Semi-Stacking for Semi-supervised Sentiment Classification

Shoushan Li[†‡], **Lei Huang**[†], **Jingjing Wang**[†], **Guodong Zhou**[†*]

[†]Natural Language Processing Lab, Soochow University, China
[‡] Collaborative Innovation Center of Novel Software Technology and Industrialization
{shoushan.li, lei.huang2013, djingwang}@gmail.com,
gdzhou@suda.edu.cn

Abstract

In this paper, we address semi-supervised sentiment learning via semi-stacking, which integrates two or more semi-supervised learning algorithms from an ensemble learning perspective. Specifically, we apply *meta*-learning to predict the unlabeled data given the outputs from the member algorithms and propose *N*-fold cross validation to guarantee a suitable size of the data for training the *meta*-classifier. Evaluation on four domains shows that such a semi-stacking strategy performs consistently better than its member algorithms.

1 Introduction

The past decade has witnessed a huge exploding interest in sentiment analysis from the natural language processing and data mining communities due to its inherent challenges and wide applications (Pang et al., 2008; Liu, 2012). One fundamental task in sentiment analysis is sentiment classification, which aims to determine the sentimental orientation a piece of text expresses (Pang et al., 2002). For instance, the sentence "*I absolutely love this product.*" is supposed to be determined as a *positive* expression in sentimental orientation.

While early studies focus on supervised learning, where only labeled data are required to train the classification model (Pang et al., 2002), recent studies devote more and more to reduce the heavy dependence on the large amount of labeled data by exploiting semi-supervised learning approaches, such as co-training (Wan, 2009), label propagation (Sindhwani and Melville, 2008), and deep learning (Zhou et al., 2013), to sentiment classification. Empirical evaluation on various domains demonstrates the effectiveness of the unlabeled data in enhancing the performance of sentiment classification. However, semi-supervised sentiment classification remains challenging due to the following reason.

Although various semi-supervised learning algorithms are now available and have been shown to be successful in exploiting unlabeled data to improve the performance in sentiment classification, each algorithm has its own characteristic with different pros and cons. It is rather difficult to tell which performs best in general. Therefore, it remains difficult to pick a suitable algorithm for a specific domain. For example, as shown in Li et al. (2013), the co-training algorithm with personal and impersonal views yields better performances in two product domains: Book and Kitchen, while the label propagation algorithm yields better performances in other two product domains: DVD and Electronic.

In this paper, we overcome the above challenge above by combining two or more algorithms instead of picking one of them to perform semi-supervised learning. The basic idea of our algorithm ensemble approach is to apply *meta*-learning to re-predict the labels of the unlabeled data after obtaining their results from the member algorithms. First, a small portion of labeled samples in the initial labeled data, namely *meta*-samples, are picked as unlabeled samples and added into the initial unlabeled data to form a new unlabeled data. Second, we use the remaining labeled data as the new labeled data to perform semi-supervised learning with each member algorithm. Third, we collect the *meta*-samples' probability results from all member algorithms to train a *meta*-learning classifier (called *meta*-classifier). Forth and finally, we utilize the *meta*-classifier to re-predict the unlabeled samples as new automatically-labeled samples. Due to the limited number of labeled data in semi-supervised learning, we use *N*-fold cross validation to obtain more *meta*-samples for better learning the *meta*-classifier. In principle,

[*] Corresponding author

the above ensemble learning approach could be seen as an extension of the famous stacking approach (Džeroski and Ženko, 2004) to semi-supervised learning. For convenience, we call it semi-stacking.

The remainder of this paper is organized as follows. Section 2 overviews the related work on semi-supervised sentiment classification. Section 3 proposes our semi-stacking strategy to semi-supervised sentiment classification. Section 4 is the experimentation. Section 5 gives the conclusion and future work.

