Sentiment Classification Considering Negation and Contrast Transition

Shoushan Li and Chu-Ren Huang

Department of Chinese and Bilingual Studies
The Hong Kong Polytechnic University
{shoushan.li, churenhuang}@gmail.com

Abstract. Negation and contrast transition are two kinds of linguistic phenomena which are popularly used to reverse the sentiment polarity of some words and sentences. In this paper, we propose an approach to incorporate their classification information into our sentiment classification system: First, we classify sentences into sentiment reversed and non-reversed parts. Then, represent them as two different bags-of-words. Third, present three general strategies to do classification with two-bag-of-words modeling. We collect a large-scale product reviews involving five domains and conduct our experiments on them. The experimental results show that incorporating both negation and contrast transition information is effective and performs robustly better than traditional machine learning approach (based on one-bag-of-words modeling) across five different domains.

Keywords: Sentiment classification, opinion mining, linear Classifier.

1 Introduction

Sentiment classification is a task to classify text according to sentimental polarities of opinions they contain (e.g., favorable or unfavorable). This task has received considerable interests in computational linguistic community due to its wide applications.

In the latest studies of this task, machine learning techniques become the state-of-the-art approach and have achieved much better results than some rule-based approaches (Kennedy and Inkpen, 2006; Pang *et al.*, 2002) . In machine learning approach, a document (text) is usually modeled as a bag-of-words, a set of words without any word order or syntactic relation information. Therefore, the whole sentimental orientation is highly influenced by the sentiment polarity of each word. Notice that although each word takes a fixed sentiment polarity itself, its polarity contributed to the whole sentence or document might be completely the opposite. Negation and contrast transition are exactly the two kinds of linguistic phenomena which are able to reverse the sentiment polarity. For example, see a sentence containing negation "*this movie is not good*" and another sentence containing contrast transition "*this mouse is good looking, but it works terribly*". The sentiment polarity of the word *good* in these two sentences is positive but the whole sentences are negative. Therefore, we can see that the whole sentiment is not necessarily the sum of the parts (Turney, 2002). This phenomenon is one main reason why machine learning often fails to classify some testing samples (Dredze *et al.*, 2008).

Fortunately, a language usually has some special words which indicate the possible polarity shift of a word or even a sentence. These words are called contextual valence shifters (CVSs) which can cause the valence of a lexical item to shift from one pole to the other or, less forcefully, even to modify the valence towards a more neutral position (Polanyi and Zaenen, 2006). Generally speaking, CVSs are classified into two categories: sentence-based and

---

discourse-based (Polanyi and Zaenen, 2006). Sentence-based CVSs are responsible for shifting valence of some words in a sentence. The most obvious shifters are negatives, such as *not*, *none*, *never*, *nothing*, and *hardly*. These shifts usually reverse the sentiment polarity of some words. Other sentence-based shifters can be intensifiers (e.g., *rather*, *very*), modal operators (e.g., *if*), etc. Discourse-based CVSs often indicate the valence shifting in the context. Some connectives, such as *however*, *but*, and *notwithstanding*, belong to this type.

In this paper, we mainly focus on sentiment shifting including negation and contrast transition because this kind of shifting often fully reverses the sentiment polarity and thus mostly reflects the weakness of those machine learning approaches based on one-bag-of-words modeling. Other types of shifting, for instance, intensification with intensifiers (e.g., *rather*, *very*) is capable of changing the intension of some words but would not reverse their polarities.

Note that contrast transition is one special type of transition and is used to express contradiction or contrast when connecting one paragraph, sentence, clause or word with the other. It is distinguished from other types of transitions by different connectives. For contrast transitions, the connectives are some CVSs like *however*, *but*, and *notwithstanding* while others use some different connectives, e.g., conclusion transition takes the connectives like *therefore*, *in a word*, *in summary*, and *in brief*.

To incorporate sentiment reversing information into a machine learning approach, we first segment the whole document into sub-sentences. We then partition them into two groups: one includes those called sentiment-reversed sentences and the other includes those called sentiment-non-reversed sentences. As a result, each document is represented as two-bags-of-words rather than traditional one-bag-of-words. Finally, we propose the classification algorithm to do the classification on the text with two-bags-of-words modeling.

