Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis

Lei Zhang, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, Bing Liu

HP Laboratories
HPL-2011-89

Abstract:
With the booming of microblogs on the Web, people have begun to express their opinions on a wide variety of topics on Twitter and other similar services. Sentiment analysis on entities (e.g., products, organizations, people, etc.) in tweets (posts on Twitter) thus becomes a rapid and effective way of gauging public opinion for business marketing or social studies. However, Twitter's unique characteristics give rise to new problems for current sentiment analysis methods, which originally focused on large opinionated corpora such as product reviews. In this paper, we propose a new entity-level sentiment analysis method for Twitter. The method first adopts a lexiconbased approach to perform entity-level sentiment analysis. This method can give high precision, but low recall. To improve recall, additional tweets that are likely to be opinionated are identified automatically by exploiting the information in the result of the lexicon-based method. A classifier is then trained to assign polarities to the entities in the newly identified tweets. Instead of being labeled manually, the training examples are given by the lexicon-based approach. Experimental results show that the proposed method dramatically improves the recall and the F-score, and outperforms the state-of-the-art baselines.

External Posting Date: June 21, 2011 [Fulltext] Approved for External Publication
Internal Posting Date: June 21, 2011 [Fulltext]

Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis

Lei Zhang[1,2], Riddhiman Ghosh[1], Mohamed Dekhil[1], Meichun Hsu[1], Bing Liu[2]

[1]Hewlett-Packard Laboratories
1501 Page Mill Rd., Palo Alto, CA
{riddhiman.ghosh, mohamed.dekhil, meichun.hsu}@hp.com

[2]University of Illinois at Chicago
851 S. Morgan St., Chicago, IL
{lzhang3, liub}@cs.uic.edu

Abstract

With the booming of microblogs on the Web, people have begun to express their opinions on a wide variety of topics on Twitter and other similar services. Sentiment analysis on entities (e.g., products, organizations, people, etc.) in tweets (posts on Twitter) thus becomes a rapid and effective way of gauging public opinion for business marketing or social studies. However, Twitter's unique characteristics give rise to new problems for current sentiment analysis methods, which originally focused on large opinionated corpora such as product reviews. In this paper, we propose a new entity-level sentiment analysis method for Twitter. The method first adopts a lexicon-based approach to perform entity-level sentiment analysis. This method can give high precision, but low recall. To improve recall, additional tweets that are likely to be opinionated are identified automatically by exploiting the information in the result of the lexicon-based method. A classifier is then trained to assign polarities to the entities in the newly identified tweets. Instead of being labeled manually, the training examples are given by the lexicon-based approach. Experimental results show that the proposed method dramatically improves the recall and the F-score, and outperforms the state-of-the-art baselines.

1 Introduction

As a microblogging and social networking website, Twitter has become very popular and has grown rapidly. An increasing number of people are willing to post their opinions on Twitter, which is now considered a valuable online source for opinions. As a result, sentiment analysis on Twitter is a rapid and effective way of gauging public opinion for business marketing or social studies. For example, a business can retrieve timely feedback on a new product in the market by evaluating people's opinions on Twitter. As people often talk about various entities (e.g., products, organizations, people, etc.) in a tweet, we perform sentiment analysis at the entity level; that is, we mine people's opinions on specific entities in each tweet rather than the opinion about each whole sentence or whole tweet. We assume that the entities are provided by the user, e.g., he/she is interested in opinions on iPhone (an entity).

