Dual Sentiment Analysis: Considering Two Sides of One Review

Rui Xia, Feng Xu, Chengqing Zong, Qianmu Li, Yong Qi, and Tao Li

Abstract—Bag-of-words (BOW) is now the most popular way to model text in statistical machine learning approaches in sentiment analysis. However, the performance of BOW sometimes remains limited due to some fundamental deficiencies in handling the polarity shift problem. We propose a model called dual sentiment analysis (DSA), to address this problem for sentiment classification. We first propose a novel data expansion technique by creating a sentiment-reversed review for each training and test review. On this basis, we propose a dual training algorithm to make use of original and reversed training reviews in pairs for learning a sentiment classifier, and a dual prediction algorithm to classify the test reviews by considering two sides of one review. We also extend the DSA framework from polarity (positive-negative) classification to 3-class (positive-negative-neutral) classification, by taking the neutral reviews into consideration. Finally, we develop a corpus-based method to construct a pseudo-antonym dictionary, which removes DSA's dependency on an external antonym dictionary for review reversion. We conduct a wide range of experiments including two tasks, nine datasets, two antonym dictionaries, three classification algorithms, and two types of features. The results demonstrate the effectiveness of DSA in supervised sentiment classification.

Index Terms—Natural language processing, machine learning, sentiment analysis, opinion mining

1 Introduction

In recent years, with the growing volume of online reviews available on the Internet, sentiment analysis and opinion mining, as a special text mining task for determining the subjective attitude (i.e., sentiment) expressed by the text, is becoming a hotspot in the field of data mining and natural language processing [1], [25], [26], [36], [46], [48]. Sentiment classification is a basic task in sentiment analysis, with its aim to classify the sentiment (e.g., positive or negative) of a given text. The general practice in sentiment classification follows the techniques in traditional topic-based text classification, where the bag-of-words (BOW) model is typically used for text representation. In the BOW model, a review text is represented by a vector of independent words. The statistical machine learning algorithms (such as naïve Bayes, maximum entropy classifier, and support vector machines) are then employed to train a sentiment classifier.

Although the BOW model is very simple and quite efficient in topic-based text classification, it is actually not very suitable for sentiment classification because it disrupts the word order, breaks the syntactic structures, and discards some semantic information. Consequently, a large number of researches in sentiment analysis aimed to enhance BOW by incorporating linguistic knowledge [7], [12], [17], [28], [30], [35], [41], [43]. However, due to the fundamental deficiencies in BOW, most of these efforts showed very slight effects in improving the classification accuracy. One of the most well-known difficulties is the polarity shift problem.

Polarity shift is a kind of linguistic phenomenon which can reverse the sentiment polarity of the text. Negation is the most important type of polarity shift. For example, by adding a negation word *"don't"* to a positive text *"I like this book"* in front of the word *"like"*, the sentiment of the text will be reversed from positive to negative. However, the two sentiment-opposite texts are considered to be very similar by the BOW representation. This is the main reason why standard machine learning algorithms often fail under the circumstance of polarity shift.

Several approaches have been proposed in the literature to address the polarity shift problem [5], [6], [14], [17], [19], [21], [35], [42]. However, most of them required either complex linguistic knowledge or extra human annotations. Such high-level dependency on external resources makes the systems difficult to be widely used in practice. There were also some efforts to address the polarity shift problem with the absence of extra annotations and linguistic knowledge [6], [19], [21], [35]. However, to the best of our knowledge, the state-of-the-art results are still far from satisfactory. For example, the improvement is only about 2 percent on the multi-domain sentiment datasets in [21].

In this paper, we propose a simple yet efficient model, called dual sentiment analysis (DSA), to address the polarity shift problem in sentiment classification. By using the property that sentiment classification has two opposite

- *R. Xia, Q. Li, and Y. Qi are with School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. E-mail: {rxia, qianmu, qyong}@njust.edu.cn.*
- *F. Xu is with School of Economics and Management, Nanjing University of Science and Technology, Nanjing, China. E-mail: breezewing@126.com.*
- *C. Zong is with National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. E-mail: cqzong@nlpr.ia.ac.cn.*
- *T. Li is with the School of Computer Science & Technology, Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, P.R. China, 210046, and the School of Computing and Information Sciences, Florida International University, Miami, FL 33199. E-mail: taoli@cs.fiu.edu.*

Manuscript received 15 May 2014; revised 24 Jan. 2015; accepted 1 Feb. 2015. Date of publication 25 Feb. 2015; date of current version 2 July 2015.
Recommended for acceptance by P.G. Ipeirotis.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2015.2407371

class labels (i.e., positive and negative), we first propose a data expansion technique by creating sentiment-reversed reviews. The original and reversed reviews are constructed in a one-to-one correspondence.

Thereafter, we propose a dual training (DT) algorithm and a dual prediction (DP) algorithm respectively, to make use of the original and reversed samples in pairs for training a statistical classifier and make predictions. In DT, the classifier is learnt by maximizing a combination of likelihoods of the original and reversed training data set. In DP, predictions are made by considering two sides of one review. That is, we measure not only how positive/negative the original review is, but also how negative/positive the reversed review is.

We further extend our DSA framework from polarity (positive-negative) classification to 3-class (positive-negative-neutral) sentiment classification, by taking the neutral reviews into consideration in both dual training and dual prediction.

To reduce DSA's dependency on an external antonym dictionary, we finally develop a corpus-based method for constructing a pseudo-antonym dictionary. The pseudo-antonym dictionary is language-independent and domain-adaptive. It makes the DSA model possible to be applied into a wide range of applications.

The organization of this paper is as follows. Section 2 reviews the related work. In Section 3, we present the data expansion technique. In Section 4, we introduce the DSA framework in detail. Section 5 presents two methods for constructing an antonym dictionary. The experimental results are reported and discussed in Section 6. Section 7 finally draws conclusions and outlines directions for the future work.

2 Related Work

We first summarize the work of sentiment analysis and polarity shift, and then review the technique of data expansion.

2.1 Sentiment Analysis and Polarity Shift

According to the levels of granularity, tasks in sentiment analysis can be divied into four categorizations: document-level, sentence-level, phrase-level, and aspect-level sentiment analysis.

Focusing on the phrase/subsentence- and aspect-level sentiment analysis, Wilson et al. [42] discussed effects of complex polarity shift. They began with a lexicon of words with established prior polarities, and identify the "contextual polarity" of phrases, based on some refined annotations. Choi and Cardie [4] further combined different kinds of negators with lexical polarity items though various compositional semantic models, both heuristic and machine learned, to improved subsentential sentiment analysis. Nakagawa et al. [29] developed a semi-supervised model for subsentential sentiment analysis that predicts polarity based on the interactions between nodes in dependency graphs, which potentially can induce the scope of negation. In aspect-level sentiment analysis, the polarity shift problem was considered in both corpus- and lexicon-based methods [8], [9], [10], [13].

For document- and sentence-level sentiment classification, there are two main types of methods in the literature: term-counting and machine learning methods. In term-counting methods, the overall orientation of a text is obtained by summing up the orientation scores of content words in the text, based on manually-collected or external lexical resources [38], [39]. In machine learning methods, sentiment classification is regarded as a statistical classification problem, where a text is represented by a bag-of-words; then, the supervised machine learning algorithms are applied as classifier [35]. Accordingly, the way to handle polarity shift also differs in the two types of methods.

The term-counting methods can be easily modified to include polarity shift. One common way is to directly reverse the sentiment of polarity-shifted words, and then sum up the sentiment score word by word [4], [16], [17], [37]. Compared with term counting methods, the machine learning methods are more widely discussed in the sentiment classification literatures. However, it is relatively hard to integrate the polarity shift information into the BOW model in such methods. For example, Das and Chen [6] proposed a method by simply attaching *"NOT"* to words in the scope of negation, so that in the text *"I don't like book"*, the word *"like"* becomes a new word *"like-NOT"*. Yet Pang et al. [35] reported that this method only has slightly negligible effects on improving the sentiment classification accuracy. There were also some attempts to model polarity shift by using more linguistic features or lexical resources. For example, Na et al. [28] proposed to model negation by looking for specific part-of-speech tag patterns. Kennedy and Inkpen [17] proposed to use syntactic parsing to capture three types of valence shifters (negative, intensifiers, and diminishers). Their results showed that handling polarity shift improves the performance of term-counting systems significantly, but the improvements upon the baselines of machine learning systems are very slight (less than 1 percent). Ikeda et al. [14] proposed a machine learning method based on a lexical dictionary extracted from General Inquirer[1] to model polarity-shifters for both word-wise and sentence-wise sentiment classification.

There were still some approaches that addressed polarity shift without complex linguistic analysis and extra annotations. For example, Li and Huang [19] proposed a method first to classify each sentence in a text into a polarity-unshifted part and a polarity-shifted part according to certain rules, then to represent them as two bags-of-words for sentiment classification. Li et al. [21] further proposed a method to separate the shifted and unshifted text based on training a binary detector. Classification models are then trained based on each of the two parts. An ensemble of two component classifiers is used to provide the final polarity of the whole text. Orimaye et al. [34] proposed a sentence polarity shift algorithm to identify consistent sentiment polarity patterns and use only the sentiment-consistent sentences for sentiment classification.

