Implicit Feature Detection via a Constrained Topic Model and SVM

Wei Wang [*], **Hua Xu** [*] and **Xiaoqiu Huang** [†]
[*]State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University,
Beijing 100084, China
[†]Beijing University of Posts and Telecommunications, Beijing 100876, China
ww880412@gmail.com, xuhua@tsinghua.edu.cn, alexalexhxqhxq@gmail.com

Abstract

Implicit feature detection, also known as implicit feature identification, is an essential aspect of feature-specific opinion mining but previous works have often ignored it. We think, based on the explicit sentences, several Support Vector Machine (SVM) classifiers can be established to do this task. Nevertheless, we believe it is possible to do better by using a constrained topic model instead of traditional attribute selection methods. Experiments show that this method outperforms the traditional attribute selection methods by a large margin and the detection task can be completed better.

1 Introduction

Feature-specific opinion mining has been well defined by Ding and Liu(2008). Example 1 is a cell phone review in which two features are mentioned.

Example 1 *This cell phone is fashion in appearance, and it is also very cheap.*

If a feature appears in a review directly, it is called an *explicit feature*. If a feature is only implied, it is called an *implicit feature*. In Example 1, *appearance* is an explicit feature while *price* is an implicit feature, which is implied by *cheap*. Furthermore, an *explicit sentence* is defined as a sentence containing at least one explicit feature, and an *implicit sentence* is the sentence only containing implicit features. Thus, the first sentence is an explicit sentence, while the second is an implicit one.

This paper proposes an approach for implicit feature detection based on SVM and Topic Model(TM). The Topic Model, which incorporated into constraints based on the pre-defined product feature, is established to extract the training attributes for SVM. In the end, several SVM classifiers are constructed to train the selected attributes and utilized to detect the implicit features.

2 Related Work

The definition of implicit feature comes from Liu et al. (2005)'s work. Su et al. (2006) used Pointwise Mutual Information (PMI) based semantic association analysis to identify implicit features, but no quantitative experimental results were provided. Hai et al. (2011) used co-occurrence association rule mining to identify implicit features. However, they only dealt with opinion words and neglected the facts. Therefore, in this paper, both the opinions and facts will be taken into account.

Blei et al. (2003) proposed the original LDA using EM estimation. Griffiths and Steyvers (2004) applied Gibbs sampling to estimate LDA's parameters. Since the inception of these works, many variations have been proposed. For example, LDA has previously been used to construct attributes for classification; it often acts to reduce data dimension(Blei and Jordan, 2003; Fei-Fei and Perona, 2005; Quelhas et al., 2005). Here, we modify LDA and adopt it to select the training attributes for SVM.

3 Model Design

3.1 Introduction to LDA

We briefly introduce LDA, following the notation of Griffiths(Griffiths and Steyvers, 2004). Given D

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 903–907,
Seattle, Washington, USA, 18-21 October 2013.

documents expressed over *W* unique words and *T* topics, LDA outputs the document-topic distribution θ and topic-word distribution φ, both of which can be obtained with Gibbs Sampling. For this scheme, the core process is the topic updating for each word in each document according to Equation 1.

$$P(z_i = j|\mathbf{z}_{-i}, \mathbf{w}, \alpha, \beta) = \\ (\frac{n_{-i,j}^{(w_i)} + \beta}{\sum_{w'}^{W} n_{-i,j}^{(w')} + W\beta})(\frac{n_{-i,j}^{(d_i)} + \alpha}{\sum_{j}^{T} n_{-i,j}^{(d_i)} + T\alpha}) \quad (1)$$

where $z_i = j$ represents the assignment of the i^{th} word in a document to topic j, z_{-i} represents all the topic assignments excluding the i^{th} word. $n_j^{(w')}$ is the number of instances of word w' assigned to topic j and $n_j^{(d_i)}$ is the number of words from document d_i assigned to topic j, the $-i$ notation signifies that the counts are taken omitting the value of z_i. Furthermore, α and β are hyper-parameters for the document-topic and topic-word Dirichlet distributions, respectively. After N iterations of Gibbs sampling for all words in all documents, the distribution θ and φ are finally estimated using Equations 2 and 3.