The remainder of this paper is organized as follows. Section 2 introduces the related work on CVS applications in sentiment classification. Section 3 presents our approach in detail. Experimental results are presented and analyzed in Section 4. Finally, Section 5 draws our conclusions and outlines the future work.

2 Related Work

During recent several years, various of issues have been studied for sentiment classification, such as feature extraction (Riloff *et al.*, 2006), domain adaptation (Blitzer *et al.*, 2007) and multi-domain learning (Li and Zong, 2008). For a detailed survey of this research field, see Pang and Lee (2008). However, most studies directly borrow machine learning approach from traditional topic-based text classification and very few work are focus on incorporating linguistic knowledge that sentiment text particularly contains, e.g., valence shifting phenomena and comparative sentences (Jindal and Liu, 2006).

Pang *et al.* (2002) first employ machine learning approach to sentiment classification and find that machine learning methods definitely outperform human-produced baselines. In their approach, they consider negation by adding the tag NOT to every word between a negation word (*not*, *isn't*, *didn't*, etc.) and the first punctuation mark following the negation word. But their results show that adding negation has a very negligible and on average slightly harmful effect on the performance.

Kennedy and Inkpen (2006) check three types of CVSs: negatives, intensifiers, and diminishers and add their valence shifting bigrams as additional features. Their results show that considering CVSs greatly improve the performances of term-counting approach. But as far as machine learning approach is concerned, the improvement is very slight (less than 1%).

Na *et al.* (2004) attempt to model negation more accurately and achieve a satisfactory improvement. However, they need to do part-of-speech to get negation phrases and their baseline performance itself is very low (less than 80%).

Different from all the above work, our approach is easy to implement and need no additional features (e.g., bi-gram, part-of-speech tag). Furthermore, our approach is capable of considering

both negation and contrast transition. In our view, only considering negation is not enough since there are some negation sentences appear in a contrast transition structure. For example, *this mouse is not good looking, but it works perfect and I like it*. Apparently, only considering negation is still difficult to give an correct sentiment classification in this case.

3   Our Approach

3.1   Classification Algorithm

In a standard machine learning classification problem, we seek a predictor f (also called a classifier) that maps an input vector x to the corresponding class label y. The predictor is trained on a finite set of labeled examples (X, Y) which are drawn from an unknown distribution **D**. The learning objective is to minimize the expected error, i.e.,

$$f = \arg\min_{f \in \mathbf{H}} \sum_{X,Y} L(f(X),Y) \tag{1}$$

where L is a prescribed loss function and **H** is a set of functions called the hypothesis space, which consists of functions from x to y.

As a linear classifier, the predictor takes the form $f(X_i) = w^T X_i$. Then a regularized form of formula (1) is often used as below, which always has a unique and numerically stable solution

$$\hat{w} = \arg\min_{w} \sum_{X,Y} L(w^T X, Y) + \frac{\lambda}{2}\|w\|_2^2 \tag{2}$$

where $\|w\|_2^2 = w^T w$ and λ is a non-negative regularization parameter. If $\lambda = 0$, the problem is un-regularized.

***

Algorithm (standard SGD)

Initialize $\hat{w}_0$

for t=1,2, ...

   Draw (X_t, Y_t) randomly from **D**.

   Update $\hat{w}_{t-1}$ as

   $\hat{w}_t = \hat{w}_{t-1} - \eta_t S^{-1}(\lambda \hat{w}_{t-1} + L_1'(\hat{w}_{t-1}^T X_t, Y_t)X_t)$

end

***

Figure 1: Standard online SGD algorithm

Solving (2) with stochastic gradient descent (SGD), we get the standard SGD online updating strategy as following (Zhang, 2004)

$$\hat{w}_t = \hat{w}_{t-1} - \eta_t S^{-1}(\lambda \hat{w}_{t-1} + L_1'(\hat{w}_{t-1}^T X_t, Y_t)X_t) \tag{3}$$

where $L_1'(p,y) = \frac{\partial}{\partial p} L(p,y)$ and (X_t, Y_t) is the instance we are observing at the t-th step. The matrix S can be regarded as a pre-conditioner. For simplicity, we assume it to be a constant matrix. $\eta_t > 0$ is a appropriately chosen learning rate parameter. The whole algorithm is described in **Figure 1** (Zhang, 2004).

