 1.

Table 1. Statistics of the review corpus

#Products	8
#Reviews	708,450
#Aspects	17

These reviews were crawled from the following websites: www.jd.com, www.pcpop.com, www.it168.com, www.zol.com, weibo.com.

The number of the sentiment phrases extracted from these reviews is 206,793. After duplicate removal, there are 7,263 phrases left, containing 64 unique aspect words and 2,941 opinion words.

The ground-truth of the clustering was labeled manually, containing 336 clusters.

3.2 Evaluation Metrics

We adapt three measures, Purity, Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI) for performance evaluation.

Given a data set D with N items in D, we suppose its gold-standard partition with a total number of J clusters is $G = \{g_1, g_2, \ldots, g_J\}$. A clustering algorithm partitions D into K clusters and $R = \{r_1, r_2, \ldots, r_K\}$ is the clustering result.

Purity: The purity of the entire clustering result is calculated by:

$$\mathrm{P}urity(G,R) = \frac{1}{N}\sum_{k}\max_{j}|r_k \cap g_j|,$$

where each individual item in this addition equation is the intersection of one cluster r_k and the gold-standard cluster by which the majority of r_k are contained.

We can see that if the clustering results perfectly match the gold-standard clusters, the purity is 1. However, it doesn't mean that if the purity is 1 the result is perfect. If K is relatively large, say K = N (the size of D) with only one element on every clusters, the purity is also 1. So by using purity as a measure, there exists a trade-off between the quality of the clustering and the cluster number.

NMI: NMI is the mutual information of the clustering result and the gold-standard clusters divided by half of the sum of their entropies. The NMI of the entire clustering result is calculated by:

$$NMI(G,R) = \frac{I(G,R)}{[H(G)+H(R)]/2},$$

where $I(G,R)$ is the mutual information of G and R, $H(G)$ and $H(R)$ is the entropies of G and R.

It can overcome the disadvantage of purity when K is very large, for $H(R)$ will increase as the K increases. Generally speaking, a larger NMI usually means a better result of clustering.

ARI: ARI is the adjusted form of Rand Index. Rand Index (RI) will give punishment to false positive decisions and false negative decisions, where false positive decision means that a pair of elements in different clusters in G was put into one cluster in R and false negative is similar. ARI is the difference of the RI and its expected value under the null hypothesis [9]. The ARI is calculated by:

$$ARI(G,R) = \frac{2(N_{00}N_{11} - N_{01}N_{10})}{(N_{00}+N_{01})(N_{01}+N_{11})+(N_{00}+N_{10})(N_{10}+N_{11})}.$$

N_{00} is the number of element pairs that are in different clusters both in G and R. N_{11} is that in same cluster both in G and R. N_{10} is that in same cluster in G but in different clusters in R. And N_{01} is similar but opposite to N_{10} in G and R.

ARI measures the degree of agreement between the gold-standard result and the clustering result by checking every pair of elements.

3.3 Evaluation Results

Comparison to Different Constraints. We apply our approach with different constraints and make comparisons among them. The given cluster number of aspect words is empirically set to 36, and that of opinion words is 100. The clusters with less than 5 sentiment phrases in the clustering results are all merged into two big clusters according to their sentiment polarity (+1 and -1) in the end.

We compare methods with different constraints. These methods are listed as follows.

- **Information-Theoretic Co-clustering (ITCC):** Information-theoretic co-clustering without any constraints. Here we use ITCC as a baseline.

- **Information-Theoretic Co-clustering with Aspect Constraints (ITCCAC):** Constrained Information-theoretic co-clustering with constraints only on aspect words. These constraints are human-labeled.
- **Information-Theoretic Co-clustering with Opinion Constraints (ITCCOC):** Constrained Information-theoretic co-clustering with constraints only on opinion words, including cannot-links from sentiment polarity and constraints from words in common.
- **Information-Theoretic Co-clustering with Aspect and Opinion Constraints (ITCCAOC):** Constrained Information-theoretic co-clustering with constraints both on aspect and opinion words.

