# Expanding Chinese sentiment dictionaries from large scale unlabeled corpus

Hongzhi Xu, Kai Zhao, Likun Qiu, Changjian Hu

NEC Laboratories China.
Tsinghua Science Park, Haidian District, Beijing 100084, China.
{xu_hongzhi, zhao_kai, qiu_likun, hu_changjian}@nec.cn

**Abstract.** Unsupervised sentiment classification usually needs a user defined sentiment dictionary. However, the existing dictionaries in Chinese are insufficient, for example, the intersection rate of two popular Chinese sentiment dictionaries HowNet and NTUSD is less than 10%. In this paper, we present a method to help expand the dictionaries with more sentiment words by ranking them through link analysis based on a word graph constructed from a large unlabeled corpus. Meanwhile, our method could compute a sentiment polarity strength for each word in the new dictionaries. Manual evaluation has shown that our method has a high precision to expand the dictionaries. Experiments for sentiment classification have shown that the new dictionaries with the polarity strength for each word given by our algorithm are effective to improve the performance. As a byproduct, our algorithm could also discover the errors existing in current dictionaries.

## 1 Introduction

Sentiment analysis becomes more and more important as various user generated content(UGC) appears on the web, such as product reviews and personal blogs. Unsupervised sentiment classification has a great advantage that it does not need a large expensive labeled corpus but only a user defined sentiment dictionary. However, the current existing dictionaries(such as HowNet and NTUSD) are insufficient of their vocabularies.

Many sentiment words are not included in the current Chinese sentiment dictionaries. For example, HowNet contains 3969 positive words and 3755 negative words; NUTSD contains 2648 positive words and 7742 negative words. Among them, only 669 positive words and 877 negative words are shared. Chinese idioms usually express strong sentiment, such as ‘励精图治’(to make great efforts to do sth.), ‘欺世盗名’(to win fame by cheating the world), however, they are not included in the dictionaries.

Typical existing algorithms for sentiment dictionary extension use patterns to construct a graph that reflects the relation between words, clustering algorithm is then applied on this graph to get positive and negative clusters. The basic idea is that multiple sentiment words are usually used together at the same time in one sentence, and they express sentiment with same polarity. The intuition is that, if a word mostly occurred with positive/negative sentiment words based on the current dictionaries, then we can guess it is also a positive/negative sentiment word. For example, ‘挺拔’(tall and straight) is usually co-occurred with positive words: ‘高大’(tall), ‘气派’(style), ‘刚强’(strong), ‘秀丽’(elegant), but rarely co-occurred with negative words, so we can guess that ‘挺拔’ is also a positive word.

In this work, we develop a new technique to deal with this problem. Similarly, by observing a large amount of sentiment word co-occurrences, we can construct a graph based on them, within which the polarity of some words that are defined in current sentiment dictionaries are known. So, our method is also based on a graph. However, we use syntactic analysis technology to get co-occurred word pairs. Meanwhile, we don't use clustering algorithm, because using patterns and clustering will both introduce errors. Then we compute a polarity strength for each word in the graph based on the link information and then rank all the words based on the strength. The top words that are mostly linked with positive words are likely to be positive words and bottom words that are mostly linked with negative words are likely to be negative words. The initial labels are given based on the current sentiment dictionary. The labels change during each iteration. Finally, the labels of the words will become stable.

In all, compared with the existing methods of sentiment dictionary extension, our method first use dependency parsing to process the unlabeled corpus and extract coordinated word pairs to construct a word graph then use the link analysis techniques to compute a polarity strength for each word and rank the words rather than clustering them. Our experiments show that the dependency parsing contribute to construct a better sentiment dictionary and the polarity strength for each word is useful to improve the performance of sentiment classification task.

The remainder of the paper is organized as follows. Section 2 is related work. Section 3 describes the main algorithm how to expand existing sentiment dictionaries. We conduct experiments to evaluate our technique in Section 4. Section 5 is the conclusion and future work.