One approach to perform sentiment analysis is based on a function of opinion words in context. Opinion words are words that are commonly used to express positive or negative sentiments, e.g., "good" and "bad". The approach generally uses a dictionary of opinion words to identify and determine sentiment orientation (positive, negative or neutral). The dictionary is called the *opinion lexicon*. The approach of using opinion words (the lexicon) to determine opinion orientations is called the *lexicon-based approach* to sentiment analysis (Ding et al, 2008; Taboada, et al., 2010). This approach is efficient and can be employed to analyze text at the document, sentence or entity level. It is thus applicable to our task as well. However, Twitter data has developed its own characteristics. Some of them are detrimental to the lexicon-based approach. For example, emoticons, colloquial expressions, abbreviations, etc. are frequently used in tweets. These expressions may possess semantic/sentiment orientation but they do not exist in a general opinion lexicon. Let us see a tweet example, "I bought iPad yesterday, just lovvee it :-)". It clearly expresses a positive opinion on iPad by the word "lovvee" and the emoticon ":-)". But the lexicon-based method would regard the tweet as expressing no/neutral opinion on iPad, since there is not a general opinion word in the tweet. This leads to the low recall problem for the lexicon-based method, which depends entirely on the presence of opinion words to determine the sentiment orientation. Although one may say that these additional expressions can be added to the opinion lexicon, such expressions change constantly and new ones are also appearing all the time following the trends and fashions on the Internet. Moreover, their polarities can be domain dependent. These problems make it hard to manually add them to the opinion lexicon. Without a comprehensive lexicon, the sentiment analysis results will suffer.

Alternatively, we can apply a machine learning-based method to perform sentiment analysis (Pang et al., 2002). That is, we train a sentiment classifier to determine positive, negative and neutral sentiments. The method has been frequently used for sentiment classification of documents or sentences. However, it is not easy to apply in our case because manual labeling of a large set of tweet examples is labor-intensive and time-consuming. Moreover, manual labeling needs to be done for each application domain, as it is well-known that a sentiment classifier may perform very well in the domain that it is trained, but performs poorly when it is applied to a different domain (Aue and Gamon, 2005). The learning-based method is thus not very scalable for Twitter sentiment analysis which covers almost all domains as people can express opinions about anything on Twitter.

In this paper, we explore an entity-level sentiment analysis approach to the Twitter data. We first employ an augmented lexicon-based method for entity-level sentiment analysis. Although this method gives good precision, the recall can be quite low. To improve the recall, we do the following: We first extract some additional opinionated indicators (e.g. words and tokens) through the Chi-square test on the results of the lexicon-based method. With the help of the new opinionated indicators, additional opinionated tweets can be identified. Afterwards, a sentiment classifier is trained to assign sentiment polarities for entities in the newly-identified tweets. The training data for the classifier is the result of the lexicon-based method. Thus, the whole process has no manual labeling. The proposed approach is an *unsupervised method* except for the initial opinion lexicon, which is publicly available. The reason that our technique works is that sentiment expressions (including domain-specific opinion words, emoticons, colloquial expressions, abbreviations, etc.) depend on the sentiment context. For example, let us see a tweet with a positive opinion, “The movie is so amazing. Harry potter is so cuteee !!!”. Although the expression “cuteee” is not a general opinion word, if we find it often co-occurs in positive opinion contexts through a statistical test, we can infer it is a positive opinion indicator. And the sentiment classifier could learn this piece of valuable information in training. The statistical test and training need a huge amount of data, which is not a problem for tweets because people produce millions of tweets every day.

Our proposed method seems to be similar to several existing techniques, e.g., using a lexicon to bootstrap learning and transfer learning. However, as we will discuss in the next section, it is entirely different from them due to some subtle but crucial differences. We believe that our method is more desirable for practical applications due to its nature of no manual involvement and its ability to automatically adapt to new fashions in language, neologisms and trends. Our experimental study shows that the proposed method dramatically improves the recall and the F-score, and outperforms the state-of-the-art baselines.

2 Related Work

The proposed research is in the area of sentiment analysis. To determine whether a document or a sentence expresses a positive or negative sentiment, two main approaches are commonly used: the lexicon-based approach and the machine learning-based approach.

The lexicon-based approach (Hu and Liu 2004, Kim and Hovy, 2004; Ding et al., 2008; Taboada, et al., 2010) determines the sentiment or polarity of opinion via some function of opinion words in the document or the sentence. As discussed earlier, this method can result in low recall for our entity-level sentiment analysis.