Table 2. Comparisons among co-clustering with different constraints with #aspect cluster = 36 and #opinion cluster = 100

	NMI	Purity	ARI
ITCC	0.707	0.469	0.204
ITCCAC	0.742	0.547	0.240
ITCCOC	0.806	0.651	0.426
ITCCAOC	**0.859**	**0.795**	**0.566**

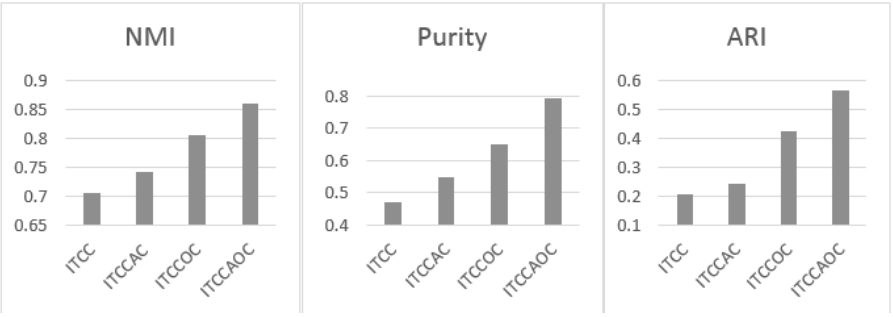

Fig. 4. NMI, Purity and ARI of the comparison

From the above table we can see that co-clustering without any constraints has the worst performance, with the lowest scores in three measures. By adding constraints to aspect words, the three measures all have a promotion. By adding constraints to opinion words, the three measures have a bigger promotion, for the number of constraints in opinion words are much more than that of aspect words so that they have stronger impact on the clustering result. By adding constraints to both the aspect words and the opinion words, the result is the best.

We can draw a conclusion that adding proper constraints to both the aspect and opinion words can help do a better job in clustering sentiment phrases.

Comparison to Other Clustering Methods. We further compare our approach with other clustering methods. We choose these clustering methods in the experiment setting of [6, 7]. The given cluster number of aspect words is empirically set to 36, and that of opinion words is 100. These methods are listed as follows. Kmeans, constrained Kmeans (CKmeans) [10], Tri-factorization of Semi-NMF (STriNMF) [11], and constrained Tri-factorization of Semi-NMF (CSTriNMF) [11]. Kmeans and CKmeans are one-way clustering methods. We apply them to both two dimensions (the aspect words and the opinion words) separately. Then we assemble sentiment phrases clusters with the aspect clusters and the opinion clusters. STriNMF, CSTriNMF, Information-Theoretic Co-Clucstering (ITCC) and constrained Information-Theoretic Co-Clustering (CITCC) are co-clustering methods. CKmeans, CSTriNMF and CITCC are all clustering methods with constraints.

Table 3. Comparison to different clustering methods

	NMI	Purity	ARI
STriNMF	0.685	0.421	0.187
CSTriNMF	0.694	0.446	0.192
Kmeans	0.708	0.459	0.215
CKmeans	0.773	0.586	0.362
ITCC	0.707	0.469	0.204
CITCC	**0.859**	**0.795**	**0.566**

From the above table we can see that we apply CITCC to our task can get the best results. Our constraints can improve the performance in KMeans and ITCC, but can do little favor to STriNMF and it can make a greater promotion in CITCC than other constrained clustering methods.

4 Related Works

The related works are in three parts: co-clustering, constrained co-clustering, words and phrases clustering.

Co-clustering algorithms deal with two-dimensional clustering. The two-dimensional data can be modeled in a co-occurrence matrix and the clustering problem can be solved by matrix factorization [12]. And it can also be modeled in a bipartite graph form and the clustering problem can be solved by graph partition [13]. It can be modeled in a joint distribution of two discrete random variables and information theory are used to partition the two sets of variables [8].

