Dual Training and Dual Prediction for Polarity Classification

Rui Xia, Tao Wang, Xuelei Hu
Department of Computer Science
Nanjing University of
Science and Technology
rxia@njust.edu.cn,
linclonwang@163.com,
xlhu@njust.edu.cn

Shoushan Li
NLP Lab
Department of
Computer Science
Soochow University
shoushan.li
@gmail.com

Chengqing Zong
National Lab of
Pattern Recognition
Institute of Automation
CAS
cqzong
@nlpr.ia.ac.cn

Abstract

Bag-of-words (BOW) is now the most popular way to model text in machine learning based sentiment classification. However, the performance of such approach sometimes remains rather limited due to some fundamental deficiencies of the BOW model. In this paper, we focus on the polarity shift problem, and propose a novel approach, called dual training and dual prediction (DTDP), to address it. The basic idea of DTDP is to first generate artificial samples that are polarity-opposite to the original samples by polarity reversion, and then leverage both the original and opposite samples for (dual) training and (dual) prediction. Experimental results on four datasets demonstrate the effectiveness of the proposed approach for polarity classification.

1 Introduction

The most popular text representation model in machine learning based sentiment classification is known as the bag-of-words (BOW) model, where a piece of text is represented by an unordered collection of words, based on which standard machine learning algorithms are employed as classifiers. Although the BOW model is simple and has achieved great successes in topic-based text classification, it disrupts word order, breaks the syntactic structures and discards some kinds of semantic information that are possibly very important for sentiment classification. Such disadvantages sometimes limit the performance of sentiment classification systems.

A lot of subsequent work focused on feature engineering that aims to find a set of effective features based on the BOW representation. However, there still remain some problems that are not well addressed. Out of them, the polarity shift problem is the biggest one.

We refer to "polarity shift" as a linguistic phenomenon that the sentiment orientation of a text is reversed (from positive to negative or vice versa) because of some particular expressions called polarity shifters. Negation words (e.g., "no", "not" and "don't") are the most important type of polarity shifter. For example, by adding a negation word "don't" to a positive text "I like this book" in front of "like", the orientation of the text is reversed from positive to negative.

Naturally, handling polarity shift is very important for sentiment classification. However, the BOW representations of two polarity-opposite texts, e.g., "*I like this book*" and "*I don't like this book*", are considered to be very similar by most of machine learning algorithms. Although some methods have been proposed in the literature to address the polarity shift problem (Das and Chen, 2001; Pang et al., 2002; Na et al., 2004; Kenndey and Inkpen, 2006; Ikeda et al., 2008; Li and Huang, 2009; Li et al., 2010), the state-of-the-art results are still far from satisfactory. For example, the improvements are less than 2% after considering polarity shift in Li et al. (2010).

In this work, we propose a novel approach, called dual training and dual prediction (DTDP), to address the polarity shift problem. By taking advantage of the unique nature of polarity classification, DTDP is motivated by first generating artificial samples that are polarity-opposite to the original ones. For example, given the original sample "*I don't like this book. It is boring*," its polarity-opposite version, "*I like this book. It is interesting*", is artificially generated. Second, the original and opposite training samples are used together for training a sentiment classifier (called dual training), and the original and opposite test samples are used together for prediction (called dual prediction). Experimental results prove that the procedure of DTDP is very effective at correcting the training and prediction errors caused

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 521–525,
Sofia, Bulgaria, August 4-9 2013.

by polarity shift, and it beats other alternative methods of considering polarity shift.

2 Related Work

The lexicon-based sentiment classification systems can be easily modified to include polarity shift. One common way is to directly reverse the sentiment orientation of polarity-shifted words, and then sum up the orientations word by word (Hu and Liu, 2004; Kim and Hovy, 2004; Polanyi and Zaenen, 2004; Kennedy and Inkpen, 2006). Wilson et al. (2005) discussed other complex negation effects by using conjunctive and dependency relations among polarity words. Although handling polarity shift is easy and effective in term-counting systems, they rarely outperform the baselines of machine learning methods (Kennedy, 2006).