The machine learning-based approach typically trains sentiment classifiers using features such as unigrams or bigrams (Pang et al. 2002). Most techniques use some form of supervised learning by applying different learning techniques such as Naïve Bayes, Maximum Entropy and Support Vector Machines. These methods need manual labeling of training examples for each application domain.

There are also some approaches that utilizes both the opinion words/lexicon and the learning approach. For example, Wiebe and Riloff (2005) used a subjectivity lexicon to identify training data for supervised learning for subjectivity classification. Our work does not do subjectivity classification. A similar idea was also applied to sentiment classification of reviews in (Tan et al., 2008), which classifies reviews into two classes, positive and negative, but no neutral class, which makes the problem much easier. These approaches are different from ours: First, we perform sentiment analysis at the entity level, thus are assignment of sentiment polarities is done on a much finer level of granularity. Second, our technique for polarity assignment is also different since we deal with three classes of sentiment (positive, negative and neutral) and thus cannot directly apply their methods. Due to low recall of the lexicon-based approach for positive and negative classes, many of the neutral tweets identified are actually opinionated. Therefore we have to identify these opinionated tweets before any classification can be performed. Both the existing methods do not have this step because their two-class classification does not need it. For us, however, this step is crucial.

While most sentiment analysis methods were proposed for large opinionated documents (e.g. reviews, blogs), some recent work has addressed microblogs.

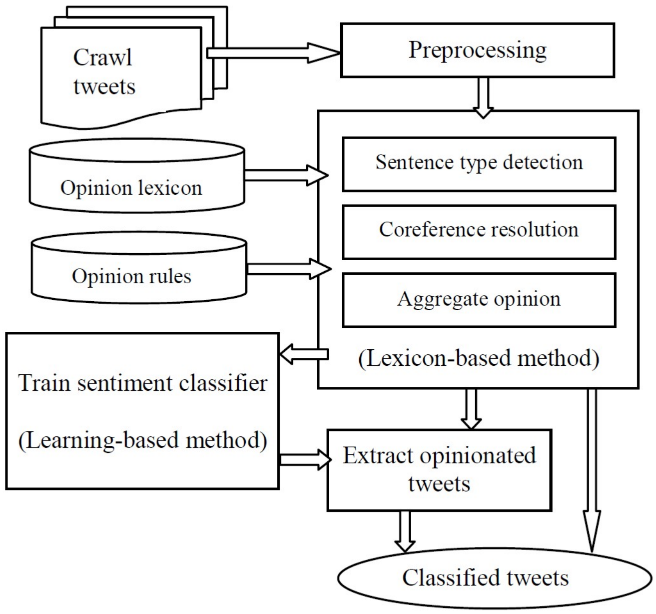

Figure 1: Algorithm architectural overview

Supervised learning is the dominant approach. (Park and Paroubek, 2010) built a sentiment classifier to classify tweets into positive, negative and neutral classes. (Barbosa and Feng, 2010) proposed a two-step classification method. It first classified tweets as subjective and objective, and then classifies the subjective tweets as positive or negative. In (Davidov et al., 2010), many Twitter characteristics and language conventions (e.g. hashtags and smiley) were utilized as features. There are also several online Twitter sentiment analysis systems (e.g. Twend[1], Twitter Sentiment[2], and TweetFeel[3]). These approaches mainly used supervised learning. Our method needs no supervision or manually labeled training data.

While our work is related to transfer learning (Pan et al., 2010; Tan et al., 2008), which uses the learning results from one domain to help learning in another domain, it is significantly different since we exploit a given sentiment lexicon and use it for classification in any domain without any labeling of training data.

3 The Proposed Technique

This section presents the proposed approach. Figure 1 gives an architectural overview of our sentiment analysis algorithm.

We will discuss the techniques in the following sections. Before we dive into the details of the algorithms, let us take a look at the Twitter data first and discuss its characteristics.

[1]http://twendz.waggeneredstorm.com/
[2]http://twittersentiment.appspot.com/
[3]http://www.tweetfeel.com/

3.1 Twitter Data

Twitter has developed its own language conventions. The following are examples of Twitter conventions.

