Interpolative Self-training Approach for Sentiment Analysis

Somayyeh Aghababaei, Masoud Makrehchi
Department of Electrical, Computer, and Software Engineering
University of Ontario Institute of Technology, Oshawa, ON, Canada
{somayyeh.aghababaei, masoud.makrehchi}@uoit.ca

Abstract—**Sentiment analysis has become one of the fundamental research areas with an objective of estimating the polarity of text documents. While sentiment analysis requires rich training resources, the number of available labeled documents is limited. The proposed interpolative self-training model is an extension of self-training as one of the most common semi-supervised learning algorithms. The proposed method is based on enlarging learning documents by interpolating data in both the training and the test phase. The method also includes a weighting strategy for data selection in each iteration. The method is evaluated using four Twitter datasets for the task of sentiment analysis. The results indicate that the proposed self-training model successfully outperforms the baseline and the standard self-training approach.**

I. Introduction

With the rapid growth of publicly available social content via social networks where users share their opinions, sentiment analysis is becoming more important in better understanding user opinions and intentions. Unlike other NLP problems, such as topic classification which performs well at general level, opinion mining and sentiment analysis are challenging tasks. Indeed, differentiating between a variety of topics can be easier than determining the sentiment of documents. Although users usually use more specific features for different topics, in sentiment analysis these variances are challenging to detect. This could be explained by the fact that users express their feelings in many different ways. Also, sentiment can be laid down in ambiguous documents such as "I am not invited to the party" which expresses sadness feeling or "feeling determined" which expresses happiness. Some documents, such as reviews, include both positive and negative sides from which it is difficult to detect any general feeling.

Supervised learning (SL) is one of the most common approaches in sentiment analysis, opinion mining, and polarity classification. Many attempts have been made to improve the performance of SL classifiers in different ways. One of the simplest and most common techniques is using subjective training set to detect objective samples applying a binary classifier [1]. Koppel and Schler [2] applied neutral instances to differentiate between positive and negative reviews. Nevertheless, the effectiveness of all the approaches using SL methods highly depend on the availability of annotated data.

While plenty of unlabeled data are available, annotating data or obtaining labeled training sets is time-consuming and expensive. Another possible solution is unsupervised learning methods such as a clustering approach to distinguish between different polarities. However, without having any knowledge of dependencies of features and different sentiments, sentiment analysis still remains as a challenging task. Semi-supervised learning (SSL) algorithms are potential solutions for the problem of insufficient annotated data and have attracted widespread research attention in many domains [3, 4]. SSL techniques utilize unlabeled data as well as a limited number of available labeled samples in a classification engine. Self-training is one of the most popular and efficient algorithms [5] in semi-supervised learning. The algorithm is based on training classifier with limited labeled data to annotate an extensive number of unlabeled documents in an iterative cycle. In every iteration, it is assumed that the predictions with high confidence scores are correct; therefore, they are added to the training set.

In this paper, a self-training model, namely interpolative self-training is proposed. The proposed algorithm is an iterative method, which contributes in two learning stages: interpolating documents and selecting the best predictions to be added to the training set. In the first stage, the training set is enlarged by concatenating insufficient labeled examples in each class with documents in the same class or different classes. In addition to enlarging the training set, insufficient labeled data are interpolated with the test data. In fact, the idea of interpolating the training set with the test set can create different test data possibilities and can aid the classifier to distinguish between those possibilities and learn more from the data. For the selection stage, we define a weighting approach to score the predicted documents and select the best predictions for adding them to the training set in each iteration.

In fact, our proposed model pursues two main contributions: enlarging insufficient training data and facilitating the predicting performance of the classier. Interpolation of documents increases the number of available

labeled documents. In addition, it helps the classifier in differentiating the polarity of the documents. In fact, the brevity of micro-blogging data causes many challenges in polarity classification. It may result in not having specific features that express emotions. Indeed, sentiment classification, which seeks to determine the polarity of a document, may fail when there are ambiguous features. Therefore, the interpolation of documents also aids the classifier in labeling the documents that are short in length or have ambiguous features.

