Sentiment Analysis by Augmenting Expectation Maximisation with Lexical Knowledge

Xiuzhen Zhang[1]*, Yun Zhou[1]**, James Bailey[2], and Kotagiri Ramamohanarao[2]

[1] School of Computer Science & IT, RMIT University, Australia
{xiuzhen.zhang}@rmit.edu.au
[2] Dept of CIS, The University of Melbourne, Australia
{baileyj,kotagiri}@unimelb.edu.au

Abstract. Sentiment analysis of documents aims to characterise the positive or negative sentiment expressed in documents. It has been formulated as a supervised classification problem, which requires large numbers of labelled documents. Semi-supervised sentiment classification using limited documents or words labelled with sentiment-polarities are approaches to reducing labelling cost for effective learning. Expectation Maximisation (EM) has been widely used in semi-supervised sentiment classification. A prominent problem with existing EM-based approaches is that the objective function of EM may not conform to the intended classification task and thus can result in poor classification performance. In this paper we propose to augment EM with the lexical knowledge of opinion words to mitigate this problem. Extensive experiments on diverse domains show that our lexical EM algorithm achieves significantly higher accuracy than existing standard EM-based semi-supervised learning approaches for sentiment classification, and also significantly outperforms alternative approaches using the lexical knowledge.

Keywords: Sentiment Analysis, Expectation Maximisation, Semi-supervised Learning, Text Classification

1 Introduction

The Web provides a platform for the public to freely express their opinions, where blogs, product reviews and movie reviews are popularly used forums. Sentiment analysis aims to identify the positive or negative opinions and sentiments expressed in documents (blog posts or reviews) [15]. Machine learning approaches have been widely used for sentiment analysis [16, 18, 12]. Especially Pang et al [16] cast the sentiment analysis of documents as a supervised text classification problem, where documents are classified as carrying *positive* or *negative* labels.

On the other hand, linguistic studies have mostly focused on identifying words and phrases that express subjectivity, either manually or automatically [7, 17, 20,

* Work was done while Zhang was on sabbatical leave at the University of Melbourne.

** Currently at The University of Melbourne. Email: yuzhou@student.unimelb.edu.au.

21]. Opinion words (also called subjectivity or sentiment words) are words that express prior, out of context positive or negative polarity. For example words like *adore* and *perfect* carry prior positive polarity whereas *abhor* and *insane* carry prior negative polarity. Some publicly available opinion word repositories include SentimentWordNet [1] and the General Inquirer positive and negative word lists [6]. As opinion words have sentiment labels, they are also called labelled words [10] or more generally labelled features [4] from the perspective of classification learning.

Generally supervised learning algorithms for text classification require a large number of labelled documents for effective learning. But labelling documents is costly. Semi-supervised learning approaches [13] that leverage unlabelled documents for effective learning from limited training documents are appealing. Recently semi-supervised learning in the new form of learning from labelled words has also attracted lots of attention from the research community [10, 12, 18].

Expectation Maximisation (EM) [3] has been employed as the key mechanism for both forms of semi-supervised learning. In [13], EM is combined with Naive Bayes to find classifier parameters that maximises the likelihood of both the labelled and unlabelled documents. In [10], word labels are used to construct fictitious exemplar positive and negative documents and unlabelled documents are "softly" labelled according to their distance to the exemplar documents. EM is then applied to build the classification model. A prominent problem with the standard EM procedure is that it may optimise parameters of a generative model whose objective function does not conform to its intended purpose of classification, which can result in poor classification performance [2, 13] .

In this paper we propose to augment the EM algorithm with the lexical knowledge of word labels. To this end, we modify the expectation computation step for the probabilistic document labels in the EM procedure by combining any given document class labels and labels derived heuristically from word labels. The Naive Bayes generative model is applied to re-estimate word distribution (data likelihood) under the adjusted document class expectation. It is typically difficult to incorporate prior knowledge into the EM process. In our approach the latent variables in the generative model have been constrained and directed by the lexical knowledge in a simple yet effective approach. Our approach can reduce the problem of mismatch of posterior distribution between latent variable for EM and the objective class label variable for classification.

We conduct experiments on sentiment classification of real-world datasets including blogs on different topics and movie reviews. Experiments show that our lexical knowledge augmented EM (Lexical EM) approach significantly improves semi-supervised classification based on labelled documents [13] as well as based on labelled words [10], where standard EM is employed. It often achieves better results than a recently proposed linear pooling approach of combining lexical knowledge with machine learning [12]. Especially our lexical knowledge augmented EM approach achieves effective learning when there are few labelled documents for training.