1. "RT" is an acronym for retweet, which is put in front of a tweet to indicate that the user is repeating or reposting.
2. "#" called the hashtag is used to mark, organize or filter tweets according to topics or categories.
3. "@username1" represents that a message is a reply to a user whose user name is "username1".
4. Emoticons and colloquial expressions are frequently used in tweets, e.g. ":-)", "lovvve", "lmao".
5. External Web links (e.g. http://amze.ly/8K4n0t) are also commonly found in tweets to refer to some external sources.
6. Length: Tweets are limited to 140 characters. This is different from usual opinionated corpora such as reviews and blogs, which are usually long.

Another unique characteristic of Twitter data compared to the other opinionated corpora is its volume. It is estimated that people post about 60 million tweets every day and the number is still increasing rapidly.

3.2 Preprocessing

Before starting sentiment analysis, we need to do some data cleansing. We removed retweets (duplicates which do not add any value for our purpose) whose text starts with "RT". We also restore popular abbreviations to their corresponding original forms using a lexicon of abbreviations (e.g. "wknd" to "weekend"). External links and user names (signified by @ sign) are eliminated. However, punctuations are kept since people often express sentiment with tokens such as ":)", ":-)". After cleaning, we perform sentence segmentation, which separates a tweet into individual sentences. Afterwards, we tokenize and perform part of speech tagging (POS) for each sentence.

3.3 Augmented lexicon-based method

In this section, we propose an augmented lexicon-based approach to sentiment analysis considering the characteristics of the Twitter data.

3.3.1 Sentence Type Detection

Sentence type detection is a special step for analyzing tweets. There are three main types of sentences in tweets:

(i) Declarative sentence: it states a view of the author, e.g. "this is a pretty good phone."

(ii) Imperative sentence: it gives a command or request, e.g. "do not buy phone xyz."

(iii) Interrogative sentence: it asks a question, e.g. "what is the best HP desktop in market?"

The first two types of sentences often express opinions. The third type, which is frequently used in Twitter, often does not express any informative opinion on entities. Thus, we need to identify and remove these sentences before analysis. We adopt the following pattern matching rules to detect the interrogative sentence in tweets. The patterns are as follows:

"model word + auxiliary verb + . . . "
". . . + question mark"

where "model word" refers to the first word in the sentence. It should belongs to the word set {what, where, when, why, who}. Auxiliary verb should belong to word set {am, is, are, was, were, am, do, did, does}. Question mark should be the last token in the sentence.

3.3.2 Coreference Resolution

We use some heuristic rules (e.g. the closest entity) to perform coreference resolution in tweets. Although this may not work well in general, it works very well for tweets because tweets are short and simple, and have few complicated sentences. For example, in a tweet, "I bought this iPhone yesterday. It is awesome!". We can resolve that "it" in the second sentence refers to "iPhone" in the first sentence as "iPhone" is the closest entity to "it".

3.3.3 Opinion Lexicon

The lexicon-based approach depends on opinion (or sentiment) words, which are words that express positive or negative sentiments. Words that encode a desirable state (e.g., "great" and "good") have a positive polarity, while words that encode an undesirable state have a negative polarity (e.g., "bad" and "awful"). Although opinion polarity normally applies to adjectives and adverbs, there are verb and noun opinion words as well. Researchers have compiled sets of opinion words and phrases for adjectives, adverbs, verbs and nouns respectively. We obtained our initial opinion lexicon from the authors of (Ding et al., 2008). We then enriched the lexicon with opinion hashtags of Twitter. As introduced before, hashtags are a convention for adding additional context and metadata to microblogs. Some tags are sentiment tags which assign sentiment orientation to the Twitter data, e.g. "#Fail", and "#sucks". We manually add such frequently used opinion hashtags into our opinion lexicon. Note that there are also many words whose polarities depend on the contexts in which they appear. For example, "unexpected" is a positive opinion word for movie domain. Our lexicon does not contain such words. However, we will discuss how to deal with them in the following section.