The machine learning methods are generally more effective for sentiment classification. However, it is difficult to handle polarity shift based on the BOW model. Das and Chen (2001) 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*" is changed to a new word "*like*-NOT". There were also some attempts to model polarity shift by using more complex linguistic features (Na et al., 2004; Kennedy and Inkpen, 2006). But the improvements upon the baselines of machine learning systems are very slight (less than 1%).

Ikeda et al. (2008) proposed a machine learning method, to model polarity-shifters for both word-wise and sentence-wise sentiment classification, based on a dictionary extracted from General Inquirer. Li and Huang (2009) 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 bag-of-words for sentiment classification. Li et al. (2010) 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 parts is used at last to get the final polarity of the whole text.

3 The Proposed Approach

We first present the method for generating artificial polarity-opposite samples, and then introduce the algorithm of dual training and dual prediction (DTDP).

3.1 Generating Artificial Polarity-Opposite Samples

Given an original sample and an antonym dictionary (e.g., WordNet [1]), a polarity-opposite sample is generated artificially according to the following rules:

1) **Sentiment word reversion:** All sentiment words out of the scope of negation are reversed to their antonyms;
2) **Handling negation:** If there is a negation expression, we first detect the scope of negation, and then remove the negation words (e.g., "no", "not", and "don't"). The sentiment words in the scope of negation are not reversed;
3) **Label reversion:** The class label of the labeled sample is also reversed to its opposite (i.e., Positive to Negative, or vice versa) as the class label of newly generated samples (called polarity-opposite samples).

Let us use a simple example to explain the generation process. Given the original sample:

The original sample
Text: *I don't like this book. It is boring.*
Label: Negative

According to Rule 1, "*boring*" is reversed to its antonym "*interesting*"; According to Rule 2, the negation word "*don't*" is removed, and "*like*" is not reversed; According to Rule 3, the class label Negative is reversed to Positive. Finally, an artificial polarity-opposite sample is generated:

The generated opposite sample
Text: *I like this book. It is interesting.*
Label: Positive

All samples in the training and test set are reversed to their polarity-opposite versions. We refer to them as "opposite training set" and "opposite test set", respectively.

3.2 Dual Training and Dual Prediction

In this part, we introduce how to make use of the original and opposite training/test data together for dual training and dual prediction (DTDP).

Dual Training: 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 opposite training set respectively, where x denotes the feature vector, y denotes the class label, and N denotes the size of training set. In dual training, $\mathcal{D} \cup \tilde{\mathcal{D}}$ are used together as training data to learn

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

a classification model. The size of training data is doubled in dual training.

Suppose the example in Section 3.1 is used as one training sample. As far as only the original sample ("*I don't like this book. It is boring.*") is considered, the feature "*like*" will be improperly recognized as a negative indicator (since the class label is Negative), ignoring the expression of negation. Nevertheless, if the generated opposite sample ("*I like this book. It is interesting."*) is also used for training, "*like*" will be learned correctly, due to the removal of negation in sample reversion. Therefore, the procedure of dual training can correct some learning errors caused by polarity shift.

Dual Prediction: Given an already-trained classification model, in dual prediction, the original and opposite test samples are used together for prediction. In dual prediction, when we predict the positive degree of a test sample, we measure not only how positive the original test sample is, but also how negative the opposite sample is.

Let x and $\tilde{x}$ denote the feature vector of the original and opposite test samples respectively; let $p_d(c|x)$ and $p_d(c|\tilde{x})$ denote the predictions of the original and opposite test sample, based on the dual training model. The dual predicting function is defined as:

$$p_d(+|x,\tilde{x}) = (1-a)p_d(+|x) + ap_d(-|\tilde{x}),$$
$$p_d(-|x,\tilde{x}) = (1-a)p_d(-|x) + ap_d(+|\tilde{x}),$$

where a ($0 \leqslant a \leqslant 1$) is the weight of the opposite prediction.