2 Related Work

Opinion mining and sentiment analysis have attracted active research recently. Overview of developments in this area has been described in [15] and [9]. Turney [19] employed lexical knowledge phrases (adjectives followed by nouns, or adverbs followed by verbs) to develop an unsupervised learning algorithm for classifying opinions of reviews. The polarity of phrases is computed by searching the Web to compute its similarity to the positive and negative reference word "excellent" and "poor" respectively, and thus the proposed approach can not be easily generalised to applications like blog sentiment analysis where there are not ready-made reference words.

Pang et al. [16] showed that using lexical information in a naive approach of counting the occurrences of positive and negative opinion words for sentiment classification is not as effective as building text classification models using training examples. It is worth noting that their conclusion is conditioned on that a large number of labelled documents are available for training an accurate classification model. But generally labelling documents is a costly task. The objective of this study is to achieve accurate sentiment classification with few labelled documents. We have shown that making use of the lexical knowledge in a more intelligent way can compensate for the shortage of labelled training documents and can achieve fairly accurate sentiment classification.

Incorporating domain knowledge into standard text classification has been actively studied recently [4, 12, 18]. In [4] and [18] word labels are used directly to constrain the class distribution computation for documents in discriminative models. In contrast the main purpose of our work is to incorporate prior lexical knowledge into the EM process which is based on a generative model such as Naive Bayes. Rather than modifying the class distribution parameters directly we modify the parameters for generating the classification model. In [12], linear pooling is used to combine the word distribution in classes estimated from the training data and the lexical knowledge. Naive Bayes is then employed for classification. Different from their approach of incorporating lexical knowledge by modifying word distributions in classes, we adjust the expected class distribution for documents making use of both the lexical knowledge, and labelled and unlabelled documents. Our experiments show that our approach often achieves better classification accuracy with fewer labelled documents.

There has been previous work on improving the EM process. Graca et al. [8] proposed a general method that incorporates prior constraints into the EM process, focusing on clustering and the alignment problem for statistical machine translation. In [5] a general semi-supervised learning framework is developed to constrain the posterior distribution of latent variables under a set of feature expectation constraints. The generative model parameters are estimated with a coordinate ascent algorithm. Our approach is consistent with this general framework but more importantly has shown a practical way of incorporating domain knowledge into this general framework for document sentiment classification.

3 The Semi-supervised Learning

In this section we describe two forms of semi-supervised learning, in the context of sentiment classification of documents with a limited number of labelled documents. EM has been employed in both semi-supervised settings.

3.1 Semi-supervised learning with labelled documents

EM is an iterative algorithm for maximum likelihood or maximum a posteriori estimation in problems with incomplete data [3]. For text classification, the data is incomplete in the sense that class labels for documents are missing. In the semi-supervised learning paradigm, both the labelled and unlabelled documents are used to derive a classification model [13]. A Naive Bayes (NB) classifier is firstly trained with the available labelled documents, and it is used to assign probabilistic class labels to unlabelled documents. The EM procedure is then employed to train a new classifier using both the originally labelled and unlabelled documents. The EM process iterates to find the word distribution for classes that maximises the likelihood of all documents.

NB is a probabilistic generative model for data, and it is the base classification model to incorporate unlabelled documents for learning. Each document is generated according to a probability distribution defined by a set of parameters — the word distribution for classes. NB estimates the word distribution for classes using only labelled training documents, and then uses the estimated word distribution to classify new documents — computing the probability for a document in each class and the most likely class is thought to have generated the document.

Consider a collection D of documents for training, where each document is labelled with a class label. For ease of discussion we assume that there are only two class labels, the positive and the negative. Suppose that V is the vocabulary for D. Given a document d and class labels C_j, $j \in \{+, -\}$, and under the assumption of independent word distribution for classes, the probability that each class has generated the document is

$$P(C_j|d) = \frac{\prod_{t \in d} P(t|C_j) * P(C_j)}{P(d)}.$$

For a class C_j, its prior probability is the proportion of documents in the collection that belong to class C_j:

$$P(C_j) = \frac{N(d, C_j)}{|D|}.$$

The term distribution in each class is computed as

$$P(t|C_j) = \frac{N(t \in C_j) + 1}{\sum_{t' \in V}(N(t' \in C_j) + 1)}$$

where $N(t \in C_j)$ and $N(t' \in C_j)$ are respectively the total number of occurrences of t and t' in documents with class label C_j. Note that the demoninator is computed from only terms appearing in class C_j.