3.3.4 Aggregating Opinions for an Entity in a Sentence

Using the above opinion lexicon with positive, negative words, we can identify opinion polarity expressed for an entity in a sentence. However, in some cases, we may need to combine several opinion words in a sentence as both positive and negative words may exist in a sentence. We use the aggregation formula in Equation (1) below (which is adapted from (Ding et al. 2008)). The basic idea is as follows. Given a sentence s containing the user-given entity, opinion words in the sentence are first identified by matching with the words in the opinion lexicon. We then compute an orientation score for the entity e. A positive word is assigned the semantic orientation score of +1, and a negative word is assigned the semantic orientation score of −1. All the scores are then summed up using the following score function:

$$score(e) = \Sigma_{w_i : w_i \in L \cap w_i \in s} \frac{w_i \cdot so}{dis(w_i, e)} \quad (1)$$

where $w_i$is an opinion word, L is the opinion lexicon and s is the sentence that contains the entity e, and $dis(w_i, e)$ is the distance between entity e and opinion word w_i in the sentence s. $w_i \cdot so$ is the semantic orientation score of the word w_i. The multiplicative inverse in the formula is used to give low weights to opinion words that are far away from the entity e.

3.3.5 Comparative Sentences

In tweets, comparative sentences are frequently used. It expresses similarity and differences of more than one entity. For example, the sentence, "iPhone is better than the HTC phone", expresses a comparative positive opinion on iPhone and negative opinion on "HTC phone". For these kind of sentences, aggregation rule will not apply. We have to use special techniques to deal with this problem. As we know, the comparison is due to the fact that positive and negative opinion words have their corresponding comparative and superlative forms indicating superior and inferior states respectively. Thus, we first detect comparative word by its corresponding POS Tagging. For example, JJR (comparative adjective), RBR (comparative verb), JJS (superlative adjective and RBS (superlative adverb) are good indicators for comparison sentences. Then we exploit the following two patterns to identify entities in a comparative sentence. Pattern (a) refers to regular comparatives and superlatives forms of comparison. The pattern (b) refers to the equative form of comparison.

(a) entities +. . . + *compword* + . . . + entities
(b) entities +. . . + *as JJ* + . . . + entities

compword is a comparative word. Entity is the entity name in the sentence, which can be identified by its POS tagging - NN or NNP.

Based on the opinion mining, if the sentence is positive, then the entities before the comparative keyword are superior and otherwise they are inferior (with the negation considered). Superlative sentences can be handled in a similar way. Note that equative comparisons do not express preferences.

3.3.6 Opinion Rules

Besides comparative sentences, some language constructs also need special handling, for which a set of rules of opinions are applied. An opinion rule is an implication with an expression on the left and an implied opinion on the right. The expression is a conceptual one as it represents a concept, which can be expressed in many ways in an actual sentence.

Negation rules: A negation word or phrase usually reverses the opinion expressed in a sentence. Negation words include "no" "not", etc. e.g. "this cellphone is not good."

But-clause rules: A sentence containing "but" also needs special treatment. The opinion before "but" and after "but" are usually the opposite to each other. Phrases such as "except that" "except for" behave similarly.

Decreasing and increasing rules: This set of rules says that deceasing or increasing the quantities associated with some opinionated items may change the orientations of the opinions. For example, "The drug eases my pain greatly". Here "pain" is a negative opinion word in the opinion lexicon, and the reduction of "pain" indicates a desirable effect of the drug. Note that we compile a corresponding verb list for these kind of actions, which include "increase", "decease", "diminish", etc. The basic rules are as follows:

Decreased Neg → Positive
e.g.: "My problem has certainly diminished."
Decreased Pos → Negative
e.g.: "The iPad costs me a fortune."

