Information Processing and Management xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Self-training from labeled features for sentiment analysis

Yulan He [a,*], Deyu Zhou [b]

[a] Knowledge Media Institute, Open University, Walton Hall, Milton Keynes MK6 6AA, UK
[b] School of Computer Science and Engineering, Southeast University, Nanjing, China

ARTICLE INFO

Article history:
Received 21 July 2010
Received in revised form 5 October 2010
Accepted 4 November 2010
Available online xxxx

Keywords:
Sentiment analysis
Opinion mining
Self-training
Generalized expectation
Self-learned features

ABSTRACT

Sentiment analysis concerns about automatically identifying sentiment or opinion expressed in a given piece of text. Most prior work either use prior lexical knowledge defined as sentiment polarity of words or view the task as a text classification problem and rely on labeled corpora to train a sentiment classifier. While lexicon-based approaches do not adapt well to different domains, corpus-based approaches require expensive manual annotation effort.

In this paper, we propose a novel framework where an initial classifier is learned by incorporating prior information extracted from an existing sentiment lexicon with preferences on expectations of sentiment labels of those lexicon words being expressed using generalized expectation criteria. Documents classified with high confidence are then used as pseudo-labeled examples for automatical domain-specific feature acquisition. The word-class distributions of such self-learned features are estimated from the pseudo-labeled examples and are used to train another classifier by constraining the model's predictions on unlabeled instances. Experiments on both the movie-review data and the multi-domain sentiment dataset show that our approach attains comparable or better performance than existing weakly-supervised sentiment classification methods despite using no labeled documents.

1. Introduction

With the explosion of people's attitudes and opinions expressed in social media including blogs, discussion forums, tweets, etc., detecting sentiment or opinion from the Web is becoming an increasingly popular way of interpreting data. The objective of sentiment analysis is to determine the overall attitude, either positive, negative, or neutral, expressed in a give piece of text. Most prior work in sentiment analysis (Blitzer, Dredze, & Pereira, 2007; Choi, Cardie, Riloff, & Patwardhan, 2005; Kim & Hovy, 2004; Narayanan, Liu, & Choudhary, 2009; Pang & Lee, 2004; Pang, Lee, & Vaithyanathan, 2002; Zhao, Liu, & Wang, 2008) view sentiment classification as a text classification problem where an annotated corpus with documents labeled with their sentiment orientation is required to train the classifiers. As such they lack of portability across different domains. Moreover, the rapid evolution of user-generated contents demands sentiment classifiers that can easily adapt to new domains with minimum supervision. This thus motivates the investigation of weakly-supervised or unsupervised sentiment analysis approaches.

While supervision for a sentiment classifier can come from labeled documents, it can also come from labeled words. For example, the word "*excellent*" typically conveys positive sentiment. A simple approach of using such polarity words for sentiment classification is to compare the frequency of occurrence of positive and negative terms in a document.

* Corresponding author. Tel.: +44 1908 858215; fax: +44 1908 653169.
E-mail addresses: y.he@cantab.net (Y. He), d.zhou@seu.edu.cn (D. Zhou).

0306-4573/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2010.11.003

However, it does not normally give good results. In recent years, much effort has been devoted to incorporate prior belief of word-sentiment associations from a sentiment lexicon into classifier learning by combining such lexical knowledge with a small set of labeled documents (Andreevskaia & Bergler, 2008; Li, Zhang, & Sindhwani, 2009; Melville, Gryc, & Lawrence, 2009).

Other weakly-supervised sentiment analysis approaches typically adopt the *self-training* strategy (Qiu, Zhang, Hu, & Zhao, 2009; Zagibalov & Carroll, 2008a, 2008b). They start with some initial seed sentiment lexicon and then use iterative training to enlarge the lexicon. Documents classified at the current iteration are used as *self-labeled instances* to train a classifier for the next iteration. Other approaches use ensemble techniques by combining lexicon-based and corpus-based algorithms (Tan, Wang, & Cheng, 2008). Nevertheless, all these approaches are either complex or require careful tuning of domain and data specific parameters. More recently, Dasgupta and Ng (2009) proposed a weakly-supervised sentiment classification algorithm by integrating user feedbacks into a spectral clustering algorithm. Features induced for each dimension of spectral clustering can be considered as sentiment-oriented topics. Nevertheless, human judgement of identifying the most important dimensions during spectral clustering is required.

In this paper,[1] we propose a simple and robust strategy that works by providing weak supervision at the level of features rather than instances. We obtain an initial classifier by incorporating prior information extracted from an existing sentiment lexicon into a sentiment classifier model learning, where preferences on expectations of sentiment labels of those lexicon words are expressed using generalized expectation criteria (Druck, Mann, & McCallum, 2008; McCallum, Mann, & Druck, 2007). Documents classified with high confidence by this initial classifier are used to derive a set of self-learned and domain-specific features that are related to the distribution of the target classes. Such self-learned features are then used to train another classifier by constraining the model's predictions on unlabeled instances.

We evaluate our proposed framework on the movie-review data and the multi-domain sentiment dataset and show that our method attains comparable or better performance than other previously proposed weakly-supervised or semi-supervised methods for sentiment classification despite using no labeled instances. The rest of the paper is structured as follows. Related work on weakly-supervised and semi-supervised sentiment classification are discussed in Section 2. The proposed framework is introduced in Section 3. The experimental setup and results are presented in Section 4. Finally, Section 5 concludes the paper and outlines directions for future research.

2. Related work

The pioneer work on sentiment classification that does not require labeled data is that of Turney (2002)[Turney's (2002)] which classifies a document as positive or negative by the average semantic orientation of the phrases in the document that contain adjectives or adverbs. The semantic orientation of a phrase is calculated as the pointwise mutual information (PMI) with a positive word "*excellent*" minus the PMI with a negative word "*poor*". His approach achieved an accuracy of 84% for automobile reviews and 66% for movie-reviews. In the same vein, Read and Carroll (2009) proposed three different ways, lexical association (using PMI), semantic spaces, and distributional similarity, to measure the similarity between words and polarity prototypes (such as "*excellent*" or "*good*"). While Turney only used one polarity prototype for each class, Read and Carroll chose seven polarity prototypes which were obtained from Roget's Thesaurus and WordNet and selected based on their respective frequency in the Gigaword corpus. Still the best result was achieved using PMI with 69.1% accuracy obtained on the movie-review data.