Now suppose the example in Section 3.1 is a test sample. As far as only the original test sample ("*I don't like this book. It is boring.*") is used for prediction, it is very likely that it is falsely predicted as Positive, since "*like*" is a strong positive feature, despite that it is in the scope of negation. While in dual prediction, we still measure the "sentiment-opposite" degree of the opposite test sample ("*I like this book. It is interesting."*). Since negation is removed, it is very likely that the opposite test sample is assigned with a high positive score, which could compensate the prediction errors of the original test sample.

Final Output: It should be noted that although the artificially generated training and testing data are helpful in most cases, they still produce some noises (e.g., some poorly generated samples may violate the quality of the original data set). Therefore, instead of using all dual predictions as the final output, we use the original prediction $p_o(c|x)$ as an alternate, in case that the dual prediction $p_d(c|x,\tilde{x})$ is not enough confident, according to a confidence threshold t. The final output is defined as:

$$p_f(c|x) = \begin{cases} p_d(c|x,\tilde{x}), & if \Delta p \geqslant t \\ p_o(c|x), & if \Delta p < t \end{cases}$$

where $\Delta p = p_d(c|x,\tilde{x}) - p_o(c|x)$.

4 Experimental Study

4.1 Datasets

The Multi-Domain Sentiment Datasets[2] are used for evaluations. They consist of product reviews collected from four different domains: Book, DVD, Electronics and Kitchen. Each of them contains 1,000 positive and 1,000 negative reviews. Each of the datasets is randomly spit into 5 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 average of 5-fold cross validation.

4.2 Evaluated Systems

We evaluate four machine learning systems that are proposed to address polarity shift in document-level polarity classification:

1) **Baseline**: standard machine learning methods based on the BOW model, without handling polarity shift;
2) **Das-2001**: the method proposed by Das and Chen (2001), where "NOT" is attached to the words in the scope of negation as a preprocessing step;
3) **Li-2010**: the approach proposed by Li et al. (2010). The details of the algorithm is introduced in related work;
4) **DTDP**: our approach proposed in Section 3. The WordNet dictionary is used for sample reversion. The empirical value of the parameter a and t are used in the evaluation.

4.3 Comparison of the Evaluated Systems

In table 1, we report the classification accuracy of four evaluated systems using unigram features. We consider two widely-used classification algorithms: SVM and Naïve Bayes. For SVM, the LibSVM toolkit[3] is used with a linear kernel and the default penalty parameter. For Naïve Bayes, the OpenPR-NB toolkit[4] is used.

[2] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/
[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[4] http://www.openpr.org.cn

Dataset	SVM				Naïve Bayes			
	Baseline	Das-2001	Li-2010	DTDP	Baseline	Das-2001	Li-2010	DTDP
Book	0.745	0.763	0.760	**0.800**	0.779	0.783	0.792	**0.814**
DVD	0.764	0.771	0.795	**0.823**	0.795	0.793	0.810	**0.820**
Electronics	0.796	0.813	0.812	**0.828**	0.815	0.827	0.824	**0.841**
Kitchen	0.822	0.820	0.844	**0.849**	0.830	0.847	0.840	**0.859**
Avg.	0.782	0.792	0.803	**0.825**	0.804	0.813	0.817	**0.834**

Table 1: Classification accuracy of different systems using unigram features

Dataset	SVM				Naïve Bayes			
	Baseline	Das-2001	Li-2010	DTDP	Baseline	Das-2001	Li-2010	DTDP
Book	0.775	0.777	0.788	**0.818**	0.811	0.815	0.822	**0.840**
DVD	0.790	0.793	0.809	**0.828**	0.824	0.826	0.837	**0.868**
Electronics	0.818	0.834	0.841	**0.848**	0.841	0.857	0.852	**0.866**
Kitchen	0.847	0.844	0.870	**0.878**	0.878	0.879	0.883	**0.896**
Avg.	0.808	0.812	0.827	**0.843**	0.839	0.844	0.849	**0.868**

Table 2: Classification accuracy of different systems using both unigram and bigram features

Compared to the Baseline system, the Das-2001 approach achieves very slight improvements (less than 1%). The performance of Li-2010 is relatively effective: it improves the average score by 0.21% and 0.13% on SVM and Naïve Bayes, respectively. Yet, the improvements are still not satisfactory.