$$\phi_j^{(w_i)} = \frac{n_j^{(w_i)} + \beta}{\sum_{w'}^{W} n_j^{(w')} + W\beta} \quad (2)$$

$$\theta_j^{(d_i)} = \frac{n_j^{(d_i)} + \alpha}{\sum_{j}^{T} n_j^{(d_i)} + T\alpha} \quad (3)$$

3.2 Framework

Algorithm 1 summarizes the main steps. When a specific product and the reviews are provided, the explicit sentences and corresponding features are extracted(Line 1) by word segmentation, part-of-speech(POS) tagging and synonyms feature clustering. Then the prior knowledge are drawn from the explicit sentences automatically and integrated into the constrained topic model((Line 3 - Line 5). The word clusters are chosen as the training attributes(Line 6). Finally, several SVM classifiers are generated and applied to detect implicit features(Line 7 - Line 12).

Algorithm 1 Implicit Feature Detection

1: $ES \leftarrow$ extract explicit sentence set
2: $NES \leftarrow$ non-explicit sentence set
3: $CS \leftarrow$ constraint set from ES
4: $CPK \leftarrow$ correlation prior knowledge from ES
5: $ETM \leftarrow$ ConstrainedTopicModel(T,ES,CS,CPK)
6: $TA \leftarrow$ select training attributes from ETM
7: **for** each f_i in feature clusters **do**
8: $TD_i \leftarrow$ GenerateTrainingData(TA_i,ES)
9: $C_i \leftarrow$ BuildClassificationModelBySVM(TD_i)
10: $PR_i \leftarrow$ positive result of Classify(C_i,NES)
11: the feature of sentence in $PR_i \leftarrow f_i$
12: **end for**

3.3 Prior Knowledge Extraction and Incorporation

It is obvious that the pre-existing knowledge can assist to produce better and more significant clusters. In our work, we use a constrained topic model to select attributes for each product features. Each topic is first pre-defined a product feature. Then two types of prior knowledge, which are derived from the pre-defined product features, are extracted automatically and incorporated: must-link/cannot-link and correlation prior knowledge.

3.3.1 Must-link and Cannot-link

Must-link: It specifies that two data instances must be in the same cluster. Here is the must-link from an observation: as "*cheap*" to "*price*", some words must be associated with a feature. In order to mine these words, we compute the co-occurrence degree by *frequency*PMI(f,w)*, whose formula is as following: $P_{f\&w} * \log_2 \frac{P_{f\&w}}{P_f P_w}$, where P is the probability of subscript occurrence in explicit sentences, f is the feature, w is the word, and $f\&w$ means the co-occurrence of f and w. A higher value of *frequency*PMI* signifies that w often indicates f. For a feature f_i, the top five words and f_i constitute must-links. For example, the co-occurrence of "*price*" and "*cheap*" is very high, then the must-link between "*price*" and "*cheap*" can be identified.

Cannot-link: It specifies that two data instances cannot be in the same cluster. If a word and a feature never co-occur in our corpus, we assume them to form a cannot-link. For example, the word *low-cost* has never co-occurred with the product feature *screen*, so they constitute a cannot-link in our cor-

pus.

In this paper, the pre-defined process, must-link, and cannot-link are derived from Andrzejewski and Zhu (2009)'s work, all must-links and cannot-links are incorporated our constrained topic model. We multiply an indicator function $\delta(w_i, z_j)$, which represents a hard constraint, to the Equation 1 as the final probability for topic updating (see Equation 4).

$$P(z_i = j|\mathbf{z}_{-i}, \mathbf{w}, \alpha, \beta) = \delta(w_i, z_j)(\frac{n_{-i,j}^{(w_i)} + \beta}{\sum_{w'}^{W} n_{-i,j}^{(w')} + W\beta})(\frac{n_{-i,j}^{(d_i)} + \alpha}{\sum_{j}^{T} n_{-i,j}^{(d_i)} + T\alpha}) \quad (4)$$

As illustrated by Equations 1 and 4, $\delta(w_i, z_j)$, which represents intervention or help from pre-existing knowledge of must-links and cannot-links, plays a key role in this study. In the topic updating for each word in each document, we assume that the current word is w_i and its linked feature topic set is $Z^{(w_i)}$, then for the current topic z_j, $\delta(w_i, z_j)$ is calculated as follows:

1. If w_i is constrained by must-links and the linked feature belongs to $Z^{(w_i)}$, $\delta(w_i, z_j|z_j \in Z^{(w_i)}) = 1$ and $\delta(w_i, z_j|z_j \notin Z^{(w_i)}) = 0$.