3.2 Text Modeling

In traditional text classification tasks, a text T (e.g., document, sentence) are modeled as one bag-of-words and the input vector of the text is constructed from weights of the words (also called terms) $(t_1,...,t_N)$. In this paper, we focus on document-based sentiment classification. Specifically, the terms are possibly words, word n-grams, or even phrases extracted from the training data, with N being the number of terms. The weights are statistic information of these terms, e.g., *tf, tf·idf* . Then the text T is represented as a vector $X(T)$, i.e.,

$$X(T) = < sta(t_1),\ sta(t_2),\ ...\ ,\ sta(t_N) > \tag{4}$$

The output label y has a value of 1 or -1 representing a positive or negative sentiment polarity.

As a special case of text classification, sentiment classification applies bag-of-words model directly for a long time. Although machine learning with this text modeling approach has shown to perform much better than some rule-based approaches, e.g., term-counting approach, the achieved performance is much worse than traditional topic-based text classification. Compared to topic-based classification, one big challenge in sentiment classification is that sentiment polarity of one word is not always consistent with the whole orientation of the text. Consider the following two sentences:

a1. *This is not a good movie and I hate it.*

a2. *This is such a good movie and I do not hate it at all.*

Because they are represented as almost the same bag-of-words, their classification results would be the same when applying machine learning with one-bag-of-words modeling. But their sentiment polarities are obviously different from each other. Therefore, traditional bag-of-words modeling is not appropriate for sentiment classification to some extent.

Instead of considering a text as a bag-of-words, we propose a new text modeling approach which considers a text as two bags-of-words. Specifically, a text T, either for training or testing, is partitioned into two sub-texts: sentiment-reversed part T_{re} and sentiment-non-reversed part T_{non}. Sentiment-reversed part ideally contains those sentences which holds words with the opposite sentiment polarity compared to the whole document's.

Formally, a text T consists of multiple sentences, i.e., $T = (s_1, s_2,...,s_m)$. Suppose each sentence takes a sentiment-reversed tagging V which represents whether it is a sentiment-reversed sentence ($V(s)=1$) or not ($V(s)=-1$). Originally, every sentence is assigned the same tagging value of -1, i.e., $V_o(s_i)=-1, i=1,2,...,m$.

3.3 Sentence Segmentation

We assume the sentences as the basic text unit and each one would be assigned a tag. Actually, the ideal basic text unit should be something like clauses rather than sentences (we call them sub-sentences). For example,

b1. *This is not a good movie and I hate it.*

b2. *I like it because I didn't want to transfer video.*

Although these two sentences contain negation, it is unsuitable to put the whole sentence into the sentiment-reversed part. A better way is to first segment the sentences into sub-sentences and assign each one the sentiment-reversed tagging.

We implement a simple approach to segment a document into sub-sentences. First, we do segmentation merely with the punctuations, such as period, comma, and interrogation mark. Then, we use some manually-collected key words, such as *and*, *because* and *since* for further segmentation. These key words are used to introduce various complex sentences with clauses.

3.4 Sentiment-reversed Sentence Detection

A language usually has some special words called CWSs to indicate possible sentiment shifting of a word or a sentence. As mentioned in the introduction, two kinds of CWSs are commonly used to indicate valence switching: negatives and contrast transition connectives. We would use these CWSs to tag sentence to be a sentiment-reversed sentence or not.