II. Overview of Previous Studies

Until now, semi-supervised learning techniques have been considered as effective learning algorithms in many applications ranging from query attempt calcification [6] to NLP [7], spam filtering, and personality prediction [8, 9]. With the popularity of sentiment detection in text analysis and the availability of abundant unlabeled data, SSL algorithms have become more popular [10, 11]. There has been widespread research attention on semi-supervised learning and sentiment analysis. A semi-supervised learning algorithm proposed by Dasgupta et al. [12] is based on using unsupervised learning for automatically labeling a training seed for an iterative algorithm, which requires careful parameter tuning. In this context, Zhou et al. [13] proposed a novel active deep network (ADN) to detect training documents for a semi-supervised algorithm. In addition, Other non iterative algorithms also have been developed [14]. Extracted patterns from a large number of unlabeled data were used as features in SL to investigate to what extent effective prior knowledge can improve the performance of a classifier. Furthermore, semi-supervised learning algorithms have been applied in lexicon-based approaches [15, 16]. As an example, He and Zhou [16] employed a general lexicon sentiment dictionary to annotate unlabeled data. Based on the automatically labeled data, a new self-learnt approach was built.

Among the most common techniques in semi-supervised learning, self-training is one of the most popular algorithms [5]. Self-training is known as one of the most efficient and simplest approach in semi-supervised learning. It works as a black box wrapper, which avoids struggling with the complexity of the systems during learning. The algorithm is based on training a classifier with limited labeled data and applying it to the majority of unlabeled data in an iterative cycle. In every iteration, it is assumed that the high confidence predictions are correct and added to the training set. However, this assumption is more effective when classes are more separated in context. This emerged approach was first adopted for the problem of sentiment bearing by Riloff and Jones [17]. Although many sentiment labeling tasks have been devoted for analyzing applicability of self-training [18, 19], there is still a lack of study on how to modify the self-training framework in order to improve its capabilities. In this study, we propose a self-training approach to better focus on the generation of the initial training set as well as the selection of the data in each iteration.

III. Interpolative Self-training Framework

We propose a self-training model, namely interpolative self-training. The model was proposed for a sentiment classification of limited labeled documents with the abundant unlabeled data. This algorithm starts with enriching data in the training and test datasets. An initial classifier is trained with the enriched dataset and is applied to label the test data. In each iteration, the best test data are selected to be added to the former training set. Training and testing sets are recombined and retrained. In the final stage, based on some predefined number of iterations, the performance is achieved. The framework of interpolative learning has been presented in Table I. In the following subsections, we highlight how data are interpolated in the training and test phases. We also explain how data are selected in each iteration based on the proposed weighting schema.

A. Data Generation

In the proposed iterative model, learning documents are enlarged with two different steps. First, documents are concatenated in the training phase and the binary classification is converted to three class problem. Second, the test data are integrated with a set of labeled documents to enable better understating and learning of the test data. In each iteration, the best representative data is selected to be added to the training set iteratively.

Data Interpolation in Training Phase: Suppose the training set consists of positive and negative documents, which may express sentiments decently, such as "I feel so sad for them". In addition, there might be some documents such as "I am not going to the party" and "I was at Starbucks yesterday", which are ambiguous and are mostly found in microblogging datasets. In order to resolve the issue of the ambiguity, a subset of training dataset are selected from each classes (positive and negative). The selected datasets are concatenated with the original training dataset and all three possibilities of categories (positive, negative, and ambiguous) are generated (see Table II). While data are enriched, the strength of positiveness or negativeness in each document is also increased. Let X^l be the training set (labeled documents) that is presented as follows:

$$X^l = \{x_1, x_2, ..., x_m\} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,|V|} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,|V|} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,|V|} \end{pmatrix} \quad (1)$$

where V is the global vocabulary and m represents the number of documents in training set. Therefore, X^{pos} and

TABLE I: Interpolative self-training framework.

1: Input: Classifier f, labeled documents consist of positive and negative samples $\{X^l : (X^{pos}, X^{neg}) \rightarrow Y\}$; test documents $\{X^u\}$.

