A Statistical NLP Approach for Feature and Sentiment Identification from Chinese Reviews

Zhen Hai[1] **Kuiyu Chang**[1] **Qinbao Song**[2] **Jung-jae Kim**[1]
[1]School of Computer Engineering, Nanyang Technological University, Singapore 639798
{haiz0001, askychang, jungjae.kim}@ntu.edu.sg
[2]Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China
qbsong@mail.xjtu.edu.cn

Abstract

Existing methods for extracting features from Chinese reviews only use simplistic syntactic knowledge, while those for identifying sentiments rely heavily on a semantic dictionary. In this paper, we present a systematic technique for identifying features and sentiments, using both syntactic and statistical analysis. We firstly identify candidate features using a proposed set of common syntactic rules. We then prune irrelevant candidates with topical relevance scores below a cut-off point. We also propose an association analysis method based on likelihood ratio test to infer the polarity of opinion word. The sentiment of a feature is finally adjusted by analyzing the negative modifiers in the local context of the opinion word. Experimental results show that our system performs significantly better than a well-known opinion mining system.

1 Introduction

There were 420 million Internet users in China by the end of June 2010. As a result, online social media in China has accumulated massive amount of valuable peer reviews on almost anything. Mining this pool of Chinese reviews to detect features (e.g. "手机" mobile phone) and identify the corresponding sentiment (e.g. positive, negative) has recently become a hot research area. However, the vast majority of previous work on feature detection only uses simplistic syntactic natural language processing (NLP) approaches, while those on sentiment identification depend heavily on a semantic dictionary. Syntactic approaches are often prone to errors due to the informal nature of online reviews. Dictionary-based approaches are more robust than syntactic approaches, but must be constantly updated with new terms and expressions, which are constantly evolving in online reviews.

To overcome these limitations, we propose a statistical NLP approach for Chinese feature and sentiment identification. The technique is in fact the core of our Chinese review mining system, called Idea Miner or iMiner. Figure 1 shows the architectural overview of iMiner, which comprises five modules, of which Module Ⅲ (Opinion Feature Detection) and Ⅳ (Contextual Sentiment Identification) are the main focus of this paper.

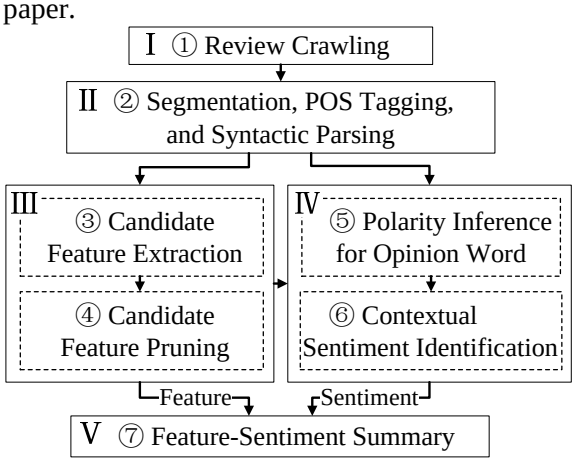

Figure 1: Overview of the iMiner system.

2 Related Work

Qiu *et al*. (2007) used syntactic analysis to identify features[1] in Chinese sentences, which is similar to the methods proposed by Zhuang *et al.* (2006) and Xia *et al.* (2007). However, syntactic analysis alone tends to extract many invalid features due to the colloquial nature of online reviews, which are often abruptly concise or

[1] A feature refers to the subject of an opinion.

grammatically incorrect. To address the issue, our approach employs an additional step to prune candidates with low topical relevance, which is a statistical measure of how frequently a term appears in one review and across different reviews.

Pang *et al.* (2002) examined the effectiveness of using supervised learning methods to identify document level sentiments. But the technique requires a large amount of training data, and must be re-trained whenever it is applied to a new domain. Furthermore, it does not perform well at the sentence level. Zhou *et al.* (2008) and Qiu *et al.* (2008) proposed dictionary-based approaches to infer contextual sentiments from Chinese sentences. However, it is difficult to maintain an up-to-date dictionary, as new expressions emerge frequently online. In contrast, to identify the sentiment expressed in a review region[2], our method first infers the polarity of an opinion word by using statistical association analysis, and subsequently analyzes the local context of the opinion word. Our method is domain independent and uses only a small set of 80 polarized words instead of a huge dictionary.