2 Related Work

Early studies on sentiment classification mainly focus on supervised learning methods with algorithm designing and feature engineering (Pang et al., 2002; Cui et al., 2006; Riloff et al., 2006). Recently, most studies on sentiment classification aim to improve the performance by exploiting unlabeled data in two main aspects: semi-supervised learning (Dasgupta and Ng, 2009; Wan, 2009; Li et al., 2010) and cross-domain learning (Blitzer et al. 2007; He et al. 2011; Li et al., 2013). Specifically, existing approaches to semi-supervised sentiment classification could be categorized into two main groups: bootstrapping-style and graph-based.

As for bootstrapping-style approaches, Wan (2009) considers two different languages as two views and applies co-training to conduct semi-supervised sentiment classification. Similarly, Li et al. (2010) propose two views, named personal and impersonal views, and apply co-training to use unlabeled data in a monolingual corpus. More recently, Gao et al. (2014) propose a feature subspace-based self-training to semi-supervised sentiment classification. Empirical evaluation demonstrates that subspace-based self-training outperforms co-training with personal and impersonal views.

As for graph-based approaches, Sindhwani and Melville (2008) first construct a document-word bipartite graph to describe the relationship among the labeled and unlabeled samples and then apply label propagation to get the labels of the unlabeled samples.

Unlike above studies, our research on semi-supervised sentiment classification does not merely focus on one single semi-supervised learning algorithm but on two or more semi-supervised learning algorithms with ensemble learning. To the best of our knowledge, this is the first attempt to combine two or more semi-supervised learning algorithms in semi-supervised sentiment classification.

3 Semi-Stacking for Semi-supervised Sentiment Classification

In semi-supervised sentiment classification, the learning algorithm aims to learn a classifier from a small scale of labeled samples, named initial labeled data, with a large number of unlabeled samples. In the sequel, we refer the labeled data as $L=\{(x_i, y_i)\}_{i=1}^{n_L}$ where $x_i \in \mathbf{R}^d$ is the d dimensional input vector, and y_i is its output label. The unlabeled data in the target domain is denoted as $U=\{(x_k)\}_{k=1}^{n_U}$. Suppose l^{semi} is a semi-supervised learning algorithm. The inputs of l^{semi} are L and U, and the output is $U'=\{(x_k, y_k)\}_{k=1}^{n_U}$ which denotes the unlabeled data with automatically assigned labels. Besides the labeled results, it is always possible to obtain the probability results, denoted as $P^{U'}$, which contains the posterior probabilities belonging to the positive and negative categories of each unlabeled sample, i.e., < $p(pos \mid x_k), p(neg \mid x_k)$ >. For clarity, some important symbols are listed in Table 1.

Table 1: Symbol definition

Symbol	Definition
L	Labeled data
U	Unlabeled data
U'	Unlabeled data with automatically assigned labels
$P^{U'}$	The probability result of unlabeled data
l^{super}	A supervised learning algorithm
l^{semi}	A semi-supervised learning algorithm
c_{meta}	The *meta*-classifier obtained from *meta*-learning
c_{test}	The test classifier for classifying the test data

3.1 Framework Overview

In our approach, two member semi-supervised learning algorithm are involved, namely, l_1^{semi} and l_2^{semi} respectively, and the objective is to leverage both of them to get a better-performed semi-supervised learning algorithm. Our basic idea is to apply *meta*-learning to re-predict the labels of the unlabeled data given the outputs from the member algorithms. Figure 1 shows the framework of our implementation of the basic idea. The core component in semi-stacking is the *meta*-classifier learned from the *meta*-learning process, i.e., c_{meta}. This classifier aims to make a better prediction on the unlabeled samples by combining two different probability results from the two member algorithms.

