Hidden Sentiment Association in Chinese Web Opinion Mining

Qi Su[1]* , Xinying Xu[1], Honglei Guo[2], Zhili Guo[2], Xian Wu[2], Xiaoxun Zhang[2], Bin Swen[1] and Zhong Su[2]

Peking University[1]
Beijing, 100871, China
{sukia, xuxinying, bswen}@pku.edu.cn

IBM China Research Lab[2]
Beijing, 100094, China
{guohl, guozhili, wuxian, zhangxx, suzhong}@cn.ibm.com

ABSTRACT

The boom of product review websites, blogs and forums on the web has attracted many research efforts on opinion mining. Recently, there was a growing interest in the finer-grained opinion mining, which detects opinions on different review features as opposed to the whole review level. The researches on feature-level opinion mining mainly rely on identifying the explicit relatedness between product feature words and opinion words in reviews. However, the sentiment relatedness between the two objects is usually complicated. For many cases, product feature words are implied by the opinion words in reviews. The detection of such hidden sentiment association is still a big challenge in opinion mining. Especially, it is an even harder task of feature-level opinion mining on Chinese reviews due to the nature of Chinese language. In this paper, we propose a novel mutual reinforcement approach to deal with the feature-level opinion mining problem. More specially, 1) the approach clusters product features and opinion words simultaneously and iteratively by fusing both their content information and sentiment link information. 2) under the same framework, based on the product feature categories and opinion word groups, we construct the sentiment association set between the two groups of data objects by identifying their strongest n sentiment links. Moreover, knowledge from multi-source is incorporated to enhance clustering in the procedure. Based on the pre-constructed association set, our approach can largely predict opinions relating to different product features, even for the case without the explicit appearance of product feature words in reviews. Thus it provides a more accurate opinion evaluation. The experimental results demonstrate that our method outperforms the state-of-art algorithms.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural language processing—*text analysis*

General Terms

Algorithms

*Part of this work was done while Qi Su and Xinying Xu were interns at IBM China Research Lab

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

Keywords

opinion mining, product feature, opinion word, association, mutual reinforcement

1. INTRODUCTION

With the dramatic growth of web's popularity, the number of freely available online reviews is increasing at a high speed. A significant number of websites, blogs and forums allow users to post reviews for various products or services (e.g., amazon.com). Such reviews are valuable resources to help the potential customers make their purchase decisions. This situation is also notable in Chinese web services. In the past few years, mining the opinions expressed in web reviews attracts extensive researches [3, 10, 13, 19]. Based on a collection of customer reviews, the task of opinion mining is to extract customers' opinions and predict the sentiment orientation. It usually can be integrated into search engines to satisfy users' search needs related to opinions, such as comparative web search(CWS)[17] and opinion question answering[9, 22]. In recent years, the Text REtrieval Conference (TREC) also held a task of finding relevant opinion sentences to a given topic in the Novelty track[9].

The task of opinion mining has been usually approached as a classification of either positive or negative on a review or its snippet. However, for many applications, simply judging the sentiment orientation of a review unit is not sufficient. Researchers[7, 8, 10, 15] began to work on finer-grained opinion mining which predicts the sentiment orientation related to different review features. The task is known as feature-level opinion mining. Take product review as an example, a reviewer may praise some features of the product and while bemoan it in other features. So it is important to find out reviewers' opinions toward different product features instead of the overall opinion in those reviews.

In feature-level opinion mining, most of the existing researches associate product features and opinions by their explicit co-occurrence. For example, for a product feature that appears explicitly in reviews, we can judge the attitude towards it by its nearest adjacent opinion words. Either we can conduct syntax parsing to judge the modification relationship between opinion words and the product feature within a review unit. However, the approaches are either crude or inefficient in time cost, thus not very fit for real-time online web applications. Moreover, real reviews from customers are usually complicated. The approaches are not effective for many cases. Look at the following automobile review sentences:

1. MiniCooper Convertible . . . 可爱、漂亮、又贵的离谱。
   MiniCooper Convertible . . . lovely, pretty but too expensive.
2. 车身**线条**流畅，雍容华贵的感觉尽显无疑。
   The car has a smooth **line** and stylish feeling.
3. 初见辉腾给人的感觉就是一款典型的德国车，很中庸很踏实。
   The first feeling which Phaeton arouses is that it's a typical Germany car, ordinary and plain.
4. 雪铁龙C5的**前脸设计**摄人心神。
   Citrön C5's **front design** is impressive.
5. 05款的东方之子表现不俗，. . . 。
   Eastar 05 has good performance, . . .

6. *The Corvette C6 is beautiful, but commonplace.*
7. *The NSX is truly one of the world's finest cars.*
8. *Parts are expensive and the car is far more complicated.*
9. *It is superbly sporty, yet elegant and understated.*
10. ***Transmission** is clunky and suspect on many older A8s.*

Take the Chinese parts in the above table[1] as examples, we use a figure (Figure 1) to show the complicated relationship between the product features and the opinion words in the sentences. Sometimes the product features is explicit in reviews. Such as the product feature "*front design*" in the sentence 4. But for many cases, product feature words are implicit in review sentences. "*MiniCooper Convertible is expensive*" has the same meaning as "*MiniCooper Convertible's price is expensive*". So it is considered that the real product feature "*price*" is left out in the review sentence. The similar situation happens in the sentence 3. Although the product feature may not appear explicitly in reviews, it is usually implied by the opinion words in its context. For example, from the opinion words of "*lovely, pretty, . . .*", we can deduce that the product feature being evaluated should be the *"appearance"* or *"design"* of the car (see the related hollow circles and dash lines between product features and opinion words in figure 1). So, hidden sentiment association essentially exists between the product feature category and the group of opinion words. There is no doubt that the approach of either explicit adjacency or syntactic analysis is not the way to deal with this kind of problem.

The basic purpose of our approach in this paper is to mine the hidden sentiment links between the groups of product feature words and opinion words, then build the association set. Using the pre-constructed association set, we can identify feature-oriented sentiment orientation of opinions more conveniently and accurately. The major contributions of our approach are as follows:

• Product feature words and opinion words are organized into categories, thus we can provide a non-trivial and more sound opinion evaluation than the existing word-based approaches.