A preliminary version of this paper was published in [44]. In this paper, we extend our previous work in three major aspects. First, we strengthen the DSA algorithm by adding a selective data expansion procedure. Second,

[1] http://www.wjh.harvard.edu/~inquirer/

TABLE 1
An Example of Creating Reversed Training Reviews

	Review Text	Class
Original review	*I don't like this book. It is boring.*	Negative
Reversed review	*I like this book. It is interesting.*	Positive

we extend the DSA framework from sentiment polarity classification to positive-negative-neutral sentiment classification. Third, we propose a corpus-based method to construct a pseudo-antonym dictionary that could remove DSA's dependency on an external antonym dictionary.

2.2 Data Expansion Technique

The data expansion technique has been seen in the field of handwritten recognition [3], [40], where the performance of the handwriting recognition systems was significantly improved by adding some synthetic training data.

In the field of natural language processing and text mining, Agirre and Martinez [2] proposed expanding the amount of labeled data through a Web search using monosemous synonyms or unique expressions in definitions from WordNet for the task of word sense disambiguation. Fujita and Fujino [11] proposed a method that provides reliable training data using example sentences from an external dictionary.

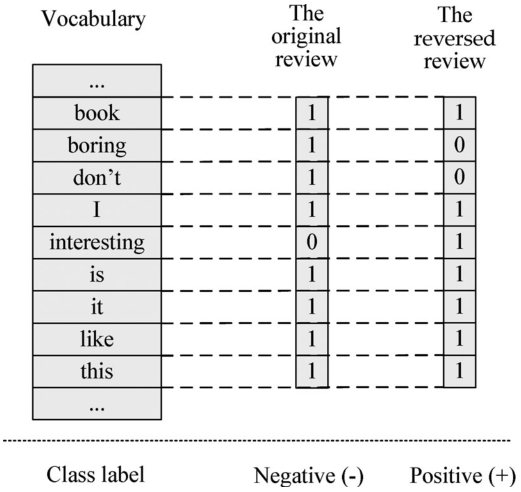
To the best of our knowledge, the data expansion technique proposed here is the first work that conducts data expansion in sentiment analysis. Different from the above mentioned techniques, the original and reversed reviews are constructed in a one-to-one correspondence. Another novel point of this work is that we expand the data set not only in the training stage, but also in the test stage. The original and reversed test review is used in pairs for sentiment prediction.

3 Data Expansion by Creating Reversed Reviews

In this section, we introduce the data expansion technique of creating sentiment-reversed reviews.

Based on an antonym dictionary, for each original review, the reversed review is created according to the following rules:

- *Text reversion*. If there is a negation, we first detect the scope of negation.[2] All sentiment words out of the scope of negation are reversed to their antonyms. In the scope of negation, negation words (e.g., "*no*", "*not*", "*don't*", etc.) are removed, but the sentiment words are not reversed;
- *Label reversion*. For each of the training review, the class label is also reversed to its opposite (i.e., positive to negative, or vice versa), as the class label of the reversed review.



Vocabulary	The original review	The reversed review
...		
book	1	1
boring	1	0
don't	1	0
I	1	1
interesting	0	1
is	1	1
it	1	1
like	1	1
this	1	1
...		
Class label	Negative (-)	Positive (+)

Fig. 1. The BOW representations of the original and the reversed reviews in TABLE 1.

Table 1 gives two simple examples of creating the reversed training reviews. Given an original training review, such as "*I don't like this book. It is boring.* (class: Negative)", the reversed review is obtained by three steps: 1) the sentiment word "*boring*" is reversed to its antonym "*interesting*"; 2) the negation word "*don't*" is removed. Since "*like*" is in the scope of negation, it is not reversed; 3) the class label is reversed from Negative to Positive. Note that in data expansion for the test data set, we only conduct Text Reversion. We make a joint prediction based on observation of both the original and reversed test reviews. Fig. 1 illustrates the BOW representations of the original and the reversed reviews.

Note that the created sentiment-reversed review might be not as good as the one generated by human beings. Since both the original and reversed review texts are represented by the BOW representation in machine learning, the word order and syntactic structure are totally ignored. Therefore, the requirement for keeping the grammatical quality in the created reviews is lower as that in human languages. Moreover, we will use a tradeoff parameter to leverage the original and reversed reviews in dual prediction. Assigning a relatively smaller weight to the reversed review can protect the model from being damaged by incorporating low-quality review examples.

4 Dual Sentiment Analysis

In this section, we present our DSA framework in detail. Fig. 2 illustrates the process of a DSA algorithm. It contains two main stages: 1) dual training and 2) dual prediction. In the following two subsections, we will introduce them respectively.

4.1 Dual Training

In the training stage, all of the original training samples are reversed to their opposites. We refer to them as "original training set" and "reversed training set" respectively. In our data expansion technique, there is a one-to-one correspondence between the original and reversed reviews. The classifier is trained by maximizing a combination of the

[2] In this work, we adopt the method in [6], [19], [35], which defined the scope of negation as the range from the negation word to the end of the sub-sentence. There were indeed some refined techniques for negation scope detection [5], [18]. But most of them depend on either human annotations of negation or very complex linguistic analysis. Even so, these problems are still not well addressed today.

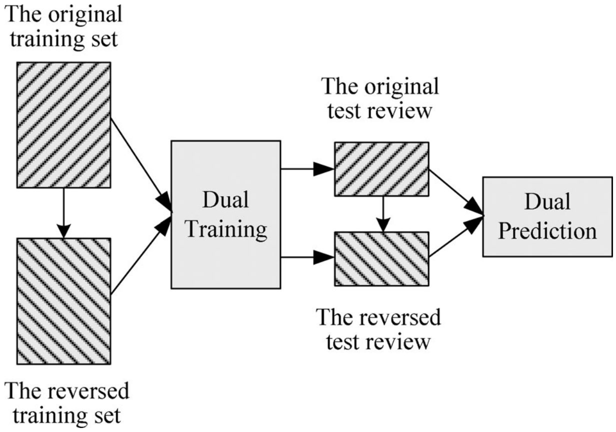


Fig. 2. The process of dual sentiment analysis. The rectangle filled with slash denotes the original data, and the rectangle filled with backslash denotes the reversed data.

likelihoods of the original and reversed training samples. This process is called dual training.

For simplicity, in this paper we derive the DT algorithm by using the logistic regression model as an example. Note that our method can be easily adapted to the other classifiers such as naïve Bayes and SVMs.[3] In the experiments, all of the three classification algorithms are examined.

Before we proceed, we first summarize in Table 2 some notations that will be used in the following descriptions. Let $\mathcal{D}=\{(x_i,y_i)\}_{i=1}^N$ and $\tilde{\mathcal{D}}=\{(\tilde{x}_i,\tilde{y}_i)\}_{i=1}^N$ be the original and reversed training sets, respectively, where *x* and $\tilde{x}$ denote the feature vector of the original and reversed reviews respectively, $y \in \{0,1\}$ denotes the original class label, $\tilde{y} = 1 - y$ denotes the reversed class label, and *N* is the number of the original training samples. Let w denote the weight of features, and *J(w)* be the cost function.

Logistic regression is a popular and widely-used statistical model for the binary classification problem. Logistic regression uses the logistic function to predict the probability of a feature vector *x* belonging to the positive class:

$$p(y=1|x) = h(x) = \frac{1}{1+e^{w^{\mathrm{T}}x}}, \tag{1}$$

where w is the weight of features remaining to be learnt.

