# Emoticon Smoothed Language Models for Twitter Sentiment Analysis

**Kun-Lin Liu, Wu-Jun Li, Minyi Guo**
Shanghai Key Laboratory of Scalable Computing and Systems
Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
liukunlin@sjtu.edu.cn, {liwujun,guo-my}@cs.sjtu.edu.cn

## Abstract

Twitter sentiment analysis (TSA) has become a hot research topic in recent years. The goal of this task is to discover the attitude or opinion of the tweets, which is typically formulated as a machine learning based text classification problem. Some methods use manually labeled data to train fully supervised models, while others use some noisy labels, such as emoticons and hashtags, for model training. In general, we can only get a limited number of training data for the fully supervised models because it is very labor-intensive and time-consuming to manually label the tweets. As for the models with noisy labels, it is hard for them to achieve satisfactory performance due to the noise in the labels although it is easy to get a large amount of data for training. Hence, the best strategy is to utilize both manually labeled data and noisy labeled data for training. However, how to seamlessly integrate these two different kinds of data into the same learning framework is still a challenge. In this paper, we present a novel model, called emoticon smoothed language model (ESLAM), to handle this challenge. The basic idea is to train a language model based on the manually labeled data, and then use the noisy emoticon data for smoothing. Experiments on real data sets demonstrate that ESLAM can effectively integrate both kinds of data to outperform those methods using only one of them.

## Introduction

Sentiment analysis (SA) (Pang and Lee 2007) (also known as opinion mining) is mainly about discovering "what others think" from data such as product reviews and news articles. On one hand, consumers can seek advices about a product to make informed decisions in the consuming process. On the other hand, vendors are paying more and more attention to online opinions about their products and services. Hence, SA has attracted increasing attention from many research communities such as machine learning, data mining, and natural language processing. The sentiment of a document or sentence can be positive, negative or neutral. Hence, SA is actually a three-way classification problem. In practice, most methods adopt a two-step strategy for SA (Pang and Lee 2007). In the *subjectivity classification* step, the target is classified to be subjective or neutral (objective), and in the *polarity classification* step, the subjective targets are further classified as positive or negative. Hence, two classifiers are trained for the whole SA process, one is called subjectivity classifier, and the other is called polarity classifier. Since (Pang, Lee, and Vaithyanathan 2002) formulated SA as a machine learning based text classification problem, more and more machine learning methods have been proposed for SA (Pang and Lee 2007).

Twitter is a popular online micro-blogging service launched in 2006. Users on Twitter write tweets up to 140 characters to tell others about what they are doing and thinking. According to the some sources [1], until 2011, there have been over 300 million users on Twitter and 300 million new tweets are generated every day. Because almost all tweets are public, these rich data offer new opportunities for doing research on data mining and natural language processing(Liu et al. 2011a; 2011b; 2011c; Jiang et al. 2011).

One way to perform Twitter sentiment analysis (TSA) is to directly exploit traditional SA methods (Pang and Lee 2007). However, tweets are quite different from other text forms like product reviews and news articles. Firstly, tweets are often short and ambiguous because of the limitation of characters. Secondly, there're more misspelled words, slang, modal particles and acronyms on Twitter because of its casual form. Thirdly, a huge amount of unlabeled or noisy labeled data can be easily downloaded through Twitter API. Therefore, many novel SA methods have been specially developed for TSA. These methods can be mainly divided into two categories: fully supervised methods and distantly supervised methods[2].

The fully supervised methods try to learn the classifiers from manually labeled data. (Jansen et al. 2009) uses the multinomial Bayes model to perform automatic TSA. (Bermingham and Smeaton 2010) compares support vector machine (SVM) and multinomial naive Bayes (MNB) for both blog and microblog SA, and finds that SVM outperforms MNB on blogs with long text but MNB outperforms SVM on microblogs with short text. One problem with the fully supervised methods is that it is very labor-intensive and time-consuming to manually label the data and hence the training data sets for most methods are often too small to



[1] http://en.wikipedia.org/wiki/Twitter

[2] We use the terminology 'distant' as that from (Go, Bhayani, and Huang 2009).

guarantee a good performance.