• We develop a mutual reinforcement principle to mine the associations between product feature categories and opinion word groups.

• We propose to enhance clustering quality by both the multi-source knowledge and the mutual reinforcement principle.

Aim at the Chinese applications, we develop the system architecture based on the specialty of Chinese language,

[1]The Chinese parts of these examples are taken from http://auto.sohu.com; the English parts are taken from http://www.carreview.com

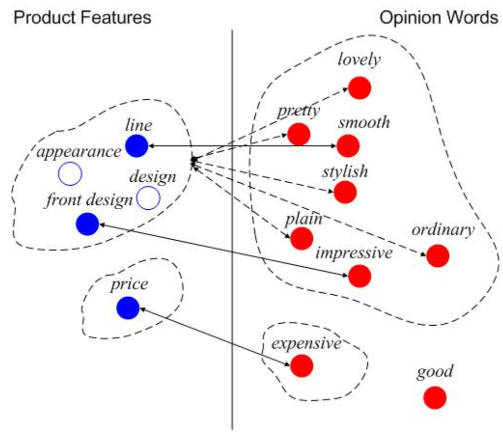

Figure 1: Complicated Relationship Between Product Features and Opinion Words in Real Reviews (solid circle/solid line represents an explicit word/relationship; hollow circle/dash line represents an implicit word/relationship)

and verify the performance on Chinese web reviews. However, the main proposed approach in this paper is language-independent in essence.

With our approach, we can get an association set for the above sentences as in table 1. It shows the advantage of our approach over the existing approaches in identifying the hidden sentiment links between product features and opinion words. Since the association set is pre-constructed, our approach is well fit for online applications.

Table 1: Identified Product Features and the Related Opinion Words Using the Existing Approaches And Our Approach (with sentence numbers in brackets)

Existing Approaches

Identified Feature	Opinion Word
MiniCooper Convertible (1)	lovely; pretty; expensive
line (2)	smooth
feeling (2)	stylish
it/Phaeton (3)	ordinary; plain
front design (4)	impressive
performance (5)	good

Our Approach

Identified Feature	Opinion Word
appearance; line; design; front design (1, 2, 3, 4)	lovely; pretty; smooth; plain; stylish; ordinary; impressive
price (1)	expensive

The remainder of the paper is organized as follows. In section 2, we introduce some related works. Our sentiment association approach based on the mutual reinforcement principle is proposed in section 3. In addition, we present the strategy of clustering optimization for the mutual reinforcement based on a combination of multi-source knowledge. Section 4 overviews the system architecture, and also concerns the extraction, pruning and representation of product features. Experiments and evaluations are reported in section 5. We conclude the paper in section 6 with future researches.

2. RELATED WORKS

Opinion mining has been extensively studied in recent years. A majority of these researches has focused on identifying the polarity expressed in various opinion units such as word, phrase, sentence or review document. While not so much work has been done on feature level opinion mining, especially for Chinese reviews[16].

Liu[10] and Hu[7, 8]'s works may be the most representative researches in this area. The appearance of implicit product feature was first showed in their papers based on English data. Obviously, if a product feature appears explicitly in review units, it is an explicit product feature. While we consider that an implicit product feature should satisfy the following two conditions: 1) the related product feature word doesn't occur explicitly; 2) the feature can be deduced by its surrounding opinion words in the review. Our definition of implicit product feature is a little different from the definition in [10]. In the paper, they gave an example to show the implicit product feature in a digital camera review:*"included 16MB is stingy"*. They considered "*16MB*" as a value of product feature *"memory"*. Since the feature word *"memory"* does not appear in the sentence segment, it is an implicit feature. For our approach, we only take the product features implied by opinion words as implicit ones. Words like "*16MB*" are treated as clustering objects to build product feature categories.

The association rule mining approach in [10] did a good job in identifying product features, but it can not deal with the identification of implicit features effectively. They also noted the cases of synonyms and granularity of features. Different words may be used to mean the same product feature. In addition, some product features may be too specific and fragment the opinion evaluation. They deal with the problems by the synonym set in WordNet and the semi-automated tagging of reviews. Our approach groups product feature words (including those which are considered to express the values of some product features in [10]) into categories. It's an unsupervised method and easy to be adapted to new domains.

Our approach associates product feature categories and opinion word groups by their interrelationship. The idea of mutual reinforcement for multi-type interrelated data objects is utilized in some applications, such as web mining and collaborative filtering [21]. We develop the idea to identify the association between product feature categories and opinion word groups, and simultaneously enhance clustering under the uniform framework.

3. ASSOCIATION APPROACH TO FIND HIDDEN LINKS BETWEEN PRODUCT FEATURES AND OPINION WORDS

In this section, we first illustrate the problem of feature level opinion mining. Then an association approach based on mutual reinforcement between product feature categories and opinion word groups is proposed. Under the framework, for improving the performance of association and product feature category construction, we propose to utilize multi-source knowledge including semantic and textual structure to enhance the algorithm.

3.1 The Problem

In product reviews, opinion words are used to express opinion, sentiment or attitude of reviewers. Although some review units may express general opinions toward a product, most review units are regarding to specific features of the product.

A product is always reviewed under a certain feature set $\mathcal{F}$. Suppose we have got a lexical list $\mathcal{O}$ which includes all the opinion expressions and their sentiment polarities. For the feature level opinion mining, identifying the sentiment association between $\mathcal{F}$ and $\mathcal{O}$ is essential. The key points in the whole process are as follows:

- get opinion word set $\mathcal{O}$ (with polarity labels)
- get product feature set $\mathcal{F}$
- identify relationships between $\mathcal{F}$ and $\mathcal{O}$

The focus of the paper is on the latter two steps. We propose an unsupervised approach to deal with the tasks. Given a product review, existing approaches identify the association between product feature words and opinion words in the review by their explicit adjacency. But for our approach, the association set is pre-constructed. The proposed approach detects the sentiment association between $\mathcal{F}$ and $\mathcal{O}$ based on product feature word categories and opinion word groups gained from the review corpus. A mutual reinforcement principle is developed to solve the task. Meanwhile, we perform clustering optimization under the unified framework. Generally speaking, the benefits of our approach are threefold.