As for our approach (DTDP), the improvements are remarkable. Compared to the Baseline system, the average improvements are 4.3% and 3.0% on SVM and Naïve Bayes, respectively. In comparison with the state-of-the-art (Li-2010), the average improvement is 2.2% and 1.7% on SVM and Naïve Bayes, respectively.

We also report the classification accuracy of four systems using both unigrams and bigrams features for classification in Table 2. From this table, we can see that the performance of each system is improved compared to that using unigrams. It is now relatively difficult to show improvements by incorporating polarity shift, because using bigrams already captured a part of negations (e.g., "*don't like*").

The Das-2001 approach still shows very limited improvements (less than 0.5%), which agrees with the reports in Pang et al. (2002). The improvements of Li-2010 are also reduced: 1.9% and 1% on SVM and Naïve Bayes, respectively.

Although the improvements of the previous two systems are both limited, the performance of our approach (DTDP) is still sound. It improves the Baseline system by 3.7% and 2.9% on SVM and Naïve Bayes, respectively, and outperforms the state-of-the-art (Li-2010) by 1.6% and 1.9% on SVM and Naïve Bayes, respectively.

5 Conclusions

In this work, we propose a method, called dual training and dual prediction (DTDP), to address the polarity shift problem in sentiment classification. The basic idea of DTDP is to generate artificial samples that are polarity-opposite to the original samples, and to make use of both the original and opposite samples for dual training and dual prediction. Experimental studies show that our DTDP algorithm is very effective for sentiment classification and it beats other alternative methods of considering polarity shift.

One limitation of current work is that the tuning of parameters in DTDP (such as a and t) is not well discussed. We will leave this issue to an extended version.

Acknowledgments

The research work is supported by the Jiangsu Provincial Natural Science Foundation of China (BK2012396), the Research Fund for the Doctoral Program of Higher Education of China (20123219120025), and the Open Project Program of the National Laboratory of Pattern Recognition (NLPR). This work is also partly supported by the Hi-Tech Research and Development Program of China (2012AA011102 and 2012AA011101), the Program of Introducing Talents of Discipline to Universities (B13022), and the Open Project Program of the Jiangsu Key Laboratory of Image and Video Understanding for Social Safety (30920130122006).

References

S. Das and M. Chen. 2001. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific Finance Association Annual Conference*.

M. Hu and B. Liu. 2004. Mining opinion features in customer reviews. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

D. Ikeda, H. Takamura L. Ratinov M. Okumura. 2008. Learning to Shift the Polarity of Words for Sentiment Classification. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*.

S. Kim and E. Hovy. 2004. Determining the sentiment of opinions. In *Proceeding of the International Conference on Computational Linguistics (COLING)*.

A. Kennedy and D. Inkpen. 2006. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22:110–125.

S. Li and C. Huang. 2009. Sentiment classification considering negation and contrast transition. In *Proceedings of the Pacific Asia Conference on Language, Information and Computation (PACLIC)*.

S. Li, S. Lee, Y. Chen, C. Huang and G. Zhou. 2010. Sentiment Classification and Polarity Shifting. In *Proceeding of the International Conference on Computational Linguistics (COLING)*.

J. Na, H. Sui, C. Khoo, S. Chan, and Y. Zhou. 2004. Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews. In *Proceeding of the Conference of the International Society for Knowledge Organization*.

B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

L. Polanyi and A. Zaenen. 2004. Contextual lexical valence shifters. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text, AAAI technical report*.

P. Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceeding of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

T. Wilson, J. Wiebe, and P. Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.