If the sentence s_i contains k negatives, we update the tagging value as following:

$$V_{Neg}(s_i) = V_o(s_i) \times (-1)^k \tag{5}$$

As for transition connectives, we first need to recognize which related sentences are possible to be sentiment-reversed. Different from negatives, each transition connective has its own rule to pick sentiment-reversed sentences around it. Here, we only focus on two transition connectives: *but* and *however* because they appear most frequently and more likely to really reverse the sentiment polarity. If the connective is *but*, the sentence before it might be sentiment reversing. If the connective is *however*, there might be not only one sentiment-reversed sentence before it. We only pick the nearest one as the sentiment-reversed sentence to avoid introducing too many noises. Overall, if the sentence s_i appears before *but* or *however*, we update its tagging value as following:

$$V_{Tran}(s_i) = V_{Neg}(s_i) \times (-1) \tag{6}$$

Then, we get the sentiment-reversed part T_{re} and sentiment-non-reversed part T_{non} as follows.

$$T_{re} = \{V_{Tran}(s) = 1, s \in (s_1, s_2, ..., s_m)\} \tag{7}$$

and,

$$T_{non} = \{V_{Tran}(s) = -1, s \in (s_1, s_2, ..., s_m)\} \tag{8}$$

It is worth pointing out that the sentiment-reversed sentences obtained by our approach sometimes are not really sentiment reversed. This is due to some mistakes in sentence segmentation and reversed-sentiment detection. Meanwhile, some real sentiment-reversed sentences are not able to be recognized. Consider the following sentence:

c1. *It could have been a great product. I dislike it, however.*

The sub-sentence (*I dislike it*) before *however* is actually not sentiment-reversed but the previous sentence (*It could have been a great product*) is. In fact, recognizing those sentiment-reversed sentences can hardly perform perfectly and it might be as difficult as sentiment classification itself. Nevertheless, our main objective here is to build an approach which is able to incorporate the sentiment reversing information. As a preliminary step, we try to recognize most sentiment-reversed sentences and decrease their influence to the whole sentiment.

3.5 Sentiment Classification

In this section, we propose three general strategies for classifying the text with two-bags-of-words modeling: (1) remove the sentiment-reversed part; (2) tune the parameters of the sentiment-reversed part according to those learned from the sentiment-non-reversed part; (3) simultaneity learn both sentiment-reversed and sentiment-non-reversed parts.

The first naive strategy, called *remove strategy*, is to directly remove the sentiment-reversed part considering that they might badly influence the whole sentiment. Accordingly, the text is represented as a bag-of-words which only contains the words in all sentiment-non-reversed text, i.e., T_{non}. Then, the words in T_{non} are used to generate input vectors X_N for each document. The learning objective is to minimize the following expected error

$$\hat{w}_n = \arg\min_{w_n} \sum_{X,Y} L(w_n^T X_N, Y) + \frac{\lambda_n}{2} \|w_n\|_2^2 \tag{9}$$

In the testing phase, the label Y' of one sample X_N' is estimated as

$$Y' = Sgn(\hat{w}_n^T X_N') \tag{10}$$

Where $Sgn(x)$ is defined as

$$Sgn(x) = \begin{cases} 1 & if \ x > 0 \\ 0 & if \ x = 0 \\ -1 & if \ x < 0 \end{cases} \quad (11)$$

The second strategy, called *shift strategy*, takes the same learning process as the first strategy in the training phase but perform different estimation in the testing phase. Since the sentences in the sentiment-reversed part are possibly expressing the reversed polarities, we would like to shift the parameters $\hat{w}_n$ when they are applied to the sentiment-reversed text. Thus the label Y' of one sample (X'_N, X'_{N-re}) is estimated as

$$Y' = Sgn(\hat{w}_n^T X'_N + (-1) \cdot \hat{w}_n^T X'_{N-re}) \quad (12)$$

where X'_{N-re} represents the input vector of the sentiment-reversed text. Here, X'_N and X'_{N-re} are generated from the same term set as the first strategy, i.e., the words in T_{non}.

The third strategy, called *joint strategy*, simultaneity learning both sentiment-reversed and sentiment-non-reversed parts. In the training phase, the learning objective is to minimize the following expected error

$$\hat{w}_n, \hat{w}_r = \underset{w_n, w_r}{\arg\min} \sum_{X,Y} L(w_n^T X_N + w_r^T X_{R-re}, Y) + \frac{\lambda_n}{2}\|w_n\|_2^2 + \frac{\lambda_r}{2}\|w_r\|_2^2 \quad (13)$$

where X_{R-re} represents the input vector of the sentiment-reversed text. Here, X_N and X_{R-re} are generated from different term sets: the words in T_{non} and in T_{re} respectively.