In standard logistic regression, the cost function is known as the log-likelihood of the training data:

$$\begin{aligned} J(w) &= \sum_{i=1}^{N} \log p(y_i|x_i) \\ &= \sum_{i=1}^{N} y_i \log h(x_i) + (1-y_i) \log(1-h(x_i)). \end{aligned} \tag{2}$$

By contrast, in DT, a combination of the original and reversed training set is used for training. Therefore, the cost function contains two component parts:

TABLE 2
Some Notations in Dual Training and Dual Prediction

Notation	Description
x	The original sample
$\tilde{x}$	The reversed sample
$y \in \{0,1\}$	The class label of the original sample
$\tilde{y} = 1 - y$	The class label of the reversed sample
$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$	The original training set
$\tilde{\mathcal{D}} = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^N$	The reversed training set
w	Weights of features in a linear model
J(w)	Log-likelihood function
$p(\cdot\|x)$	Prediction for the original sample
$p(\cdot\|\tilde{x})$	Prediction for the reversed sample
$p(\cdot\|x,\tilde{x})$	Dual prediction based on a pair of samples

$$\begin{aligned} J_d(w) &= \sum_{i=1}^{N} \log p(y_i|x_i) + \sum_{i=1}^{N} \log p(\tilde{y}_i|\tilde{x}_i) \\ &= \sum_{i=1}^{N} \log p(y_i|x_i) + \log p(\tilde{y}_i|\tilde{x}_i) \\ &= \sum_{i=1}^{N} y_i \log h(x_i) + (1-y_i)\log(1-h(x_i)) \\ &\quad + \tilde{y}_i \log h(\tilde{x}_i) + (1-\tilde{y}_i)\log(1-h(\tilde{x}_i)). \end{aligned} \tag{3}$$

In polarity reversion, the class label of the training sample is reversed to its opposite. Therefore we have $\tilde{y}_i = 1 - y_i$. By using this property, we can further get the following cost function:

$$\begin{aligned} J_d(w) &= \sum_{i=1}^{N} [y_i \log h(x_i) + (1-y_i)\log(1-h(x_i))] \\ &\quad + [(1-y_i)\log h(\tilde{x}_i) + y_i \log(1-h(\tilde{x}_i))] \\ &= \sum_{i=1}^{N} y_i[\log h(x_i) + \log(1-h(\tilde{x}_i))] \\ &\quad + (1-y_i)[\log(1-h(x_i)) + \log h(\tilde{x}_i)] \\ &= \sum_{i=1}^{N} y_i \log[h(x_i)(1-h(\tilde{x}_i))] \\ &\quad + (1-y_i)\log[(1-h(x_i))h(\tilde{x}_i)]. \end{aligned} \tag{4}$$

Comparing the cost functions of standard logistic regression (Equation (2)) and our DT algorithm (Equation (4)), we can get more profound insights as follows:

- If x_i is a positive training sample, the standard likelihood score with respect to x_i is $\log h(x_i)$ While in DT, the likelihood score becomes $\log[h(x_i)(1-h(\tilde{x}_i))]$ That is, the feature weights in DT are learnt by considering not only how likely is x_i to be *positive*, but also how likely is x_i to be *negative*.
- If x_i is a negative training sample, the standard likelihood score with respect to x_i is $\log(1-h(x_i))$. While in DT, the likelihood becomes $\log[(1-h(x_i))h(\tilde{x}_i)]$. That is, the feature weights in DT are learnt by considering not only how likely is x_i to be *negative*, but also how likely is x_i to be *positive*.

[3] Dual training in naïve Bayes and SVMs could be conducted with the same manner as in logistic regression. The former uses a combined likelihood for training parameters, and the latter optimizes a combined hinge loss function.

Now let us use the example in Table 1 to explain the effectiveness of dual training in addressing the polarity shift problem. We assume "*I don't like this book. It is boring.* (class label: negative)" is the original training review. Hence, "*I like this book. It is interesting.* (class label: positive)" is reversed training review. Due to negation, the word "*like*" is (incorrectly) associated with the negative label in the original training sample. Hence, its weight will be added by a negative score in maximum likelihood estimation. Therefore, the weight of "*like*" will be falsely updated. While in DT, due to the removal of negation in the reversed review, "*like*" is (correctly) associated with the positive label, and its weight will be added by a positive score. Hence, the learning errors caused by negation can be partly compensated in the dual training process.

4.2 Dual Prediction

In the prediction stage, for each test sample x, we create a reversed test sample $\tilde{x}$. Note that our aim is not to predict the class of $\tilde{x}$. But instead, we use $\tilde{x}$ to assist the prediction of x. This process is called dual prediction.

Let $p(\cdot|x)$ and $p(\cdot|\tilde{x})$ denote posterior probabilities of x and $\tilde{x}$ respectively. In DP, predictions are made by considering two sides of one review:

- When we want to measure how *positive* a test review x is, we not only consider how *positive* the original test review is (i.e., $p(+|x)$), but also consider how *negative* the reversed test review is (i.e., $p(-|\tilde{x})$);
- Conversely, when we measure how *negative* a test review x is, we consider the probability of x being *negative* (i.e., $p(-|x)$), as well as the probability of $\tilde{x}$ being *positive* (i.e.,$p(+|\tilde{x})$).

A weighted combination of two component predictions is used as the dual prediction score:

$$\begin{cases} p(+|x,\tilde{x}) = (1-\alpha)\cdot p(+|x) + \alpha \cdot p(-|\tilde{x}), \\ p(-|x,\tilde{x}) = (1-\alpha)\cdot p(-|x) + \alpha \cdot p(+|\tilde{x}), \end{cases} \tag{5}$$

where α is a tradeoff parameter ($0 \leq \alpha \leq 1$). The weight of $p(\cdot|\tilde{x})$ is increased by choosing a larger α. In our experiments, α is quite stable. The best performance can always be obtained when $\alpha \in [0.5, 0.7]$.

Using $y \in \{0, 1\}$ to represent the negative class $(-)$ and positive class $(+)$, we get a compact form of the dual prediction function:

$$\begin{aligned} p(y|x,\tilde{x}) &= (1-\alpha)\cdot p(y|x) + \alpha \cdot p(1-y|\tilde{x}) \\ &= (1-\alpha)\cdot p(y|x) + \alpha \cdot [1 - p(y|\tilde{x})]. \end{aligned} \tag{6}$$

Let $p_d(y|x,\tilde{x})$ denote the dual prediction of x based on an already-trained DT model. In order to prevent DP algorithm from being damaged by low-confident predictions, instead of using all dual predictions $p_d(y|x,\tilde{x})$ as the final output, we use the original prediction $p_o(y|x)$ as an alternate, in case that the dual prediction $p_d(y|x,\tilde{x})$ is not enough confident. The final prediction is therefore defined as:

$$p_f(y|x) = \begin{cases} p_d(y|x,\tilde{x}), & \textit{if } \Delta p \geqslant t \\ p_o(y|x), & \text{otherwise} \end{cases} \tag{7}$$

where $p = p_d(y|x|x,\tilde{x}) - p_o(y|x)$, and t is a threshold parameter. In the experiments, we set t to be close to zero. That is, the prediction with a higher posterior probability will be chosen as the final prediction.

Let us use the example in Table 1 again to explain why dual prediction works in addressing the polarity shift problem. This time we assume "*I don't like this book. It is boring*" is an original test review, and "*I like this book. It is interesting*" is the reversed test review. In traditional BOW, "*like*" will contribute a high positive score in predicting overall orientation of the test sample, despite of the negation structure "*don't like*". Hence, it is very likely that the original test review will be mis-classified as Positive. While in DP, due to the removal of negation in the reversed review, "*like*" this time the plays a positive role. Therefore, the probability that the reversed review being classified into Positive must be high. In DP, a weighted combination of two component predictions is used as the dual prediction output. In this manner, the prediction error of the original test sample can also be compensated by the prediction of the reversed test sample. Apparently, this can reduce some prediction errors caused by polarity shift. In the experimental study, we will extract some real examples from our experiments to prove the effectiveness of both dual training and dual prediction.

4.3 DSA with Selective Data Expansion

In Section 4.1, we have introduced the dual training procedure, where all of the training reviews are used in data expansion. However, in many cases, not all of the reviews have such distinct sentiment polarity as the examples in Table 1 have. A natural question is hence: Is there the need to use all of the labeling reviews for data expansion and dual training? In this part, we further investigate this problem and subsequently propose a selective data expansion procedure to select a part of training reviews for data expansion.

Let us first observe two reviews which are a bit more complex than the previous examples:

- *Review (a). The book is very interesting, and the price is very cheap. I like it.*
- *Review (b). The book is somehow interesting, but the price is too expensive. I don't dislike it.*

In review (a), the sentiment is very strong and the polarity shift rate is low. In this case, the original review itself is a good labeling instance, and the reversed review will also be a good one. In review (b), the sentiment polarity is less distinct. In this case, the sentiment polarity of the reversed review is also not distinct and confident. Therefore, creating reversed review for review (b) is not that necessary in comparison with review (a).

Consequently, we propose a sentiment degree metric for selecting the most sentiment-distinct training reviews for data expansion. The degree of sentiment polarity could be measured by

$$m(x) = |p(+|x) - p(-|x)| \tag{8}$$

where $p(+|x)$ and $p(-|x)$ are the posterior probabilities predicted on the training review x.

TABLE 3
An Example of Data Expansion for Neutral Reviews

	Review Text	Class
Original review	*The room is large. But it is not clean.*	Neutral
Reversed review	*The room is small. But it is clean.*	Neutral

We use $m(x)$ as a criterion for selective data expansion. A threshold s will be set to select a percentage of original reviews with higher sentiment degree for data reversion and dual training. The cost function in DT then becomes

$$\begin{aligned} J_d(w) = \sum_{i=1}^{N} y_i \log\left[h(x_i)(1-h(\tilde{x}_i))\right] \\ + I(m(x) \geq s)(1-y_i) \log\left[(1-h(x_i))h(\tilde{x}_i)\right]. \end{aligned} \tag{9}$$

where $I(\cdot)$ is an indicator function.