When NB is given just a small set of labelled training documents, classification accuracy will suffer since variance in the parameter estimates $P(t|C_j)$, $j \in \{+,-\}$ is high. EM can improve the estimation for $P(t|C_j)$ making use of the unlabelled documents. For the originally labelled documents, $P(C_j|d)$ is already known:

$$P(C_j|d) = \begin{cases} 1 & \text{if } d \in C_j \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

For each originally unlabelled document d, $P(C_j|d)$, $j \in \{+,-\}$ is estimated using the EM process by iterating the following two steps, until $P(t|C_j)$ and $P(C_j)$ converge.

- The Expectation step (E-step): For each document d and a class C_j, $P(C_j|d)$ is estimated as follows:

$$P(C_j|d) = \frac{P(d|C_j) * P(C_j)}{P(d)} = \frac{\prod_{t \in d} P(t|C_j) * P(C_j)}{P(d)} \tag{2}$$

  In the above equation $P(d)$ does not need to be computed. Rather for two classes, $P(C_j|d)$ is normalised by the sum of the numerator for all classes.
- The Maximisation step (M-step): At this step, model parameters term distribution $P(t|C_j)$ and class priors $P(C_j)$ are re-computed. For each word t, $P(t|C_j)$ is re-computed based on $P(C_j|d)$:

$$P(t|C_j) = \frac{\sum_{d \in D} N(t \in d) * P(C_j|d) + 1}{\sum_{t' \in V} \sum_{d \in D} N(t' \in d) * P(C_j|d) + |V|}$$

  where $|V|$ is the vocabulary size. Note that Laplace smoothing is applied to avoid zero-probabilities. Class priors $P(C_j)$ are re-computed as follows:

$$P(C_j) = \frac{\sum_{d \in D} P(C_j|d)}{|D|}.$$

3.2 Semi-supervised learning with labelled words

For sentiment classification of documents, some opinion words express prior, out of context sentiment polarities. For example "excellent" is typically associated with the positive polarity whereas "abhor" is associated with the negative polarity. For topic classification of documents, some words are strong indicators for topics. For example, to classify documents into the topics "baseball" versus "hockey", the word "puck" strongly indicates that the document is about "hockey". In text classification words that are strongly associated with class labels are called labelled words. Labelling words are reported to be less costly than labelling documents [4, 10].

Another semi-supervised document classification scheme is to use a lexicon of labelled words rather than labelled documents. A typical approach of incorporating labelled words into the learning model is by creating pseudo documents using the labelled words, and a representative piece of work is by Liu et al. [10]. In their approach, given a lexicon of representative set of words for each class, representative documents are constructed containing all representative words for each class. The cosine similarity between each unlabelled document and the representative documents are computed. As a result unlabelled documents are softly assigned the label of the class with the highest similarity. Using the soft labels as a start, the EM process is then employed to iteratively improve the word distribution computation and the class probability computation for documents.

It has been shown in [13] that semi-supervised classification based on the standard EM procedure can significantly improve classification accuracy when there are only limited labelled documents. Note that EM finds the word distribution for classes with locally maximal likelihood given both the labelled and unlabelled document. Note also that the underlying assumption for semi-supervised classification based on the standard EM procedure is that components of the generative model correspond to classes for our intended text classification task. However such assumptions do not always hold in real applications. It has been shown that the model may degrade the classification performance when the model parameters are misspecified [2]. In Section 5, we describe making use of the domain knowledge of word labels to modify the basic EM algorithm, aiming to address the issue of performance degradation due to violated assumptions.

4 The Lexical Knowledge Model

When no labelled data or other domain knowledge are available in a new domain, a simple document sentiment classification model can be built using the lexical knowledge of labelled opinion words. Given a lexicon of positive and negative words, the probability that a document d belongs to the positive class can be computed as

$$P'(C_+|d) = \frac{a}{a+b}$$

where a and b represent respectively the number of occurrences of positive and negative words in the document d. Without any prior information regarding the relative frequency of positive and negative words in a domain, a document with more positive words is likely to express overall positive sentiment, while a document with more negative words expresses overall negative feeling. In particular, a document d is classified as positive if $P(C_+|d) \geq 0.5$, and negative otherwise.

In this study we use the comprehensive and generic opinion lexicon developed by Wilson et al. [21]. Words in the lexicon are categorised as strongly subjective or weakly subjective. Words that are subjective in most contexts are *strongly subjective* and those that may only have certain subjective usages are *weakly subjective*. Subjective words are tagged with their prior polarity. The *positive* and *negative* tags are for positive and negative polarities respectively. The *both* tag is

Positive	acclaim, adore, affirm, befit, catalyst, dear, defer, encourage, fantastic, good, hero, loyalty, marvel, nice, perfect, radiant, sane, thrill, understand, want, yearn, zest
Negative	abash, abhor, accuse, admonish, agonize, beg, chao, defunct, excess, fear, grief, hell, insane, lose, malignant, nightmare, object, penalty, quarrel, racist, scandal, thwart, unfair, virus, yawn, zealous

Table 1. Some words in our opinion lexicon

for words that can express both positive and negative polarities. The *neutral* tag is for words expressing subjectivity but not obvious positive or negative polarity.