- It groups the product feature terms in reviews if they have similar meaning or refer to the same topic. Thus it can provide users a more sound and non-trivial opinion evaluation.
- Based on a pre-constructed association set, our approach is effective in finding the implicit product features, and well fit for online applications.
- Also, we can largely identify the related explicit product features which an opinion word is attached in reviews. What is more, the approach is easy to be combined with the existing explicit adjacency approaches to optimize the performance.

3.2 Associate Product Feature Categories and Opinion Word Groups By Mutual Reinforcement

We first consider two sets of association objects: the set of product feature words $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$ and the set of opinion words $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$. A weighted bipartite graph from $\mathcal{F}$ and $\mathcal{O}$ can be built, denoted by $\mathcal{G}(\mathcal{F}, \mathcal{O}, \mathcal{R})$. Here $\mathcal{R} = [r_{ij}]$ is the $m \times n$ link weight matrix containing all the pairwise weights between set $\mathcal{F}$ and $\mathcal{O}$. The weight can be calculated with different weighting schemes. For example, if a product feature word f_i and an opinion word o_j co-occur in a sentence, we set the weight $r_{ij} = 1$, otherwise $r_{ij} = 0$. In this paper, we set r_{ij} by the co-appearance frequency of f_i and o_j in clause level. The main idea of association approach is shown as figure 2.

The co-appearance of product feature words and opinion words may be incidental in review corpus and without essential semantic relatedness. Meanwhile, for the real se-

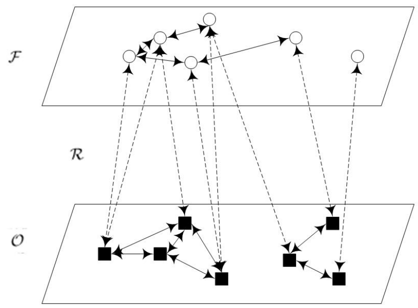

Figure 2: Association Approach Using the set of Product Feature Words $\mathcal{F}$ and the set of Opinion Words $\mathcal{O}$

mantic relatedness between product feature words and opinion words, the co-appearance may be quantitatively sparse. Statistics based on word-occurrence loses semantic related information. While by clustering, we can organize product feature words or opinion words if they have similar meaning or refer to the same concept. So the judgement of association can be more effective if it is applied with product feature categories and opinion word groups. The association set between product features and opinion word groups will be constructed according to the interrelated pairwise weight between the two types of object groups.

To form the two kinds of groups, the general approach is to cluster the objects in $\mathcal{F}$ and $\mathcal{O}$ separately. However, the two types of objects are highly interrelated. It is obvious that surrounding opinion words play an important role in clustering product feature words. Similarly, when clustering opinion words, the product feature words co-occurred should also be important. So we consider both *intra* relationship from single type homogeneous data objects and *inter* relationship from different type interrelated data objects. This updated relationship space is utilized to perform clustering on those related types of objects in set $\mathcal{F}$ and $\mathcal{O}$.

The purpose of clustering data objects is to partition each object into one cluster so that objects in the same cluster have high similarity, and objects from different clusters are dissimilar. Using the updated relationship space, the similarity between two objects of the same type is defined as:

$$\begin{gathered} S(X_i, X_j) = \alpha S_{intra}(X_i, X_j) + (1-\alpha) S_{inter}(X_i, X_j) \\ where, \{X_i \in \mathcal{F} \wedge X_j \in \mathcal{F}\} \vee \{X_i \in \mathcal{O} \wedge X_j \in \mathcal{O}\} \end{gathered} \tag{1}$$

In equation 1, the similarity between two data objects X_i and X_j is denoted as a linear combination of *intra* similarity and *inter* similarity. The parameter α reflects the weight of different relationship spaces. $S_{intra}(X_i, X_j)$ is the similarity of homogeneous data object X_i and X_j calculated by traditional approach. This kind of similarity can be considered based on the *content information* between two data objects. While $S_{inter}(X_i, X_j)$ determines the similarity of homogeneous data object X_i and X_j by their respective heterogeneous relationships, which are based on the degree of interrelated association between product features and opinion words. It can be considered based on the *link information* between two data objects. For example, suppose $X_i \in \mathcal{F}$, the interrelated relationship feature of data object X_i is represented as $\mathcal{R}_i = [r_1^{(i)}, r_2^{(i)}, \ldots, r_n^{(i)}]^T$. And the interrelated relationship feature of X_j ($X_j \in \mathcal{F}$) is represented as $\mathcal{R}_j = [r_1^{(j)}, r_2^{(j)}, \ldots, r_n^{(j)}]^T$. $r^{(i)}$ and $r^{(j)}$ are the entries in the link weight matrix $\mathcal{R}$ of product feature set $\mathcal{F}$ and opinion word set $\mathcal{O}$. Then, the inter similarity between X_i and X_j can be calculated by:

$$S_{inter}(X_i, X_j) = \cos(\mathcal{R}_i, \mathcal{R}_j) \tag{2}$$

The basic idea of the mutual reinforcement principle is to propagate the clustered results between different type data objects by updating their inter- relationship spaces, that is, the link information between two data object groups. The clustering process can begin from an arbitrary type of data object. The clustering results of one data object type update the link information thus reinforce the data object categorization of another type. The process is iterative until clustering results of both object types converge. Suppose we begin the clustering process from data objects in set $\mathcal{F}$, then the steps can be expressed as follows.

---

step 1. Cluster the data objects in set $\mathcal{F}$ into k clusters according to the intra relationship;
step 2. Update the interrelated relationship space of data objects in set $\mathcal{O}$. $\forall\ X_i$ ($X_i \in \mathcal{O}$), the interrelated relationship feature is replaced with $\mathcal{R}'_i = [u_1^{(i)}, u_2^{(i)}, \ldots, u_k^{(i)}]$. Where $u_x(x \in [1, k])$ is an updated pairwise weight with each component in the vector corresponding to one of the k clusters of $\mathcal{F}$ layer;
step 3. Cluster the data objects in set $\mathcal{O}$ into l clusters based on the updated inter-type relationship space;
step 4. Update the interrelated relationship space of data objects in set $\mathcal{F}$. $\forall\ Y_i$ ($Y_i \in \mathcal{F}$), the interrelated relationship feature is replaced with $\mathcal{R}'_i = [v_1^{(i)}, v_2^{(i)}, \ldots, v_l^{(i)}]^T$. Where $v_x(x \in [1, l])$ is an updated pairwise weight with each component in the vector corresponding to one of the l clusters of $\mathcal{O}$ layer;
step 5. Re-cluster the data objects in set $\mathcal{F}$ into k clusters based on the updated inter-type relationship space;
step 6. Iterative the steps 2-5 until clustering results in both object types converge.