2: **Repeat**

3: Generate training and testing documents:

a: Concatenate positive samples with training set $\{X^l, X^{pos}\}$; labeled as positive.

b: Concatenate negative samples with training set $\{X^l, X^{neg}\}$; labeled as negative.

c: Concatenate positive samples with negative samples $\{X^{pos}, X^{neg}\}$; labeled as ambiguous.

d: Concatenate positive samples with test set $\{X^{pos}, X^u\}$.

e: Concatenate negative samples with test set $\{X^{neg}, X^u\}$.

3: Train classier f with the data created by steps a, b, c .

4: Apply classifier f on the generated unlabeled data by steps d and e.

5: Select X^s: the most confidently labeled examples based on the weights assigned to each test example.

6: Remove X^s from X^u; add X^s to training set.

7: **Until** predefined number of iterations

8: Return labels of the test dataset

X^{neg} are the subsets of positive and negative documents that are selected from the training set (X^l). In interpolation phase, the training documents are integrated to generate three groups of possibilities as follows:

$$\begin{aligned} X^{pos}_{m',|V|} \times X^{pos}_{m_1,|V|} &: m' \times m_1 \quad \text{positive} \\ X^{neg}_{m'',|V|} \times X^{neg}_{m_2,|V|} &: m'' \times m_2 \quad \text{negative} \\ X^{pos}_{m_1,|V|} \times X^{neg}_{m_2,|V|} &: m_1 \times m_2 \quad \text{ambiguous} \end{aligned} \tag{2}$$

Where m' and m'' are the sizes of the positive and negative documents in training set, while m_1 and m_2 are the sizes of the subsets (randomly selected) of the positive and negative documents from the training set. We assume the combination of the positive documents generates positive sentiment and the negative documents make negative combinations. However, the combination of X^{pos} and X^{neg} creates the ambiguous documents.

Furthermore, interpolation of each two vectors is the concatenation of their features. As an example, suppose A is a vector space of a document from training set, which is represented by a vector of features, $A_i = \{a_1, a_2, ..., a_{|V_1|}\}$. B is also a selected document for concatenation and is denoted as $B_i = \{ b_1, b_2, ..., b_|V_2|\}$. Therefore, the combination of these two vectors generates a new data consist of a set of features, which is presented as follows:

$$A \bigcup B = \{f_1, f_2, ..., f_{|V_1+V_2|}\} \tag{3}$$

Where V_1 and V_2 are the subsets of the global vocabulary (V). The concatenation results in generating more labeled documents as well as increasing the degree of positiveness and negativeness of each document.

Data Interpolation in Test Phase: In contrast to the common self-training, in which the test set remains unchanged, in the proposed algorithm the test data is also merged with the selected training samples. In this context, by having no knowledge about test dataset, the documents in test set are concatenated with the selected positive or negative from the training set. This approach helps the classifier to distinguish more about sentiment of the data. Additionally, in each iteration, the best test documents with their predicted labels are reinforced in the training data. Therefore, if X^u is the test set, all the possibilities of the generated unlabeled data are as follows:

$$\begin{aligned} X^u_{x,|V|}, X^{pos}_{m_1,|V|} &: (n-m) \times m_1 \\ X^u_{x,|V|}, X^{neg}_{m_2,|V|} &: (n-m) \times m_2 \end{aligned} \tag{4}$$

where $n - m$ is the size of test dataset.

B. Data Selection

In each iteration, the best labeled data, which is annotated by the classifier, is selected to be added to the training set. In this regard, four weights (w) are assigned to each test data based on the score given by the classifier. The weights are calculates as follows:

$$\begin{aligned} W_a = \sum_1^j \frac{(w_a)_j}{j+1}, W_c = \sum_1^k \frac{(w_c)_k}{k+1} \\ W_b = \sum_1^q \frac{(w_b)_q}{q+1}, W_d = \sum_1^r \frac{(w_d)_r}{r+1} \end{aligned} \tag{5}$$

Where w_a is the score given by the classifier to the document that has **positive** label, given by the classifier, and was interpolated with the **positive** documents. Therefore, j is the total number of documents in this

TABLE II: Sample of integrated documents in training phase.