We apply some filtering criteria to the original subjectivity lexicon to construct the sentiment lexicon for our research. We first remove the small fraction of words with the *both* or *neutral* tags from the original lexicon. As a result only positive and negative words are kept for further consideration. For the positive and negative words, we further remove the weak subjective words. Our final opinion lexicon consists of 3218 unique words after stemming. In total there are 1067 positive and 2151 negative words in our final sentiment lexicon.

Some opinion words randomly chosen from our opinion lexicon are listed in Table 1. The opinion lexicon we use is a generic lexicon without any specific domain in mind. As a result, depending on how applicable the lexicon is to different domains, performance of the simple lexical knowledge-based classification model can vary in different domains. As will be discussed in Section 6, our experiments show that such a simple lexical knowledge classification model achieves modest classification accuracy for blogs while significantly more accurate classification for movie reviews.

5 Augmenting EM with Lexical Knowledge

We now describe how to modify the standard EM process described in Section 3 to effectively incorporate the lexical knowledge model for more accurate sentiment classification. With standard EM, the labelled documents are used to initialise the parameters for the EM hill climbing process. When the labelled documents are limited, during the iteration the unlabelled documents have significant effect on setting the parameters $p(t|C_j)$ $(t \in V,\ j \in \{+,-\})$. As the EM process is set to maximise the likelihood of labelled as well as unlabelled documents, this standard hill-climbing process may result in maximum data likelihood estimation that leads to latent variable estimation drifting away from our target classification function .

Our main idea of modifying EM is to modify the class distribution for unlabelled documents at the expectation step. Our adjustment is intended to constrain the EM process towards generating document class distributions more consistent with our target classification task, and the adjustment is achieved by incorporating the lexical knowledge model. In particular, we modify the pos-

Algorithm 1 The Lexical EM algorithm

```
Input:
    A set of labelled documents $D^L$ and a set of unlabelled documents $D^U$.
    Documents are defined on vocabulary $V$, and class labels are $\{+, -\}$.
    An opinion lexicon $X$.
Output:
    A classification model $\theta$ with parameters $P(C_j)$ and $P(t|C_j)$, $t \in V$, $j \in \{+, -\}$.
    {// In the description next $t \in V$ and $j \in \{+, -\}$.}
 1: Train a Naive Bayes classifier $\theta^0 = \langle P^0(C_j), P^0(w_i|C_j) \rangle$ from $D^L$
 2: for ($k = 1$; $\theta^k$ improves over $\theta^{k-1}$; $k++$) do
 3:    Compute the weight $\alpha^{k-1}$ in Equation 3 for classifier $\theta^{k-1}$
       {// Lines 4–8: E-step}
 4:    Compute $P^{k-1}(C_j|d \in D^L)$ from class labels
 5:    for each document $d \in D^u$ do
 6:       Compute $P^{k-1}(C_j|d)$ using classifier $\theta^{k-1}$
          {// Line 7: Equation 3}
 7:       Adjust $P^{k-1}(C_j|d)$ by $\alpha^{k-1}$, $\theta^{k-1}$ and the lexical knowledge model using $X$
 8:    end for
       {// Line 9: M-step}
 9:    Compute classifier $\theta^k = \langle P^k(C_j), P^k(t|C_j) \rangle$ from $P^{k-1}(C_j|d \in D^L \cup D^U)$
10: end for
11: Return the final classification model $\theta^k = \langle P^k(C_j), P^k(t|C_j) \rangle$
```

teriori class distribution for documents as in Equation 3, where $P(C_j|d)$ and $P'(C_j|d)$ represent respectively the class probability for documents computed from the generative model and lexical model.

$$P(C_j|d) = \begin{cases} P(C_j|d) & \text{if } d \text{ is a labelled document} \\ \alpha P(C_j|d) + (1-\alpha)P'(C_j|d) & \text{otherwise.} \end{cases} \tag{3}$$

- If a document d has a label, its class probability $P(C_j|d)$ remains unchanged. Following Equation 1, if d has a positive class label $P(C_+|d) = 1$ and $P(C_-|d) = 0$; otherwise $P(C_+|d) = 0$ and $P(C_-|d) = 1$.
- If a document d is an originally unlabelled document, $P(C_j|d)$ is adjusted in each iteration as a weighted sum of $P(C_j|d)$ computed from the current estimation of $P(t|C_j)$ and $P(C_j)$, and $P'(C_j|d)$ according to the lexical knowledge model (Section 4).