There have also been much interests in incorporating prior information from sentiment lexicon containing a list of words bearing positive or negative polarity into sentiment model learning, which we call weakly-supervised sentiment classification. Sentiment lexicons can be constructed in many different ways, ranging from manual approaches (Whitelaw, Garg, & Argamon, 2005), to semi-automated approaches (Abbasi, Chen, & Salem, 2008; Argamon, Bloom, Esuli, & Sebastiani, 2007; Kim & Hovy, 2004), and even almost fully automated approaches (Kaji & Kitsuregawa, 2006; Kanayama & Nasukawa, 2006; Turney & Littman, 2002). When incorporating such prior information into model learning, Andreevskaia and Bergler (2008) integrate a corpus-based classifier trained on a small set of annotated in-domain data and a lexicon-based system trained on WordNet for sentence-level sentiment annotation across different domains. Li et al. (2009) employ lexical prior knowledge for semi-supervised sentiment classification based on non-negative matrix tri-factorization, where the domain-independent prior knowledge was incorporated in conjunction with domain-dependent unlabeled data and a few labeled documents. Melville et al. (2009) also combine lexical information from a sentiment lexicon with labeled documents where word-class probabilities in Naïve Bayes classifier learning are calculated as a weighted combination of word-class distributions estimated from the sentiment lexicon and labeled documents respectively. Lin and He (2009) proposed a joint sentiment-topic (JST) model to model both sentiment and topics from text and they incorporate sentiment prior information by modifying conditional probabilities used in Gibbs sampling during JST model learning.

Instead of incorporating prior information into model learning through sentiment lexicons, Dasgupta and Ng (2009) proposed an unsupervised sentiment classification algorithm where user feedbacks are provided in the spectral clustering process in an interactive manner to ensure that text are clustered along the sentiment dimension. Features induced for each dimension of spectral clustering can be considered as sentiment-oriented topics. Nevertheless, human judgement of

[1] The paper is a substantial extension of He (2010).

Please cite this article in press as: He, Y., & Zhou, D. Self-training from labeled features for sentiment analysis. *Information Processing and Management* (2010), doi:10.1016/j.ipm.2010.11.003

identifying the most important dimensions during spectral clustering is required. We compare with the methods of (Dasgupta & Ng, 2009; Lin & He, 2009; Li et al., 2009) in Section 4.3 and show that our proposed approach achieves comparable or better performance on the movie-review data and outperforms their on the multi-domain sentiment dataset.

Other weakly-supervised sentiment classification approaches typically adopt the *self-training* strategy. Zagibalov and Carroll (2008b) start with a one-word sentiment seed vocabulary and use iterative retraining to gradually enlarge the seed vocabulary by adding more sentiment-bearing lexical items based on their relative frequency in both the positive and negative parts of the current training data. Sentiment direction of a document is then determined by the sum of sentiment scores of all the sentiment-bearing lexical items found in the document. The problem with this approach is that there is no principal way to set the optimal number of iterations. They then suggested an iteration control method in (Zagibalov & Carroll, 2008a) where iterative training stops when there is no change to the classification of any document over the previous two iterations. However, this does not necessarily correlate to the best classification accuracy.

Similar to Zagibalov and Carroll (2008b), Qiu et al. (2009) also use a lexicon-based iterative process as the first phase to iteratively enlarge an initial sentiment dictionary. But instead of using a one-word seed dictionary as in (Zagibalov & Carroll, 2008b), they start with a much larger HowNet Chinese sentiment dictionary[2] as the initial lexicon. Documents classified by the first phase are taken as the training set to train the SVMs which are subsequently used to revise the results produced by the first phase. Tan et al. (2008) proposed a combination of lexicon-based and corpus-based approaches that first labels some examples from a give domain using a sentiment lexicon and then trains a supervised classifier based on the labeled ones from the first stage.

The above self-training approach utilizes *self-labeled instances* in the training loop. While the current model could be improved by iteratively adding the most confident self-labeled examples generated at each iteration, this is not always true since *self-labeled instances* might suffer from the incestuous learning bias problem as instances might be consistently mislabeled which makes the model even worse in the next iteration. Much recent work has thus been conducted to explore *labeled features* in model learning without labeled instances. For example, some approaches use human annotated labeled features to generate pseudo-labeled examples that are subsequently used in standard supervised learning (Schapire, Rochery, Rahim, & Gupta, 2002; Wu & Srihari, 2004). Druck et al. (2008) proposed training discriminative probabilistic models with labeled features and unlabeled instances using generalized expectation (GE) criteria. Labeled features can come from human annotations or through unsupervised feature clustering with latent Dirichlet allocation (LDA). For LDA-generated features, the feature labels are generated by an oracle which assumes the availability of labeled instances. These soft constraints are then expressed as GE criteria.

In contrast to the aforementioned methods, our proposed framework does not use human annotations to generate labeled features. Instead, we use the generalized expectation criteria to express preferences on expectations of sentiment labels of those lexicon words from a sentiment lexicon. Moreover, our framework further induces domain-specific features automatically from a large corpus of un-annotated data.

3. Proposed framework

We propose a novel framework for sentiment classifier learning from unlabeled documents as shown in Fig. 1. The process begins with a collection of un-annotated text and a sentiment lexicon such as the MPQA subjectivity lexicon.[3] An initial classifier is trained by incorporating prior information from the sentiment lexicon which consists of a list of words marked with their respective polarity. For example, the word "*good*" typically conveys a positive sentiment. We refer such prior information as *labeled features* and use them directly to constrain model's predictions on unlabeled instances using generalized expectation (GE) criteria. The initially-trained classifier using GE is then applied on the un-annotated text and the documents labeled with high confidence are fed into the self-learned features extractor to acquire domain-dependent features automatically. Such self-learned features are subsequently used to train another classifier which is then applied on the test set to obtain the final results.

The remainder of the section will describe the proposed framework in details.

3.1. Classifier training using generalized expectation criteria

Assuming that we have a total number of S sentiment labels denoting by $\mathscr{S} = \{positive, negative\}$ in a typical sentiment classification task; a corpus with a collection of D documents is denoted by $\mathscr{D} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_D\}$ where the bold-font variables denote the vectors; each document in the corpus is a sequence of N_d words denoted by $\boldsymbol{w} = (w_1, w_2, \ldots, w_{N_d})$, and each word in the document is an item from a vocabulary $\mathscr{V}$ index with V distinct terms denoted by {1,2,...,V}.