More recent works have focused on distantly supervised methods which learn the classifiers from data with noisy labels such as emoticons and hashtags. The distant supervision method (Go, Bhayani, and Huang 2009) uses the emoticons like ":)" and ":(" as noisy labels for polarity classification. The basic assumption is that a tweet containing ":)" is most likely to have a positive emotion and that containing ":(" is assumed to be negative. Experiments show that these emoticons do contain some discriminative information for SA. Hashtags (e.g., #sucks) or Smileys are used in (Davidov, Tsur, and Rappoport 2010) to identify sentiment types. (Barbosa and Feng 2010) uses the noisy data collected from some Twitter sentiment detection web sites, such as the Twitter Sentiment[3]. (Kouloumpis, Wilson, and Moore 2011) investigates both hashtags and emoticons and finds that combining both of them can get better performance than using only hashtags. The advantage of these distantly supervised methods is that the labor-intensive manual annotation can be avoided and a large amount of training data can be easily built, either from Twitter API or existing web sites. However, due to the noise in the labels, the accuracy of these methods is not satisfactory.

Considering the shortcomings of the fully supervised and distantly supervised methods, we argue that the best strategy is to utilize both manually labeled data and noisy labeled data for training. However, how to seamlessly integrate these two different kinds of data into the same learning framework is still a challenge. In this paper, we propose a novel model, called emoticon smoothed language model (ESLAM), to handle this challenge. The main contributions of ESLAM are outlined as follows:

- ESLAM uses the noisy emoticon data to smooth the language model trained from manually labeled data. Hence, ESLAM seamlessly integrate both manually labeled data and noisy labeled data into a probabilistic framework. The large amount of noisy emoticon data gives ESLAM have the power to deal with misspelled words, slang, modal particles, acronyms, and the unforseen test words, which cannot be easily handled by fully supervised methods.
- Besides the polarity classification, ESLAM can also be used for subjectivity classification which cannot be handled by most existing distantly supervised methods.
- Rather than crawling a large amount of noisy data to local disks which is a typical choice by existing distantly supervised methods, we propose an efficient and convenient way to directly estimate the word probabilities from Twitter API without downloading any tweet. This is very promising because it is very expensive in terms of time and storage to download and process large amount of tweets.
- Experiments on real data sets demonstrate that ESLAM can effectively integrate both manually labeled data and noisy labeled data to outperform those methods using only one of them.

[3]http://twittersentiment.appspot.com/

## Related Work

SA (Pang and Lee 2007) has a long history in natural language processing. Before (Pang, Lee, and Vaithyanathan 2002), almost all methods are partially knowledge-based. (Pang, Lee, and Vaithyanathan 2002) shows that machine learning techniques, such as naive Bayes, maximum entropy classifiers, and SVM can outperform the knowledge-based baselines on movie reviews. After that, the machine learning based methods have become the mainstream for SA.

Earlier works on TSA follow the methods of traditional SA on normal text forms like movie reviews. These methods are mainly fully supervised (Jansen et al. 2009; Bermingham and Smeaton 2010) which have been introduced in the Introduction section. Most recent works include target-dependent SA based on SVM (Jiang et al. 2011), user-level SA based on social networks (Tan et al. 2011), sentiment stream analysis based on association rules (Silva et al. 2011), and real-time SA (Guerra et al. 2011).

Recently, more and more distantly supervised methods are proposed. (Go, Bhayani, and Huang 2009)'s training data consist of tweets with emoticons like ":)" and ":(" and they use these emoticons as noisy labels. (Davidov, Tsur, and Rappoport 2010) uses 50 Twitter tags and 15 smileys as noisy labels to identify and classify diverse sentiment types of tweets. Other methods with noisy labels (Barbosa and Feng 2010; Kouloumpis, Wilson, and Moore 2011) are also proposed. All these methods cannot handle subjectivity classification well. Furthermore, these methods need to crawl all the data and store them in the local disks. This is very inefficient when millions or even billions of tweets are used because request rate for crawling tweets is limited by Twitter server.

Although a lot of TSA methods have been proposed, few of them can effectively integrate both manually labeled data and noisy labeled data into the same framework, which motivates our ESLAM work in this paper.

## Our Approach

In this section, first we present how to adapt language models (Manning, Raghavan, and Schutze 2009) for SA. Then we propose a very effective and efficient way to learn the emoticon model from Twitter API. Finally, we will introduce the strategy to seamlessly integrate both manually labeled data and emoticon data into a probabilistic framework which is our ESLAM method.