In adjusting the class probability for unlabelled documents, generally weighting factors should be set according to the classification accuracy of each component. α is a normalised weight factor according to the classification accuracies of the NB generative and lexical knowledge models of a current iteration. Our lexical EM algorithm is as shown in Algorithm 1. The algorithm starts with initialisation by a Naive Bayes classifier θ^0 trained on the set of labelled documents D^L (line 1). Parameters for classifier θ^0 include $P^0(C_j)$ and $P^0(t|C_j)$, $t \in V$, $j \in \{+, -\}$. The accuracy of θ^0 is estimated from the labelled documents in D^L. The initial class distribution expectation for the unlabelled documents is

Dataset	Description	#Pos	#Neg
Cartoon Blog	Worldwide opinions to the cartoons depicting the Muslim prophet Muhammad printed in a Danish newspaper.	107	107
McDonald's Blog	Opinions regarding the food at McDonald's restaurants.	41	41
Economic Forum Blog	Opinions on the World Economic forum in Davos, Switzerland.	38	38
Challenger Blog	Opinions about the Challenger space shuttle disaster.	104	81
Bolivia Blog	Documents that show opinions about Bolivia.	41	41
Movie Review	Movie reviews from the Internet Movie Database.	1000	1000

Table 2. Experimental data sets

computed from θ^0 — the unlabelled documents in D^U are assigned conditional labels by applying θ^0, and then adjusted by the accuracy of θ^0 and prior lexical knowledge using Equation 3 (lines 5–8). From the class distribution expectations of both the labelled and unlabelled training documents, new parameters $P^1(C_j)$ and $P^1(t|C_j)$ with the maximal likelihood are computed. The new parameters form a new classifier θ^1. The expectation and maximisation steps are iterated until the parameters for model θ^k are not improving.

6 Experiments

We conduct experiments to examine the performance of our approach to augmenting EM with lexical knowledge. Our experimental data sets are extracted from the TREC Blog06 collection [14, 11] that was used in the TREC-2006 and TREC-2007 conferences (Blog Track). NIST organised the relevance assessment for the opinion finding task. Given a target topic, if a blog post or its comments is not only on target, but also contains an explicit expression of opinion or sentiment towards the target, showing some personal attitude of the writer(s) then the document is judged as negatively opinionated, mixed or positively opinionated. On five topics from the Blog06 collection where our opinion lexicon has relatively good coverage, we randomly selected a roughly equal number of positive and negative documents for each topic. Table 2 summarises the five blog datasets used in our experiments, and they cover opinion analysis on a wide range of topics, including political figures, restaurants and political events. Table 2 also includes the movie review dataset that is popularly used in literature [16] for sentiment analysis. While the casual writing style is popular in blogs formal language expressions are mostly used in movie reviews. As will be seen next sentiment classification for blogs is significantly more challenging than that for movie reviews.

Dataset	Lexical EM	Linear Pooling	Word Supervsion	Doc Supervision
Cartoon Blog	**56.54**	54.73	51.53	53.05
McDonald's Blog	64.70	**65.09**	54.02	58.17
Economic Forum Blog	**70.17**	66.99	59.94	67.68
Challenger Blog	**58.71**	56.56	52.65	49.33
Bolivia Blog	**64.24**	60.46	53.83	53.65
Movie Review	**82.23**	78.61	61.96	76.27

Table 3. The average accuracies on each dataset for all models

6.1 Overview of results

We implemented Lexical EM using the accuracy-based weighting function. We compare our lexical EM algorithm with the two semi-supervised learning algorithms based on standard EM (Section 3). We also compare our lexical EM algorithm against the linear pooling algorithm by Melville et al. [12]. All EM processes are set to finish after 10 iterations when the EM parameters start to converge. Classification accuracy is obtained by 10-fold cross validation experiments, where for each fold only a proportion (5%–70%) of the hold-out documents are used as labelled training data. The final results are the average of 10 runs of cross validation.

The average accuracies for all models using different proportions of labelled documents are shown in Table 3, where Word Supervision and Doc Supervision refer to the semi-supervised learning algorithms using labelled words and labelled documents respectively. Overall Lexical EM outperforms other models and has the highest accuracy in five out of 6 domains. According to the paired Wilcoxon signed rank test, the improvements in accuracy of Lexical EM over Word Supervision and Document Supervision are statistically significant ($p < 0.05$) on all datasets. Lexical EM outperforms Linear Pooling statistically significantly on five out of six datasets while shows similar accuracy on the McDonald's blog.