Suppose we have a classifier parameterized by Λ, the sentiment label s of a document $\boldsymbol{w}$ is found by maximizing the following equation:

$$\tilde{s} = \arg\max_{s} P(s|\boldsymbol{w}; \Lambda) \tag{1}$$

[2] http://www.keenage.com/download/sentiment.rar.
[3] http://www.cs.pitt.edu/mpqa/.

Please cite this article in press as: He, Y., & Zhou, D. Self-training from labeled features for sentiment analysis. *Information Processing and Management* (2010), doi:10.1016/j.ipm.2010.11.003

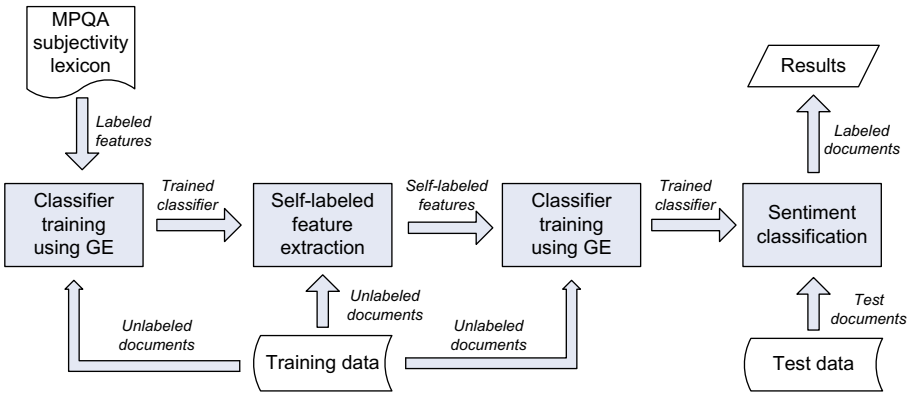

Fig. 1. A framework for sentiment classifier training.

Assume we have some labeled features where words are given with their prior sentiment orientation, we could construct a set of real-valued features of the observation to expresses some characteristic of the empirical distribution of the training data that should also hold of the model distribution.

$$f_{jk}(\boldsymbol{w}, s) = \sum_{d=1}^{D} \delta(s_d = j)\delta(k \in \boldsymbol{w}_d) \tag{2}$$

where $\delta(x)$ is an indicator function which takes a value of 1 if x is true, 0 otherwise. Eq. (2) calculates how often feature k and document label j co-occur in an instance.

We define the expectation of the features as

$$E_{\Lambda}[\boldsymbol{f}(\boldsymbol{w}, s)] = E_{\widetilde{P}(\boldsymbol{w})}[E_{P(s|\boldsymbol{w};\Lambda)}[\boldsymbol{f}(\boldsymbol{w}, s)]] \tag{3}$$

where $\widetilde{P}(\boldsymbol{w})$ is the empirical distribution of $\boldsymbol{w}$ in document corpus $\mathscr{D}$, and $P(s|\ \boldsymbol{w};\Lambda)$ is a conditional model distribution parameterized at Λ.

$E_{\Lambda}[\boldsymbol{f}(\boldsymbol{w},s)]$ is a matrix of size $S \times K$ where S is the total number of sentiment labels and K is the total number of features or constraints used in model learning. The jkth entry denotes the expected number of instances that contain feature k and have label j.

By adding a normalization term $z_k = \sum_{d=1}^{D}\delta(k \in \boldsymbol{w}_d)$ into f_{jk}, the feature expectation becomes the predicted label distribution on the set of instances containing feature k, i.e.

$$\widetilde{P}(j|k;\Lambda) = \frac{\sum_{d=1}^{D}\delta(s_d = j)\delta(k \in \boldsymbol{w}_d)}{z_k} \tag{4}$$

We define a criterion that minimizes the KL divergence of the expected label distribution and a target expectation $\hat{\boldsymbol{f}}$, which is essentially an instance of generalized expectation criteria that penalizes the divergence of a specific model expectation from a target value.

$$G(E_{\Lambda}[\boldsymbol{f}(\boldsymbol{w}, s)]) = \mathrm{KL}(\hat{\boldsymbol{f}} \| E_{\Lambda}[\boldsymbol{f}(\boldsymbol{w}, s)]) \tag{5}$$

We can use the target expectation $\hat{\boldsymbol{f}}$ to encode human or task prior knowledge. For example, the word "*excellent*" typically represent a positive orientation. We would expect that this word more likely appears in positive documents.

For each labeled feature $k \in K$, a single GE term is

$$\sum_{s} \hat{f}_{sk} \log \frac{\hat{f}_{sk}}{E_{\Lambda}[\boldsymbol{f}_{\ sk}(\boldsymbol{w}, s)]} \tag{6}$$

The gradient of Eq. (6) with respect to the model parameter for feature k and label j is:

$$\begin{aligned}\frac{\partial}{\partial\lambda_{jk}} \sum_{s} \hat{f}_{sk} \log \frac{\hat{f}_{sk}}{E_{\Lambda}[\boldsymbol{f}_{\ sk}(\boldsymbol{w}, s)]} &= -\frac{\partial}{\partial\lambda_{jk}} \sum_{s} \hat{f}_{sk} \log E_{\Lambda}[\boldsymbol{f}_{\ sk}(\boldsymbol{w}, s)] = -\sum_{s} \frac{\hat{f}_{sk}}{E_{\Lambda}[\boldsymbol{f}(\boldsymbol{w}, s)]} \frac{\partial}{\partial\lambda_{jk}} E_{\Lambda}[\boldsymbol{f}_{\ sk}(\boldsymbol{w}, s)] \\ &\propto -\sum_{s} \frac{\hat{f}_{sk}}{\widetilde{P}(s|k;\Lambda)} \sum_{d=1}^{D} \frac{\partial}{\partial\lambda_{jk}} P(s|\boldsymbol{w}_d)\boldsymbol{f}_{\ sk}(\boldsymbol{w}_d, s)\end{aligned} \tag{7}$$

Please cite this article in press as: He, Y., & Zhou, D. Self-training from labeled features for sentiment analysis. *Information Processing and Management* (2010), doi:10.1016/j.ipm.2010.11.003