2.1 Topic Detection and Tracking

The task of Topic Detection and Tracking is to find and follow new events in a stream of news stories. Fukumoto and Suzuki (2000) proposed a domain dependence criterion to discriminate a topic from an event, and find all subsequent similar news stories. Our idea of topical relevance is related but different; we only focus on the relevance of a candidate feature with respect to a review topic, so as to extract the features on which sentiments are expressed.

2.2 Polarity Inference for Opinion Word

Turney (2002) used point-wise mutual information (PMI) to predict the polarity of an opinion word *O*, which is calculated as MI_1-MI_2, where MI_1 is the mutual information between word *O* and positive word "excellent", and MI_2 denotes the mutual information between *O* and negative word "poor". Instead of PMI, our method uses the likelihood ratio test (LRT) to compute the semantic association between an opinion word and each seed word, since LRT leads to better results in practice. Finally, the polarity is calculated as the weighted sum of the polarity values of all seed words, where the weights are determined by the semantic association.

2.3 Feature-Sentiment Pair Identification

Turney (2002) proposed an unsupervised learning algorithm to identify the overall sentiments of reviews. However, his method does not detect features to associate with the sentiments. Shi and Chang (2006) proposed to build a huge Chinese semantic lexicon to extract both features and sentiments. Other lexicon-based work for identifying feature-sentiment pair was proposed by Yi *et al.* (2003) and Xia *et al.* (2007). We propose a new statistical NLP approach to identify feature-sentiment pairs, which uses not only syntactic analysis but also data-centric statistical analysis. Most importantly, our approach requires no semantic lexicon to be maintained.

3 Feature Detection

Module Ⅲ in iMiner aims to detect opinion features, which are subjects of reviews, such as the product itself like "手机" (mobile phone) or specific attributes like "屏幕" (screen).

Example 1: "我喜欢这款手机的颜色" (I like the color of this mobile phone).

In example 1, the noun "颜色" (color) indicates a feature. Some features are expressed implicitly in review sentences, as shown below.

Example 2: "太贵了，我买不起" (Too expensive, I cannot afford it).

In example 2, the noun "价格" (price) is the opinion feature of this sentence, but it does not occur explicitly. In this paper, we do not deal with implicit features, but instead focus on the extraction of explicit features only.

3.1 Candidate Feature Extraction

According to our observation, features are generally expressed as nouns and occur in certain patterns in Chinese reviews. Typically, a noun acting as the object or subject of a verb is a potential feature. In addition, when a clause contains only a noun phrase without any verbs, the headword of the noun phrase is also a candidate. Due to the colloquial nature of online reviews, it is complicated and nearly impossible to collect all possible syntactic roles of features. Thus, we

[2]A review region is a sentence or clause which contains one and only feature.

Table 1: Dependence relations and syntactic rules for candidate feature extraction.

Relation	Rule	Interpretation	Example (3-5)
VOB	$(N, VOB) \Rightarrow (N, C)$	If term is noun (N) and depends on another component with relation VOB, extract as candidate (C).	"我喜欢这款**手机**" (I like the **mobile phone**). The noun "手机" relies on the word "喜欢" with relation VOB, thus, "手机" is extracted as candidate.
SBV	$(N, SBV) \Rightarrow (N, C)$	If term is noun (N) and depends on another component with relation SBV, extract as candidate (C).	"**屏幕**太小了" (The **screen** is too small). The noun "屏幕" depends on the word "小" with relation SBV, thus "屏幕" is extracted as candidate.
HED	$(N, HED) \Rightarrow (N, C)$	If term is noun (N) and governs another component with relation HED, extract as candidate (C).	"漂亮的**外观**" (beautiful **exterior**). The noun "外观" governs the word "漂亮" with relation HED, thus, "外观" is extracted as candidate.

only use the aforementioned three primary patterns to extract an initial set of candidates.

Dependence Grammar (Tesniere, 1959) explores asymmetric governor-dependent relationship between words, which are then combined into the dependency structure of sentences. The three dependency relations SBV, VOB, and HED correspond to the three aforementioned patterns. For each relation, we define a rule with additional restrictions for candidate feature extraction, as shown in Table 1.

Candidate features are extracted in the following manner: for each word, we first determine if it is a noun; if so, we apply the VOB, SBV, and HED rules sequentially. A noun matching any of the rules is extracted as a candidate feature.

3.2 Candidate Feature Pruning

Due to the informal nature of online reviews, a large number of irrelevant candidates are extracted by the three syntactic rules. Thus, we need to further prune them by using additional techniques.