The learning curves for all models using 5%–70% of labelled documents are plotted in Fig. 1. The figure clearly demonstrates the effectiveness of incorporating prior knowledge into the standard EM process, especially when there are limited labelled documents. When there are only 5% labelled training documents Lexical EM shows substantial performance improvement over Linear Pooling, Document Supervision or Word Supervision ($p < 0.05$ for the paired Wilcoxon signed rank test). The lexical knowledge used in the Lexical EM approach provides a reliable source to boost the estimation of $P(C_j|d)$ for a document d than trying to estimate model parameters from a limited set of labels. On the McDonald's blog, with only 8 (10%) labelled documents Lexical EM achieves 14.31% increment in accuracy over Document Supervision, equivalent to relative improvement of 28.37% in accuracy; with only 4 (5%) labelled documents, Lexical EM achieves 17.36% increment in accuracy over NB using labelled documents, equivalent to relative improvement of 47.17% in accuracy.

Not using any labelled documents, the classification accuracy of Word Supervision using only labelled words does not show improvement with more labelled

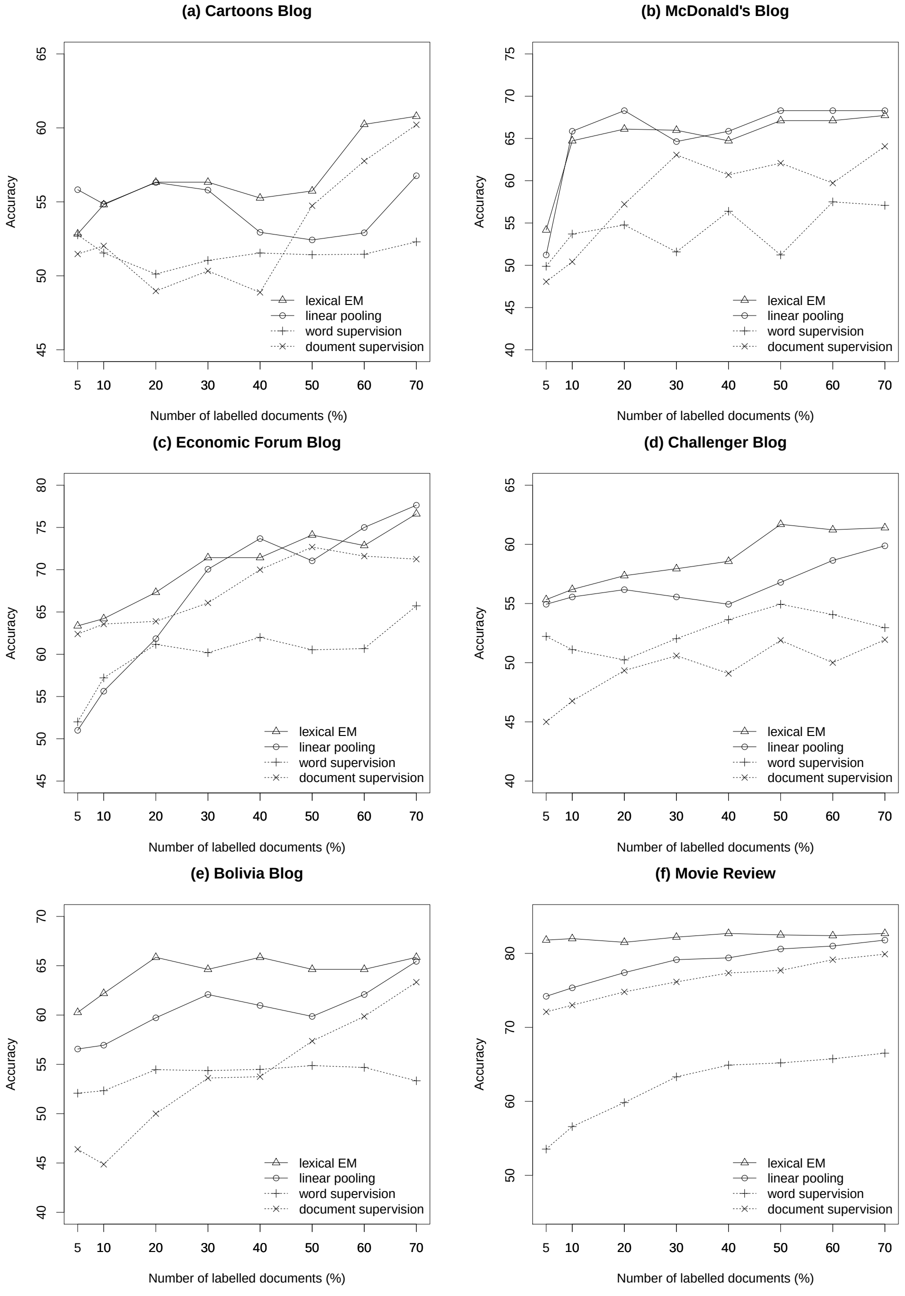

Fig. 1. Learning curves for all models

documents. We also observe that Word Supervision very often has the lowest accuracy compared with other models. This result is different from that in [10] where Word Supervision is reported to outperform Document Supervision. Note that however, there are several main differences in our study: First we are using a generic labelled lexicon whereas manually compiled lexicons are used in [10]. Secondly our lexicon has thousands of general labelled words whereas in [10] there are only tens of manually selected words.

We also examined performance of the Lexical Knowledge model described in Section 4. Not surprisingly performance of the Lexical Knowledge model varies considerably on different datasets, from an accuracy of 56.26% on the Cartoon blog to an accuracy of 66.96% on the Economic forum blog. In contrast the Lexical Knowledge model performs