3.3.7 Handling Context-Dependent Opinions

Context-dependent opinion words must be determined by its context. We solve this problem by using global information rather than only local information. We use a conjunction rule to determine the opinion polarity. For example, if in a tweet, people write a sentence like "The movie is really fun and the plot was unexpected". From this example, we can discover that "unexpected" is positive for "plot" because it is conjoined with the positive opinion word "fun". With this idea, we can determine part context-dependent opinion word polarity. For others, we let the sentiment classifier to determine polarity, which we will discuss in the following section.

4 Opinionated Tweet Extraction

As discussed in the introduction, the lexicon-based method can cause low recall. This section proposes a technique to extract additional opinionated tweets. We first extract opinion indicators and then determine whether a tweet is opinionated or not by checking whether it has indicators in the context. The indicator could be a word or a token, which is not in the original opinion lexicon.

Table 1: Contingency table for chi-square test

	With w	Without w	Row Total
Positive set	f_{11}	f_{12}	$f_{11} + f_{12}$
Negative set	f_{21}	f_{22}	$f_{21} + f_{22}$
Column total	$f_{11} + f_{21}$	$f_{12} + f_{22}$	

We use Pearson's chi-square test to identify indicators. Pearson's chi-square test has been popularly used for feature selection in machine learning. We can apply it to our case as well. The basic idea is that if a term is more likely to occur in positive or negative opinion sentences, it is more likely to be an opinion indicator. That is, we need to find out how dependent a term w is with respect to the positive tweets or negative tweets. Such tweets have already been labeled by the lexicon-based method. We first set up a null hypothesis that the candidate indicator w is independent of the positive/negative tweets with respect to its occurrences in the two sets. The Pearson's chi-square test compares observed frequencies of w to its expected frequencies to test this hypothesis. Table 1 shows the content of a contingency table. In the table, f_{ij} represents indicator frequency in the positive/negative tweets set, for example, f_{11} indicates the count of tweet which contains the candidate indicator w in the positive tweet set.

The chi-square value is computed as follows:

$$\chi^2(w) = \Sigma_{i=1,2}\Sigma_{j=1,2}\frac{(f_{ij} - Eij)^2}{Eij} \quad (2)$$

where E_{ij} is the expected frequency of f_{ij} calculated by,

$$Eij = \frac{row\ total_i \times column\ total_j}{f_{11} + f_{12} + f_{21} + f_{22}}, i, j \in \{1, 2\} \quad (3)$$

The larger the chi-square value, the more dependent w is with respect to the positive tweet set or negative tweet set. We select an opinion indicator if it has a chi-square value no less than 6.63, which is at the significance level of 0.01.

5 Sentiment Classifier

In this section, we train a binary classifier to assign sentiment polarity to newly-identified opinionated tweets in the above section. We use Support Vector Machines (SVM) as our learning algorithm.

5.1 Training Data

Training data are the tweets labeled by of the lexicon-based method. We use positive and negative opinion tweets as training examples.

5.2 Classification Feature

Our basic features are unigrams (with negations considered). We also utilize emoticons and hashtag as features which are specific to the Twitter data. All feature types are combined into a single feature vector. (Pang et. al, 2002) shows that feature presence (binary value) is more useful than feature frequency for the SVM classifier. Therefore, we use binary feature values for each feature instead of feature frequency. In order to prevent training bias problem, we remove all the opinion words in the training examples. Stripping out the opinion words causes the classifier to learn from domain-specific words, emoticons, hashtags, etc. The classifier uses these features to determine the sentiment.

5.3 Test Data

Test data is newly-identified opinion tweets from section 4. In order to perform the entity-level analysis, the feature vector of an entity is the context in a text window centered at the entity (the window size is 8 in our case, i.e., 4 words before and 4 words after the entity).

6 Empirical Evaluations

For evaluation, we compare experimental results of the following sentiment analysis methods:

ME: a state-of-the-art learning-based method used by the website "Twitter Sentiment", which uses Maximum Entropy as the supervised learning algorithm. The API[4] of the sentiment classifier is publicly available.

FBS: a lexicon-based method proposed in (Ding et al, 2008) for feature-based sentiment analysis.