In the experiments, we will discuss the effect of selective data expansion. We will show that using a selected part of training reviews for data expansion can achieve better performance than that using all reviews.

4.4 DSA for Positive-Negative-Neutral Sentiment Classification

Polarity classification is the most classical sentiment analysis task which aims at classifying reviews into either positive or negative. However, in many cases, in addition to the positive and negative reviews, there still exist many neutral reviews. The abovementioned DSA system does not have the ability to classify the neutral reviews. In this section, we extend the DSA framework to the scenario of three-class (positive-neutral-negative) sentiment classification. We call the DSA approach in three-class sentiment classification DSA3.

Naturally, neural review contains two main situations: 1) Neither positive nor negative (objective texts without expressing sentiment); 2) Mixed positive and negative (texts expressing mixed or conflicting sentiment). For both of the two cases, it is reasonable for us to assume that the sentiment of the reversed review is still neutral. Based on this assumption, in data expansion for neutral reviews, we only reverse the review text but keep its class label as neutral still. Table 3 gives an example of creating the reversed reviews for sentiment-mixed neutral reviews.

In DSA3, we first conduct training data expansion by creating reversed reviews. For a *negative* review, we create a *positive* one; for a *positive* review, we create a *negative* one; for a *neutral* review, we create a *neutral* one. The selective data expansion procedure is still used in this case, i.e., only the labeled data with high posterior probability will be used for data expansion.

In the training stage, a multi-class machine learning models, such as multi-class logistic regression (also called softmax regression), is trained based on the expanded dual training set.

In the prediction stage, for each original test sample x, we create an reversed one $\tilde{x}$. In order to take into account the neutral reviews, we update the previous prediction algorithm in Equation (5) as follows:

$$\begin{cases} p(+|x,\tilde{x}) = (1-\alpha)\cdot p_d(+|x) + \alpha \cdot p(-|\tilde{x}), \\ p(-|x,\tilde{x}) = (1-\alpha)\cdot p_d(-|x) + \alpha \cdot p(+|\tilde{x}), \\ p(*|x,\tilde{x}) = (1-\alpha)\cdot p(*|x) + \alpha \cdot p(*|\tilde{x}). \end{cases} \tag{10}$$

where $\{+,-,*\}$ denote the class labels of positive, negative and neutral, respectively. Specifically, we add one prediction rule for the neutral reviews. It is a weighted combination of the prediction of the original and reversed test reviews. Note that in this case, we can still guarantee that

$$p(+|x,\tilde{x}) + p(-|x,\tilde{x}) + p(*|x,\tilde{x}) = 1.$$

As we have mentioned in Section 4.2 that in dual prediction, when we measure how *positive/negative* a test review is, we not only consider how *positive/negative* the original review is, but also how *negative/positive* the reversed review is. In addition to that, in three-class sentiment classification, when we measure how *neutral* a test review is, we not only consider how *neutral* the original review is, but also how *neutral* the reversed review is.

5 The Antonym Dictionary for Review Reversion

So far we have presented the DSA model. However, we notice that DSA highly depends on an external antonym dictionary for review reversion. How to construct a suitable antonym dictionary by applying DSA into practice? It still remains an important problem.

5.1 The Lexicon-Based Antonym Dictionary

In the languages where lexical resources are abundant, a straightforward way is to get the antonym dictionary directly from the well-defined lexicons, such as WordNet[4] in English. WordNet is a lexical database which groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets. Using the antonym thesaurus it is possible to obtain the words and their opposites.

The WordNet antonym dictionary is simple and direct. However, in many languages other than English, such an antonym dictionary may not be readily available. Even if we can get an antonym dictionary, it is still hard to guarantee vocabularies in the dictionary are domain-consistent with our tasks.

To solve this problem, we furthermore develop a corpus-based method to construct a pseudo-antonym dictionary. This corpus-based pseudo-antonym dictionary can be learnt using the labeled training data only. The basic idea is to first use mutual information (MI) to identify the most positive-relevant and the most negative-relevant features, rank them in two separate groups, and pair the features that have the same level of sentiment strength as pair of antonym words.

5.2 The Corpus-Based Pseudo-Antonym Dictionary

In information theory, the mutual information of two random variables is a quantity that measures the mutual dependence of the two random variables. MI is widely used

[4] http://wordnet.princeton.edu/

TABLE 4
The Datasets in Sentiment Classification

Dataset	#positive	#negative	#neutral	average length	#features
Book	1,000	1,000	–	201	23,833
DVD	1,000	1,000	–	197	23,216
Electronics	1,000	1,000	–	126	12,148
Kitchen	1,000	1,000	–	105	10,260
Hotel (Chinese)	2,000	2,000	–	85	18,900
Notebook (Chinese)	2,000	2,000	–	38	10,402
Kithcen (3-class)	736	728	719	138	11,238
Network (3-class)	483	482	435	141	8,832
Health (3-class)	857	854	856	108	10,638

as a feature selection method in text categorization and sentiment classification [20].

First, we choose all adjectives, adverbs and verbs in the training corpus as candidate features, and use the MI metric to calculate the relevance of each candidate feature w_i to the positive (+) and negative (-) class, respectively:

$$\begin{cases} MI(w_i,+) = \log\frac{p(w_i,+)}{p(w_i)p(+)} \\ MI(w_i,-) = \log\frac{p(w_i,-)}{p(w_i)p(-)} \end{cases} \tag{11}$$

Then, we rank two groups of features in a decreasing order of $MI(w_i,+)$ and $MI(w_i,-)$ respectively:

$$\begin{cases} \mathcal{W}^+ = [w_1^+, w_2^+, \cdots, w_D^+] \\ \mathcal{W}^- = [w_1^-, w_2^-, \cdots, w_D^-] \end{cases} \tag{12}$$

Finally, we obtain the pseudo-antonym dictionary by zipping w^+ and w^i. Specifically, a positive-relevant word and a negative-relevant word that have the same ranking positions (e.g., $\{w_i^+, w_i^-\}$) are matched as a pair of antonym words.

It is important to notice that, rather than a common-sense antonym dictionary, it is a “pseudo” antonym dictionary, Here, “pseudo” means a pair of antonym words are not really semantic-opposite, but have opposite sentiment strength. As we have stated in Section 3, both the original and created reviews are represented as a vector of independent words in the BOW representation. Therefore, it is not that important whether the created review is grammatically correct or not. We just need to maintain the level of sentiment strength in review reversion. Apparently, the mutual information provides a good measure of the contextual sentiment strength. Therefore, the condition of the same level sentiment strength can be required by pairing the positive- and negative-relevant words with the same ranking positions as antonyms.

Moreover, because the pseudo-antonym dictionary is learnt from the training corpus, it has a good property: language-independent and domain-adaptive. This property makes the DSA model possible to be applied into a wider range, especially when the lexical antonym dictionary is not available across different languages and domains.

In the experimental study, we will evaluate the effect of the MI-based pseudo-antonym dictionary by conducting experiments on two Chinese datasets to. We also compare the results of two kinds of antonym dictionaries on the English multi-domain sentiment datasets, and provide some discussions on the choice of them in real practice.

6 Experimental Study

In this section, we systematically evaluate our approach on two tasks including polarity classification and positive-negative-neutral sentiment classification across nine sentiment datasets, three classification algorithms, two types of features and two kinds of antonym dictionaries.

6.1 Datasets and Experimental Settings

For polarity classification, we use four English datasets and two Chinese datasets. The Multi-Domain Sentiment Datasets[5] are used as the English datasets. They contain product reviews taken from Amazon.com including four different domains: Book, DVD, Electronics and Kitchen. Each of the reviews is rated by the customers from Star-1 to Star-5. The reviews with Star-1 and Star-2 are labeled as Negative, and those with Star-4 and Star-5 are labeled as Positive. Each of the four datasets contains 1,000 positive and 1,000 negative reviews. The Chinese datasets contain two domains extracted from the ChnSentiCorp corpus[6]: Hotel and Notebook. Each of them contains 2,000 positive and 2,000 negative reviews.

For 3-class (positive-negative-neutral) sentiment classification, we collect three datasets of reviews taken from three product domains (Kitchen, Network and Health) of Amazon.com, which are similar to the Multi-Domain Sentiment Datasets. But we do not only extract reviews with Star-1, Star-2, Star-4 and Star-5, but also reviews with Star-3. The reviews with Star-3 are labeled as the Neutral category. Table 4 summarizes some detailed information of the nine datasets.

In our experiments, reviews in each category are randomly split up into five folds (with four folds serving as training data and the remaining one fold serving as test data). All of the following results are reported in terms of an averaged accuracy of five-fold cross validation.