In the testing phase, the label Y' of one sample (X'_N, X'_{R-re}) is estimated as

$$Y' = Sgn(\hat{w}_n^T X'_N + \hat{w}_r^T X'_{R-re}) \quad (14)$$

Although all strategies are expressed in terms of linear classifiers, the corresponding ideas for the first and third strategies are general for any other classification algorithms. Overall speaking, only the third one really utilizes both the reversed-sentiment and non-reversed sentiment information for learning. Also, it shares the similar computational complexity as traditional machine learning approaches based on one-bag-of-words modeling.

4 Experimental Studies

4.1 Experimental Setup

Data Set: There are some famous public data sets available for sentiment classification studies. Among them, Cornell movie-review dataset[1] (Pang and Lee, 2004) and product reviews[2] (Blitzer *et al.*, 2007) are most popularly used. Both of them are 2-category (positive and negative) tasks and each consists of 2,000 reviews in a domain. The results in some previous work are sometimes not consistent due to the application of different domains of reviews when negation is considered (Pang *et al.*, 2002 and Na *et al.*, 2004). Thus we follow the way of Blitzer *et al.* (2007) to collect more data involving data in our experiments. Specifically, we totally collect 5 domains of reviews from *Amazon.cn*, namely Book, Camera, HD (Hard Disk), Health and Kitchen. Each domain consists of 2,400 reviews and each category (negative or positive) contains 1,200 reviews.

Experiment Implementation: We perform 5-fold cross validation in all experiments. That is to say, the dataset in each domain is randomly and evenly split into 5 folds. Then we use each

---

[1] http://www.cs.cornell.edu/People/pabo/movie-review-data/
[2] http://www.seas.upenn.edu/~mdredze/datasets/sentiment/

4 folds for training and the remaining 1 fold for testing. We use accuracy to measure the classification performances.

Features: The features are single words with a BOOL weight (0 or 1), representing the presence or absence of a feature.

Classification Algorithm: We use SGD linear predictors with Huber function as the loss function (Zhang, 2004). Compared to support vector machine (SVM), SGD linear classifier not only performs online learning but also gives comparable or even better results. We compare the two classification algorithms with the Cornell movie-review data set (Pang and Lee, 2004). The 5-fold cross validation average results are 0.843 by SVM and 0.859 by SGD, from which we can see that SGD outperforms SVM (implemented with LIBSVM[3] with linear kernel). Actually, similar conclusion can be found in Dredze *et al.* (2008).

4.2 Distribution of Negation and Contrast Transition Sentences

Before classification, each document is necessarily partitioned into two sub-texts: sentiment-reversed part and sentiment-non-reversed part. To achieve that, we use some CVSs to classify those segmented sub-sentences into two categories: sentiment-reversed and sentiment-non-reversed. Specifically, negatives are used to recognize the negation sentences and the connectives of '*but*' and '*however*' are used to recognize contrast transition sentences. First of all, let us see the distribution of these negation sentences and contrast transition sentences in our review corpus.
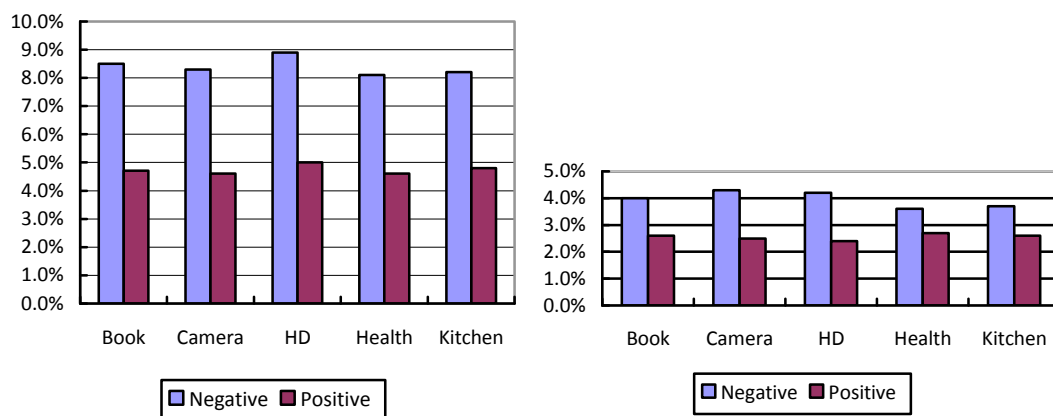
Figure 2: The proportion of negation (left) and transition (right) sentences in negative and positive reviews