### Language Models for SA

Language models (LM) can be either probabilistic or non-probabilistic. In this paper, we refer to probabilistic language models which are widely used in information retrieval and natural language processing (Ponte and Croft 1998; Zhai and Lafferty 2004; Manning, Raghavan, and Schutze 2009). A LM assign a probability to a sequence of words. In information retrieval, first we estimate a LM for each document, then we can compute a likelihood measuring how likely a query is generated by each document LM and rank the documents with respect to the likelihoods.

TSA is actually a classification problem. To adapt LM for TSA, we concatenate all the tweets from the same class to form one synthetic document. Hence, for the polarity classification problem, one document is constructed from positive training tweets, and the other document is constructed from negative training tweets. Then we learn two LMs, one for positive class and the other for negative class. The LM learning procedure for subjectivity classification is similar. During the test phase, we treat each test tweet as a query, and then we can use the likelihoods to rank the classes. The class with the highest likelihood will be chosen as the label of the test tweet.

We use $c_1$ and $c_2$ to denote the two language models. In polarity classification, $c_1$ is the language model for positive tweets and $c_2$ is for negative tweets. In subjectivity classification, $c_1$ is for subjective class and $c_2$ is for objective (neutral) class. In order to classify a tweet $t$ to $c_1$ or $c_2$, we need to estimate the tweet likelihoods computed by $P(t|c_1)$ and $P(t|c_2)$. By using the common unigram assumption, we get:

$$P(t|c) = \prod_{i=1}^{n} P(w_i|c),$$

where $n$ is the number of words in tweet $t$ and $P(w_i|c)$ is a multinomial distribution estimated from the LM of class $c$. This probability simulates the generative process of the test tweet. Firstly, the first word ($w_1$) is generated by following a multinomial distribution $P(w_i|c)$. After that, the second word is generated independently of the previous word by following the same distribution. This process continues until all the words in this tweet have been generated.

One commonly used method to estimate the distributions is maximum likelihood estimate (MLE), which computes the probability as follows:

$$P_a(w_i|c) = \frac{N_{i,c}}{N_c},$$

where $N_{i,c}$ is the number of times word $w_i$ appearing in training data of class $c$ and $N_c$ is the total number of words in training data of class $c$.

In general, the vocabulary is determined by the training set. To classify tweets in test set, it is very common to encounter words that do not appear in training set especially when there are not enough training data or the words are not well-formed. In these cases, smoothing (Zhai and Lafferty 2004) plays a very important role in language models because it can avoid assigning zero probability to unseen words. Furthermore, smoothing can make the model more accurate and robust. Representative smoothing methods include Dirichlet smoothing and Jelinek-Mercer (JM) smoothing (Zhai and Lafferty 2004). Although the original JM smoothing method is used to linear interpolation of the MLE model with the collection model (Zhai and Lafferty 2004), we use JM smoothing method to linearly interpolate the MLE model with the emoticon model in this paper.

## Emoticon Model

From the emoticon data, we can also build the LMs for different classes. We propose a very effective and efficient way to estimate the emoticon LM $P_u(w_i|c)$ from Twitter Search API. Twitter Search API [4] is a dedicated API for running searches against the real-time index of recent tweets. Its index includes tweets between 6-9 days. Given a query which consists of one or several words, the API returns up to 1500 relevant tweets and their posting time.

**Polarity Classification** To get $P_u(w_i|c_1)$, the probability of $w_i$ in positive class, we make an assumption that all tweets containing ":)" are positive. We build a query "$w_i$ :)" and input it to the Search API. Then it returns tweets containing both $w_i$ and ":)" with their posting time. After summarization, we get the number of tweets $nw_i$ and the time range of these tweets $tw_i$. Then we build another query ":)" and get the number of returned tweets $ns$ and the time range $ts$. Some estimations [5] show that a tweet contains 15 words on average.

Assume that the tweets on Twitter are uniformly distributed with respect to time. Similar to the rule of getting $P_a(w_i|c)$, we can estimate $P_u(w_i|c_1)$ with the following rule:

$$P_u(w_i|c_1) = \frac{\frac{nw_i}{tw_i}}{\frac{ns}{ts} \times 15} = \frac{nw_i \times ts}{15 \times tw_i \times ns}.$$

The term $\frac{nw_i}{tw_i}$ is roughly the number of times word $w_i$ appearing in class $c$ per unit time, and the term $\frac{ns}{ts} \times 15$ is roughly the total number of words in class $c$ per unit time.