## 2 Related Work

There are generally two different methods for dictionary extension: dictionary based method and corpus based method. The strategy of dictionary based method is to first collect a small set of sentiment words manually with known polarities, and then to grow this set by searching in an existing dictionary, such as WordNet in English or HowNet in Chinese, for their synonyms and antonyms. The newly found words are then added to the seed list. The process iterates until no more new words are found. This approach is used in (Hu and Liu, 2004; Kim and Hovy, 2004). After the process completes, manual inspection can be carried out to remove and correct errors. For English, researchers have also used additional information (e.g., glosses) in WordNet and machine learning techniques to generate better lists (Andreevskaia and Bergler, 2006)(Esuli and Sebastiani, 2005)(Esuli and Sebastiani, 2006)(Esuli and Sebastiani, 2007)(Kamps et al., 2004).

For corpus based methods, one of the key ideas is proposed by Hazivassiloglou and McKeown (1997). The technique starts with a list of seed opinion adjective words, and uses a set of linguistic constraints or conventions on connectives to identify additional adjective opinion words and their polarities. For example, in the sentence, “This car is beautiful and spacious,” if ‘beautiful’ is known to be positive, it can be inferred that ‘spacious’ is also positive. Rules or constraints are also designed for other connectives, OR, BUT, EITHER-OR, and NEITHER-NOR. Same and different polarity links between adjectives forms a graph. Finally, clustering is performed on the graph to produce two sets of words: positive and negative.

Kanayama and Nasukawa (2006) expanded this approach by introducing the idea of intra-sentential (within a sentence) and inter-sentential (between neighboring sentences) sentiment consistency to extract domain specific polar atoms. The intra-sentential consistency is similar to that in (Hatzivassiloglou and McKeown, 1997). Inter-sentential consistency applies the idea to neighboring sentences. That is, the same sentiment polarity (positive or negative) is usually expressed in a few consecutive sentences. Opinion changes are indicated by adversative expressions such as but and however. Some criteria to determine whether to add a word to the positive or negative lexicon are also proposed. This study was based on Japanese text. Other related work include (Kaji and Kitsuregawa, 2007)(Wiebe and Wilson, 2002).

Our method is related to but different from (Hatzivassiloglou and McKeown, 1997). First, their method only deals with adjectives, while our method can extract verbs, adverbs, adjectives and Chinese idioms; second, they use patterns to extract word pairs, our method uses dependency parsing to get word pairs with COO (coordination) relationship, so the syntactic information can be utilized; third, our method utilizes the link analysis to rank the words rather than clustering them. Finally, our method can compute polarity strength for each sentiment word that appears in corpus which is useful to improve the sentiment classification performance.

Qiu (2009a) proposed another method to extract domain specific sentiment words from reviews using also some seed sentiment words. The main idea is to exploit certain syntactic relations of sentiment words and object features for extraction. Ding and Liu (2008) explores the idea of intra-sentential and inter-sentential sentiment consistency further. Instead of finding domain dependent sentiment words, they showed that the same word might have different polarities in different contexts even in the same domain.

## 3 A Method to Expand Chinese Sentiment Dictionaries

In this section, we present our method of how to effectively expand the current sentiment dictionaries. The basic idea is that if a word often co-occurred with a sentiment word, then it is probably a sentiment word too. The polarity of a word could be computed by the statistical information of how many positive, negative and neutral sentiment words co-occurred with it.

Based on this idea, we first utilize a large unlabeled corpus to get a set of coordinated word pairs, then each pair of words are linked with an undirected arc. All the words construct an undirected graph. Then, the polarity strength for each word is computed based on the link information. For instance, if a word is mostly linked with a positive sentiment words, then we can assign positive label to this word. After computing the polarity strength, we could rank all the words. The top ranked words are likely to be positive words, the bottom ranked words are likely to be negative words. Simply, we assign m% of the top words with positive label, n% of the bottom words with negative label. Then the polarity strength of each word can be computed again. The process iterates until it reaches a maximum M times or the label of the words don't change any more.