In the procedure, a basic clustering algorithm is needed to cluster objects in each layer based on the defined similarity function (equation 1). In the first step of iterative reinforcement, we cluster data objects only by their intra relationship without interrelated link information, since in most cases link information is too sparse in the beginning to help the clustering [23]. Then both intra- and inter- relationships are combined in the subsequent steps to iteratively enhance reinforcement.

After the iteration, we can get the strongest n links between product feature categories and opinion word groups. That constitutes our set of sentiment association.

3.3 Product Feature Category Optimization Based on Semantic and Textual Structural Knowledge

In the process of mutual reinforcement, any traditional clustering algorithm can be easily embedded into the iterative process, such as the K-Means algorithm[12] and other state-of-art algorithms. Take the plain K-Means algorithm as example, it is an unsupervised learning based on iterative relocation to partition a dataset into k clusters of similar datapoints, typically by minimizing an objective function of average squared distance. The algorithm utilizes the constructed instance representation to conduct the process of clustering. As an unsupervised learning, its performance is usually not comparable with supervised learning. However, the performance of mutual reinforcement of multi-type data objects is effected by the embedded clustering. Usually, background knowledge about the application is useful in clustering. If we add more background knowledge for the clustering algorithm, we may expect to get a better clustering result.

Our basic idea of clustering enhancement by background knowledge comes from COP-KMeans. COP-KMeans [20] is a semi-supervised variant of K-Means. Background knowledge, provided in the form of constraints between data objects, is used to generate the partition in the clustering process. Two types of constraints are used in COP-KMeans, including:

Compatibility. two data objects have to be in the same cluster.

Incompatibility. two data objects must not to be in the same cluster.

The compatibility and incompatibility constraints in COP-KMeans are checked by human labeler. Here we employ a clustering optimization method in which background knowledge are extracted automatically from several knowledge resources. Then we construct the knowledge-based constraints to improve the primary clustering similarity measure based on content information.

• **Semantic Class.** We use a WordNet-like semantic lexicon, *Chinese Concept Dictionary*(*CCD*)[11], to obtain coarse semantic class information for each data object. Generally speaking, the noun network is richly developed in most of electronic lexicon like WordNet. Comparing with nouns, researches on semantic relatedness using WordNet performed far worse for words with other part-of-speeches [1]. And in our research, the extracted product feature words are included in the set of nouns and noun phrases (see section 4.2). So we only generate constraints based on semantic relatedness of nouns. There are totally 25 semantic class tags in *CCD*. We use the noun part to provide semantic class constraints for clustering enhancement.

Two kinds of automatic constraint generation strategies are proposed. First, some words may belong to multi semantic classes simultaneously. For each such word, set $\mathbf{A}$ is generated by pairing any two of the elements in its semantic class set. $\bigcup_{word} \mathbf{A}$ denotes the complete set of all the word with multi semantic classes. By pairing any two of all the semantic classes, we get a set $\mathbf{B}$. Then the incompatibility table is constructed by the difference set of $\mathbf{B}$ and $\bigcup_{word} \mathbf{A}$.

In addition, we utilize the information of the common father node of two instances. If we cannot find their common father node in the semantic lexicon or the level of their common father node is too low, e.g. in the first level of the lexicon, we consider the two instances incompatible.

• **Textual Structure.** The semantic class information of an data object is context-independent. However, the context-dependent information is also useful to construct constraints. In general, paragraph is a collection of related sentences with a single focus, which locates a rough semantic boundary. Semantic coherence usually can be assessed within a paragraph. Our observation of product review corpus largely meet the point. For example, for an editor review on automobile, reviewers may usually present their opinions on the *power* of the automobile in a paragraph, followed by their opinions on the *appearance* in another paragraph. That's a common case in reviews on kinds of products. So we propose to calculate the textual structure based similarity between product feature word X_1 and X_2 by their paragraphic co-occurrence. It is denoted by equation 3.

$$\mathrm{sim}(X_1, X_2) = \frac{\mathrm{pf}(X_1, X_2)}{\mathrm{pf}(X_1) \times \mathrm{pf}(X_2)} \times \frac{\sum_N \frac{N_p}{\mathrm{pf}_{\mathrm{doc}}(X_1, X_2)}}{N} \times \mathrm{df}(X_1, X_2) \tag{3}$$

Here $\mathrm{pf}(w)$ is the paragraphic frequency of word w by counting the number of paragraphs in the corpus containing word w. $\mathrm{pf}_{\mathrm{doc}}(w)$ is the paragraphic frequency of word w within a document. N denotes the total number of documents in the corpus. While N_p is the number of paragraphs in a document. The equation indicates the similarity between two words according to their positional relationship based on paragraph structure. Utilizing their similarity, we augment the distance metric between the two data objects with a weighting function according to equation 4.

$$\begin{aligned}\mathrm{dist}'(X_1, X_2) = \mathrm{incom}(X_1, X_2) \times \left(1 - \log^{-1} \mathrm{sim}(X_1, X_2)\right) \\ \times \mathrm{dist}(X_1, X_2)\end{aligned} \tag{4}$$

The first two items denote the constraints which incorporate prior knowledge from both universal language resources and corpus. They alter the original distance measure $\mathrm{dist}(X_1, X_2)$ for the embedded clustering algorithm. $incom(X_1, X_2) = 0$ represents the semantic class based incompatibility of two data objects X_1 and X_2. Since they are incompatible, we can first rule out of these impossible matches. The item of $sim(X_1, X_2)$ increases or decreases the similarity measure of the original vector distance according to their paragraphic distribution features.

4. SYSTEM ARCHITECTURE

Based on the approach proposed in section 3, we construct a feature-level opinion mining system to conduct sentiment analysis on Chinese web reviews. Some modules in our system are considered based on the specialty of Chinese language, including product feature extraction & filtering, named entity identification and etc. While in essence, the main proposed approach in this paper is language independent. It is easy to adapt our system to different applications.