We implement the naïve Bayes Classifier based on a multinomial event model with Laplace smoothing [31]. The LibSVM[7] toolkit is chosen as the SVM classifier. Setting of kernel function is linear kernel, the penalty parameter is set as the default value (i.e., one), and the Platt’s probabilistic

[5] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/
[6] http://www.searchforum.org.cn/tansongbo/corpus/
[7] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

TABLE 5
Classification Accuracy of Polarity Classification Using Linear SVM Classifier

Dataset	Features: unigrams					Features: unigrams and bigrams				
	Baseline	**DS**	**LSS**	**DSA-WN**	**DSA-MI**	**Baseline**	**DS**	**LSS**	**DSA-WN**	**DSA-MI**
Book	0.745	0.763	0.760	0.792	**0.801**	0.775	0.777	0.788	0.809	**0.816**
DVD	0.764	0.771	0.795	0.803	**0.811**	0.790	0.793	0.809	0.816	**0.823**
Electronics	0.796	0.813	0.812	0.829	**0.838**	0.818	0.834	0.841	0.849	**0.851**
Kitchen	0.822	0.820	0.844	**0.864**	0.861	0.847	0.844	0.870	**0.879**	0.875
Avg.	0.782	0.792	0.802	0.822	**0.828**	0.808	0.812	0.827	0.838	**0.841**
Hotel (Chinese)	0.827	0.833	0.847	–	**0.877**	0.862	0.866	0.872	–	**0.886**
Notebook (Chinese)	0.883	0.893	0.895	–	**0.918**	0.910	0.914	0.917	–	**0.927**
Avg.	0.855	0.863	0.871	–	**0.898**	0.886	0.890	0.895	–	**0.907**

TABLE 6
Classification Accuracy of Polarity Classification Using Naïve Bayes Classifier

Dataset	Features: unigrams					Features: unigrams and bigrams				
	Baseline	**DS**	**LSS**	**DSA-WN**	**DSA-MI**	**Baseline**	**DS**	**LSS**	**DSA-WN**	**DSA-MI**
Book	0.779	0.783	0.792	**0.818**	0.808	0.811	0.815	0.822	**0.837**	0.828
DVD	0.795	0.793	0.810	**0.824**	0.821	0.824	0.826	0.837	**0.844**	0.840
Electronics	0.815	0.828	0.824	**0.844**	0.843	0.841	0.857	0.852	0.859	**0.860**
Kitchen	0.830	0.847	0.840	**0.864**	**0.864**	0.878	0.879	0.883	**0.895**	0.893
Avg.	0.804	0.813	0.817	**0.838**	0.834	0.838	0.844	0.848	**0.859**	0.855
Hotel (Chinese)	0.844	0.858	0.855	–	**0.873**	0.869	0.876	0.876	–	**0.886**
Notebook (Chinese)	0.899	0.905	0.906	–	**0.915**	0.915	0.920	0.920	–	**0.923**
Avg.	0.872	0.881	0.881	–	**0.894**	0.892	0.898	0.898	–	**0.905**

output for SVM is applied to approximate the posterior probabilities. The LibLinear[8] toolkit is used as the logistic regression model with all parameters set to be the default value (e.g., the regularization parameter is one). Following the standard experimental settings in sentiment classification, we use term presence (i.e., Boolean value) as the weight of feature, and evaluate two kinds of features, 1) unigrams, 2) both unigrams and bigrams. Note that we do not aim to compare different classification algorithms and different features. Our aim in this work is to evaluate our DSA model under various settings. The paired t-test [45] is performed for significant testing with a default significant level of 0.05.

6.2 Experiments on Polarity Classification

In this section, we first report the experimental results on the polarity classification task. For this task, we evaluate the following five systems that are proposed in the literature with the aim at addressing polarity shift.

1) *Baseline*. The standard machine learning methods based on the BOW representation;
2) *DS*. The method proposed by [6], where *"NOT"* is attached to the words in the scope of negation, e.g., *"The book is not interesting"* is converted to *"The book is interesting-NOT"*;
3) *LSS*. The method proposed by [21], where each text is split up into two parts: polarity-shifted and polarity-unshifted, based on which two component classifiers are trained and combined for sentiment classification. To our knowledge, this is the state-of-the-art approach of considering polarity shift without using external resources;
4) *DSA-WN*. The DSA model with selective data expansion and the WordNet antonym dictionary;
5) *DSA-MI*. The DSA model with selective data expansion and the MI-based pseudo-antonym dictionary.

In Sections 6.2.1 and 6.2.2, we report the results on four English datasets and two Chinese datasets, respectively.

6.2.1 Results on the Multi-Domain Sentiment Datasets

From Tables 5, 6, and 7 (in the next page), we report the classification accuracy of five evaluated systems using 1) unigram features and 2) both unigram and bigram features, based on three classifiers, i.e, linear SVM, naïve Bayes, and logistic regression, respectively.

We first observe the results on linear SVM. In Table 5, we can see that compared to the Baseline system, the average improvements of the DS approach are very limited (1.0 and 0.4 percent on two kinds of features respectively). The performance of LSS is more effective, but the improvements are limited. It improves the average score by 2.0 percent on unigram features, and 1.9 percent on both unigram and bigram features. By contrast, our DSA approach achieves the best performance. As for DSA-WN, in comparison with the Baseline system, the improvements on unigram features are 4.7, 3.9, 3.3 and 4.2 percent (4.0 percent on average) across four datasets. On unigram and bigram features, it outperforms the Baseline system by 3.4, 2.6, 3.1 and 3.2 percent (3.0 percent on average). Compared with the LSS

[8] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

TABLE 7
Classification Accuracy of Polarity Classification Using Logistic Regression Classifier

Dataset	Features: unigrams					Features: unigrams and bigrams				
	Baseline	DS	LSS	DSA-WN	DSA-MI	Baseline	DS	LSS	DSA-WN	DSA-MI
Book	0.771	0.775	0.784	0.809	**0.815**	0.779	0.789	0.809	0.823	**0.824**
DVD	0.785	0.800	0.815	0.826	**0.827**	0.801	0.802	0.823	0.831	**0.836**
Electronics	0.803	0.815	0.823	**0.842**	**0.842**	0.826	0.833	0.844	**0.857**	0.856
Kitchen	0.835	0.841	0.851	**0.875**	0.872	0.851	0.858	0.872	**0.886**	0.883
Avg.	0.798	0.808	0.818	0.838	**0.839**	0.814	0.821	0.837	0.849	**0.850**
Hotel (Chinese)	0.856	0.867	0.864	–	**0.879**	0.876	0.877	0.883	–	**0.888**
Notebook (Chinese)	0.904	0.907	0.911	–	**0.922**	0.913	0.914	0.919	–	**0.927**
Avg.	0.880	0.887	0.888	–	**0.901**	0.895	0.8955	0.901	–	**0.908**

system, the improvements of the average score are 2.0 and 1.1 percent on the two kinds of features respectively. All of the improvements are significant according to the paired *t*-test. As for DSA-MI, it gains even higher improvements than DSA-WN on linear SVM classifier. For unigram features, compared with the Baseline system, the improvements are 5.6, 4.7, 4.2 and 3.9 percent (4.5 percent on average) across the Multi-domain datasets. For unigram and bigram features, it improves the Baseline system by 3.3 percent on the average score. In comparison with LSS, the average improvements are 2.6 and 1.4 percent on two kinds of features. All of the differences are significant according to the paired *t*-test.

Apart from the linear SVM classifier, we also report the classification accuracy based on naïve Bayes and logistic regression in Tables 6 and 7 respectively. As we can see, the DS approach still achieves very slight improvements (less than 1 percent). The improvements of LSS are also limited: for using unigram features, 1.3 and 2.0 percent on naïve Bayes and logistic regression, respectively; for using both unigram and bigram features, 1.0 and 2.3 percent, respectively on naïve Bayes and logistic regression. While the previous two systems are not effective, the improvements of our DSA approach are significant. In Table 6, DSA-WN improves the Baseline system by 3.4 and 2.1 percent, and outperforms LSS by 2.1 and 1.1 percent on average on two kinds of features respectively. In Table 7, we could also observe that for unigram features, it improves the average score by 4.0 and 2.0 percent compared with the Baseline system and the LSS approach respectively; for both unigram and bigram features, it improves Baseline and LSS by 3.5 and 1.2 percent, respectively. The improvements of DSA-MI are also sound on naïve Bayes and logistic regression. In Table 6, it improves the Baseline system by 3.0 and 1.7 percent on average, and outperforms LSS by 1.7 and 0.7 percent on two kinds of features respectively. In Table 7, for unigram features, it improves the average score by 4.1 and 2.1 percent compared with the Baseline system and LSS; for using both unigram and bigram features, it improves the Baseline System by on average 3.6 percent, and outperforms LSS by 1.3 percent across the Multi-domain datasets. All of the improvements are significant in the paired *t*-test.

6.2.2 Results on two Chinese Sentiment Datasets

In the previous part, we have compared our DSA approach with three related systems on four English datasets. In this part, we will further report the experimental results on two Chinese datasets. It is worthy noting that compared with DSA-WN, DSA-MI it is a totally corpus-based method which does not rely on external lexicons. Thus, DSA-MI could be applied into a wide range, especially when the lexical antonym dictionary is not available. In this experiment, we did not resort to a Chinese antonym dictionary for DSA. We focus on the performance of DSA-MI.