If a maximum entropy model is used as the classifier, the probability of sentiment label s conditioned on document $\boldsymbol{w}$ is given by

$$P(s|\boldsymbol{w};\Lambda) = \frac{\exp\left(\sum_i \lambda_i g_i(\boldsymbol{w},s)\right)}{Z(\boldsymbol{w})} \tag{8}$$

where each $g_i(\boldsymbol{w},s)$ is a model feature function, λ_i is a parameter to be estimated and $Z(\boldsymbol{w})$ is the normalizing factor to ensure a proper probability. Plug Eq. (8) into (7), we get

$$-\sum_s \frac{\hat{f}_{sk}}{\widetilde{P}(s|k;\Lambda)} \sum_{d=1}^{D} \delta(k \in \boldsymbol{w}_d)(g_{jk} f_{jk} P(s|\boldsymbol{w}_d) - g_{jk} f_{jk} P(j|\boldsymbol{w}_d) P(s|\boldsymbol{w}_d)) \tag{9}$$

The final objective function consists of a GE term for each labeled features $k \in K$ with a zero-mean σ^2 variance Gaussian prior on parameters for regularization.

$$Q = -\mathrm{KL}(\hat{\boldsymbol{f}} \| E_\Lambda[\boldsymbol{f}(\boldsymbol{w},s)]) - \frac{\sum_k \lambda_k^2}{2\sigma^2} \tag{10}$$

In our experiments, we set $\sigma = 0.1$ and use L-BFGS to estimate model parameters.

In order to build a classifier based on GE, we need to first select the indicative feature words for each class, decide on their respective class labels, and suggest the target or reference word-class distribution for each feature. We will investigate various ways in doing these in Section 4.1.

3.2. Self-Learned features extraction

An initial classifier is built using the labeled features obtained from a sentiment lexicon and is subsequently applied on the document collection to infer sentiment labels. Documents with their labels inferred with high confidence from the initial classifier are added into a labeled document pool which is used to extract self-learned features following Algorithm 1. There is then a question on how to automatically select high quality classification results. This is more crucial to *self-training* since the model prediction is error prone and therefore pseudo-labeled examples with wrong labels might degrade the model performance during the iterative training process. Various pseudo-labeled example selection strategies (Medlock & Briscoe, 2007; Daumé, 2008) have been proposed. But these heuristic choices often require careful parameter tuning. In our algorithm presented here, we simply choose pseudo-labeled examples based on their posterior probability of class membership and use the class prediction probability threshold τ to filter out low confidence pseudo-labeled examples. As will be shown in Section 4.4, our proposed algorithm is not sensible to the setting of τ. Hence, we can simply ignore this parameter and select all the self-labeled examples for word-class distributions estimation.

Given the pseudo-labeled examples generated, we first select the most indicative feature words for each class based on information gain. We set the information gain threshold γ as the mean of the information gain scores of the top 200 most predictive features. The expected word-class distribution for a given word $w \in \mathscr{V}$ is defined as a vector $\hat{\boldsymbol{f}}(w) \in \mathbb{R}^S$ where

$$f(w,j) = \widetilde{P}(j|w;\Lambda). \tag{11}$$

That is, the jth element is the probability of a sentiment label $s = j$ being assigned given that word w is present in a document. Such a distribution can be estimated from pseudo-labeled examples as defined in Eq. (4).

4. Experiments

We evaluate our proposed framework on the two datasets, the movie-review (MR)dataset[4] and the multi-domain sentiment (MDS) dataset.[5] The MR dataset consists of 1000 positive and 1000 negative movie-reviews downloaded from the IMDB movie archive. The MDS dataset contains four different types of product reviews extracted from Amazon.com including Books, DVDs, Electronics and Kitchen appliances, with 1000 positive and 1000 negative reviews for each domain.

Preprocessing was performed on both of the datasets by removing punctuation, numbers, non-alphabet characters and stopwords. Summary statistics of the datasets before and after preprocessing are shown in Table 1. It can be seen that the MR data appears to be the largest dataset, nearly doubling in its vocabulary size compared to that of Books and DVDs. The Electronics and Kitchen datasets are smaller with their vocabulary size being only half of that of Books and DVDs. The MPQA subjectivity lexicon is used as a sentiment lexicon in our experiments. It contains 2,718 positive and 4,911 negative words. It should be noted that the MPQA subjectivity lexicon is domain-independent and does not bear any domain-specific information about the datasets used here.

[4] http://www.cs.cornell.edu/People/pabo/movie-review-data/.
[5] http://www.cs.jhu.edu/mdredze/datasets/sentiment/.

Please cite this article in press as: He, Y., & Zhou, D. Self-training from labeled features for sentiment analysis. *Information Processing and Management* (2010), doi:10.1016/j.ipm.2010.11.003

Table 1
Dataset and sentiment lexicon statistics in number of words.

Dataset	MR	MDS			
		Books	DVDs	Electronic	Kitchen
Corpus size[†]	674,662	214,350	212,413	146,159	129,587
Corpus size[*]	450,032	120,553	119,887	74,996	64,443
Vocabulary[†]	38,911	22,497	21,976	11,060	9809
Vocabulary[*]	38,408	21,998	21,488	10,585	9332
Matched polarity Words (pos./neg.)[*]	1091/1951	792/1285	773/1190	305/445	331/398

[*] Denotes after preprocessing.
[†] Denotes before preprocessing.

Algorithm 1. (Self-learned features extraction)

Input: The document collection $\mathscr{D} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_D\}$, the sentiment lexicon $\mathscr{L}$, the class prediction probability threshold τ, and the information gain threshold γ
Output: Self-learned features with their expected distribution, $\mathscr{F} = \{(w_1, \hat{\boldsymbol{f}}(w_1)), (w_2, \hat{\boldsymbol{f}}(w_2)), \ldots\}$
Construct an initial classifier upon samples of $\mathscr{D} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_D\}$ with prior information obtained from $\mathscr{L}$
for each document $\boldsymbol{w}_i \in \mathscr{D}$ **do**
  Infer its sentiment class label as $s_i = \arg\max_s P(s|w_i;\Lambda)$
  if $P(s_i|\, w_i;\Lambda) > \tau$ **then**
    Add labeled sample (w_i, s_i) into a labeled document pool $\mathscr{B}$
  end if
end if
for each distinct word w_t from the labeled document pool $\mathscr{B}$ **do**
  Calculate the information gain $IG(w_t)$ based on $\mathscr{B}$
  if $IG(w_t) > \gamma$ **do**
    Calculate the target expectation of $w_t, \hat{\boldsymbol{f}}(w_t)$ from $\mathscr{B}$
    Add $(w_t, \hat{\boldsymbol{f}}(w_t))$ into the self-learned feature list $\mathscr{F}$
  end if
end for

We randomly split the data with 90% data used as the training set and the remaining 10% data used as the test set. For all the results reported in this section, we performed 10 random splits and report the classification accuracy averaged over 10 such runs. It has to be noted that no labeled instances were used in the classifier training in our proposed framework.