Let $F_u = \sum_{j=1}^{|V|} P_u(w_j|c)$ be the normalization factor where $|V|$ is the size of vocabulary containing both seen and unseen words. Then each estimated $P_u(w_i|c)$ should be normalized to make them sum up to one:

$$P_u(w_i|c) := P_u(w_i|c)/F_u = \frac{P_u(w_i|c)}{\sum_{j=1}^{|V|} P_u(w_j|c)}$$
$$= \frac{\frac{nw_i \times ts}{15 \times tw_i \times ns}}{\sum_{j=1}^{|V|} \frac{nw_j \times ts}{15 \times tw_j \times ns}} = \frac{\frac{nw_i}{tw_i}}{\sum_{j=1}^{|V|} \frac{nw_j}{tw_j}}.$$

We can find that there is no need to get $ts$ and $ns$, because $P_u(w_i|c)$ can be determined only by $nw_i$ and $tw_i$.

For the LM of negative class, we assume that the negative tweets are those containing ":(". The estimate procedure for $P_u(w_i|c_2)$ is similar to that for $P_u(w_i|c_1)$. The only difference is that the query should be changed to "$w_i$ :(".

**Subjectivity Classification** For subjectivity classification, the two classes are subjective and objective. The assumption for subjective tweets is that tweets with ":)" or ":(" are assumed to carry subjectivity of the users. So we build the query "$w_i$:) OR :(" for the subjective class.

As for the objective LM, getting $P_u(w_i|c_2)$, the probability of $w_i$ in objective class, is much more challenging than that in subjective class. To the best of our knowledge, no general assumption for objective tweets has been reported by researchers. We tried the strategy which treats tweets without emoticons as objective but the experiments showed that

[4] https://dev.twitter.com/docs/using-search

[5] http://blog.oup.com/2009/06/oxford-twitter/

the results were not satisfactory, which implies that this assumption is unreasonable. (Kouloumpis, Wilson, and Moore 2011) tries to use some hashtags like "#jobs" as indicators for objective tweets. However, this assumption is not general enough because the number of tweets containing specific hashtags is limited and these tweets' sentiment may be biased to certain topics like "jobs".

Here we present a novel assumption for objective tweets that tweets containing an *objective url link* is assumed to be objective. Based on our observation, we find that urls linking to the picture sites (e.g., twitpic.com) or video sites (e.g., youtube.com) are often subjective and other urls like those linking to news articles are usually objective. Hence, if a url link doesn't represent pictures or videos, we call it an *objective url link*. Based on the above assumption, we build the query "$w_i filter : links$" [6] to get the statistics about the objective class.

## ESLAM

After we have estimated the $P_a(w_i|c)$ from manually labeled data and $P_u(w_i|c)$ from the noisy emoticon data, we can integrate them into the same probabilistic framework $P_{co}(w_i|c)$. Before combining $P_a(w_i|c)$ and $P_u(w_i|c)$, there's another important step: smoothing $P_u(w_i|c)$. Because $P_u(w_i|c)$ is estimated from noisy emoticon data, it can be biased. We adopt Dirichlet smoothing (Zhai and Lafferty 2004) to smooth $P_u(w_i|c)$.

By following the JM smoothing principle (Zhai and Lafferty 2004), our ESLAM model $P_{co}(w_i|c)$ can be computed as follows:

$$P_{co}(w_i|c) = \alpha P_a(w_i|c) + (1-\alpha)P_u(w_i|c), \qquad (1)$$

where $\alpha \in [0,1]$ is the combination parameter controlling the contribution of each component.

# Experiments

## Data Set

The publicly available Sanders Corpus[7] is used for evaluation. It consists of 5513 manually labeled tweets. These tweets were collected with respect to one of the four different topics (Apple, Google, Microsoft, and Twitter). After removing the non-English and spam tweets, we have 3727 tweets left. The detailed information of the corpus is shown in Table 1. As for the noisy emoticon data, theoretically we use all the data existing in Twitter by sampling with its API.