Constrained co-clustering incorporates prior knowledge constraints to co-clustering to have a promotion. Wang et al. [11] proposed a constrained co-clustering by matrix factorization, and so is Shi et al. [14]. The objective functions of all these constrained co-clustering methods are all sum squared residue-based in Euclidean distance. However, Song et al. [6, 7] use ITCC framework which use KL divergence and proposed constrained co-clustering method. It's more proper for sparse and high dimensional data.

Words and phrases clustering is also very much related to our work. Matsuo et al. [15] proposed a method of using web search engines as a corpus to perform a graph-based word clustering. SanJuan et al. [16] proposed a method for clustering phrases based on general lexico-syntactic relations without prior knowledge. Zhai et al. [3] proposed their EM based unsupervised methods for aspect expressions clustering. Zhao et al. [4] used a soft constraint with the PR framework to cluster aspect-related phrases.

5 Conclusion

Clustering sentiment phrases in product reviews is an important and useful task for sentiment analysis. In order to cluster sentiment phrases, this paper applies constrained co-clustering and incorporates rich constrained knowledge. We obtain most of the constraints automatically. Experiments show that our constraints are proper and useful and our method is superior to the baselines.

References

1. Yutaka, M., Takeshi, S., Koki, U., and Mitsuru, I.: Graph-based word clustering using a web search engine. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 542–550 (2006)
2. Lin, D., Wu, X.: Phrase clustering for discriminative learning. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP), pp. 1030–1038 (2009)
3. Zhai, Z., Liu, B., Xu, H., Jia, P.: Clustering product features for opinion mining. In: Proceedings of the 4th ACM International Conference on Web Search and Data Mining, pp. 347–354 (2011)
4. Zhao, L., Huang, M., Chen, H., Cheng, J., Zhu, X.: Clustering aspect-related phrases by leveraging sentiment distribution consistency. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1614–1623 (2014)
5. Li, H., Abe, N.: Word clustering and disambiguation based on co-occurrence data. In: Proceedings of the 17th International Conference on Computational Linguistics, pp. 749–755 (1998)
6. Song, Y., Pan, S., Liu, S.: Constrained co-clustering for textual documents. In: Proceedings of the 24th AAAI conference on Artificial Intelligence, pp. 581–586 (2010)
7. Song, Y., Pan, S., Liu, S., Wei, F., Zhou, M.X., Qian, W.: Constrained text coclustering with supervised and unsupervised constraints. IEEE Trans. Knowl. Data Eng. **25**(6), 1227–1239 (2013)
8. Dhillon, I.S., Mallela, S., Modha, D. S.: Information-Theoretical Coclustering. In: Proceedings of the Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD 2003), pp. 89–98 (2003)
9. Wagner, S., Wagner, D.: Comparing clusterings - an overview. Technical report 2006-04, Faculty of Informatics, Universitat Karlsruhe (TH) (2006)
10. Basu, S., Bilenko, M., Mooney, R. J.: A probabilistic framework for semi-supervised clustering. In: Proceedings of the 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 59–68 (2004)

11. Wang, F., Li, T., Zhang, C.: Semi-supervised clustering via matrix factorization. In: Proceedings of SIAM Int'l Conf. Data. Mining (SDM), pp. 1–12 (2008)
12. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix trifactorizations for clustering. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 126–135 (2006)
13. Dhillio, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001), pp. 269–274 (2001)
14. Shi, X., Fan, W., Yu, P.S.: Efficient semi supervised spectral co-clustering with constraints. In: Proceedings of IEEE 10th International Conf. Data Mining (ICMD), pp. 1043–1048 (2010)
15. Matsuo, Y., Sakaki, T., Uchiyama, k., Ishizuka, M.: Graph based word clustering using web search engine. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 542–550 (2006)
16. SanJuan, E., Fidelia I.: Phrase clustering without document context. In: Proceedings of the 28th European Conference on Information Retrieval, pp. 494–497 (2006)