2. If w_i is constrained by cannot-links and the linked feature belongs to $Z^{(w_i)}$, $\delta(w_i, z_j|z_j \in Z^{(w_i)}) = 0$ and $\delta(w_i, z_j|z_j \notin Z^{(w_i)}) = 1$.

3. In other cases, $\delta(w_i, z_j|j = 1, \ldots, T) = 1$.

3.3.2 Correlation Prior Knowledge

In view of the explicit product feature of each topic, the association of the word and the feature to topic-word distribution should be taken into account. Therefore, Equation 2 is revised as the following:

$$\phi_j^{(w_i)} = \frac{(1 + C_{w_i,j})(n_j^{(w_i)}) + \beta}{\sum_{w'}^{W}(1 + C_{w',j})(n_j^{(w')}) + W\beta} \quad (5)$$

where $C_{w',j}$ reflects the correlation of w' with the topic j, which is centered on the product feature f_{z_j}. The basic idea is to determine the association of w' and f_{z_j}, if they have the high relevance, $C_{w',j}$ should be set as a positive number. Otherwise, if we can determine w' and f_{z_j} are irrelevant, $C_{w',j}$ should be set as a positive number. In this paper, we attempt to using PMI or dependency relation to judge the relevance. For word w' and feature f_{z_j}:

1. Dependency relation judgement: If w' as parent node in the syntax tree mainly co-occurs with f_{z_j}, $C_{w',j}$ will be set positive. If w' mainly co-occurs with several features including f_{z_j}, $C_{w',j}$ will be set negative. Otherwise, $C_{w',j}$ will be set 0.

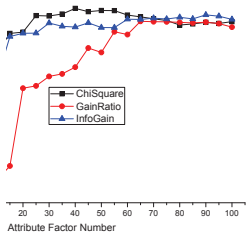
2. PMI judgement: If w' mainly co-occurs with f_{z_j} and $PMI(w', f_{z_j})$ is greater than the given value, $C_{w',j}$ will be set positive. Otherwise, $C_{w',j}$ will be set negative.

3.4 Attribute Selection

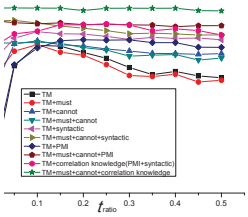
Some words, such as "*good*", can modify several product features and should be removed. In the result of run once, if a word appears in the topics which relates to different features, it is defined as a **conflicting word**. If a term is thought to describe several features or indicate no features, it is defined as a **noise word** .

When each topic has been pre-allocated, we run the explicit topic model 100 times. If a word turns into a conflicting word T_{cw} times(T_{cw} is set to 20), we assume that it is a noise word. Then the noise word collection is obtained and applied to filter the explicit sentences. Actually, here 100 is just an estimated number. And for T_{cw}, when it is between 15 and 25, the result is same, and when it exceeds 25, the result does not change a lot. The most important part to filter noise words is the correlation computation. So the experiment can work well with only estimated parameters.

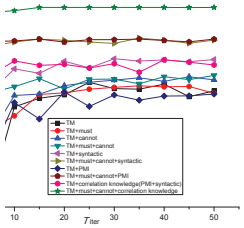
Next, By integrating pre-existing knowledge, the explicit topic model, which runs T_{iter} times, severs as attribute selection for SVM. In every result for each topic cluster, we remove the least four probable of word groups and merge the results by the pre-defined product feature. For a feature, if a word appears in its topic words more than $T_{iter} * t_{ratio}$ times, it is selected as one of the training attributes for the feature. In the end, if an attribute associates with different features, it is deleted.



(a) SVM based on traditional attribute selection method



(b) our constrained topic model by different t_{ratio} ($T_{iter} = 20$)



(c) our constrained topic model by different T_{iter} ($t_{ratio} = 0.1$)

Figure 1: Performance of different cases

3.5 Implicit Feature Detection via SVM

After completing attribute selection, vector space model(VSM) is applied to the selected attributes on the explicit sentences. For each feature f_i, a SVM classifier C_i is adopted. In train-set, the positive cases are the explicit sentences of f_i, and the negative cases are the other explicit sentences. For a non-explicit sentence, if the classification result of C_i is positive, it is an implicit sentence which implies f_i.