4.1 Architecture

The architecture of our approach is illustrated (see Figure 3) in this section. Given a specific product topic, the system first crawls the related reviews and puts them in the review database. Then parsing is conducted, including splitting review texts into sentences/clauses, Chinese word segment and part-of-speech tagging. After that, candidate product feature words and opinion words are extracted from reviews. Then we prune the candidates to generate the set of product feature words. The product feature words and opinion words are represented by Vector Space Model. According to the representation, we conduct the mutual reinforcement approach to construct product feature categories and realize sentiment association. Using the pre-constructed sentiment

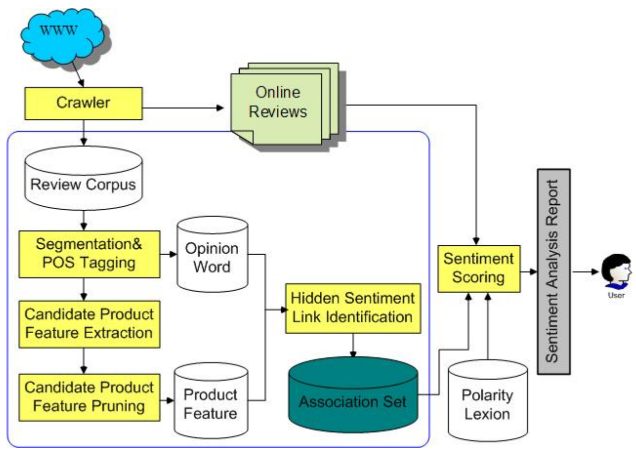

Figure 3: Architecture of The System

association set, we can then deal with feature-level opinion mining effectively.

Below, we discuss the steps of candidate product feature extraction and pruning, followed by their representation for the mutual reinforcement based sentiment association.

4.2 Candidate Product Feature Word Extraction and Pruning

Usually, adjectives are normally used to express opinions in reviews[6]. Therefore, most of the existing researches take adjectives as opinion words. In the research of[7, 8], they proposed that other components of a sentence are unlikely to be product features except for nouns and noun phrases. In the paper of [4], they targeted nouns, noun phrases and verb phrases. The adding of verb phrases caused the identification of more possible product features, while brought lots of noises. So in this paper, we follow the points in[7, 8], extract nouns and noun phrases as candidate product feature words.

In grammatical theory, a noun phrase consists of a pronoun or noun with any associated modifiers, including adjectives, adjective phrases, adjective clauses, and other nouns. Since many adjectives are evaluative indicators, we do not want to include the components in our candidate product features. For our extraction, two or more adjacent nouns are identified as the candidate "noun phrases". The strategy is effective for some cases. While it may bring many noises.

The noises may come from two aspects. 1) Some candidates may not be the integrated phrases. 2) It's obvious that not all the nouns or noun phrases could be product feature words. We propose methods to prune the candidate product feature words from the two aspects.

The BD (boundary dependency) algorithm is proposed to verify the phrase boundary of candidates. The definition of BD is shown as equation 5.

$$\mathrm{BD}(w_1 \ldots w_n) = \frac{f(\mathrm{bdw} + w_1 \ldots w_n) f(w_1 \ldots w_n + \mathrm{bdw})}{f(w_1 \ldots w_n)^2} \quad (5)$$

In the equation, $w_1 \ldots w_n$ denotes an extracted adjacent noun in the specific product reviews. $f(w_1 \ldots w_n)$ is its frequency. To avoid data sparseness and get a more reliable frequency statistic, we use the number returned by a search engine query to estimate the frequency, instead of our existing corpus (We use Google in our experiment). The BD method is proposed based on the following consideration: some specific adjacent words or characters indicate a rough phrasal boundary. Such as "de" (*'s*) in Chinese. We name these words boundary indicators (bdw). In addition, some words usually cannot be prefix word or suffix word of noun phrases. The above two points can help to determine the phrasal boundaries. If boundary indicators appear on the left and right of a noun phrase and the BD is higher than a threshold δ, we consider it correct noun phrase. Whereas, if impossible prefix or suffix words/characters appear, we judge the extracted phrases is not completed ones.

Some completed noun phrases and nouns may not be real product features. Such as *car, BMW, driver...* in automobile reviews. We filter out part of the non-product feature words by their sense. Named Entity Recognition (NER) is utilized in the process. We use a NER system developed by IBM[5]. The system can recognize four types of NEs: person (PER), location (LOC), organization (ORG), and miscellaneous NE (MISC) that does not belong to the previous three groups (e.g. products, brands, conferences etc.). Since the NEs have little probability of being product features, we prune the candidate nouns or noun phrases which have the above NE taggers.

By the pruning of candidate product feature words, we get the set of product feature words $\mathcal{F}$. And the set of opinion words $\mathcal{O}$ is composed by all the adjectives in reviews.

4.3 Representation of Product Features and Opinion Words for Sentiment Association

Product feature words and opinion words are clustered respectively in the iterative process of mutual reinforcement. To conduct the procedure, we represent each data object instance(including product feature word and opinion word) by a feature vector and then conduct clusterings and the mutual reinforcement. Data from online customer product reviews are preprocessed in several steps, including sentence segmentation, stop words elimination and etc. Then, we get the second-order substantival context of each product feature instance and opinion word instance in reviews, say, the [-2, +2] substantival window around the instance after stop words elimination. The context is requested to be in the same clause of the instance. We represent an instance as a set of following features.

- Pointwise Mutual information (PMI) between the instance and its context.
- For phrases, we also calculate the inner word PMI within the phrases.
- Part-of-speech tagger of the context is another feature we used in the instance representation.

Eg. for the noun phrase "*battery life*" in the sentence "*The battery life of this camera is too short.*", the instance's [-2, +2] substantival window should be [NULL, NULL, *camera*, *short*]. The inner words are "*battery*" and "*life*".

Let w_1, w_2 be two words or phrases. The pointwise mutual information[18] between w_1 and w_2 is defined as:

$$\mathrm{PMI}(w_1, w_2) = \log \frac{\mathrm{P}'(w_1, w_2)}{\mathrm{P}(w_1)\mathrm{P}(w_2)} \quad (6)$$

where $\mathrm{P}(w_1)$ and $\mathrm{P}(w_2)$ are the frequency of w_1 and w_2 in the corpus. While $\mathrm{P}'(w_1, w_2)$ is the co-occurrent frequency of w_1 and w_2 in a certain position. For example, when calculating the inner word PMI within a phrase, $P'(w_1, w_2)$ denotes the co-occurrence frequency of w_1 and w_2 within a phrase's range.