4.1. Overall comparison

We compare our proposed approach with several other methods as described below:

- *Lexicon labeling*. We implemented a baseline model which simply assigns a score +1 and −1 to any matched positive and negative word respectively based on a sentiment lexicon. A review document is then classified as either positive or negative according to the aggregated sentiment score. Thus, in this baseline model, a document is classified as positive if there are more positive words than negative words in the document and vice versa.
- *Heuristic labeling*. For a dataset, the matched polarity words from a sentiment lexicon are extracted as features which are assumed to be highly predictive of their corresponding polarity class. It should be noted that features generated in this way is domain-independent and does not bear any domain specificity. For GE-based training, a target or reference expectation for each feature is required by the KL divergence calculation as in Eq. (5). A simple heuristic approach (Druck et al., 2008; Schapire et al., 2002) is adopted that a majority of the probability mass for a feature is distributed uniformly among its associated class(es), and the remaining probability mass is distributed uniformly among the other non-associated class(es). Since we are dealing with the binary classification problem here, the target expectation of a feature having its prior polarity (or associated class label) is 0.9 and 0.1 for its non-associated class.
- *Self-labeled instances*. This resembles the traditional *self-training* approach that documents labeled by the initial classifier trained using *Heuristic labeling* are taken as training examples to train a supervised classifier. The results reported here are from the Naïve Bayes classifier trained using document vectors with binary features. There are many different strategies in selecting the pseudo-labeled examples to be added into training set. We chose those self-labeled instances whose label prediction probability exceeds 0.8.

Please cite this article in press as: He, Y., & Zhou, D. Self-training from labeled features for sentiment analysis. *Information Processing and Management* (2010), doi:10.1016/j.ipm.2010.11.003

Table 2
Sentiment classification accuracy (%).

Method	MR	MDS				
		Books	DVDs	Electronics	Kitchen	Overall
Lexicon labeling	66.90	64.50	65.00	62.70	64.70	64.23
Heuristic labeling	71.20	66.75	70.95	71.30	71.10	70.03
Self-labeled instances	73.20	67.50	71.25	73.80	71.95	71.13
Self-learned features	**74.70**	**70.10**	**74.30**	**79.60**	**76.40**	**75.10**
Oracle labeling	81.36	76.10	78.52	81.05	80.80	79.12
Naïve Bayes	82.53	81.58	83.87	82.75	84.85	83.26

- *Self-learned features*. This is our proposed approach that an initial classifier is trained using *Heuristic labeling*. Documents labeled by the initial classifier are taken as labeled instances. Features are selected based on the information gain (IG) of the feature with the class label and the target expectation of each feature is re-estimated from the pseudo-labeled examples. A second classifier is then trained using these *self-learned features*.
- *Oracle labeling*. Similar to *Heuristic labeling*, features used here are also extracted from a sentiment lexicon. However, instead of using the simple heuristic to specify the target expectation of the features, we assume the availability of document labels and compute the exact target expectations from the labeled instances.
- *Naïve Bayes*. For comparison purposes, we also trained supervised classifiers including Naïve Bayes (NB), support vector machines (SVMs), and maximum entropy (ME) models. We preprocessed documents by stopword removal and stemming, and performed 10-fold cross validation. The results show that NB consistently outperforms SVM and ME and representing document vectors by binary features (word presence and absence) gives better results than using TFIDF features. Thus, we only report results from NB trained on document vectors with binary features.

Table 2 shows sentiment classification accuracies on the MR and MDS datasets using the different methods mentioned above. All the approaches outperform the baseline model which classifies documents based solely on the aggregated sentiment scores calculated from a sentiment lexicon. Incorporating the prior knowledge from the sentiment lexicon and training a classifier using these labeled features based on the GE criterion brings the classification accuracy to over 70% for almost all the datasets except the Books data. Using the *Self-learned features*, the classification accuracy is further improved by 3.35–8.95% with all improvements being statistically significant compared to other methods according to paired t-tests ($p < 0.05$).

We also notice that *Self-labeled instances* only improves the classification accuracy marginally compared to *Heuristic labeling* and it performs consistently worse than *Self-learned features*. The results suggest that *Self-learned features* appears to be a better choice since it does not introduce new examples with incorrect labels. Instead, it calculates word-class association probabilities by averaging over many pseudo-labeled examples which essentially has a smoothing effect and makes it more tolerant to class prediction errors. Thus, contrary to self-training, it avoids the incestuous bias problem.

If the true document labels are revealed, the exact target expectation for each feature can be calculated from the labeled corpus and we observe that *Oracle labeling* performs better than *Self-learned features* and achieves similar classification accuracies on the MR and Electronics datasets compared to the supervised Naïve Bayes approach.

4.2. Results by filtering polarity words by frequency

We have also conducted experiments by filtering infrequent polarity words. Different word frequency cutoff threshold has been tested, ranging between 0 and 50. It can be observed from Fig. 2 that without any filtering, the total number of matched polarity words in the MR dataset against the MPQA subjectivity lexicon is about 3000. After removing the polarity words that occurred less than five times in MR, the total number of matched polarity words is reduced to 1500 and the classification accuracy using either *Heuristic labeling* or *Self-learned features* improves. The accuracy saturates at the word frequency cutoff point 20 and the total number of matched polarity words is nearly 600. The performance gap between *Heuristic labeling* and *Self-learned features* seems to diminish when too few features words are selected.

We also notice that varying the polarity word frequency cutoff does not affect the performance using *Oracle labeling* much. The performance is the best when no filtering was done and only drops slightly with more polarity words filtered. This is not surprising since true class labels were used in *Oracle labeling* and the target expectations of the polarity word features reflect the true distributions over class labels.