Figure 2 (left) shows the proportions of negation sentences to all sentences in negative and positive reviews respectively. The proportion is computed in each domain. From Figure 2, we can see that negation sentences occur frequently in reviews and are more likely expressed in negative reviews. The proportion of negation sentences in negative reviews is about 8%, which is about twice as the one in positive reviews. This result agrees with our general knowledge that people are more likely to use negation sentences when expressing their negative opinions.

Figure 2 (right) shows the proportions of contrast transition sentences to all sentences in negative and positive reviews respectively. From this figure, we can also see that contrast sentences are more likely expressed in negative reviews than in positive reviews. Compared to negation sentences, contrast transition sentences are much fewer.

---

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

4.3   Classification Results with Different Strategies

Table 1: The classification results of different strategies when only considering negation

Domain	Baseline	Negation		
		Remove	Switch	Joint
Book	0.849	0.845	0.834	**0.860**
Camera	0.920	0.912	0.907	**0.924**
HD	0.934	0.929	0.917	**0.946**
Health	0.841	0.830	0.819	**0.854**
Kitchen	0.860	0.861	0.858	**0.872**

Table 2: The classification results of different strategies when only considering contrast transition

Domain	Baseline	Transition		
		Remove	Switch	Joint
Book	0.849	**0.850**	0.843	0.848
Camera	0.920	0.924	0.917	**0.930**
HD	0.934	0.934	0.930	**0.939**
Health	0.841	0.848	0.845	**0.854**
Kitchen	0.860	**0.865**	0.855	0.864

Table 1 shows the classification results of different strategies when only negation is considered for sentiment-reversed sentence detection. Baseline shows the results of using all unigrams with one-bag-of-words modeling. Let us compare the results between the baseline and each strategy.

First, comparing baseline and *remove* strategy, we find that simply removing all negation sentences is not helpful. Sometimes, the performances even decrease more than one percent (see the domain of *health*).

Second, comparing baseline and *switch* strategy, we find that *switch* strategy is worse and always harmful for sentiment classification. This is different from our first thought of this strategy. But after close thinking of it, we would notice that assigning all the words a negative parameter, i.e., $(-1) * \hat{w}_N$ in a sentiment-reversed sentence is not reasonable. In fact, it is only necessary to assign a positive parameter to those words which express sentiment. Moreover, some words are commonly used in both negation and non-negation sentences for expressing the same sentiment polarity. For example, see the word *waste* in the following two sentences.
d1. *It is a waste of your money.*

d2. *Do not waste your money.*

Third, comparing to baseline, we find that *joint* strategy is successful and consistently improves the performance. But the improved performances in some domains are insignificant (less than 0.5% in *camera*). Therefore, it is not strange that the conclusions in Pang et al. (2002) and Na et al. (2004) is a little different from each other. Whether inducing negation is effective or not is influenced by the application domains.

Table 2 shows the classification results of different strategies when only contrast transition is considered for sentiment-reversed sentence detection. Let us compare the results between the baseline and each strategy respectively.

First, quite different from the case of negation, simply removing the contrast transition sentences can always improve classification performances. We think this is mainly because the amount of transition sentences is much less than negation sentences. Removing them is beneficial for deleting classification noise without losing too much useful classification information.