Intuitively, candidates that are found in many reviews should be more representative compared to candidates that occur in only a few reviews. This characteristic of candidates can be captured by the topical relevance (TR) score. TR can be used to measure how strongly a candidate feature is relevant to a review topic. The TR of a candidate is described by two indicators, i.e., dispersion and deviation. Dispersion indicates how frequently a candidate occurs across different reviews, while deviation denotes how many times a candidate appears in one review. The topical relevance score (TRS) is calculated by combining both dispersion and deviation. Candidate features with high TRS are supposed to be highly relevant, while those with TRS lower than a specified threshold are rejected.

Formally, let the *i*-th candidate feature be denoted by T_i, and the *j*-th review document[3] by D_j. The weight of feature T_i in document D_j is denoted by W_{ij}, which could be computed based on *TF.IDF* (Luhn, 1957) shown in formula (1):

$$W_{ij} = \begin{cases} (1 + \log TF_{ij}) * \log \dfrac{N}{DF_i} & if\ TF_{ij} > 0 \\ 0 & otherwise \end{cases} \quad (1)$$

TF_{ij} denotes the term frequency of T_i in D_j, and DF_i denotes the document frequency of T_i; N indicates the number of documents in the corpus. We compute the standard deviation S_i:

$$S_i = \sqrt{\frac{\sum_{j=1}^{N} (W_{ij} - \overline{W}_i)^2}{N-1}} \quad (2)$$

where the average weight of T_i across all documents is calculated as follows:

$$\overline{W_i} = \frac{1}{N} \sum_{j=1}^{N} W_{ij} .$$

The dispersion $Disp_i$ of T_i is then calculated:

$$Disp_i = \frac{\overline{W}_i}{S_i} \quad (3)$$

The deviation $Devi_{ij}$ of T_i in D_j is computed:

[3] A review document refers to a forum review, which tends to be shorter than full length editorial articles.

$$Devi_{ij} = W_{ij} - \overline{W_j'} \quad (4)$$

The average scalar weight of all candidate features in D_j is calculated as follows:

$$\overline{W_j'} = \frac{1}{M}\sum_{i=1}^{M} W_{ij}$$

where M is the vocabulary size of D_j.

We can obtain the topical relevance score TRS_{ij} of T_i in D_j finally as follows:

$$TRS_{ij} = Disp_i * Devi_{ij} \quad (5)$$

By combining the dispersion and deviation, the quantity TRS_{ij} thus captures the topical relevance strength of T_i with respect to the topic of document D_j.

All candidates of a document are then sorted in descending order of TRS, and those with TRS above a pre-specified threshold are extracted as opinion features. In fact, we can extract candidates at the document, paragraph, or sentence resolution. In practice, we observe no significant performance differences at the various resolutions.

3.3 Experimental Evaluation

We collected 2,986 real-life review documents about mobile phones from major online Chinese forums. Each document corresponds to a forum topic, where each paragraph in a document matches a thread under the topic. Of these, we manually annotated the features and sentiment orientations expressed in 219 randomly selected documents, which include 600 review sentences.

To evaluate the performance of our approach, we first conducted an experiment for extracting candidate features. We then performed three other experiments for pruning the candidates at the document, paragraph, and sentence levels, respectively. For each experiment, we tried several different thresholds, i.e., percentage of TRS mean (TRSM) of all candidates. The average F-measure (F), precision (P), and recall (R) of the results at the three levels are shown in Figure 2. The highest F-measure results of feature detection with and without pruning are shown in Table 2 for easy comparison.

Table 2: Feature detection results.

Feature Detection	P (%)	R (%)	F (%)
No Pruning	71.61	90.69	80.03
Pruning (33% TRSM)	81.56	85.22	83.35

As line 2 of Table 2 shows, feature detection without pruning achieves 90.69% recall, which shows that the proposed syntactic rules have excellent coverage. However, its precision is not so promising, achieving only 71.61%, which means that many irrelevant candidates are also extracted by our rules. Thus, relying on syntactic analysis alone is not good enough, and we need to take one more step to prune the candidate features.

As line 3 of Table 2 shows, after pruning the candidate set, precision improved remarkably by 10% to 81.56%, while recall dropped slightly to 85.22%. For online review mining, precision is much more important than recall, because users' confidence in iMiner rely heavily on the accuracy of the results they see (precision), and not on what they don't see (recall).