### 3.1 Construct word graph

To get the co-occurrence information of word pairs, we first parse all the sentences into dependency trees. Dependency tree is a syntactic structure that reflects grammatical relations between constituents in a sentence. Each arc in dependency tree is labeled with a dependency type. In our task, we only extract the word pairs with COO(coordination) dependency type as the co-occurring word pairs. Each word with its part-of-speech tag corresponds to a node, and each co-occurring pair of words are linked. All the words then construct a word graph.

To study whether dependency parsing will contribute to the final result, we also use simple rules to get co-occurring word pairs. In detail, we use the Chinese punctuation '、' as the coordination indicator and extract the words that have the same part-of-speech tag near the punctuation as word pairs. Other rules could also be used such as '和', '与'(and), '或'(or), however we have found that '、' covers most of the cases in real data because in Chinese adjectives, adverbs are rarely coordinated with '和', '与' and '或'. So, for simplification we only used '、' as coordination indicator.

### 3.2 Compute the polarity strength of words

After constructing the graph, the words that are contained in current sentiment dictionary are firstly assigned with positive or negative labels. For each word in this graph including those with labels, the polarity strength could be computed based on the label distribution of its neighbors. Different from web page link graph with no label information where algorithms such as Pagerank or HITS could be used to compute the authority strength of each web page, the word graph is undirected, so we devise a new metric to compute the polarity strength for each node. Suppose $N_+$ is the number of positive sentiment words that are linked with $w$; $N_-$ is the number of negative sentiment words that are linked with $w$; $N_0$ is the number of neutral words that are linked with $w$, then the polarity strength of $w$ is computed through equation 1.

$$v = \frac{N_+ - N_-}{N_+ + N_0 + N_-} \times log(1 + N_+ + N_-) \tag{1}$$

Note that, we discriminate the part-of-speech tag of words. If a word have different pos tags, it can correspond to several different nodes in the graph for each pos tag. Here we only consider verbs, adverbs, adjectives and Chinese idioms that usually express sentiment. We also consider the negation problem, e.g. if a word is negated, its label will be reversed.

Based on the polarity strength, all the words are then ranked with a descending order. Then the top m% of the word lists are assigned with positive labels and bottom n% of the word lists are assigned with negative labels. Then the polarity strength of each word is computed again. In our algorithm, the labels assigned are soft labels, i.e. the label of a word may be changed during next iteration. The process iterates until M times or the labels of words don't change any more.

### 3.3 Correcting and Expanding the Dictionaries

After iteration ends, we organize all the words into 3 different ranked list: positive list, negative list and neutral list. Positive list contains all the positive words that are contained in the original dictionary; negative list contains all the negative words that are contained in original dictionary; neutral list contains all the words that are not contained in original dictionary.

For positive list, the words with strong negative polarity value if there are any at the bottom of the list are possibly wrongly defined words. Similarly, the words with positive polarity strength at the top of the negative list are also possibly wrongly defined. So, our algorithm can potentially discover the errors existing in the current dictionaries. As we will show later, the wrongly defined sentiment words are distributed at the sides of the positive and negative lists in a limited range. To correct the dictionary, it is easy for us to check the list manually. Here, we just simply filter the bottom $x\%$ of the positive list, top $y\%$ of the negative list.

For neutral list, we can add top $m\%$ words of the list to positive list and add bottom $n\%$ words of the list to negative list. This is how to extend the original dictionary. The algorithm is shown in Figure 1.

```
Input: An initial sentiment dictionary.
Output: A new dictionary.

1. Find out all co-occurring word pairs.
2. Construct word graph based on the word pairs.
3. Compute polarity strength for each word by equation 1.
4. Rank all words by polarity strength.
5. Filter bottom x% words of the positive list, top y% of the negative list.
6. Assign top m% words with positive label and bottom n% words with negative label.
7. Go to step 3 until $M$ times.
8. Output the new dictionary.
```

**Figure 1:** Main Algorithm of Sentiment Dictionary Correcting and Expanding.

## 4 Experiments

In this section, we evaluate our algorithm in three ways. First, we manually examine the precision of correcting and expanding of the dictionaries. Second, we examine the intersection rate of two new dictionaries generated based on HowNet and NTUSD in order to evaluate whether our method is sensitive to the initial dictionary. Third, we use the new dictionaries to sentiment classification and observe the performance compared with the original dictionaries.