Although mutual information weight is biased towards infrequent words [14], it can utilize more relatedness and restriction than other weight settings such as instance's document frequency (DF) and etc. So we represent the instances by the PMI weight in this research.

5. EXPERIMENT AND EVALUATION

We evaluate our approach from three perspectives: 1) effectiveness of product feature category construction by mutual reinforcement based clustering; 2) precision of sentiment association between product feature categories and opinion word groups; 3) performance of our association approach to apply for feature level opinion mining.

5.1 Data

Our experiments take automobile reviews (in Chinese) as example. The corpus used in the experiments is composed by 300 editor reviews on automobile, including 806,923 Chinese characters. They are extracted from several specialized auto review websites. Editor reviews are usually long in length, so a completed editor review may be distributed over multiple web pages. For our corpus, the largest number of distribution is 14 web pages. The number of candidate product feature words and opinion words extracted from the corpus are shown as Table 2.

Table 2: Number of Candidate Product Features and Opinion Words in Our Corpus

Extracted Instance	Total	Non-Repetitive
Candidate Product Feature	89,542	18,867
Opinion Word	27,812	1,343

We use both the BD algorithm and NER based method to prune the candidate product features. Precision and Recall for the pruning strategy is shown in table 3

Table 3: Results of Candidate Product Feature Filtering Using Different Pruning Strategy

Pruning Strategy	Precision (P)	Recall (R)	Number of Remained Product Features
BD_{np}	78.94%	90.73%	9,389
$BD_{feature}$	47.11%	88.57%	9,389
+NER	52.49%	86.48%	7,660

We use two pruning strategies on candidate product feature words. The BD algorithm is effective to locate phrase boundary, thus identify correct noun phrases. Its performance in identifying noun phrases is shown in table 3 as BD_{np}. However, it cannot be used to judge whether a noun phrase is a product feature (shown as $BD_{feature}$). Named entity tagging helps to filter out noisy candidates, but does not show significant improvement. In fact, finding real product feature words in reviews is still an issue in related researches. Most of existing research just simply use nouns and noun phrases as candidate product features, then conduct frequency based filtering. This problem will be studied in our future research. Actually, since we choose the strongest n links between product feature categories and opinion word groups to construct the sentiment association set, part of the noisy candidates may be excluded in the process.

To conduct evaluations, we pre-construct an evaluation set. The extracted product feature words and opinion words are checked manually. If the word satisfies the specification of some automobile review categories, we give it the relevant labels. A word may have multiple labels. For example, the word "*color*" may be associated with both *"exterior"* and *"interior"*. In our labeled set, the average number of labels for product feature words is 1.135. The average label number per opinion word is 1.556. We utilize the set to conduct evaluations on both product feature categorization and sentiment association.

5.2 Evaluation of Product Feature Category Construction

The performance of product feature categorization is evaluated using the measure of Rand index [2, 20]. In equation 7, P_1 and P_2 respectively represents the partition of an algorithm and manual labeling. The agreement of P_1 and P_2 is checked on their $n*(n-1)/2$ pairs of instances, where n is the size of data set D. For each two instances in D, P_1 and P_2 either assigns them to the same cluster or to different clusters. Let a be the frequency where pairs belong to the same cluster of both partitions. Let b be the frequency where pairs belong to the different cluster of both partitions. Then the Rand index is calculated by the proportion of total agreement.

$$\mathrm{Rand}(P_1, P_2) = \frac{2(a+b)}{n \times (n-1)} \tag{7}$$

The parts of product feature words in the pre-constructed evaluation set are used to represent the data set D. Partition agreements between the pairs of any two words in the parts and in the clustering results are checked automatically.

Our approach of mutual reinforcement can easily integrate any traditional clustering algorithm. The parameter α reflects the relative importance of content information and link information in the iteration. $\alpha = 0$ denotes only link information is utilized. When $\alpha = 1$, the approach is similar to traditional content-based clustering. Those can be taken as the baselines.

We fix several value of parameter α ($\alpha \in [0, 1]$, stepped by 0.2) to conduct the experiments. Figure 4 shows the clustering results by different parameter α. We can find from the results that the iterative mutual reinforcement achieves a higher performance than both the content-based ($\alpha = 1$) and link-based ($\alpha = 0$) approach. The reason for the improvement lies in the fact that the mutual reinforcement approach can fully exploit the relationship between product features and opinion words. Comparing with the two parts, the content-based ($\alpha = 1$) method gets a higher performance than the link-based ($\alpha = 0$) method. The improvement is also in that the former utilizes more context information than the latter.

A comparative experiment is conducted to show the impact of background knowledge on the clustering quality. Figure 5 shows the performance of similar experiment settings as in figure 4 but without introducing background knowledge. Seen from figures 4 and 5, the approach utilizing background knowledge get higher precision than the approach

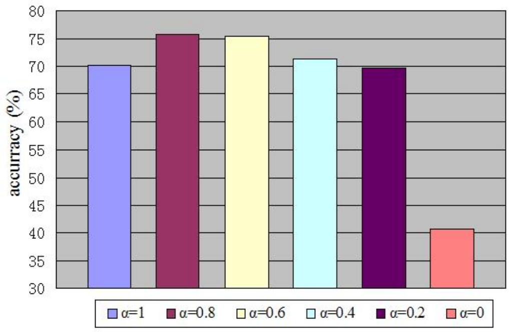

Figure 4: Performance of Product Feature Categorization by the Iterative Mutual Reinforcement Approach

without it. In addition, the proposed approach of combining both content information and link information between two data objects always outperforms the two baselines, which use either content information ($\alpha = 1$) or link information ($\alpha = 1$) in the two experiment settings. The two groups of experiment results get their best results at $\alpha > 0.5$. So, although link information can help to improve clustering per- forr ı the

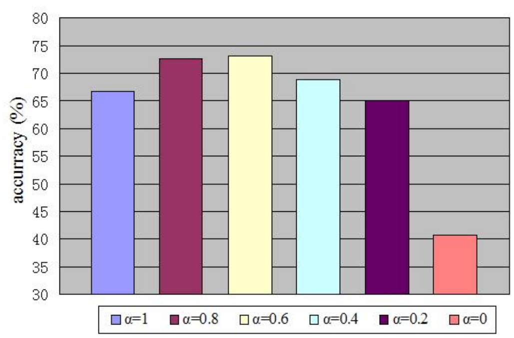

Figure 5: Performance of Product Feature Categorization by the Iterative Mutual Reinforcement Approach (Without Background Knowledge)

5.3 Evaluation of Sentiment Association

In the process of iterative mutual reinforcement between product features and opinion words, clusterings of both data objects converge with iteration. Simultaneously, the interrelationship information between product feature categories and opinion word groups tend to be stable.