surprisingly well on the movie review dataset, with an accuracy of 71.55%. The reason for this modest performance on blogs may be due to the different language style for blogs. The language used in blogs is typically very informal and the vocabulary is far different from the vocabulary for formal writing. As a result our opinion lexicon has poor representation for blogs.

6.2 Analysis of Lexical EM

The only parameter in Lexical EM is the weight α in Equation 3 when the knowledge-based estimation for class distribution is combined with the generative model. Figure 2 plots the accuracies of Lexical EM with different weighting functions for α. We compare our default accuracy-based weighting function against two other weighting functions. "Equal Weight" denotes that $\alpha = 0.5$. "Percentage Weight" denotes that the percentage of labelled training documents is used as the weight for the generative model. The rationale for this strategy is that with an increasing number of labelled documents, the generative model becomes more accurate. The figure shows that Lexical EM is fairly robust with respect to different settings of weight. All three weighting functions lead to the same trend of learning curve – the more available training instances the more accurate is Lexical EM. The Percentage weighting function almost always gives the same accuracy as the accuracy-based weighting function.

7 Conclusions

In this paper we have proposed a simple and effective approach to incorporating the knowledge of word labels into the EM process for document sentiment classification. Experiments show that combining limited domain-specific labelled training documents with general lexical knowledge can achieve significantly better performance than the model derived from only labelled documents. More generally our study strongly suggests that the marriage of limited domain-specific information and domain-independent knowledge is a cost-effective approach to sentiment classification in new domains. In future work we will focus on developing more advanced lexical knowledge models to improve the EM process for document sentiment analysis.