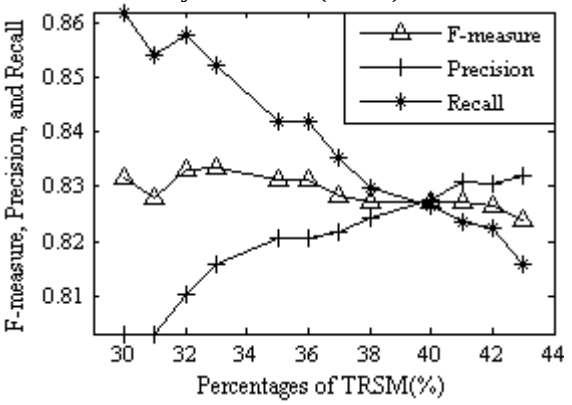

Figure 2: iMiner feature pruning results.

Figure 2 plots the results of pruning at various TRSM thresholds. The best F-measure of 83.35% was achieved with a 33% TRSM. If we increase the threshold to 43%, the precision increases to 83.19%, while the recall drops to 81.57%. By exploring the distribution of a candidate in corpus, its topical relevance with respect to the review topic can be measured statistically, which allows the noisy candidates to be pruned effectively. From the results in Figure 2, our idea of topical relevance is shown to be highly effective in detecting features.

Table 3: Characteristics of FBS and iMiner.

Aspects	**FBS**	**iMiner**
Candidates	Nouns from POS tagger	Nouns from synthetic analysis
Pruning	Association Mining	Topical Relevance
Opinion word	Adjectives	Adjectives, verbs
Polarity inference	Dictionary based	LRT association based
Sentiment Resolution	Sentence	Sentence, clause
Negation	Single	Single, double

We compared our results with that of the association mining-based method in Feature-based Summarization (FBS) (Hu and Liu, 2004) on the same dataset. Table 3 summarizes the differences between FBS and iMiner, parts of which are elaborated in Section 4. The results of FBS with various support thresholds are shown in Figure 3. The support corresponds to the percentage of total number of review sentences. FBS attained the highest F-measure of 76.35% at a support of 0.4% with 79.6% precision and 73.36% recall. As the support increases, the precision also increases from 62.99% to 86.92%, while the recall decreases from 91.61% to 61.86%. Comparing the best results of the two systems, iMiner beats FBS by 7% in F-Measure, 1.96% in precision, and 11.86% in recall.

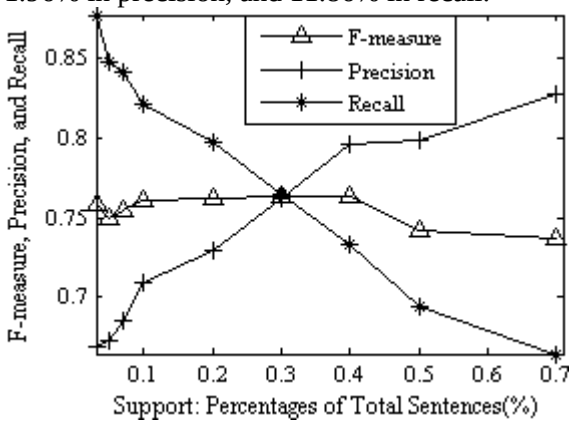

Figure 3: FBS feature extraction results.

We find that FBS suffers from the following limitations: (1) FBS extracted an additional 14.11% noisy candidate features due to the lack of syntactic analysis, which requires more extensive pruning; and (2) FBS only considers sentence frequency in computing the support to identify frequent candidate features, ignoring the candidate frequency within the sentence.

3.4 Feature Extraction Error Analysis

We categorize our feature extraction errors into 4 main types, FE1 to FE4, as follows.

FE1: When more than one candidate exists in a review region, our algorithm may pick the wrong features due to misplaced priorities. Note that we assume only one (dominant) feature per region in both our algorithm and the labeled dataset. A total of 43% errors were due to picking the wrong dominant candidate.

Example 6: “声音太小，让人听不清楚” (The sound is too weak, people cannot listen clearly).

In example 6, both “声音” and “人” are extracted as features. However, the noun “人” is an incorrect feature detected by our algorithm.

FE2: The proposed set of common syntactic rules is not comprehensive, missing out 23% of true features.

Example 7: “对于这个机子我很讨厌” (I am sick about this phone).