### 4.1 Prepare the corpus

We build the needed corpus based on the corpus that are released by Sogou cooperation[1]. To use all these documents is difficult. Since we want to get the co-occurrence information of words with sentiment words, we only extract sentences that at least contain one sentiment word of the

[1] http://www.sogou.com/labs/dl/t.html

sentiment dictionary. In detail, for each sentiment word, we retrieve top 10000 sentences that contain this word.

We use HowNet and NTUSD as the original dictionaries. HowNet contains 3969 positive words and 3755 negative words(excluding the single character words), then there are totally about 77M sentences extracted from the Sogou corpus. NTUSD contains 2648 positive words and 7742 negative words, then there are totally about 103M sentences extracted from the Sogou corpus. Finally, the two sets of sentences are combined together, which contain 180M sentences. Then we do dependency parsing with LTP[2] software.

## 4.2 Experimental setting

In this work, we use three original sentiment dictionaries: HowNet, NTUSD and HowNet+NTUSD (HowNTU) in order to see whether our algorithm is stable to get consistent extended dictionaries. After examining the two dictionaries carefully, we found that NTUSD contains a lot of phrases rather than words. Because our corpus are word segmented and parsed, so, in our extracted corpus of COO pairs, 1175 positive words and 2728 negative words in NTUSD appear. For HowNet, about 3668 positive words and 3374 negative words appear in the corpus.

There are also contradictory definitions between HowNet and NTUSD. For example, there are 87 words that are defined in positive list in HowNet, but defined in negative list in NTUSD; there are 27 words that are defined in negative list in HowNet, but defined in positive list in NTUSD. In this work, we filter these words and don't use them. We let the algorithm iterate 20 times at most ($M = 20$) for all running.

## 4.3 Manual examination result

As we have mentioned above, the final list is organized in three lists: positive, negative and neutral list. The list sizes when we use different original dictionaries are shown in Table 1 .

| | Positive | Negative | Neutral |
|---|---|---|---|
| HowNet | 3771 | 3427 | 25215 |
| NTUSD | 1237 | 2796 | 28380 |
| HowNTU | 4323 | 5266 | 22824 |

**Table 1:** Size of Positive, Negative and Neutral lists with different original dictionaries.

Here, we manually evaluate our new dictionaries in terms of precision with original dictionary HowNTU. For positive list, the precision is the percentage of correctly discovered false items from all the discovered false items. All the discovered false items in positive list include the words whose polarity strength values are negative. The definition is similar for negative list. For neutral list, we examine two sub lists. The first sub list contains the top 2000 words that are used as positive words. The second sub list contains the bottom 2000 words that are used as negative words. Then, the precision is defined as the percentage of true positive/negative words among the positive/negative sub list.

In all, there are four lists to be evaluated: $false\_in\_positive$ contains 393 words whose polarity strengths are less than 0, $false\_in\_negative$ contains 567 words whose polarity strengths are larger than 0, $pos\_in\_neu$ contains 2000 words and $neg\_in\_neu$ contains 2000 words.

For manual annotation, if a word is positive, it is labeled 1; if a word is negative, it is labeled -1, otherwise, it is labeled 0. We have 3 annotators, and the coherence in term of agreement rate and Kappa value between annotators is shown in Table 2. We can see that the agreement between annotators is not high, this is because of some words are really ambiguous whether to express sentiment. Adjective words such as '火红'(red), '洁白'(white), '空荡'(void) sometimes express positive/negative sentiment when authors use them to describe a beautiful scene or a desolated place and sometimes not, and annotators may have different thinking when labeling such words.