First, we take the results on the linear SVM for observation. In Table 5, we can easily observe that the performance of DSA-MI is sound. For unigram features, the improvements are 5.0 and 3.5 percent on the two datasets, in comparison with the Baseline system. For unigram and bigram features, it improves the Baseline system by 2.4 and 1.7 percent on the two datasets. In comparison with LSS, the improvements on the Hotel dataset are 3.0 and 1.4 percent on the two kinds of features. On the Notebook dataset, although the accuracies of LSS are already very high (0.895 and 0.917), DSA-MI still achieves significant improvements (2.3 and 1.0 percent).

As for naïve Bayes and logistic regression classifiers, the improvements of DSA-MI are relatively smaller, yet still significant. In Table 6, it improves the Baseline system by 2.2 and 1.2 percent on the average score on two kinds of features, and performs relatively better than the LSS system (0.881 versus 0.894, and 0.898 versus 0.905). In Table 7, for unigram features, DSA-MI improves the average score by 2.0 and 1.3 percent compared with Baseline and LSS; for that using both unigram and bigram features, the average improvements are a bit small (0.895 versus 0.901 versus 0.908). It is acceptable because the baselines are already very high (e.g., 0.904 and 0.913 on the Notebook dataset).

Generally speaking, on two Chinese sentiment datasets, although we do not use an external dictionary, our DSA-MI approach is still effective and it outperforms alternative systems significantly. The results prove the feasibility and effectiveness of our DSA model, in case that we do not have a lexical antonym dictionary for data expansion.

6.3 Experiments on Positive-Negative-Neutral Sentiment Classification

In this section, we will report the experimental results on the 3-class (positive-neutral-negative) sentiment classification task. For this task, we evaluate the following five systems.

1) *Multi-class*. The direct multi-class classification algorithm such as multi-class logistic regression;

TABLE 8
Classification Accuracy of Three-Class (Positive-Negaitve-Neutral) Sentiment Classification

Dataset	Multi-class	Hierarchy	OVO	OVA	DSA3
Health	0.735	0.721	0.626	0.720	**0.758**
Kitchen	0.710	0.685	0.615	0.701	**0.739**
Network	0.674	0.648	0.580	0.664	**0.711**
Avg.	0.706	0.685	0.607	0.695	**0.736**

2) *Hierarchy*. A hierarchical classification system used in [42], where the neutrality is determined first and sentiment polarity is determined second.
3) *OVO*. An ensemble system of one-vs-one base classifiers proposed in [49]. In OVO, three binary classifiers (including positive/negative, positive/neutral, and negative/neutral) are trained at first. Then a special ensemble rule is applied to yield the final prediction.
4) *OVA*. An ensemble system of one-versus-all base classifiers.[9] In OVA, two binary classifiers (positive/non-positive, and positive/non-positive) are trained at first. Then a four-way classification rule is used: positive (+pos, -neg), negative (-pos, +neg), neutral (-pos, -neg or +pos, +neg).
5) *DSA3*. The extended DSA model for positive-neutral-negative sentiment classification, proposed in Section 4.4.

In Table 8, we compare the three-class sentiment classification accuracy of the five systems on three datasets. In Hierarchy, OVO and OVA where the three-class classification is converted to several binary classification subtasks, we use the logistic regression classifier. In Multi-class and DSA3, we use the multi-class logistic regression classifier. Due to space limitation, we only report the result on unigram features, similar conclusions can be drawn by using unigram and bigram feature together.

Seen from Table 8, we can find that the OVO method fails in three-class sentiment classification. It performs consistently the worst among five systems. The Hierarchy and OVA methods yield comparative performance (0.685 versus 0.695), which are significantly higher than OVO. The Multi-class method is the best in the previous four base systems. It shows that directly modeling 3 categories is better than a vote (or ensemble) of several binary subtasks in positive-negative-neutral sentiment classification. As for our DSA3 model, it outperforms the Multi-class, Hierarchy, OVO and OVA by 3, 5.1, 12.9 and 4.1 percent, respectively. All of the improvements are significant according to the paired *t*-test. It shows that the extended DSA model is quite efficient in positive-negative-neutral sentiment classification.

6.4 Discussion on the Effects of DSA in Addressing Polarity Shift

In this subsection, we try to explain why the DSA model could address the polarity shift problem, based on both artificial examples and some real examples extracted from our experiments.

6.4.1 A Case when Dual Training Works

We first discuss the effectiveness of dual training. Let us take a look at a real test sample extracted from the Electronics dataset:

- *Original review*. I found that these dvd-rs *did not work* well in my system, were *unreliable* and slow. I *can not recommend* them.
- *Reversed review*. I found that these dvd-rs *work* well in my system, were *excellent* and slow. I *can recommend* them.

We use the underlines to denote the changes in polarity reversion. Note that in the reversed sample, two negations (*"did not work well"* and *"can not recommend"*) are removed, and some new pseudo-opposite words are introduced (*"unreliable"* -> *"excellent"*).

We observe the results of the traditional method (i.e., the Baseline model) and our DSA model, respectively. The prediction of the traditional method is false ($p(+|x) = 0.58$), probably because two negations are not handled ("well" and *"recommend"* contribute high positive scores in prediction). Based on dual training, two component predictions in our DSA (predictions of the original sample and its reversed one) are $p_d(+|x) = 0.38$ and $p_d(-|\tilde{x}) = 0.30$ respectively. Note that both of them are correct, even without dual prediction. The dual prediction $p_d(-|\tilde{x})$ is more confident than the original predition $p_d(+|x)$, due to the removal of negation in polarity reversion. As a weighted combination of two component predictions, the final dual prediction makes the result more robust:

$$p_d(+|x, \tilde{x}) + 0.5p_d(-|x) + 0.5p_d(-|\tilde{x}) = 0.34.$$

6.4.2 A Case when Dual Prediction Works

Now let us take a look at another real example, where $p_d(+|x)$ is still false, but $p_d(-|\tilde{x})$ is correct. That is, only applying DT is not enough to correct the error. We observe how DP corrects the error in this case.

- *Original review*: *Sorry*, the speakers *don't attach* well, and the quality of these stands *is not* what I'm used to with a bose system.
- *Reversed review*. *Pleasantly*, the speakers *attach* well, and the quality of these stands is what I'm used to with a bose system.

In this example, the traditional prediction is incorrect ($p(-|x) = 0.54$). This time, two component predictions of the DSA model are contradictory: $p_d(+|x) = 0.60$ and $p_d(-|\tilde{x}) = 0.16$. The original prediction $p_d(+|x)$ is still false. But $p_d(-|\tilde{x})$ is correct and it is more confident than $p_d(+|x)$, because the negation is removed. Finally, the dual prediction, as a weighted combination of $p_d(+|x)$ and $p_d(-|\tilde{x})$, is correct:

$$p_d(+|x, \tilde{x}) + 0.5p_d(-|x) + 0.5p_d(-|\tilde{x}) = 0.38.$$

6.5 Discussion on the Effectiveness of Selective Data Expansion

In this section, we discuss the effect of selective dual training. In Fig. 3, we report the performance of DSA by selecting an increasing percentage of training reviews for data

[9] This method was proposed by Breckbaldwin in http://lingpipe-blog.com/2008/01/02/positive-negative-and-neutral-sentiment/

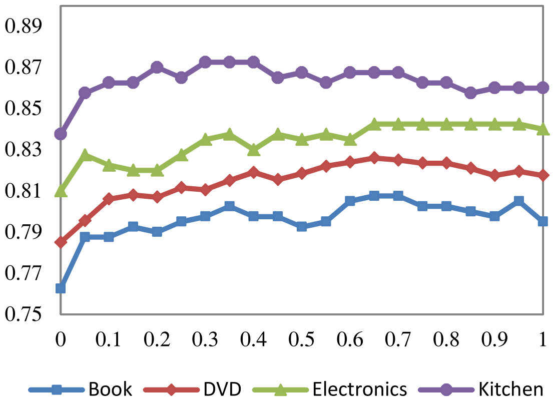


Fig. 3. The effect of selective data expansion in DSA. The *x*-axis denotes the percentage of selected samples. The *y*-axis denotes the sentiment classification accuracy of DSA.

TABLE 9
The Top-10 Pairs of Pseudo-Antonym Words Learnt from the Book Domain and the Electronics Domain

Book domain		Electronics domain	
Positive	**Negative**	**Posotive**	**Negative**
beautifully	*weak*	*great*	*unacceptable*
straight	*dull*	*excellent*	*unreliable*
vivid	*whatsoever*	*crisp*	*back*
gorgeous	*boring*	*easy*	*terrible*
universal	*mediocre*	*vivid*	*sadly*
visual	*repectitive*	*highly*	*painful*
wonderful	*credible*	*best*	*fatal*
excellent	*vague*	*good*	*blank*
easy	*bad*	*perfect*	*repeatddly*
great	*instead*	*terrific*	*broken*

expansion. Note that due to space limitation, we only present a representative result of DSA on the multi-domain sentiment datasets by using the logistic regression classifier and unigram features. Similar conclusions can be drawn in the other experimental settings.