For the MDS dataset, a similar trend is observed (Fig. 3) as in the MR dataset that the accuracy using *Heuristic labeling* or *Self-learned features* increases with more infrequent polarity words removed and it levels off before drops when too few polarity words are used as the initial labeled features. As mentioned earlier, Books and DVDs are larger corpora and thus the number of matched polarity words without filter is about 2000. While for Electronics and Kitchen, only about 750 polarity words can be found in the MPQA subjectivity lexicon. By filtering the polarity words that occurred less than five times in the corpus, the number of matched polarity words drops dramatically with only about 500 matched words for Books and DVDs, and 160 for Electronics and Kitchen. This results in a significantly increase in accuracy for all the datasets except Books in the range of 5.32% and 8.1%. The optimal polarity word frequency cutoff point is 10 for Books and DVDs and five for

Please cite this article in press as: He, Y., & Zhou, D. Self-training from labeled features for sentiment analysis. *Information Processing and Management* (2010), doi:10.1016/j.ipm.2010.11.003

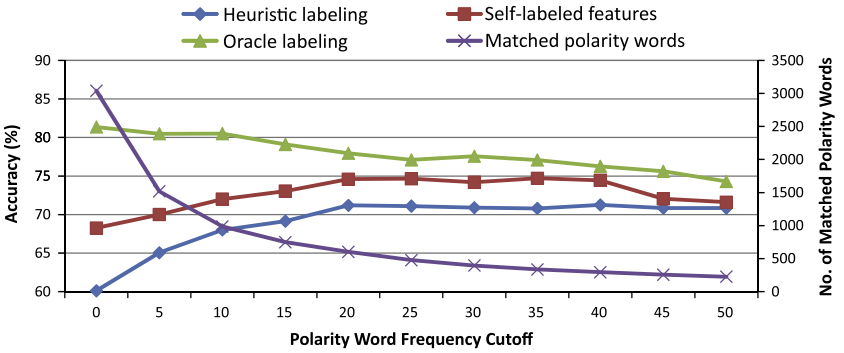

Fig. 2. Classification accuracy on MR vs. polarity words frequency cutoff.

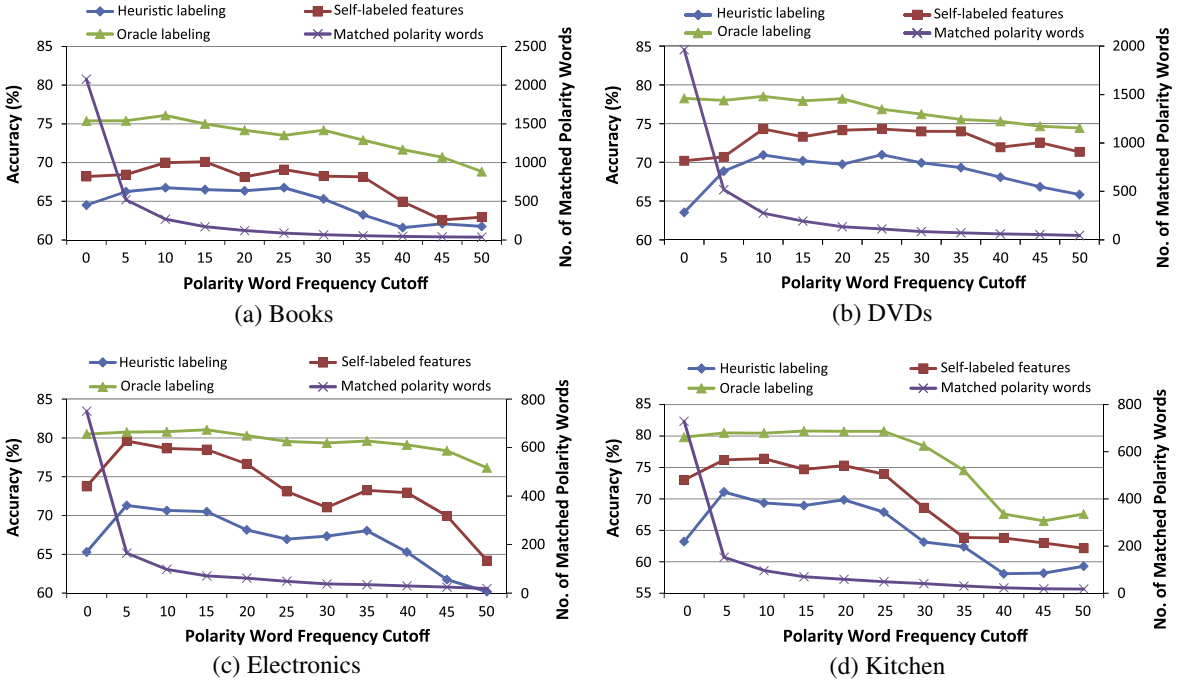

Fig. 3. Classification accuracy on the MDS dataset vs. polarity word frequency cutoff.

Electronics and Kitchen. At this point, the total number of matched polarity words is about 270 for Books and DVDs, and 160 for Electronics and Kitchen.

Using as few as 50 polarity words as features when the polarity word frequency cutoff point is 40 for Books and DVDs and 25 for Electronics and Kitchen, both *Heuristic labeling* and *Self-learned features* still outperform the baseline using *Lexicon labeling* for all the datasets.

Oracle labeling seems more robust to changes of the polarity word frequency cutoff. It performs fairly stable and only drops dramatically when too few polarity words were incorporated as prior knowledge, for example, when there are only 23 polarity words were selected at the cutoff point 40 for the Kitchen dataset.

4.3. Comparison with existing approaches

Li et al. (2009) employed lexical prior knowledge extracted from a sentiment lexicon that was developed in the IBM India Research Labs (Ramakrishnan, Jadhav, Joshi, Chakrabarti, & Bhattacharyya, 2003) for semi-supervised sentiment classification based on non-negative matrix tri-factorization. Such domain-independent prior knowledge was incorporated in conjunction with domain- dependent unlabeled data and a few labeled documents for model learning. With 10% of labeled documents for training, the non-negative matrix tri-factorization approach performed much worse than our approach with

Please cite this article in press as: He, Y., & Zhou, D. Self-training from labeled features for sentiment analysis. *Information Processing and Management* (2010), doi:10.1016/j.ipm.2010.11.003

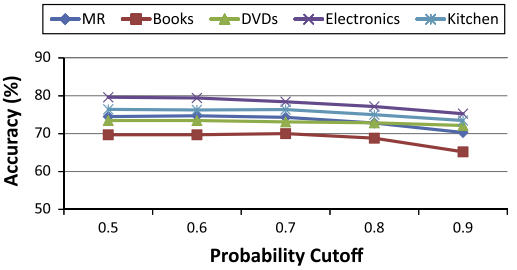

Fig. 4. Sensitivity of the self-learned feature extraction algorithm to the class prediction probability threshold.

a difference of 8–11% for *Heuristic labeling* and 13–15% for *Self-learned features* on both MR and MDS. Even with 40% labeled documents, their approach is still slightly worse than *Self-learned features* which uses no labeled documents.