[2] http://ir.hit.edu.cn/ltp/

| false_in_pos | | | | false_in_neg | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | | 1 | 2 | 3 |
| 1 | | 0.672 | 0.555 | 1 | | 0.646 | 0.547 |
| 2 | 0.509 | | 0.639 | 2 | 0.463 | | 0.589 |
| 3 | 0.357 | 0.445 | | 3 | 0.327 | 0.371 | |
| pos_in_neu | | | | neg_in_neu | | | |
| | 1 | 2 | 3 | | 1 | 2 | 3 |
| 1 | | 0.815 | 0.778 | 1 | | 0.885 | 0.859 |
| 2 | 0.335 | | 0.919 | 2 | 0.482 | | 0.893 |
| 3 | 0.150 | 0.313 | | 3 | 0.268 | 0.278 | |

**Table 2:** Agreement (top) and Kappa (bottom) values between annotators.

To integrate the results of three annotators, we sum up three different values from the annotators for each word, and if the sum is larger than 0, it is positive, if the sum is less than 0, it is negative, else it is neutral. Then we can evaluate the correcting and expanding precision based on the annotated result.

To reflect the change of precision value, we split each list into 10 pieces based on the ranked order. The precisions on each piece of the four lists are shown in Figure 2. The $x$ axis is each piece of an evaluation list. We can see that for $false\_in\_positive$ and $false\_in\_negative$ lists, the precision is high at first and then drops in a sharp rate. It shows that our algorithm ranks the false items at bottom of $false\_in\_positive$; ranks the false items at top of $false\_in\_negative$ in a limited range. Table 3 lists bottom 30 words in $false\_in\_positive$ and top 30 words in $false\_in\_negative$ of HowNTU. Some words are wrongly computed of their polarity strength values because it only occurred several times in the corpus. Since there are only few hundreds words in these two lists, we can manually revise the dictionary based on the ranked results.

| | |
|---|---|
| $false\_in\_positive$ | 不整齐 任性 自鸣得意 有口无心 碍事 松动 公然 唯唯诺诺 挑三拣四 松动 妄想 自满 泼悍 吐口 吃醋 贱价 眼红 脸红 狂乱 投机 忌妒 妒嫉 惨淡 目眩 任意 缩手缩脚 可笑 妒忌 嫉妒 自卑 |
| $false\_in\_negative$ | 蕴藉 恬淡 超然 变化无穷 骄人 严格 变化万千 绵软 慧黠 莽莽苍苍 娇嫩 嶙峋 忧国忧民 凛然 酥松 不断 默默 浓烈 险峻 与世无争 傲岸 硬气 净余 淡然 洁身自好 膻气 高昂 娇憨 险峭 什锦 |

**Table 3:** Bottom 30 words in $false\_in\_positive$ and top 30 words in $false\_in\_negative$ based on HowNTU.

Although the agreement of the annotators are not high, after integrating the annotation results, the dictionary extension gives a high precision (98% for $pos\_in\_neu$ and 97.3% for $neg\_in\_neu$ on average). Table 4 lists the top 30 words in $pos\_in\_neu$ and bottom 30 words in $neg\_in\_neu$ generated with HowNTU sentiment dictionary.

| | |
|---|---|
| $pos\_in\_neu$ | 挺拔 饱满 简捷 爱岗敬业 翠绿 精炼 适中 宽广 清爽 奋发向上 盎然 浓郁 洁白 助人为乐 精诚团结 高挑 畅通 团结友爱 深厚 积极向上 灵动 直观 丰厚 轻便 雄厚 悠久 乐于助人 有度 鲜艳 鲜嫩 |
| $neg\_in\_neu$ | 巧取豪夺 头晕目眩 欺上瞒下 乏力 倒行逆施 营私舞弊 斗殴 贩私 枪杀 目无法纪 出言不逊 无所适从 为非作歹 断章取义 尔虞我诈 坑蒙拐骗 贪污腐化 偷盗 弄虚作假 知法犯法 诬陷 拖拉 言行不一 寡廉鲜耻 仗势欺人 纵火 贪赃枉法 食欲不振 敲诈勒索 焦虑不安 |