Note that when the percentage is 0, no training samples are used for data expansion. In this case, DSA equals the standard Baseline system. When the percentage is 1, all of the training samples are used for data expansion.

We first observe the performance using all training reviews. It yields significantly better classification performance in comparison with the Baseline system that does not use data expansion. We can further observe that with a percentage of selected training reviews for data expansion, DSA can achieve comparative or even better performance than that using all reviews. For example, in the Book, DVD and Electronics domains, the best classification performance can be obtained by using 60-70 percent selected training reviews. In the Kitchen domain, using 30-40 percent of the training reviews can even obtain significantly better performance than that using all the ones. It suggests that it is not case that the more training reviews are used in data expansion, the better system performance DSA has. With a selected part of training reviews for data expansion, we might get better classification results.

6.6 Discussion on Two Types of Antonym Dictionaries

We further compare the performance of two different antonym dictionaries (i.e., DSA-WN and DSA-MI) and discuss the choice of them in real practice. DSA-WN uses the lexical antonym dictionary extracted from WordNet, and DSA-MI uses the corpus-based pseudo-antonym dictionary learnt from the training corpus based on the MI metric. In Table 9, we display two real pseudo-antonym dictionaries learnt from the training corpus of two domains (Book and Electronics) of the Multi-domain sentiment datasets.

From Tables 5, 6 and 7, we could easily compare the classification accuracy of DSA-WN and DSA-MI. It can be observed from the three tables that, DSA-WN and DSA-MI gain comparable performances on the Multi-domain sentiment datasets. Across different classifiers and datasets, there is no consistent winner between the two methods. Take the linear SVM classifier as an example. For unigram features, DSA-WN yields a better result on the Kitchen dataset (0.864 versus 0.861), but on the Book, DVD and Electronics datasets, DSA-MI outperforms DSA-WN slightly. For using both unigrams and bigrams, DSA-WN still performs better on the Kitchen dataset compared with DSA-MI, but slightly worse on the Book, DVD and Electronics datasets. As for the naïve Bayes and logistic regression classifiers, we can find that the conclusions are similar to linear SVM classifier, and the difference between the accuracy of the two algorithms is less than 1 percent across most of the datasets. It is reasonable because although the lexical antonym dictionary includes more standard and precise antonym words, the corpus-based pseudo-antonym dictionary is also good at obtaining more domain-relevant antonym words by learning from the corpus. Most of the differences of two antonym systems are not significant in the paired *t*-test.

In general, we can conclude that the performances of two types of antonym dictionaries are comparable. But we should note that we do not always have a good lexical dictionary. Given that a myriad of languages do not have good antonym dictionaries, or these dictionaries, if any, cannot match the vocabularies of the specific domain in our task, the corpus-based pseudo-antonym dictionary is a better choice for DSA. In comparison with the lexical antonym dictionary, the corpus-based pseudo-antonym dictionary is language independent and domain adaptive, The two advantages make the DSA algorithm more convenient to use and more applicable across different languages and domains.

6.7 Discussion on the Applicability of DSA in More Generalized Situations

In this paper, we focus on supervised sentiment classification. It should be noted that the DSA framework could be applied into a wider range of sentiment analysis tasks, such as unsupervised, semi-supervised sentiment classification, as well as class-imbalanced sentiment classification.

In case of unsupervised sentiment classification, we can create the reversed reviews for each testing example, integrate the dual prediction rule into a term counting methods and make a joint prediction based on two sides of one review.

In case of semi-supervised sentiment classification, in addition to conduct dual training and dual prediction respectively on the labeled and test data, we could also create the reversed reviews for each unlabeled example and select some reliable ones that are measured by the original and reversed views together for constructing extra labeled traning data.

In case of imbalanced sentiment classification, a commonly-used method is to conduct re-sampling technique (e.g., under-sampling and over-sampling) to construct a number of balanced datasets, and then use the ensemble technique to combine the component results. Each component task is a balanced sentiment classification problem, where we can directly apply the DSA algorithm proposed here.

7 Conclusions and Future Work

In this work, we propose a novel data expansion approach, called DSA, to address the polarity shift problem in sentiment classification. The basic idea of DSA is to create reversed reviews that are sentiment-opposite to the original reviews, and make use of the original and reversed reviews in pairs to train a sentiment classifier and make predictions. DSA is highlighted by the technique of one-to-one correspondence data expansion and the manner of using a pair of samples in training (dual training) and prediction (dual prediction). A wide range of experiments demonstrate that the DSA model is very effective for polarity classification and it significantly outperforms several alternative methods of considering polarity shift.

In addition, we strengthen the DSA algorithm by developing a selective data expansion technique that chooses training reviews with higher sentiment degree for data expansion. The experimental results show that using a selected part of training reviews for data expansion can yield better performance than that using all reviews.

We furthermore extend the DSA algorithm to DSA3, which could deal with three-class (positive-negative-neutral) sentiment classification. We update the dual training and dual prediction algorithm by taking the neutral reviews into consideration. The experimental results also prove the effectiveness of DSA3 in three-class sentiment classification.

Finally, to remove DSA's dependency on an external antonym dictionary, we propose a corpus-based method to construct a pseudo-antonym dictionary. The experiments on four English sentiment datasets show that DSA using the pseudo-antonym dictionary (DSA-MI) can yield comparable performance that using the WordNet antonym dictionary (DSA-WN). In terms of practical applicability, DSA-MI has major implications especially for sentiment analysis tasks with limited lexical resource and domain knowledge. We also conduct experiments on two Chinese sentiment datasets without using external antonym dictionary, and the results prove the feasibility of the DSA-MI approach.

In this paper, we focus on creating reversed reviews to assist supervised sentiment classification. In the future, we can generalize the DSA algorithm to a wider range of sentiment analysis tasks. We also plan to consider more complex polarity shift patterns such as transitional, subjunctive and sentiment-inconsistent sentences in creating reversed reviews.

Acknowledgments

The work was supported by the Natural Science Foundation of China (61305090 and 61272419), the Jiangsu Provincial Natural Science Foundation of China (BK2012396), the Jiangsu Provincial Social Science Foundation of China (14SZB018), the US National Science Foundation (CNS-1126619 and IIS-1213026), and the Research Fund for the Doctoral Program of Higher Education of China (20123219120025). Tao Li is the corresponding author.

References

[1] A. Abbasi, S. France, Z. Zhang, and H. Chen, "Selecting attributes for sentiment classification using feature relation networks," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 3, pp. 447–462, Mar. 2011.

[2] E. Agirre, and D. Martinez, "Exploring automatic word sense disambiguation with decision lists and the web," in *Proc. COLING Workshop Semantic Annotation Intell. Content*, 2000, pp. 11–19.

[3] J. Cano, J. Perez-Cortes, J. Arlandis, and R. Llobet, "Training set expansion in handwritten character recognition," in *Proc. Struct., Syntactic, Statistical Pattern Recognit.*, 2002, pp. 548–556.

[4] Y. Choi and C. Cardie, "Learning with compositional semantics as structural inference for subsentential sentiment analysis," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2008, pp. 793–801.

[5] I. Councill, R. MaDonald, and L. Velikovich, "What's great and what's not: Learning to classify the scope of negation for improved sentiment analysis," in *Proc. Workshop Negation Speculation Natural Lang. Process.*, 2010, pp. 51–59.

[6] S. Das and M. Chen, "Yahoo! for Amazon: Extracting market sentiment from stock message boards," in *Proc. Asia Pacific Finance Assoc. Annu.Conf.*, 2001.

[7] K. Dave, S. Lawrence and D. Pen-nock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proc. Int. World Wide Web Conf.*, 2003, pp. 519–528.

[8] X. Ding and B. Liu, "The utility of linguistic rules in opinion mining," in *Proc. 30th ACM SIGIR Conf. Res. Development Inf. Retrieval*, 2007, pp. 811–812.

[9] X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining," in *Proc. Int. Conf. Web Search Data Mining*, 2008, pp. 231–240.

[10] X. Ding, B. Liu, and L. Zhang, "Entity discovery and assignment for opinion mining applications," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 1125–1134.

[11] S. Fujita and A. Fujino, "Word sense disambiguation by combining labeled data expansion and semi-supervised learning method," in *Proc. Int. Joint Conf. Natural Language Process.*, 2011, pp. 676–685.

[12] M. Gamon, "Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis," in *Proc. Int. Conf. Comput. Linguistics*, 2004, pp. 841–847.

[13] M. Hu and B. Liu, "Mining opinion features in customer reviews," in *Proc. AAAI Conf. Artif. Intell.*, 2004, pp. 755–760.