Documents before integration	Documents after integration	Assigned label
"At starbucks with my new sister learning her new phone."	"At starbucks with my new sister learning her new phone.loving life... and loving you "	positive
"I scratched my iPod, "	" I scratched my iPod, forgot about my english coursework amp; today is just not my day "	negative
" Hayfever time not good!"	" Hayfever time not good!starbucks amp; tanning. good start for today"	ambiguous

combination. Moreover, w_c is the score of the document that was labeled **ambiguous** and was concatenated with the **positive** documents. In addition, w_b and w_d are the scores of the **negative** and **ambiguous** documents respectively, which were interpolated with the **negative** documents. After having all the mentioned weights for each test data, a total weight is assigned to each test data as follows:

$$w_i = \sqrt{(W_a - W_d)^2 + (W_c - W_b)^2} \tag{6}$$

In each iteration, the top number of test data based on their wights are selected for joining to the former training set. Each label (y_i) is assigned to the selected documents as follows:

$$y_i = \begin{cases} if\ w_a + w_c > w_d + w_b\ then & \text{positive} \\ if\ w_a + w_c \leq w_d + w_b\ then & \text{negative} \end{cases} \tag{7}$$

IV. Experimental Results

We selected Twitter sentiment analysis datasets for evaluating our proposed approach. The benchmark datasets are as follows:

1) **The STS-Gold corpus [20]:** This dataset contains 2,034 tweets which were labeled by experts as positive and negative.

2) **Stander-Twitter sentiment corpus:** [1] This corpus consists of 5,513 hand-classified tweets of product reviews related to Apple, Google, Microsoft, and Twitter.

3) **Twitter Sentiment Analysis Training Corpus (sentiment-1,sentiment-2):** [2] This dataset consist of 1,578,627 classified tweets with positive and negative labels. The dataset was divided into two different corpus: sentiment-1 and sentiment-2.

A. Experiment Setup

We compared the proposed interpolative self-training model with a standard self-training and SL as the baseline. The stopping criteria for both self-training and the proposed model is 20 number of iterations. Macro-average F-measure was calculated to evaluate the performance of the applied model with liblinear SVM classifier. In every iteration i% of the best representative data are added to the training set, where i=5,10, and 20 percent were considered. In the preprocessing phase, we filtered out tweets with neutral labels as well as non-English tweets. We removed stopwords, punctuations, hashtags, urls, and numbers. The documents were employed with a binary representation.

[1]http://www.sananalytics.com/lab/twitter-sentiment/

[2]http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/

B. Results and Discussions

The F-measure results have been presented in Figure 1. The baseline is the supervised learning considering 10% of data as the training and 90% as test sets. The results of the baseline have shown that SL does not perform well because training sample is small and insufficient. In fact, microblogging datasets are mostly unstructured and the classifier needs a rich training set to achieve a high performance. The self-training has improved the performance slightly, but not significant changes were captured. Interpolative self-training achieved almost a comparable result compared with the self-training, and boosted the result of baseline significantly (Figure 1). As an example, for dataset STS-Gold, interpolative-learning performed 24% and 16% better than the baseline and the self-training respectively. Significant differences of interpolative learning performance compared with other methods can be related to the enrichment approach in the number of data and features. In every concatenation in training phase, not only data is enlarged, also the degree of positiveness and negativeness of a document is increased. In addition, changing two polarity classes problem to three classes gives this ability to classifier to discriminate the differences between positive and negative classes when there are ambiguous documents. In fact, in binary classification the classifier tries to find a separating hyperplane while some ambiguous documents would lie on different side.

Figure 2 represents the performance of three algorithms over the considered 20 iterations. It can be observed that, in three datasets, a significant improvement is achieved in the first iteration for interpolative learning and performance was slightly changed for other iterations. However, the performance of self-training was increased with the number of iterations. From the results we can conclude that interpolative self-training reduces the number of learning iterations significantly compared to the self-training.