**Table 4:** Top 30 words in $pos\_in\_neu$ and bottom 30 words in $neg\_in\_neu$ based on HowNTU.

## 4.4 Intersection of Two New Sentiment Dictionaries

Here, we examine the intersection rate of the two new dictionaries generated based on HowNet and NTUSD to see whether our algorithm is sensitive to original dictionary. The intersection rate

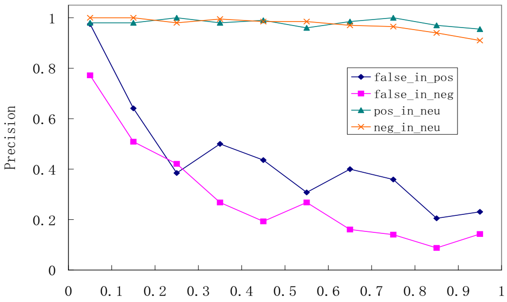

**Figure 2:** Precision of Dictionary Correcting and Expanding.

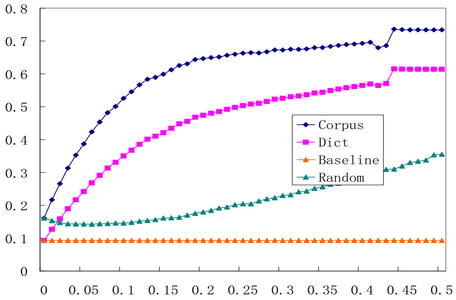

**Figure 3:** Intersection Rate of New Dictionaries.

of two dictionaries $D_1$ and $D_2$ is defined as follows, where $D_i^+$ is the positive vocabulary and $D_i^-$ is the negative vocabulary.

$$Inter(D_1, D_2) = \frac{(D_1^+ \cap D_2^+) \cup (D_1^- \cap D_2^-)}{D_1 \cup D_2} \tag{2}$$

As described in Section 3, top $m\%$ words and bottom $n\%$ words are added in new dictionaries, here we set $m = n$ for simplification. For the $false\_in\_positive$ and $false\_in\_negative$ lists, we simply filter $x\% = y\% = 10\%$ of the lists for HowNet, and do not filter any words for NTUSD. The intersection rate result is shown in Figure 3. The $x$ axis is the parameter $m, n$. $Baseline$ is the intersection rate of HowNet and NTUSD; $Corpus$ represents the intersection rate of the new dictionaries generated based on HowNet and NTUSD respectively; $Dict$ means that the sentiment words that do not appear in the unlabeled corpus (the words that do not appear in the graph) have been added to the new dictionaries when computing the intersection rate. $Random$ means we randomly add words in the word lists to dictionaries. We can see that the initial intersection rate of two dictionaries is very low ($< 10\%$). As we add more words from their neutral lists into the dictionaries, the intersection rate is gradually improved and reaches to more than $70\%$. It proves that our algorithm is relatively stable for different initial dictionaries.

### 4.5 Experimental Result for Sentiment Classification

Here, we examine the new dictionaries by using them in unsupervised sentiment classification. Two different algorithms are used here. The first is simply counting the numbers of positive and negative words contained in a target sentence or alternatively accumulating the polarity strengths of all the sentiment words. We use the result of simply counting with the original dictionary as baseline denoted as $Baseline$, and compare it with result of the simply counting on the new dictionary denoted as $Binary$ and the result of polarity strength accumulating on the new dictionary denoted as $Weight$. To evaluate whether dependency parsing is useful, we add experiments where the word pairs are extracted by simple rules when constructing the word graph as described in Section 3.1, denoted as $Rule$.

The second algorithm is $Selc$, proposed by Qiu(2009b). $Selc$ also uses an iterative scheme to extend the initial dictionary, however, it optimizes the dictionary based on the target corpus, while our algorithm try to construct a dictionary based on an independent corpus. The SELC Model is built based on a lexicon-based approach. Therefore it is domain independent. Several innovations, including positive/negative ratio control, the use of a general sentiment dictionary and the enlargement of negation word list, are adopted to overcome the positive classification bias of lexicon-based approaches. Then the SELC Model introduces a corpus-based approach to revise the result of the lexicon-based one. In the revising process, the corpus-based approach takes some reviews classified by the lexicon-based one as training data. Although the training data is machine-generated, most reviews of the training data are correctly labeled (above 93% precision in the experiments), because the lexicon-based approach is well designed. As such, the performance of the corpus-based approach is still reliable, and the whole performance is improved. For $Selc$ we compare its result on the original dictionary, denoted as $SelcBase$ with the result on our new constructed dictionary, denoted as $Selc$.