Table 1: Corpus Statistics

| Corpus | # Positive | # Negative | # Neutral | # Total |
|---|---|---|---|---|
| Sanders | 570 | 654 | 2503 | 3727 |

We adopt the following strategies to preprocess the data:

- Username. Twitter usernames which start with @ are replaced with "twitterusername".
- Digits. All Digits in tweets are replaced with "twitterdigit".
- Links. All urls in tweets are replaced with "twitterurl".
- Stopwords. Stopwords like "the" and "to" are removed.
- Lower case and Stemming. All words are changed to their lower cases and stemmed to terms.
- Retweets and Duplicates. Retweets and duplicate tweets are removed to avoid giving extra weight to these tweets in training data.

## Evaluation Scheme and Metrics

After removing the retweets or duplicates and setting the classes to be balanced, we randomly choose 956 tweets for polarity classification, including 478 positive tweets and 478 negative ones. For the subjectivity classification, we also set the classes to be balanced and randomly choose 1948 tweets for evaluation, including 974 subjective tweets and 974 objective (neutral) ones.

The evaluation schemes for both polarity and subjectivity classification are similar. Assume the total number of manually labeled tweets, including both training and test data, is $X$. Each time we randomly sample the same amount of tweets (say $Y$) for both classes (e.g., positive and negative) for training, and use the rest $X - 2Y$ tweets for test. This random selection and testing is carried out 10 rounds independently for each unique training set size, and the average performance is reported. We perform experiments with different sizes of training set, i.e., $Y$ is set to different values, such as 32, 64, and 128.

As in (Go, Bhayani, and Huang 2009) and (Kouloumpis, Wilson, and Moore 2011), we adopt accuracy and F-score as our evaluation metrics. Accuracy is a measure of what percentage of test data are correctly predicted, and F-score is computed by combining precision and recall.

## Effect of Emoticons

We compare our ESLAM method to the fully supervised language model (LM) to verify whether the smoothing with emoticons is useful or not. Please note that the fully supervised LM uses only the manually labeled data for training while ESLAM integrates both manually labeled data and the emoticon data for training. Figure 1 and Figure 2 respectively illustrate the accuracy and F-score of the two methods with different number of manually labeled training data, i.e., $2Y = 32, 64, 128, 256, 512, 768$.

From Figure 1 and Figure 2, we can see that as the number of manually labeled data increases, the performance of both methods will also increase, which is reasonable because the manually labeled data contain strong discriminative information. Under all the evaluation settings, ESLAM consistently outperforms the fully supervised LM, in particular for the settings with small number of manually labeled data. This implies that the noisy emoticon data do have some useful information and our ESLAM can effectively exploit it to achieve good performance.

Figure 3 and Figure 4 demonstrate the accuracy and F-score of the two methods on subjectivity classification with

[6] filter:links means returning tweets containing urls.

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

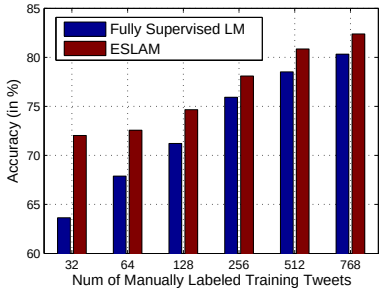


Figure 1: Effect of emoticons on accuracy of polarity classification.

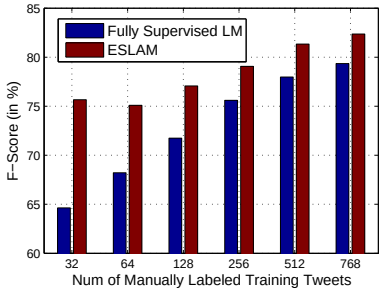


Figure 2: Effect of emoticons on F-score of polarity classification.

different number of manually labeled training data, respectively. The results are similar to those for polarity classification which once again verifies the effectiveness of our ESLAM to utilize the noisy emoticon data. The good performance of ESLAM also verifies that our url link based method is effective to find objective tweets, which is a big challenge for most existing distantly supervised methods.

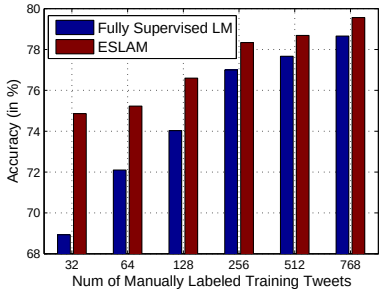


Figure 3: Effect of emoticons on accuracy of subjectivity classification.