In example 7, the noun “机子” is a missed feature. This is in fact a POB dependence relation, which is outside the scope of our three rules.

FE3: About 22% errors are due to irrelevant features possessing high TR scores, and therefore which are not pruned subsequently.

Example 8: “我好喜欢的哦没有钱买” (I like it very much, but I have no money to buy it).

In example 8, the noun “钱” is incorrectly confirmed as a feature due to its high TR score.

FE4: About 9% errors are due to incorrect POS tags.

Example 9: “听电话时它老卡” (Consistent interruption during phone calls).

In example 9, the verb “卡” is extracted incorrectly as a feature, since it is incorrectly tagged as a noun. The remaining 3% of the errors are due to the system incorrectly extracting features from sentences that contain no opinions.

4 Contextual Sentiment Identification

The main task of module Ⅳ in iMiner is to identify the contextual sentiment of a feature. A two-step approach is proposed: (1) The polarity of an opinion word within a review region is inferred via association analysis based on the likelihood ratio test; and (2) the sentiment is validated against the contextual information of the opinion word in the region and finalized.

4.1 Polarity Inference for Opinion Word

To infer polarity, an opinion word is first identified in a review region, as described in Figure 4. Note that we consider not only adjectives but also verbs as opinion words. We then measure the association between the opinion word and each seed word. We calculate the polarity value

of the opinion word as the association weighted sum of polarities of all seed words.

Example 10: "这款机子价格很便宜" (The price of this mobile phone is very cheap).

Example 10 contains an adjective "便宜" (cheap) that governs the feature "价格" (price); thus "便宜" is extracted as an opinion word.

1. feature T_i and word W_j in the same region
2. if (W_j = adjective and depends on T_i)
3. extract W_j as opinion word;
4. else if (W_j = adjective and governs T_i)
5. extract W_j as opinion word;
6. else if (W_j = verb and governs T_i)
7. extract W_j as opinion word;

Figure 4: Extracting Opinion Word

A set of polarized words were collected from corpus as seed words, including 35 positive words, 36 negative words, and 9 neutral words. Each seed word is assigned a polarity weight from -10 to 10. For example, "漂亮" (lovely) has a score of 10, "普通" (common) has a score of 0, and "差劲" (lousy) has a score of -10.

To measure the semantic association A_{ij} between an opinion word O_i and each seed word S_j, we propose a formula based on the likelihood ratio test (Dunning, 1993), as follows:

$$A_{ij} = 2[\log L(p_1, k_1, n_1) + \log L(p_2, k_2, n_2) - \log L(p, k_1, n_1) - \log L(p, k_2, n_2)] \quad (6)$$

where

$$L(p,k,n) = p^k (1-p)^{n-k};$$

$$p = \frac{k_1 + k_2}{n_1 + n_2},\ p_1 = \frac{k_1}{n_1},\ p_2 = \frac{k_2}{n_2};$$

$$n_1 = k_1 + k_3,\ n_2 = k_2 + k_4.$$

The variable $k_1(O, S)$ in Table 4 refers to the count of documents containing both opinion word O and seed word S, $k_2(O, \bar{S})$ indicates the number of documents containing O but not S, $k_3(\bar{O}, S)$ counts the number of documents containing S but not O, while $k_4(\bar{O}, \bar{S})$ tallies the count of documents containing neither O nor S.

Table 4: Document counts.

	S	$\bar{S}$
O	$k_1(O, S)$	$k_2(O, \bar{S})$
$\bar{O}$	$k_3(\bar{O}, S)$	$k_4(\bar{O}, \bar{S})$

The higher the quantity A_{ij}, the stronger the semantic association is between the opinion word and the seed word.

The polarity value OV_i of the opinion word O_i is computed as the association weighted average of all seed word polarity values:

$$OV_i = \sum_{j=1}^{L} \frac{A_{ij}}{A_i} * SV_j \quad (7)$$

The sum A_i of all association strength is calculated as follows:

$$A_i = \sum_{j=1}^{L} A_{ij};$$

where A_{ij} denotes the association between O_i and S_j, SV_j indicates the polarity value of S_j, and L is the size of the seed word list.

After performing association analysis, we then classify the polarity value OV_i using an upper bound $V+$ and lower bound $V-$, such that if V_i is larger than $V+$, then the polarity is inferred as positive; conversely if V_i is smaller than $V-$, then the polarity is inferred as negative; otherwise, it is neutral. Here, the $V+$ and $V-$ boundaries refer to thresholds that can be determined experimentally.