We use the data set that is labeled by Fuji Ren(Quan and Ren, 2009) for both emotion and sentiment analysis. It is constructed with Chinese personal blogs and contains 34295 sentences, including 16117 positive sentences, 15827 negative sentences and 2351 neutral sentences. In our experiment, we only evaluate the overall performance on positive and negative sentiment classification.

For the original dictionary, we use three different settings: HowNet, NTUSD and HowNet+NTUSD (HowNTU). The parameter setting is the same as in Section 4.4. The results in terms of precision, recall and F1 on the three settings are shown in Figure 4. The results have shown a constant trend with the change of parameter $m$ and $n$. Before $m = n = 0.1$, precision and recall are both improved. This could prove that about top 2000 words (10% of the neutral list) and bottom 2000 words are relative more reliable. After that, the precision begins to drop, while the recall and F1 still go up for $Weight$; the F1 measure for $Binary$ still goes up for a while and then drops. For $Selc$, the F1 also first goes up and then drops, this is because of the reliability of our new dictionaries with the change of parameter $m$ and $n$, and $Selc$ is sensitive to the reliability of the initial dictionary. $Weight$ is much better than $Rule$, this has shown that dependency parsing is useful since it could capture more complex relations between words than simple rules. The best result is given by $Weight$ with a high value of $m$ and $n$ on the combined dictionary HowNTU. This could prove that the polarity strength that our algorithm gives are really useful.

## 5 Conclusion and Future Work

In this paper, we present a method that utilizes a large unlabeled corpus to correct and expand Chinese sentiment dictionaries. Meanwhile, our method could compute a polarity strength for each keyword that appears in the corpus. Experiments that use our new dictionary in sentiment classification have shown that a simple method that adds the polarity strength of all words to get the polarity of a sentence gave better result than state-of-the-art method (e.g. Selc (Qiu et al., 2009b)). In future, we will continue to explore more information such as semantic similarity to integrate into the graph to improve our algorithm.

## References

Andreevskaia, Alina and Sabine Bergler. 2006. Mining wordnet for a fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL).*