## Effect of Manually Labeled Data

We compare our ESLAM method to the distantly supervised LM to verify whether the manually labeled data can provide extra useful information for classification. Please note that the distantly supervised LM uses only the noisy emoticon data for training, while ESLAM integrates both manually labeled data and the emoticon data for training.

Figure 5 and Figure 6 illustrate the accuracy and F-score of the two methods on polarity classification with different number of manually labeled training data, respectively. The

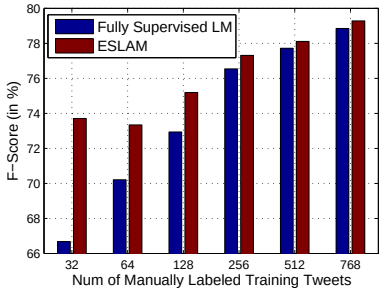


Figure 4: Effect of emoticons on F-score of subjectivity classification.

blue line corresponds to the performance of distantly supervised LM, which also corresponds to the case of zero manually labeled data. The red line is the results of ESLAM. We can find that ESLAM achieves better performance than the distantly supervised LM. With the increase of manually labeled data, the performance gap between them will become larger and larger. This verifies our claim that it is not enough to use only the data of noisy labels for training.

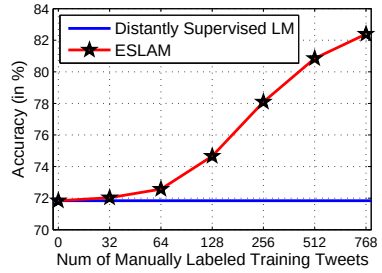


Figure 5: Effect of manually labeled data on accuracy of polarity classification.

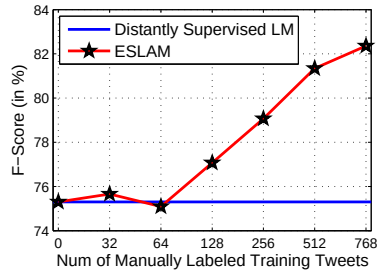


Figure 6: Effect of manually labeled data on F-score of polarity classification.

Figure 7 and Figure 8 illustrate the accuracy and F-score of the two methods on subjectivity classification with different number of manually labeled training data, respectively. The results are similar to those for polarity classification.

## Sensitivity to Parameters

The parameter $\alpha$ in (1) plays a critical role to control the contribution between the manually labeled information and noisy labeled information. To show the effect of this parameter in detail, we try different values for polarity classifica-

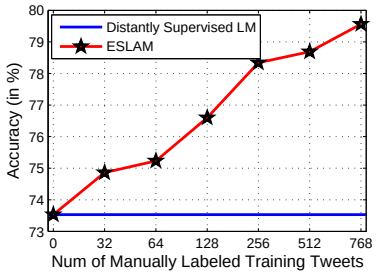


Figure 7: Effect of manually labeled data on accuracy of subjectivity classification.

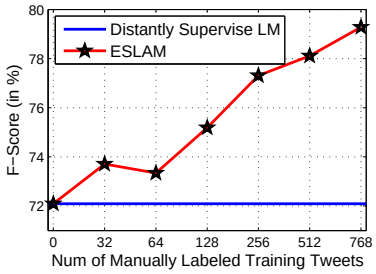


Figure 8: Effect of manually labeled data on F-score of subjectivity classification.

tion. Figure 9 and Figure 10 show the accuracy of ESLAM with 128 and 512 labeled training tweets, respectively.

The case $\alpha = 0$ means only noisy emoticon data are used and $\alpha = 1$ is the fully supervised case. The results in the Figures clearly show that the best strategy is to integrate both manually labeled data and noisy data into training. We also notice that with 512 labeled training data ESLAM achieves its best performance with relatively bigger $\alpha$ than the case of 128 labeled data, which is obviously reasonable. Furthermore, we find that ESLAM is not sensitive to the small variations in the value of parameter $\alpha$ because the range for $\alpha$ to achieve the better performance is large.