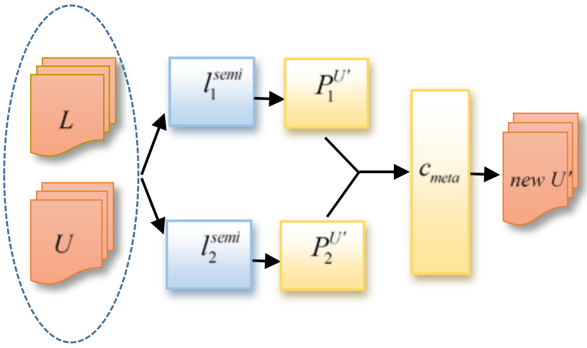

Figure 1: The framework of *semi-stacking*

3.2 *Meta*-learning

As shown above, *meta*-classifier is the core component in *semi-stacking*, trained through the *meta*-learning process. Here, *meta*- means the learning samples are not represented by traditional descriptive features, e.g., bag-of-words features, but by the result features generated from member algorithms. In our approach, the learning samples in *meta*-learning are represented by the posterior probabilities of the unlabeled samples belonging to the *positive* and *negative* categories from member algorithms, i.e.,

$$x^{meta} =< p_1(pos \mid x_k), p_1(neg \mid x_k), p_2(pos \mid x_k), p_2(neg \mid x_k) > \qquad (1)$$

Where $p_1(pos \mid x_k)$ and $p_1(neg \mid x_k)$ are the posterior probabilities from the first semi-supervised learning algorithm while $p_2(pos \mid x_k)$ and $p_2(neg \mid x_k)$ are the posterior probabilities from the second semi-supervised learning algorithm.

The framework of the *meta*-learning process is shown in Figure 2. In detail, we first split the initial labeled data into two partitions, L_{new} and L_{un} where L_{new} is used as the new initial labeled data while L_{un} is merged into the unlabeled data U to form a new set of unlabeled data $L_{un} + U$. Then, two semi-supervised algorithms are performed with the labeled data L_{new} and the unlabeled data $L_{un} + U$. Third and finally, the probability results of L_{un}, together with their real labels are used as *meta*-learning samples to train the *meta*-classifier. The feature representation of each *meta*-sample is defined in Formula (1).

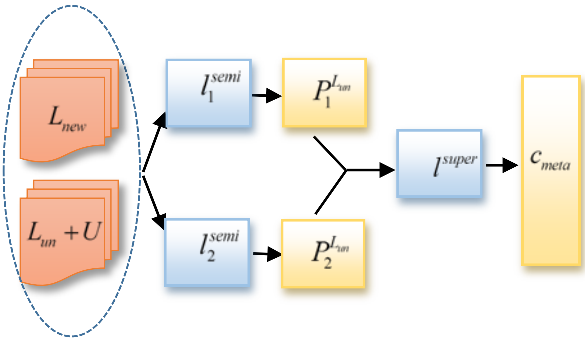

Figure 2: The framework of *meta*-learning

3.3 *Meta*-learning with *N*-fold Cross Validation

```
Input: Labeled data $L$, Unlabeled data $U$
Output: The meta-classifier $c_{meta}$
Procedure:
  (a) Initialize the meta-sample set $S_{meta} = \varnothing$
  (b) Split $L$ into $N$ folds, i.e., $L = L_1 + L_2 + \ldots L_N$
  (c) For $i$ in $1:N$:
      c1) $L_{new} = L - L_i$, $L_{un} = L_i$
      c2) Perform $l_1^{semi}$ on $L_{new}$ and $L_{un} + U$
      c3) Perform $l_2^{semi}$ on $L_{new}$ and $L_{un} + U$
      c4) Generate the meta-samples, $S_{meta}^i$, from the probability results of $L_{un}$ in the above two steps.
      c5) $S_{meta} = S_{meta} + S_{meta}^i$
  (d) Train the meta-classifier $c_{meta}$ with $S_{meta}$ and $l^{super}$
```

Figure 3: The algorithm description of meta-learning with *N*-fold cross validation

One problem of *meta*-learning is that the data size of L_{un} might be too small to learn a good meta-classifier. To better use the labeled samples in the initial labeled data, we employ *N*-fold cross validation to generate more meta- samples. Specifically, we first split L into N folds. Then, we select one of them as L_{un} and consider the others as L_{new} and generate the meta-learning samples as described in Section 3.2; Third and finally, we repeat the above step $N-1$ times by selecting a different fold as L_{un} in each time. In this way, we can obtain the *meta*-learning samples with the same size as the initial labeled data. Figure 3 presents the algorithm description of *meta*-learning with *N*-fold cross validation. In our implementation, we set N to be 10