Lin, He, and Everson (2010) conducted a comparative study of three closely related Bayesian models for sentiment classification, namely, the latent sentiment model (LSM), the joint sentiment-topic (JST) model, and the Reverse-JST model. They incorporated sentiment prior information extracted from both the MPQA subjectivity lexicon and the appraisal lexicon[6] by modifying conditional probabilities used in Gibbs sampling during model learning. The best sentiment classification results were obtained using LSM with 74.1% on MR and 69.3% on MDS. Our *Heuristic labeling* slightly outperforms LSM on MDS. *Self-learned features* gives a similar result on MR, but outperforms LSM on MDS by nearly 6%.

Dasgupta and Ng (2009) proposed a weakly-supervised sentiment classification algorithm where user feedbacks are provided on the spectral clustering process in an interactive manner to ensure that text are clustered along the sentiment dimension. Users are allowed to specify the dimension along which they want the data points to be clustered via inspecting a small number of words. They removed words that occur in only a single review and the top 1.5% words after sorting the vocabulary by document frequency. And we did not perform such preprocessing. Their proposed approach achieved 70.9% classification accuracy on MR and an average of 68.95% on the MDS dataset. Our *Heuristic labeling* gives slightly better performance on both datasets and *Self-learned features* outperforms their approach by a margin of nearly 4% on MR and 6% on MDS.

4.4. Sensitivity analysis

In this section, we explore the sensitivity to the class prediction probability threshold τ used in Algorithm 1. We measure the accuracy of our proposed approach with τ varying between 0.5 and 0.9. When τ is set to 0.5, essentially there is no filtering and all the pseudo-labeled examples are selected. By increasing τ, more pseudo-labeled examples with low confidence scores are filtered and less labeled instances are used to estimate word-class distributions. Fig. 4 shows that Algorithm 1 is fairly robust to changes of τ varying between 0.5 and 0.7 with accuracies changing in the range of 0.15% and 0.8%. However, as τ is increased beyond 0.8, accuracies drop more noticeably with the biggest drop of 4.5% observed for the Books dataset. This is in contrast to *self-training* approaches that it is important to only add the most confident pseudo-labeled examples into the training set in order to iteratively improve the model. Our proposed approach does not utilize pseudo-labeled examples directly and instead use them to estimate the word-class distributions which makes it more robust to the pseudo-labeled example selection strategies. The results also indicate that we can essentially ignore τ and simply choose all the pseudo-labeled examples for automatic feature acquisition.

4.5. Domain-specific polarity words

While a generic sentiment lexicon provides useful prior knowledge for sentiment analysis, the contextual polarity of a word may be quite different from its prior polarity. Positive words may appear in sentences describing negative sentiment, and vice versa. Also, the same word might have different polarity in different domain. For example, the word "*small*" is positive when used to describe a mobile phone, but it is negative if it is used to describe a SUV. Thus, it is worth to automatically distinguish between prior and contextual polarity. Our proposed framework starts with a generic sentiment lexicon and estimates word-polarity association probabilities from pseudo-labeled examples. Indeed, as seen from Table 3, the proposed framework is able to extract domain-dependent feature words and estimate the word-class probabilities from a particular domain and thereby reflect a domain-specific sentiment polarity for each word.

Table 3 lists some extra polarity words extracted by our approach which are not found in the MPQA subjectivity lexicon. We can see that the proposed framework is able to identify domain-specific polarity words. For example, *complicated* is generally considered as negative in other context, but it expresses positive opinion in describing movie plots. We also observe other domain-specific terms for the MR dataset, such as the actress name *winslet* (kate Winslet) with positive polarity and

[6] http://lingcog.iit.edu/arc/appraisal_lexicon_2007b.tar.gz.

Please cite this article in press as: He, Y., & Zhou, D. Self-training from labeled features for sentiment analysis. *Information Processing and Management* (2010), doi:10.1016/j.ipm.2010.11.003

Table 3
Extracted example polarity words by our approach.

Corpus		Extracted polarity words
MR	Pos	colorful, complicated, effective, enchanting, excels, finest, lively, natural, surprisingly, winslet
	Neg	badness, batman, crap, dull, fails, incoherent, lousy, miserably, pointless, unfunny
Books	Pos	accessible, accurately, cool, easy, gift, golden, helpful, meaningful, variety, simple
	Neg	boredom, contradictory, dark, dull, foreign, inaccurate, repeating, sex, sick, tiresome, unbalanced, violent
DVDs	Pos	academy, amazingly, classic, finest, heartwarming, imaginative, oscar, shine, timeless, winner
	Neg	budget, complain, fails, ignorance, pointless, sucks, unconvincing, waste, weak, wrong
Elec.	Pos	cheaper, easy, fast, light, plasma, promptly, quickly, satisfied, small, wide
	Neg	avoid, broken, confused, crashes, dangerous, failure, garbage, ineffective, junk, plastic, useless
Kitchen	Pos	balanced, comfortable, contemporary, easy, handless, heavy, large, loves, practical, sharp
	Neg	broke, burned, dirty, disappointed, failure, junk, leaky, noisy, risk, sticky

the movie name *batman* bearing negative polarity. For the MDS datasets, example domain-specific terms include *foreign* for Books, *oscar* for DVDs, *small* and *crashes* for Electronics, and *contemporary* and *burned* for Kitchen.

5. Conclusions and future work

In this paper, we have proposed a novel framework where prior knowledge from a generic sentiment lexicon is used to build a classifier where preferences on expectations of sentiment labels of those lexicon words are expressed using generalized expectation criteria. Pseudo-labeled documents by this classifier are used to automatically acquire domain-specific feature words whose word-class distributions are estimated and are subsequently used to train another classifier by constraining the model's predictions on unlabeled instances. Experiments on both the movie-review data and the multi-domain sentiment dataset show that our approach attains comparable or better performance than existing weakly-supervised sentiment classification methods despite using no labeled documents. Moreover, our approach is simple and robust and does not require careful parameter tuning. Although this paper primarily studies sentiment analysis, the proposed approach is applicable to any text classification task where some relevant prior knowledge is available.