4.2 Contextual Sentiment Identification

Apart from inferring the polarity of opinion words, we also examine additional contextual information around the opinion words. In fact, the final sentiment is determined by combining the polarity with the contextual information. In this work, we focus on negative modifiers, as shown in the examples below.

Example 11: "我不喜欢这款手机" (I do not like this mobile phone).

In example 11, the polarity of the opinion word "喜欢" (like) is inferred as positive, but the review region expresses a negative orientation to the feature "手机", because a negation word "不" (not) modifies "喜欢". Thus, it is important to locate negative modifiers.

Example 12: "手机屏幕不是不漂亮" (The screen of this mobile phone is not unlovely).

In example 12, the polarity of opinion word "漂亮" (lovely) is inferred as positive. By examining its direct modifier, i.e., "不" (un-), we identify the sentiment of "不漂亮" (unlovely) as negative. However, the final sentiment about the feature "屏幕" (screen) is actually positive due to the earlier negation "不是" (not), which modifies the latter "不漂亮" (unlovely). This is what we call a double negation sentence, which is not

uncommon in reviews. Therefore, it is necessary to take two additional steps to capture the double negation as follows.

Figure 5 shows the main steps of identifying contextual sentiment. For an opinion word O_i in the review region, we first determine if there exists an adverb modifying it. If so, we extract the adverb as the direct modifier. If the modifier has a negative meaning, then we reverse the prior polarity of O_i. Similarly, we can take one additional step to locate the double negation modifier and finally identify the contextual sentiment orientation.

1. for each opinion word O_i
2. if (a word W_j = adverb and depends on O_i)
3. extract W_j as direct modifier;
4. if (word W_j = negation word)
5. reverse the prior polarity of O_i;
6. if (word W_k = adverb and relies on W_j)
7. extract W_k as indirect modifier;
8. if (word W_k = negation word)
9. reverse the current polarity of O_i;
10. output the current polarity of O_i;

Figure 5: Identifying the Contextual Sentiment

4.3 Experimental Evaluation

Since features are detected prior to the sentiments, there is a possibility for an erroneous feature (i.e., a false positive feature) to be associated with a sentiment. We thus conducted two different experiments. In the first case, we enumerate all extracted feature-sentiment pairs, including the wrong features. In the second scenario, we enumerate the feature-sentiment pairs only for those correctly extracted features. For each experiment, we further evaluated the result with (C) and without (N.C.) contextual information.

We select the best case of feature detection and then run our sentiment identification algorithm on the review dataset described in section 3.3; the polarity thresholds *V*- and *V*+ are set to 0.45 and 0.5, respectively.

Table 5: Results for all features.

Systems		P (%)	R (%)	F (%)
iMiner	N.C.	57.07	58.21	57.63
	C.	**70.3**	**71.72**	**71**
FBS		49.70	45.80	47.67

Table 5 shows the results for all detected features (correct and incorrect). As shown in line 2, our method achieved an F-measure of 57.63% without considering contextual information, while precision and recall are 57% and 58.21%, respectively. Adding contextual information, as line 3 shows, boosts the F-measure to 71%, a remarkable 13.37% improvement.

Table 6 shows the results for just the correctly extracted features. As shown in line 2, in the case of not considering contextual information, our method achieved an F-measure of 63.17%, while precision and recall were 69.05% and 58.21%, respectively. By considering contextual information, line 3 shows that the F-measure improved to 77.82% which is 14.65% better, with precision and recall at 85.06% and 71.72%, respectively. The above results show that local contextual analysis of double and single negation can significantly improve the accuracy of sentiment orientation identification.

Table 6: Results for correctly detected features.

Systems		P (%)	R (%)	F (%)
iMiner	N.C.	69.05	58.21	63.17
	C.	**85.06**	**71.72**	**77.82**
FBS		62.45	45.80	52.84

By examining the results shown in line 3 (in bold) of both Tables 5 and 6, the F-measure on correctly identified features increases from 71% to 77.82%, while the precision increases drastically from 70.3% to 85.06%. The results show that our two-step approach of identifying sentiment orientation is reasonable and effective and that a great many of sentiments can be identified correctly for related features, especially for those correctly detected one. However, in practice there is no way to tell the correctly identified features from the incorrect ones, thus Table 5 is a more realistic gauge of our approach..