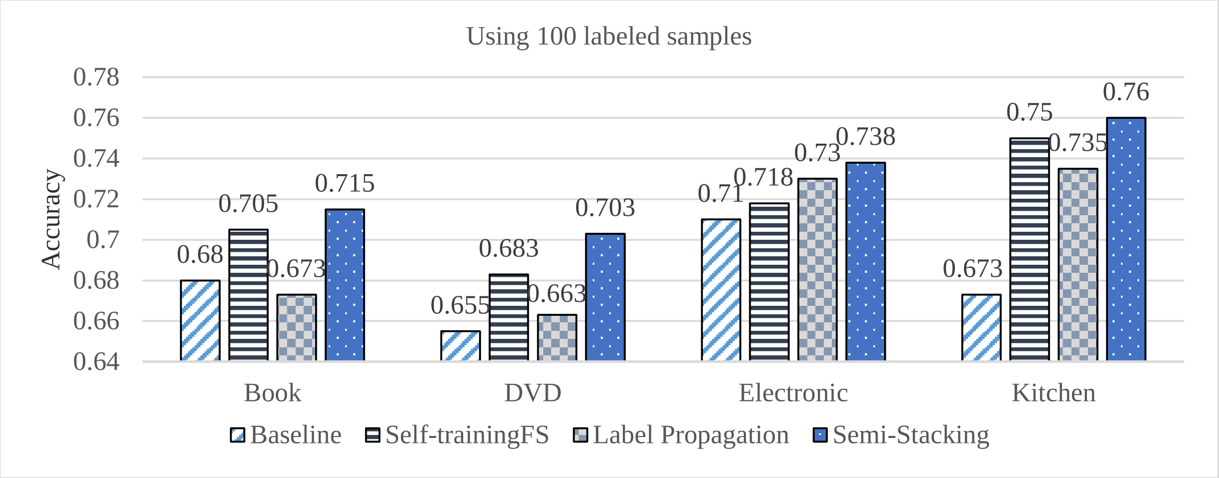

Figure 4: Performance comparison of baseline and three semi-supervised learning approaches

4 Experimentation

Dataset: The dataset contains product reviews from four different domains: Book, DVD, Electronics and Kitchen appliances (Blitzer et al., 2007), each of which contains 1000 *positive* and 1000 *negative* labeled reviews. We randomly select 100 instances as labeled data, 400 instances are used as test data and remaining 1500 instances as unlabeled data.

Features: Each review text is treated as a bag-of-words and transformed into binary vectors encoding the presence or absence of word unigrams and bigrams.

Supervised learning algorithm: The maximum entropy (ME) classifier implemented with the public tool, Mallet Toolkits (http://mallet.cs.umass.edu/), where probability outputs are provided.

Semi-supervised learning algorithms: (1) The first member algorithm is called self-trainingFS, proposed by Gao et al. (2014). This approach can be seen as a special case of self-training. Different from the traditional self-training, self-trainingFS use the feature-subspace classifier to make the prediction on the unlabeled samples instead of using the whole-space classifier. In our implementation, we use four random feature subspaces. (2) The second member algorithm is called label propagation, a graph-based semi-supervised learning approach, proposed by Zhu and Ghahramani (2002). In our implementation, the document-word bipartite graph is adopted to build the document-document graph (Sindhwani and Melville, 2008).