A promising direction for future work is to incorporate ontology engineering into weakly-supervised model learning. By incorporating domain-independent knowledge from a sentiment lexicon as well as domain knowledge from ontologies, we are hoping to reveal both topics and sentiment labels of a document simultaneously.

References

Abbasi, A., Chen, H., & Salem, A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems, 26*(3), 1–34.

Andreevskaia, A., & Bergler, S. (2008). When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. In *Proceedings of the association for computational linguistics and the human language technology conference* (ACL-HLT) (pp. 290–298).

Argamon, S., Bloom, K., Esuli, A., & Sebastiani, F. (2007). Automatically determining attitude type and force for sentiment analysis. In *Proceedings of the 3rd language and technology conference* (pp. 218–231).

Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the association for computational linguistics* (ACL) (pp. 440–447).

Choi, Y., Cardie, C., Riloff, E., & Patwardhan, S. (2005). Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing* (HLT/EMNLP) (pp. 355–362).

Dasgupta, S., & Ng, V. (2009). Topic-wise, sentiment-wise, or otherwise? Identifying the hidden dimension for unsupervised text classification. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 580–589).

Daumé III, H. (2008). Cross-task knowledge-constrained self-training. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 680–688).

Druck, G., Mann, & G., McCallum, A. (2008). Learning from labeled features using generalized expectation criteria. In The 31st annual international ACM SIGIR conference (pp. 595–602).

He, Y. (2010). Learning sentiment classification model from labeled features. In *Proceeding of the 19th ACM conference on information and knowledge management* (CIKM).

Kaji, N., & Kitsuregawa, M. (2006). Automatic construction of polarity-tagged corpus from html documents. In *Proceedings of the COLING/ACL on main conference poster sessions* (EMNLP) (pp. 452–459).

Kanayama, H., & Nasukawa, T. (2006). Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the conference on empirical methods in natural language processing* (EMNLP) (pp. 355–363).

Kim, S.-M., & Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the international conference on computational linguistics* (COLING) (pp. 1367–1373).

Li, T., Zhang, Y., & Sindhwani, V. (2009). A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In *Proceedings of the joint conference of the annual meeting of the association for computational linguistics and the international joint conference on natural language processing of the asian federation of natural language processing* (ACL-IJCNLP) (pp. 244–252).

Lin, C., & He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM international conference on information and knowledge management* (CIKM) (pp. 375–384).

Lin, C., He, Y., & Everson, R. (2010). A comparative study of bayesian models for unsupervised sentiment detection. In *Proceedings of the 14th conference on computational natural language learning* (CoNLL) (pp. 144–152).

McCallum, A., Mann, G., & Druck, G. (2007). Generalized expectation criteria. Tech. Rep. 2007-60, University of Massachusetts Amherst.

Please cite this article in press as: He, Y., & Zhou, D. Self-training from labeled features for sentiment analysis. *Information Processing and Management* (2010), doi:10.1016/j.ipm.2010.11.003

Medlock, B., & Briscoe, T. (2007). Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of the annual meeting on association for computational linguistics* (ACL) (pp. 992–999).

Melville, P., Gryc, W., & Lawrence, R. D. (2009). Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD conference on knowledge discovery and data mining* (KDD) (pp. 1275–1284).

Narayanan, R., Liu, B., & Choudhary, A. (2009). Sentiment analysis of conditional sentences. In *Proceedings of the conference on empirical methods in natural language processing* (EMNLP) (pp. 180–189).

Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the annual meeting on association for computational linguistics* (ACL) (pp. 271–278).

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the conference on empirical methods in natural language processing* (EMNLP) (pp. 79–86).

Qiu, L., Zhang, W., Hu, C., & Zhao, K. (2009). Selc: A self-supervised model for sentiment classification. In *Proceeding of the 18th ACM conference on information and knowledge management* (CIKM) (pp. 929–936).

Ramakrishnan, G., Jadhav, A., Joshi, A., Chakrabarti, S., & Bhattacharyya, P. (2003). Question answering via Bayesian inference on lexical relations. In *Proceedings of the ACL 2003 workshop on multilingual summarization and question answering* (pp. 1–10).

Read, J., & Carroll, J. (2009). Weakly supervised techniques for domain-independent sentiment classification. In *Proceeding of the 1st international CIKM workshop on topic-sentiment analysis for mass opinion* (pp. 45–52).

Schapire, R., Rochery, M., Rahim, M., & Gupta, N. (2002). Incorporating prior knowledge into boosting. In *Proceedings of the 19th international conference on machine learning* (ICML) (pp. 538–545).

Tan, S., Wang, Y., & Cheng, X. (2008). Combining learn-based and lexicon-based techniques for sentiment detection without using labeled examples. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 743–744).

Turney, P. D. (2002). Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics* (ACL) (pp. 417–424).

Turney, P. D., & Littman, M. L. (2002). Unsupervised learning of semantic orientation from a hundred-billion-word corpus. ArXiv Computer Science e-prints cs.LG/0212012.

Whitelaw, C., Garg, N., & Argamon, S. (2005). Using appraisal groups for sentiment analysis. In *Proceedings of the ACM international conference on information and knowledge management* (CIKM) (pp. 625–631).

Wu, X., & Srihari, R. (2004). Incorporating prior knowledge with weighted margin support vector machines. In *Proceedings of the 10th ACM SIGKDD conference on knowledge discovery and data mining* (KDD) (pp. 326–333).

Zagibalov, T., & Carroll, J. (2008a). Automatic seed word selection for unsupervised sentiment classification of Chinese text. In *Proceedings of the 22nd international conference on computational linguistics* (COLING) (pp. 1073–1080).

Zagibalov, T., & Carroll, J. (2008b). Unsupervised classification of sentiment and objectivity in chinese text. In *Proceedings of the 3rd international joint conference on natural language processing* (IJCNLP) (pp. 304–311).

Zhao, J., Liu, K., & Wang, G. (2008). Adding redundant features for CRFs-based sentence sentiment classification. In *Proceedings of the conference on empirical methods in natural language processing* (EMNLP) (pp. 117–126).

Please cite this article in press as: He, Y., & Zhou, D. Self-training from labeled features for sentiment analysis. *Information Processing and Management* (2010), doi:10.1016/j.ipm.2010.11.003