The result of interpolative on Stander-Twitter dataset has been presented in Figure 2 (c). In this case, the performance of interpolative has not been significantly increased in the first iteration. Also, performance of self-training has been dropped in early iterations. It could

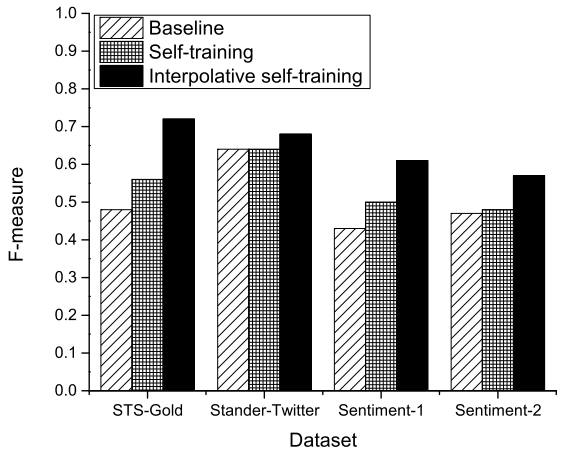

Fig. 1: *F-measure of the different approaches.*

be possibly attributed to the topics of this dataset. This corpus consist of four different products: Twitter, Apple, Microsoft, and Google. In fact, different product reviews contain different features to express emotions. As an example, some specific features, which are used to express opinion about Twitter, are not used to express sentiment about Google. Therefore, less shared common words between the same class of different products creates some challenges.

V. Conclusions

In this study, we presented a semi-supervised learning model for the problem of sentiment analysis. Interpolative self-training is the modified version of the self-training, which is based on concatenating data in both the training and the test datasets, while changing a binary classification to a three classification problem. In the presented approach, the training and the test datasets were concatenated with the selected positive and negative samples from the training set. In fact, the proposed model enriches the training dataset, while effectively enhances the predictability in test phase. The performance of interpolative learning has been compared with the self-training and the supervised learning as the baseline. According to the preliminary results, interpolative self-training outperformed the other approaches. The best result has shown 16% improvement compared to the baseline. In addition, the performance of interpolative learning has been increased in the early iterations compared with the self-training.

In the presented approach, the training and test datasets were concatenated with the randomly selected positive and negative samples from the training set. In the future, we have plan to apply a selection method to select the best representative of positive and negative samples as a concatenation set. Enriching datasets with better samples in initial iterations may increase the performance of the developed method. Moreover, we would like to evaluate the effectiveness of interpolative in domain adaptation. In this approach, other domains can be considered as a concatenation set to integrate with a target domain. In addition, we are also interested to evaluate our method for other problems such as deception detection which suffers from insufficient labeled data.

References

[1] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004, p. 271.

[2] M. Koppel and J. Schler, "The importance of neutral examples for learning sentiment," *Computational Intelligence*, vol. 22, no. 2, pp. 100–109, 2006.

[3] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," *Machine learning*, vol. 39, no. 2-3, pp. 103–134, 2000.

[4] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," 2005.

[5] H. J. Scudder, "Probability of error of some adaptive pattern-recognition machines," *Information Theory, IEEE Transactions on*, vol. 11, no. 3, pp. 363–371, 1965.

[6] A. Fuxman, A. Kannan, A. B. Goldberg, R. Agrawal, P. Tsaparas, and J. Shafer, "Improving classification accuracy using automatically extracted training data," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 1145–1154.

[7] B. Krishnapuram, D. Williams, Y. Xue, L. Carin, M. Figueiredo, and A. J. Hartemink, "On semi-supervised classification," in *Advances in neural information processing systems*, 2004, pp. 721–728.

[8] A. C. E. Lima and L. N. De Castro, "A multi-label, semi-supervised classification approach applied to personality prediction in social media," *Neural Networks*, vol. 58, pp. 122–130, 2014.

[9] J. Ortigosa-Hernández, J. D. Rodríguez, L. Alzate, M. Lucania, I. Inza, and J. A. Lozano, "Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers," *Neurocomputing*, vol. 92, pp. 98–115, 2012.