Significance testing: We perform *t*-test to evaluate the significance of the performance difference between two systems with different approaches (Yang and Liu, 1999)

Figure 5 compares the performances of the baseline approach and three semi-supervised learning approaches. Here, the baseline approach is the supervised learning approach by using only the initial labeled data (i.e. no unlabeled data is used). From the figure, we can see that both Self-trainingFS and label propagation are successful in exploiting unlabeled data to improve the performances. Self-trainingFS outperforms label propagation in three domains including Book, DVD, and Kitchen but it performs worse in Electronic. Our approach (semi-stacking) performs much better than baseline with an impressive improvement of 4.95% on average. Compared to the two member algorithms, semi-stacking always yield a better performance, although the improvement over the better-performed member algorithm is slight, only around 1%-2%. Significance test shows that our approach performs significantly better than worse-performed member algorithm (*p*-value<0.01) in all domains and it also performs significantly better than better-performed member algorithm (*p*-value<0.05) in three domains, i.e., Book, DVD, and Kitchen.

5 Conclusion

In this paper, we present a novel ensemble learning approach named semi-stacking to semi-supervised sentiment classification. Semi-stacking is implemented by re-predicting the labels of the unlabeled samples with *meta*-learning after two or more member semi-supervised learning approaches have been performed. Experimental evaluation in four domains demonstrates that semi-stacking outperforms both member algorithms.

Acknowledgments

This research work has been partially supported by three NSFC grants, No.61273320, No.61375073, No.61331011, and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

Blitzer J., M. Dredze and F. Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of ACL-07*, pp.440-447.

Blum A. and T. Mitchell. 1998. Combining Labeled and Unlabeled Data with Co-training. In *Proceedings of COLT-98*,pp. 92-100.

Cui H., V. Mittal and M. Datar. 2006. Comparative Experiments on Sentiment Classification for Online Product Reviews. In *Proceedings of AAAI-06*, pp.1265-1270.

Dasgupta S. and V. Ng. 2009. Mine the Easy, Classify the Hard: A Semi-Supervised Approach to Automatic Sentiment Classification. In *Proceedings of ACL-IJCNLP-09*, pp.701-709, 2009.

Džeroski S. and B. Ženko. 2004. Is Combining Classifiers with Stacking Better than Selecting the Best One? *Machine Learning*, vol.54(3), pp.255-273, 2004.

Gao W., S. Li, Y. Xue, M. Wang, and G. Zhou. 2014. Semi-supervised Sentiment Classification with Self-training on Feature Subspaces. In *Proceedings of CLSW-14*, pp.231-239.

He Y., C. Lin and H. Alani. 2011. Automatically Extracting Polarity-Bearing Topics for Cross-Domain Sentiment Classification. In *Proceedings of ACL-11*, pp.123-131.

Li S., C. Huang, G. Zhou and S. Lee. 2010. Employing Personal/Impersonal Views in Supervised and Semi-supervised Sentiment Classification. In *Proceedings of ACL-10*, pp.414-423.

Li S., Y. Xue, Z. Wang, and G. Zhou. 2013. Active Learning for Cross-Domain Sentiment Classification. In *Proceedings of IJCAI-13*, pp.2127-2133.

Liu B. 2012. *Sentiment Analysis and Opinion Mining (Introduction and Survey)*. Morgan & Claypool Publishers, May 2012.

Pang B. and L. Lee. 2008. Opinion Mining and Sentiment Analysis: Foundations and Trends. *Information Retrieval*, vol.2(12), pp.1-135.

Pang B., L. Lee, and S. Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of EMNLP-02*, pp.79-86.

Riloff E., S. Patwardhan and J. Wiebe. 2006. Feature Subsumption for Opinion Analysis. In *Proceedings of EMNLP-06*, pp.440-448.

Sindhwani V. and P. Melville. 2008. Document-Word Co-Regularization for Semi-supervised Sentiment Analysis. In *Proceedings of ICDM-08*, pp.1025-1030.

Wan X. 2009. Co-Training for Cross-Lingual Sentiment Classification. In *Proceedings of ACL-IJCNLP-09*, pp.235-243.

Yang Y. and X. Liu. 1999. A Re-Examination of Text Categorization Methods. In *Proceedings of SIGIR-99*.

Zhu X. and Z. Ghahramani. 2002. Learning from Labeled and Unlabeled Data with Label Propagation. *CMU CALD Technical Report*. CMU-CALD-02-107.