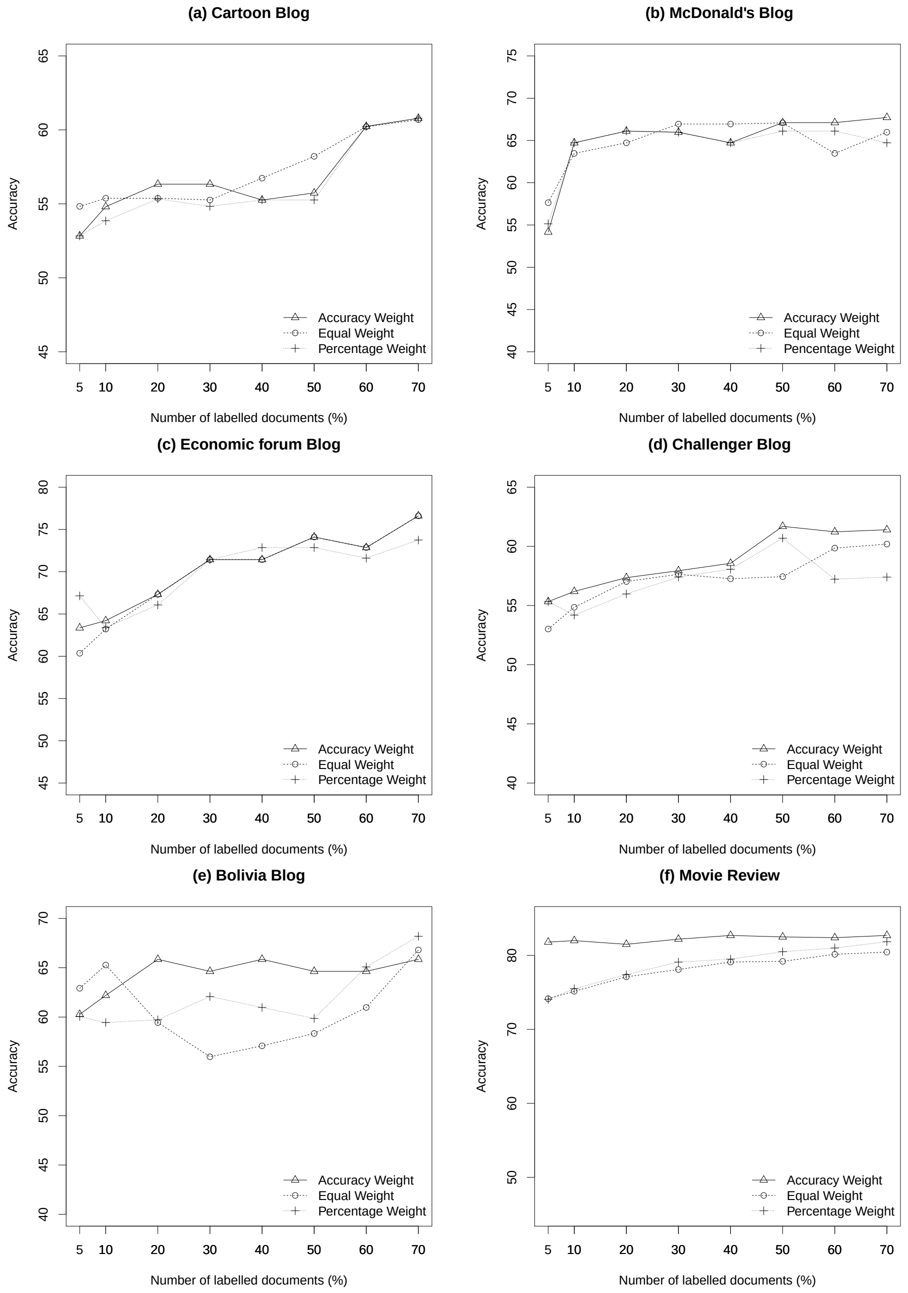

Fig. 2. Performance of Lexical EM with regard to weighting functions

References

1. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: LREC'08 (2008)
2. Cozman, F., Cohen, I.: Risks of semi-supervised learning: How unlabeled data can degrade performance of generative classifiers. Semi-Supervised Learning. Vol 4. pp. 57–72 (2006)
3. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977)
4. Druck, G., Mann, G., McCallum, A.: Learning from labeled features using generalized expectation criteria. In: SIGIR'08 (2008)
5. Druck, G., McCallum, A.: High-performance semi-supervised learning using discriminatively constrained generative models. In: ICML'10, Haifa, Israel (2010)
6. General-Inquirer: The General Inquirer Home Page. `http://www.wjh.harvard.edu/~inquirer/` (2010), [Online; accessed 22-September-2010]
7. Hatzivassiloglou, V., McKeown, K.R.: Predicting the semantic orientation of adjectives. In: LREC'97 (1997)
8. J. Graca, K.G., Taskar, B.: Expectation maximization and posterior constraints. In: Proceedings of NIPS (2007)
9. Liu, B.: Sentiment analysis and subjectivity. Handbook of Natural Language Processing, 2nd edition (2009)
10. Liu, B., Li, X., Lee, W., Yu, P.: Text classification by labelling words. In: AAAI'04 (2004)
11. Macdonald, C., Ounis, I., Soboroff, I.: Overview of the TREC 2007 blog track. In: TREC'07 (2007)
12. Melville, P., Gryc, W., Lawrence, R.D.: Sentiment analysis of blogs by combining lexical knowledge with text classification. In: ACM SIGKDD'09 (2009)
13. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Machine Learning 39(2), 103–134 (2000)
14. Ounis, I., Rijke, M.D., Macdonald, C., Mishne, G., Soboroff, I.: Overview of the TREC-2006 blog track. In: TREC'06 (2006)
15. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2008)
16. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: ACL'02. pp. 79–86 (2002)
17. Riloff, E., Wiebe, J.: Learning extraction patterns for subjective expressions. In: EMNLP'03. pp. 105–112 (2003)
18. Sindhwani, V., Melville, P.: Document-word co-regularization for semi-supervised sentiment analysis. In: ICDM'08. pp. 1025–1030 (2008)
19. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: ACL'02. pp. 417–424 (2002)
20. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. Language Resources and Evaluation 39(2), 165–210 (2005)
21. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: EMNLP'05. pp. 347–354 (2005)