Second, switch strategy generally fails to improve the performance. It can only make very small improvement in the domain of *health*

Third, *joint* strategy is still effective in dealing with contrast transition. However, some results are no better than *remove* strategy.

Overall speaking, contrast transition is also helpful for classification. But the improved performances are a little lower than the ones by using negation. This is mainly because negation appears more often than contrast transition, which makes the sentences' sentiment reversed more frequently.

Table 3: The classification results of different strategies when considering both negation and contrast transition

Domain	Baseline	Negation + Transition		
		Remove	Switch	Joint
Book	0.849	0.847	0.821	**0.863**
Camera	0.920	0.919	0.900	**0.930**
HD	0.934	0.923	0.913	**0.946**
Health	0.841	0.848	0.812	**0.864**
Kitchen	0.860	0.861	0.852	**0.873**

Table 3 shows the classification results of different strategies when both negation and contrast transition are considered for sentiment-reversed sentence detection. Apparently, *joint* strategy is more powerful than the other two strategies and consistently achieves much better classification results than the baseline (The improved accuracy is no less than 1% in all domains).

Comparing the results in Table 3 to the results in Table 1 or Table 2, we can conclude that considering both negation and contrast transition is generally a better choice than considering only one of them.

5   Conclusion

In this paper, we propose an approach for incorporating sentiment reversing information into machine learning based sentiment classification system. Specifically, we consider two kinds of linguistic phenomena: negation and contrast transition, which are popularly used to reverse the sentiment polarity. Experimental results on a newly collected corpus show that simply removing the contrast transition sentences is helpful but it is not effective for negation. Furthermore, we see that our approach with *joint* strategy is able to robustly improve the performances across all five domains.

In our approach, we only use negation and contrast transition keywords to detect sentiment reversed sentences. In addition, there certainly exist some other structures which can reverse the sentiment polarity of a word or sentence. In our future work, we hope to find some more

effective detection approaches and consider more structures to recognize sentiment-reversed sentences.

References

Blitzer, J., M. Dredze and F. Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of Annual Meeting on Association for Computational Linguistics (ACL-07)*.

Dredze, M., K. Crammer and F. Pereira. 2008. Confidence-weighted Linear Classification. In *Proceedings of International Conference on Machine Learning (ICML-08)*.

Jindal, N. and B. Liu. 2006. Identifying Comparative Sentences in Text Documents. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval (SIGIR-06)*.

Kennedy, A. and D. Inkpen. 2006. Sentiment Classification of Movie Reviews using Contextual Valence Shifters . *Computational Intelligence*, Vol. 22, No. 2, pp. 110-125.

Li, S. and C. Zong. 2008. Multi-domain Sentiment Classification. In *Proceedings of Annual Meeting of the Association for Computational Linguistics: Human Language Technology (ACL-08: HLT)*.

Na, J., H. Sui, C. Khoo, S. Chan and Y. Zhou. 2004. Effectiveness of Simple Linguistic Processing in Automatic Sentiment Classification of Product Reviews. In *Conference of the International Society for Knowledge Organization (ISKO)*, pages 49–54.

Pang, B., L. Lee and S. Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-02)*.

Pang, B. and L. Lee. 2004. A Sentimental Education: Sentiment Analysis using Subjectivity Summarization based on Minimum Cuts. In *Proceedings of Annual Meeting on Association for Computational Linguistics (ACL-04)*.

Pang, B. and L. Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundation and Trends in Information Retrieval*, 2(1-2):1–135.

Polanyi, L. and A. Zaenen. 2006. Contextual Valence Shifters. In *Computing attitude and affect in text: Theory and application*. Springer Verlag.

Riloff, E., S. Patwardhan and J. Wiebe. 2006. Feature Subsumption for Opinion Analysis. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-06)*.

Turney, P. 2002. Thumbs Up or Thumbs Down? Sentiment Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of Annual Meeting on Association for Computational Linguistics (ACL-02)*.

Zhang, T. 2004. Solving Large Scale Linear Prediction Problems using Stochastic Gradient Descent Algorithms. In *Proceedings of International Conference on Machine Learning (ICML-04)*.