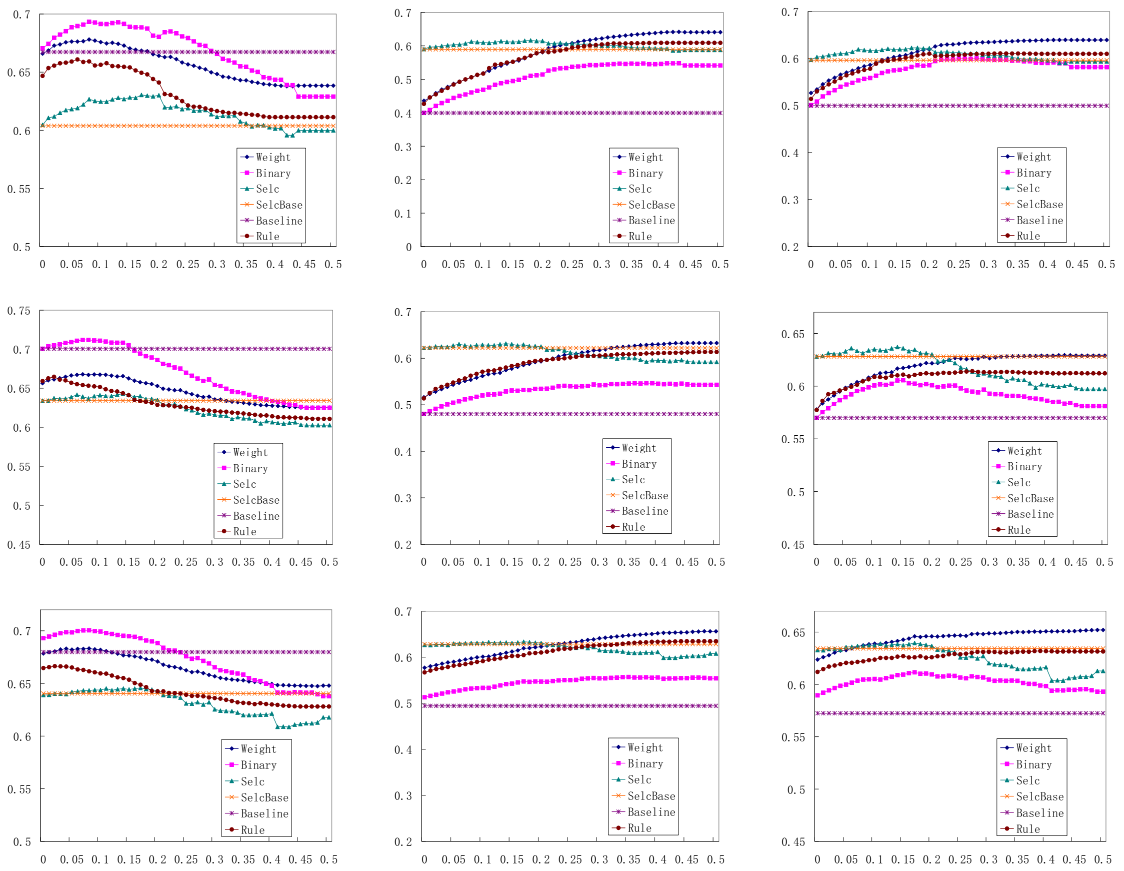

**Figure 4:** Sentiment Classification Result (Precision:Left, Recall:Middle, F1:Right) with Different Original Dictionaries (HowNet:Top, NTUSD:Middle, HowNTU:Bottom).

Ding, Xiaowen, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the Conference on Web Search and Web Data Mining (WSDM)*.

Esuli, Andrea and Fabrizio Sebastiani. 2005. Determining the semantic orientation of terms through gloss classification. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*.

Esuli, Andrea and Fabrizio Sebastiani. 2006. Determining term subjectivity and term orientation for opinion mining. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.

Esuli, Andrea and Fabrizio Sebastiani. 2007. Pageranking wordnet synsets: An application to opinion mining. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Hatzivassiloglou, Vasileios and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the Joint ACL/EACL Conference*, page 174 – 181.

Hu, Minqing and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, page 168 – 177.

Kaji, Nobuhiro and Masaru Kitsuregawa. 2007. Building lexicon for sentiment analysis from massive collection of html documents. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and ComputationalNatural Language Learning (EMNLP-CoNLL)*, page 1075 – 1083.

Kamps, Jaap, Maarten Marx, Robert J. Mokken, and Maarten de Rijke. 2004. Using wordnet to measure semantic orientation of adjectives. In *Proceedings of Language Resources and Evaluation (LREC)*, pages 1115–1118.

Kanayama, Hiroshi and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 355 – 363, July.

Kim, Soo-Min and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the International Conference on Computational Linguistics (COLING).*

Qiu, Guang, Bing Liu, Jiajun Bu, and Chun Chen. 2009a. Expanding domain sentiment lexicon through double propagation. In *International Joint Conference on Artificial Intelligence (IJCAI).*

Qiu, Likun, Weishi Zhang, Changjian Hu, and Kai Zhao. 2009b. Selc: a self-supervised model for sentiment classification. In *Proceeding of ACM conference on Information and knowledge management (CIKM)*, pages 929–936.

Quan, Changqin and Fuji Ren. 2009. Construction of a blog emotion corpus for chinese emotional expression analysis. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 1446 – 1454.

Wiebe, Janyce and Theresa Wilson. 2002. Learning to disambiguate potentially subjective expressions. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*, page 112 – 118.