We evaluate the association set by the precision measurement. A comparable evaluation is made on the original extracted pairs based on the explicit adjacency. Since our purpose is to find the association between opinion words and product feature categories, both of the two evaluations utilize the product feature categories generated by the same grouping method.

For a detected pair of product feature word and opinion word by the explicit adjacency or our association approach, we first judge the category which the product feature word belongs to. The headword (word with the highest frequency) of a product feature category is used to represent the category. We check the labels of the headword and the detected opinion word in the pre-constructed evaluation set. If the two labels are different, we judge the detected pair as illegal. Otherwise, there is a logical sentiment association in the pair. We define the precision as:

$$\text{precision} = \frac{\text{number of correctly associated pairs}}{\text{number of detected pairs}} \tag{8}$$

Precision measures the proportion of correct sentiment association in the detected pairs. Since we use the same product feature grouping result on both evaluations of explicit adjacency and our association approach, it does not skew the evaluation comparison. The precisions are calculated on the pre-constructed evaluation set. So we did not check all the detected association pairs. Only the pairs which product feature word parts and opinion word parts are in the evaluation set are checked.

Table 4: Impact of Sentiment Association By Explicit Adjacency and Mutual Reinforcement Approach

Approach	Detected Number	Precision
Explicit Adjacency	28,976	68.91%
Association Approach	294,965	81.90%

Table 4 shows the advantage of our association approach over the extraction by explicit adjacency. Using the same product feature categorization, our sentiment association approach get a more accuracy pair set than the direct extraction based on explicit adjacency. The precision we obtained by the mutual reinforcement approach is 81.90%, almost 13 points higher than the adjacency approach. Number of detected association by our approach shows its ability to finding hidden sentiment association.

5.4 Evaluation of Opinion Mining Relating to Different Product Features

We use a new test corpus to evaluate the ability of our association approach on feature level opinion mining. The corpus is composed by 50 automobile reviews(161,205 characters). In the reviews, automobile review features are rated on a 5 star scale (in half star increments) respectively. There is usually large variation in the sentiment scoring criterion for different automobile websites. We extract automobile reviews from the same websites for keeping a consistent scoring system. To validate the usefulness of the hidden sentiment link identification in the feature level opinion mining, we design the following experiments to predict sentiment on different product features in our test corpus:

• **By the Explicit Adjacency:** for a product feature word in reviews, we first find its nearest neighboring opinion word within the clause. The distance between a product feature word and its nearest opinion word may be equal for the two conditions of left adjacency and right adjacency. So we try out two sentiment attachment strategies for the case.
- **Left Adjacency First:** We attach the nearest opinion word in its left context to the product feature word. The setting is denoted by "adjacency(L)" in figure 6.
- **Right Adjacency First:** We attach the nearest opinion

word in its right context to the product feature word. The setting is denoted by "adjacency(R)" in figure 6.

We check a polarity lexicon for the sentiment polarity of the opinion word, and attach the sentiment polarity to the feature category which the product feature word belongs to. The sentiment strength for a feature category is obtained by summing up all the attached sentiment orientation with the category.

• **By the Pre-constructed Association Between Product Feature Categories and Opinion Word Groups:** by our association approach, we have constructed a sentiment association set between product feature categories and opinion word groups. So in this experiment setting, we evaluate feature-oriented sentiment without using product feature words in reviews. Utilizing the association set, we directly attach the sentiment orientation of an opinion word to it related product feature category. The sentiment strength for a feature category is obtained by summing up values of each related sentiment orientation. If an opinion word is associated with several product feature categories, we attach its sentiment orientation to all the related product feature categories. The approach is denoted as "our approach" in the experimental figure.

• **By the Combination of Explicit Adjacency and Pre-constructed Association:** we evaluate the combination of both explicit adjacency and pre-constructed association set. If there are opinion words but no product feature words in a clause, we attach the sentiment orientation to the related product feature category by our association set. Otherwise, if a clause includes both kinds of words, we attach the sentiment orientation to the related product feature category by the adjacency. Similarly, we try out two strategies to identify the nearest neighbors of opinion words and product feature words.

- **Left Adjacency First**, denoted by "combination(L)"
- **Right Adjacency First**, denoted by "combination(R)"

As we have mentioned in section 2, three key points in the feature level opinion mining are opinion word list, product feature category and their association. In this paper, we deal with the latter two tasks. To evaluate the sentiment orientation of opinions, we have constructed a polarity lexicon. The lexicon consists of 1,000 opinion words with polarity labels as 1 (positive) or -1 (negative). We predict sentiment strength for different product features in reviews by adding the polarity of the related opinion words. The semantic relatedness between product features and opinion words is judged with the above mentioned methods.

In our test corpus, product features involved in each review are rated on a 1-5 star scale rating. 1 star is the lowest rating of positive sentiment; 5 stars is the highest one in the rating system. We compare the relative ranking of different scoring methods with the standard answer set. For each product feature, we rank the 50 reviews according to their sentiment evaluation on the product feature. Then the corresponding ranking is extracted from the standard evaluation set. We check the coincidence between a generated ranking with the standard ranking. Given a reviewed product feature f_j and a review set $\mathcal{X}$ which is composed by n product reviews, we can get a ranked review sequence of Ranking$(f_j, \mathcal{X})$. The sequence is obtained according to their sentiment strength for f_j. We use Ranking$(f_j, \mathcal{X})_i$ $(i < n)$ to denote the i position in the ranking. If a generated ranking has the same member with the standard ranking in their Ranking$(f_j, \mathcal{X})_i$, it is considered having a correct output in the position. We measure the ratio of correct output with the length of a ranking. For our experiments, the ranking length is the same as the number of product reviews in the test corpus.