Lastly, we compared our approach to sentiment identification with FBS (see Table 3). The best results are used, as shown in the last rows of Table 5 and 6. When considering all features extracted, the F-measure of FBS is only 47.67%, which is 23.33% lower than that of iMiner, where both precision and recall are 49.70% and 45.80%, respectively. Considering only the correctly detected features, iMiner widens its lead over FBS to 25% in terms of F-measure.

There are several explanations for the poor results of FBS: (1) The inferior results of feature detection affect the subsequent task of sentiment identification; and (2) the polarity inference depends heavily on a semantic dictionary WordNet. In our experiments for FBS, we used an

extended version of the "同义词词林" Thesaurus containing 77,492 words, and a sentiment lexicon with 8,856 words that is part of mini (free) HowNet, and lastly our seed word list containing 80 words.

4.4 Sentiment Identification Error Analysis

We classify our sentiment identification errors into 5 main types, SE1 to SE5, as follows.

SE1: Sentiment identification relies heavily on feature extraction, which means that if features are detected wrongly, it is impossible for the sentiment identified to be correct. About 49% of false sentiments are due to incorrectly extracted features.

Even for the correctly extracted features, there are still several errors as listed below.

SE2: Incorrectly identified opinion words can lead to mistakes in inferring sentiments, accounting for 14% of the errors.

SE3: Errors in detecting contextual information about opinion words led to 12% of the wrong sentiment identification results.

SE4: Both the quality and quantity of seed words influence sentiment identification.

SE5: The threshold choices for V+ and V- directly impact the polarity inference of opinion words, affecting the sentiment identification.

SE4 and SE5 errors account for the remaining 25% of the erroneous sentiment results.

5 Conclusion

The main contribution of this paper is the proposed systematic technique of identifying both features and sentiments for Chinese reviews. Our proposed approach compares very favorably against the well-known FBS system on a small-scale dataset. Our feature detection is 7% better than FBS in terms of F-measure, with significantly higher recall. Meanwhile, our approach of identifying contextual sentiment achieved around 23% better F-measure than FBS.

We plan to further explore effective methods to deal with the various feature and sentiment errors. In addition, we plan to explore the extraction of implicit features, since a significant number of reviews express opinion via implicit features. Lastly, we plan to test out these improvements on a large-scale dataset.

Acknowledgement

We thank Harbin Institute of Technology's Center for Information Retrieval in providing their Language Technology Platform (LTP) software. This research was supported in part by Singapore Ministry of Education's Academic Research Fund Tier 1 grant RG 30/09.

References

Dunning, T. E. 1993. Accurate methods for the statistics of surprise and coincidence. Computational Linguistics 19(1).

Fukumoto, Fumiyo, and Yoshimi Suzuki. 2000. Event Tracking based on Domain Dependence, SIGIR.

Hu, Minqing, and Bing Liu. 2004. Mining and summarizing customer reviews, SIGKDD, Seattle, WA, USA.

Luhn, Hans Peter. 1957. A statistical approach tomechanized encoding and searching of literary information. IBM Journal of Research and Development 1 (4):309-17.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques, EMNLP.

Qiu, Guang, Kangmiao Liu, Jiajun Bu, Chun Chen, and Zhiming Kang. 2007. Extracting opinion topics for Chinese opinions using dependence grammar, ADKDD, California, USA.

Qiu, Guang, Can Wang, Jiajun Bu, Kangmiao Liu, and Chun Chen. 2008. Incorporate the Syntactic Knowledge in Opinion Mining in User-generated Content, WWW, Beijing, China.

Shi, Bin, and Kuiyu Chang. 2006. Mining Chinese Reviews, ICDM Data Mining on Design and Marketing Workshop.

Tesniere, L. 1959. Elements de Syntaxe Structurale: Librairie C. Klincksieck, Paris.

Turney, Peter D. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews, ACL, Philadelphia.

Xia, Yunqing, Ruifeng Xu, Kam-Fai Wong, and Fang Zheng. 2007. The Unified Collocation Framework for Opinion Mining. International Conference on Machine Learning and Cybernetics.

Yi, Jeonghee, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. 2003. Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques, ICDM.

Zhou, Chao, Guang Qiu, Kangmiao Liu, Jiajun Bu, Mingcheng Qu, and Chun Chen. 2008. SOPING : a Chinese customer review mining system, SIGIR, Singapore.

Zhuang, Li, Feng Jing, and Xiaoyan Zhu. 2006. Movie Review Mining and Summarization, CIKM.