[10] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

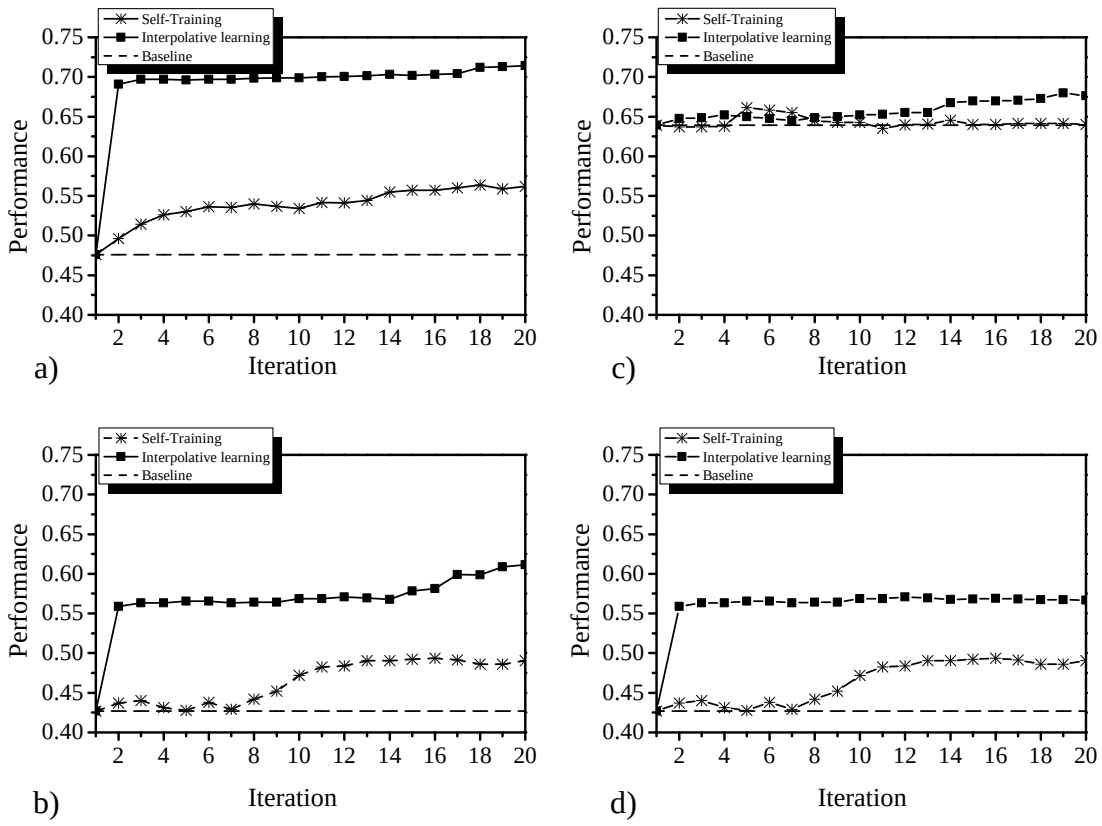

Fig. 2: *F-measure over 20 iterations: (a) STS-Gold; (b) Sentiment-1; (c) Stander-Twitter; (d) Sentiment-2.*

[11] A. B. Goldberg and X. Zhu, "Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization," in *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*. Association for Computational Linguistics, 2006, pp. 45–52.

[12] S. Dasgupta and V. Ng, "Mine the easy, classify the hard: a semi-supervised approach to automatic sentiment classification," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 701–709.

[13] S. Zhou, Q. Chen, and X. Wang, "Active deep learning method for semi-supervised sentiment classification," *Neurocomputing*, vol. 120, pp. 536–546, 2013.

[14] A. B. Goldberg, "New directions in semi-supervised learning," Ph.D. dissertation, University of Wisconsin–Madison, 2010.

[15] P. Melville, W. Gryc, and R. D. Lawrence, "Sentiment analysis of blogs by combining lexical knowledge with text classification," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 1275–1284.

[16] Y. He and D. Zhou, "Self-training from labeled features for sentiment analysis," *Information Processing & Management*, vol. 47, no. 4, pp. 606–616, 2011.

[17] E. Riloff, R. Jones *et al.*, "Learning dictionaries for information extraction by multi-level bootstrapping," in *AAAI/IAAI*, 1999, pp. 474–479.

[18] L. Qiu, W. Zhang, C. Hu, and K. Zhao, "Selc: a self-supervised model for sentiment classification," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 929–936.

[19] N. Yu, "Domain adaptation for opinion classification: A self-training approach," *Journal of Information Science Theory and Practice*, vol. 1, no. 1, pp. 10–26, 2013.

[20] H. Saif, M. Fernandez, Y. He, and H. Alani, "Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold," 2013.