Figure 6 shows the ranking precision of the 50 reviews on different product features.

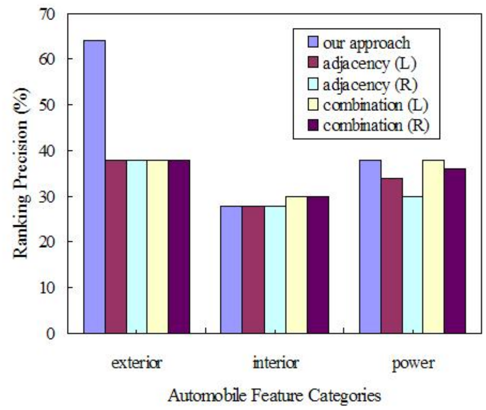

Figure 6: Ranked Sentiment Strength of Reviews on Different Product Feature Categories

From the figure, we see a remarkable effect of our association set for identifying sentiment related to the product feature of "*exterior*". A similar but not so significant effect can be seen on the product feature of *"power"*. For the product feature of "*exterior*", We can get a quite more accurate ranking by our pre-constructed association set than both the adjacency method and the combination method. We believe that the advantage comes from the ability of our approach to identify implicit product features. Product feature words which have the similar meaning of *"exterior"* are usually implicitly expressed in reviews. People seldom explicitly use the words like *"exterior"* to comment the appearance aspect of a product or other topics. They just have comment on the exterior of a thing by saying "*it's beautiful, elegant...*" or something like that. So our association approach can get an amazing performance on the sentiment evaluation of this kind of product feature. For the automobile feature of *"interior"*, our association approach shows a little worse performance than the adjacency based shallow extraction. Through a checking of the corpus, we find it's a common case that people review the product feature by lots of explicit feature words, such as *"seat"*, *"acoustics"* and so on. If the related opinion is expressed in such a sentence like *"The acoustics is excellence."*, our approach is less effective than the approach by explicit adjacency.

For all the product features, the combination approach always get better performance than both the adjacency methods. That shows the contribution of our pre-constructed association set. The set can provide hidden sentiment identification to help in getting a more accurate feature level opinion mining.

6. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel algorithm to deal with the feature-level product opinion mining problem. An unsupervised approach based on the mutual reinforcement principle is proposed. The approach clusters product features and opinion words simultaneously and iteratively by fusing both their content information and link information. Based on the clusterings of the two interrelated data objects, we construct an association set between product feature categories and opinion word groups by identifying the strongest n sentiment links. Thus we can exploit the sentiment association hidden in reviews. Moreover, knowledge from multi-source is used to enhance clustering in the procedure. Our approach can largely predict opinions relating to different product features, even for the case without explicit appearance of product feature words in reviews. The experimental results based on real Chinese web reviews demonstrate that our method outperforms the state-of-art algorithms.

Although our methods of candidate product feature extraction and filtering can partly identify real product features, it may lose some data and remain some noises. We'll conduct deeper research in this area in future works.

7. ACKNOWLEDGMENTS

The work described here was supported in part by PKU-IBM Innovation Institute and the National Natural Science Foundation of China (Project No. 60435020).

8. REFERENCES

[1] A. Budanitsky and G. Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[2] C. Cardie, K. Wagstaff, and et al. Noun phrase coreference as clustering. In *Proceedings of the Joint Conf on Empirical Methods in NLP and Very Large Corpora*, pages 82–89, 1999.

[3] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of 12th International Conference on the World Wide Web (WWW'03)*, pages 519–528, 2003.

[4] A. Fujii and T. Ishikawa. A system for summarizing and visualizing arguments in subjective documents: Toward supporting decision making. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text, ACL2006*, pages 15–22, 2006.

[5] H. Guo, J. Jiang, G. Hu, and T. Zhang. Chinese named entity recognition based on multilevel linguistic features. *Lecture Notes in Artificial Intelligence*, 3248:90–99, 2005.

[6] V. Hatzivassiloglou and K. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the ACL and the 8th Conference of the European Chapter of the ACL*, pages 174–181, 1997.

[7] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2004)*, pages 761–769, 2004.

[8] M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proceedings of Nineteeth National Conference on Artificial Intellgience (AAAI-2004)*, pages 755–760, 2004.

[9] S.-M. Kim and E. Hovy. Identifying opinion holders for question answering in opinion texts. In *Proceedings of AAAI Workshop on Question Answering in Restricted Domains*, 2005.

[10] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web (WWW'05)*, pages 1024–1025, 2005.

[11] Y. Liu and et al. The CCD construction model & its auxiliary tool vacol. *Applied Linguistics*, 45(1):83–88, 2003.

[12] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[13] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of 43nd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 2005.

[14] P. Pantel and D. Lin. Discovering word senses from text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 613–619, 2002.

[15] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of Human Language Technology Conference/Conference on Empirical Methods in Natural Language Processing(HLT-EMNLP -05)*, Vancouver, CA, 2005.

[16] Q. Su, Y. Zhu, B. Swen, and S.-W. Yu. Mining feature based opinion expressions by mutual information approach. *International Journal of Computer Processing of Oriental Languages*, 20(2/3):137–150.

[17] J.-T. Sun, X. Wang, D. Shen, H.-J. Zeng, and Z. Chen. CWS: a comparative web search system. In *Proceedings of the 15th international conference on World Wide Web*, 2006.

[18] M. C. Thomas and J. A. Thomas. *Elements of Information Theory*. Number 10. 1991.

[19] P. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 417–424, 2002.

[20] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, 2001.

[21] G.-R. Xue, Y. Yu, D. Shen, Q. Yang, H.-J. Zeng, and Z. Chen. Reinforcing web-object categorization through interrelationships. Number 12(2-3), 2006.

[22] H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of EMNLP 2003*, 2003.

[23] H.-J. Zeng, Z. Chen, and W.-Y. Ma. A unified framework for clustering heterogeneous web objects. In *Proceedings of the 3rd International Conference on Web Information Systems Engineering*, pages 161–172, 2002.