4 Evaluation of Experimental Results

4.1 Data Sets

There has no standard data set yet, we crawled the experiment data, which included reviews about a cellphone, from a famous Chinese shopping website[1]. The data contains 14218 sentences. The feature of each sentence was manually annotated by two research assistants. A handful of sentences which were annotated inconsistently were deleted. Table 1 depicts the data set which is evaluated. Other features were ignored because of their rare appearance.

Here are some explanations: (1)The sentences containing several explicit features were not added to the train-set. (2) A tiny number of sentences contain both explicit and implicit features, and they can only be regarded as explicit sentences. (3) The training set contains 3140 explicit sentences, the test set contains 7043 non-explicit sentences and more than 5500 sentences have no feature. (4) According to the ratio among the explicit sentences(6:1:2:3:1:2), it is reasonable that the most suitable number of topics should be 14. For example, the ratio of the product feature *screen* is 6, so we can assign the feature to topic 0,1,2,3,4,5. In our experiment, the performance of algorithm 1 is evaluated using F-measure. (5) Although the size of dataset is limited, out proposed is based on the constraint-based topic model, which has been widely used in different NLP fields. So, our approach can generalize well in different datasets. Of course, more high quality data will be collected to do the experiment in the future.

[1] http://www.360buy.com/

Table 1: Experiment data

Features	Explicit	Implicit	Total
screen	1165	244	1409
quality	199	83	282
battery	456	205	661
price	627	561	1188
appearance	224	167	391
software	469	129	598

4.2 Experimental Results

Figure 1a depicts the performance of using traditional attribute selection methods on SVM. Using χ^2 *test* on SVM can achieve the best performance, which is about 66.7%. In our constrained topic model, we use different T_{iter} and t_{ratio}. We conducted experiments by incorporating different types prior knowledge. From Figure 1b and 1c, we conclude that: (1)All these methods perform much better than the traditional feature selection methods, the improvements are more than 6%. (2)The reason for the little improvement of must-links is that the topic clusters have already obtained these linked word-

s. (3)All the pre-existing knowledge performs best and shows 3% improvement over non prior knowledge. (4)Different types of prior knowledge have different impact on the stabilities of different parameters. (5)As we have expected, by combing all prior knowledge, the best performance can reach 77.78%. Furthermore, as t_{ratio} or T_{iter} changes, our constrained topic model incorporating all prior knowledge look like very stable.

5 Conclusions

In this paper, we adopt a constrained topic model incorporating prior knowledge to select attribute for SVM classifiers to detect implicit features. Experiments show this method outperforms the attribute feature selection methods and detect implicit features better.

6 Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant No: 61175110) and National Basic Research Program of China (973 Program, Grant No: 2012CB316305).

References

David Andrzejewski and Xiaojin Zhu. 2009. Latent dirichlet allocation with topic-in-set knowledge. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, pages 43–48. Association for Computational Linguistics.

D.M. Blei and M.I. Jordan. 2003. Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134. ACM.

D.M. Blei, A.Y. Ng, and M.I. Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the international conference on Web search and web data mining*, WSDM '08, pages 231–240, New York, NY, USA. ACM.

L. Fei-Fei and P. Perona. 2005. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE.

T.L. Griffiths and M. Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235.

Z. Hai, K. Chang, and J. Kim. 2011. Implicit feature identification via co-occurrence association rule mining. *Computational Linguistics and Intelligent Text Processing*, pages 393–404.

B. Liu, M. Hu, and J. Cheng. 2005. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, pages 342–351. ACM.

P. Quelhas, F. Monay, J.M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool. 2005. Modeling scenes with local descriptors and latent aspects. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 883–890. IEEE.

Q. Su, K. Xiang, H. Wang, B. Sun, and S. Yu. 2006. Using pointwise mutual information to identify implicit features in customer reviews. *Computer Processing of Oriental Languages. Beyond the Orient: The Research Challenges Ahead*, pages 22–30.