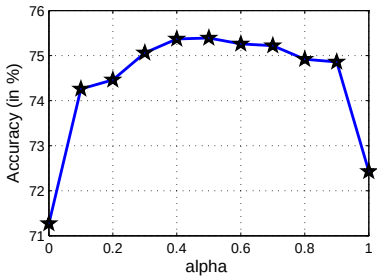


Figure 9: Effect of the smoothing parameter $\alpha$ with 128 labeled training tweets.

## Conclusion

Existing methods use either manually labeled data or noisy labeled data for Twitter sentiment analysis, but few of them utilize both of them for training. In this paper, we propose a novel model, called emoticon smoothed language model (ESLAM), to seamlessly integrate these two kinds of data into the same probabilistic framework. Experiments on real data sets show that our ESLAM method can effectively integrate both kinds of data to outperform those methods using only one of them.

Our ESLAM method is general enough to integrate other kinds of noisy labels for model training, which will be pursued in our future work.

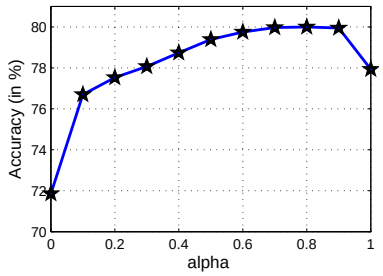


Figure 10: Effect of the smoothing parameter $\alpha$ with 512 labeled training tweets.

## Acknowledgments

This work is supported by the NSFC (No. 61100125) and the 863 Program of China (No. 2011AA01A202, No. 2012AA011003).

## References

Barbosa, L., and Feng, J. 2010. Robust sentiment detection on twitter from biased and noisy data. In *COLING*, 36–44.

Bermingham, A., and Smeaton, A. F. 2010. Classifying sentiment in microblogs: is brevity an advantage? In *CIKM*, 1833–1836.

Davidov, D.; Tsur, O.; and Rappoport, A. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In *COLING*, 241–249.

Go, A.; Bhayani, R.; and Huang, L. 2009. Twitter sentiment classification using distant supervision. *Technical report*.

Guerra, P. H. C.; Veloso, A.; Jr., W. M.; and Almeida, V. 2011. From bias to opinion: a transfer-learning approach to real-time sentiment analysis. In *KDD*, 150–158.

Jansen, B. J.; Zhang, M.; Sobel, K.; and Chowdury, A. 2009. Twitter power: Tweets as electronic word of mouth. *JASIST* 60(11):2169–2188.

Jiang, L.; Yu, M.; Zhou, M.; Liu, X.; and Zhao, T. 2011. Target-dependent twitter sentiment classification. In *ACL*, 151–160.

Kouloumpis, E.; Wilson, T.; and Moore, J. 2011. Twitter sentiment analysis: The good the bad and the omg! In *ICWSM*, 538–541.

Liu, X.; Li, K.; Zhou, M.; and Xiong, Z. 2011a. Collective semantic role labeling for tweets with clustering. In *IJCAI*, 1832–1837.

Liu, X.; Li, K.; Zhou, M.; and Xiong, Z. 2011b. Enhancing semantic role labeling for tweets using self-training. In *AAAI*.

Liu, X.; Zhang, S.; Wei, F.; and Zhou, M. 2011c. Recognizing named entities in tweets. In *ACL*, 359–367.

Manning, C. D.; Raghavan, P.; and Schutze, H. 2009. *An Introduction to Information Retrieval*. Cambridge University Press.

Pang, B., and Lee, L. 2007. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2(1-2):1–135.

Pang, B.; Lee, L.; and Vaithyanathan, S. 2002. Thumbs up? sentiment classification using machine learning techniques. In *EMNLP*, 79–86.

Ponte, J. M., and Croft, W. B. 1998. A language modeling approach to information retrieval. In *SIGIR*, 275–281.

Silva, I. S.; Gomide, J.; Veloso, A.; Jr., W. M.; and Ferreira, R. 2011. Effective sentiment stream analysis with self-augmenting training and demand-driven projection. In *SIGIR*, 475–484.

Tan, C.; Lee, L.; Tang, J.; Jiang, L.; Zhou, M.; and Li, P. 2011. User-level sentiment analysis incorporating social networks. In *KDD*, 1397–1405.

Zhai, C., and Lafferty, J. D. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.* 22(2):179–214.