[14] D. Ikeda, H. Takamura, L. Ratinov, and M. Okumura, "Learning to shift the polarity of words for sentiment classification," in *Proc. Int. Joint Conf. Natural Language Process.*, 2008.

[15] W. Jin, H. H. Ho, and R. K. Srihari, "OpinionMiner: A novel machine learning system for web opinion mining and extraction," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 1195–1204.

[16] S. Kim and E. Hovy, "Determining the sentiment of opinions," in *Proc. Int. Conf. Comput. Linguistic*, 2004, pp. 1367–1373.

[17] A. Kennedy and D. Inkpen, "Sentiment classification of movie reviews using contextual valence shifters," *Comput. Intell.*, vol. 22, pp. 110–125, 2006.

[18] J. Li, G. Zhou, H. Wang, and Q. Zhu, "Learning the scope of negation via shallow semantic parsing," in *Proc. Int. Conf. Comput. Linguistics*, 2010, pp. 671–679.

[19] S. Li and C. Huang, "Sentiment classification considering negation and contrast transition," in *Proc. Pacific Asia Conf. Lang., Inf. Comput.*, 2009, pp. 307–316.

[20] S. Li, R. Xia, C. Zong and C. Huang, "A framework of feature selection methods for text categorization," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2009, pp. 692–700.

[21] S. Li, S. Lee, Y. Chen, C. Huang and G. Zhou, “Sentiment classification and polarity shifting,” in *Proc. Int. Conf. Comput. Linguistics*, 2010, pp. 635–643.

[22] T. Li, Y. Zhang, and V. Sindhwani, “A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge,” in *Proc. 47th Annu. Meeting Assoc. Comput. Linguistics*, 2009, pp. 244–252.

[23] T. Li, V. Sindhwani, C. Ding, and Y Zhang, “Bridging domains with words: Opinion analysis with matrix tri-factorizations,” in *Proc. 10th SIAM Conf. Data Mining*, 2010, pp. 293–302.

[24] C. Lin and Y. He, “Joint sentiment/topic model for sentiment analysis,” in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 375–384.

[25] C. Lin, Y. He, R. Everson, and S. Ruger, “Weakly supervised joint sentiment-topic detection from text,” *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 6, pp. 1134–1145, Jun. 2012.

[26] B. Liu, “Sentiment analysis and opinion mining,” in *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1. San Rafael, CA, USA: Morgan & Claypool, 2012, pp. 1–165.

[27] R. Morante and W. Daelemans, “A metalearning approach to processing the scope of negation,” in *Proc. 30th Conf. Comput. Natural Language Learning*, 2009, pp. 21–29.

[28] J. Na, H. Sui, C. Khoo, S. Chan, and Y. Zhou, “Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews,” in *Proc. Conf. Int. Soc. Knowl. Org.*, 2004, pp. 49–54.

[29] T. Nakagawa, K. Inui, and S. Kurohashi. “Dependency tree-based sentiment classification using CRFs with hidden variables,” in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2010, pp. 786–794.

[30] V. Ng, S. Dasgupta and S. Arifin, “Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews,” in *Proc. Int. Conf. Comput. Linguistics Annu. Meeting Assoc. Comput. Linguistics*, 2006, pp. 611–618.

[31] A. McCallum and K. Nigam, “A comparison of event models for naive Bayes text classification,” in *Proc. AAAI Workshop Learn. Text Categorization*, 1998, pp. 307–314.

[32] P. Melville, W. Cryc, and R. D. Lawrence, “Sentiment analysis of blogs by combining lexical knowledge with text classification,” in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 1275–1284.

[33] R. Mihalcea and D. Moldovan, “An automatic method for generating sense tagged corpora,” in *Proc. AAAI Conf. Artif. Intell.*, 1999, pp. 461–466.

[34] S. Orimaye, S. Alhashmi, and E. Siew, “Buy it—don’t buy it: Sentiment classification on Amazon reviews using sentence polarity shift,” in *Proc. 12th Pacific Rim Int. Conf. Artif. Intell.*, 2012, pp. 386–399.

[35] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: Sentiment classification using machine learning techniques,” in *Proc. Conf. Empirical Methods Natural Language Process.*, 2002, pp. 79–86.

[36] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” *Found. Trends Inf. Retrieval*, vol. 2, no. 1/2, pp. 1–135, 2008.

[37] L. Polanyi and A. Zaenen, “Contextual lexical valence shifters,” in *Proc. AAAI Spring Symp. Exploring Attitude Affect Text*, 2004, pp. 1–10.

[38] P. Turney, “Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews,” *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 417–424.

[39] P. Turney and M. L. Littman, “Measuring praise and criticism: Inference of semantic orientation from association,” *ACM Trans. Inf. Syst.*, vol. 21, no. 4, pp. 315–346, 2003.

[40] T. Varga and H. Bunke, “Generation of synthetic training data for an HMM-based handwriting recognition system,” in *Proc. IEEE Int. Conf. Document Anal. Recognit.*, 2003, pp. 618–622.

[41] C. Whitelaw, N. Garg, S. Argamon, “Using appraisal groups for sentiment analysis,” in *Proc. ACM Conf. Inf. Knowl. Manag.*, 2005, pp. 625–631.

[42] T. Wilson, J. Wiebe, and P. Hoffmann, “Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis,” *Comput. Linguistics*, vol. 35, no. 3, pp. 399–433, 2009.

[43] R. Xia, C. Zong, and S. Li, “Ensemble of feature sets and classification algorithms for sentiment classification,” *Inf. Sci.*, vol. 181, no. 6, pp. 1138–1152, 2011.

[44] R. Xia, T. Wang, X. Hu, S. Li, and C. Zong, “Dual training and dual prediction for polarity classification,” in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2013, pp. 521–525.

[45] Y. Yang and X. Liu, “A re-examination of text categorization methods,” in *Proc. ACM SIGIR Conf. Res. Development Inf. Retrieval*, 1999, pp. 42–49.

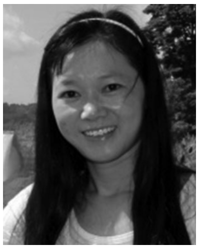
[46] X. Yu, Y. Liu, X. Huang, and A. An, “Mining online reviews for predicting sales performance: A case study in the movie domain,” *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 4, pp. 720–734, Apr. 2012.

[47] L. Zhuang, F. Jing, and X. Zhu, “Movie review mining and summarization,” in *Proc. 15th ACM Int. Conf. Inf. Knowl. Manage.*, 2006, pp. 43–50.

[48] Z. Hai, K. Chang, J. Kim, and C. C. Yang, “Identifying features in opinion mining via intrinsic and extrinsic domain relevance,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 447–462, Mar. 2014.

[49] M. Koppel and J. Schler, “Using neutral examples for learning polarity,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2005, pp. 1616–1617.

Rui Xia received the BSc degree from Southeast University, Nanjing, China, in 2004, the MSc degree from East China University of Science and Technology, Shanghai, China, in 2007, and the PhD degree from the Institute of Automation, Chinese Academy of Sciences in 2011. He is currently an assistant professor in the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. His research interests include natural language processing, machine learning, and data mining.



Feng Xu received the BSc degree from Nanjing University Information Science and Technology, China, in 2004, and the MSc degree from Southeast University, China, in 2007. She is currently working toward the PhD degree in the School of Economics and Management, Nanjing University of Science and Technology, China. She is currently a lecturer in Chengxian College at Southeast University. Her research interests include financial management, social computing, and data mining.

Chengqing Zong received the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences in 1998. He is a professor in the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include natural language processing, machine translation, and sentiment analysis. He is a director in the Chinese Association of Artificial Intelligence and the Society of Chinese Information Processing, and a member of International Committee on Computational Linguistics (ICCL). He is an associate editor of the *ACM Transactions on Asian Language Information Processing* and an editorial board member of *IEEE Intelligent Systems*, *Machine Translation*, and the *Journal of Computer Science and Technology*.

Qianmu Li received the BSc and PhD degrees from Nanjing University of Science and Technology, China, in 2001 and 2005, respectively. He is currently a professor in the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. His research interests include machine learning, data mining, network and distributed system, and information security.

Yong Qi received the BSc degree from East China Institute of Technology, China, in 1992, and the MSc and PhD degrees from Nanjing University of Science and Technology, China, in 1999 and 2005, respectively. He is currently a professor in the School of Computer Science and Engineering, and School of Economics and Management, Nanjing University of Science and Technology, China. His research interests include machine learning, data mining, and social computing.

Tao Li received the PhD degree in computer science from the Department of Computer Science, University of Rochester, Rochester, NY, in 2004. He is currently a professor with the School of Computing and Information Sciences, Florida International University, Miami, FL. He is also a professor with the School of Computer Science & Technology, Nanjing University of Posts and Telecommunications (NJUPT). His research interests include data mining, computing system management, information retrieval, and machine learning. He received the US National Science Foundation (NSF) CAREER Award and multiple IBM Faculty Research Awards.