AFBS: the augmented lexicon-based method for tweets described in Section 3, without utilizing the final SVM sentiment classifier.

LLS: After opinion indicators are identified in Section 4, we put them into the original general opinion lexicon, and run AFBS again. This method also does not use the final SVM sentiment classifier.

LMS: Our proposed method that utilizes all the techniques described in this paper.

6.1 Data Sets

We used five diverse Twitter data sets obtained from the Twitter API by searching some query entities. The entity terms and the corresponding tweet counts are listed in Table 2. For each data set, we randomly selected five hundred tweets as the test set and the rest is used in training. No manual labeling is involved except the test set.

Table 2: Twitter data sets

Query Entity	**Tweet Count (before preprocessing)**	**Tweet Count (after preprocessing)**
Obama	1,001,879	191,942
Harry Potter	2,216,451	413,001
Tangled	163,569	42,967
iPad	477,324	57,985
Packers	1,614,193	266,319

6.2 Evaluation Measures

We first use accuracy to evaluate the whole classification performance of each method with three classes, positive, negative and neutral (30% - 70% tweets have no opinions, i.e., neutral). For positive and negative sentiments on entities, we employ the standard evaluation measures of precision, recall and F-score.

6.3 Evaluation Results

We manually evaluated the result of each method. An issue in judging opinions for tweets is that the decisions can be subjective. Thus a consensus had to be reached between two annotators.

Table 3 shows the accuracy for all three classes positive, negative and neutral for each method. We can see that the accuracy of our method **LMS** is better than every baseline method.

Table 4 shows the evaluation results for positive and negative opinions on entities. The precision and recall are computed based on both the correctly identified positive and negative sentiments on the entities. From the table, we can see that the supervised method **ME** performs poorly. **AFBS** outperforms **FBS** by considering the characteristics of Twitter data. For F-

Table 3: Accuracy results

Entity	**ME**	**FBS**	**AFBS**	**LLS**	**LMS**
Obama	0.756	0.878	0.868	0.880	0.888
Harry Potter	0.764	0.862	0.880	0.902	0.910
Tangled	0.630	0.794	0.818	0.720	0.882
iPad	0.628	0.642	0.692	0.764	0.810
Packers	0.620	0.720	0.736	0.756	0.780
Average	**0.679**	**0.779**	**0.798**	**0.804**	**0.854**

[4]http://sites.google.com/site/twittersentimenthelp/api

Table 4: Precision, Recall and F-Score Results

	ME			**FBS**			**AFBS**			**LLS**			**LMS**		
Query Entity	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
Obama	0.170	0.202	**0.184**	0.564	0.556	**0.560**	0.522	0.582	**0.569**	0.569	0.708	**0.631**	0.595	0.708	**0.647**
Harry Potter	0.456	0.418	**0.436**	0.822	0.631	**0.714**	0.864	0.641	**0.736**	0.715	0.860	**0.781**	0.751	0.902	**0.820**
Tangled	0.454	0.510	**0.481**	0.927	0.627	**0.732**	0.884	0.679	**0.768**	0.636	0.851	**0.728**	0.827	0.928	**0.874**
iPad	0.263	0.294	**0.278**	0.360	0.352	**0.356**	0.436	0.356	**0.392**	0.576	0.802	**0.671**	0.636	0.831	**0.721**
Packers	0.247	0.327	**0.282**	0.550	0.445	**0.492**	0.672	0.484	**0.563**	0.551	0.714	**0.622**	0.629	0.753	**0.686**
Average	0.318	0.350	**0.332**	0.644	0.522	**0.570**	0.675	0.548	**0.605**	0.609	0.787	**0.686**	0.687	0.827	**0.749**

score, our **LMS** method outperforms **AFBS** by a large margin. The reason is that many opinionated tweets are identified and classified correctly by **LMS**. **LMS** also performed significantly better than **LLS** because the method for identifying sentiment indicators can get many sentiment orientations wrong, which causes mistakes for the subsequent step of sentiment identification using the lexicon-based method in **LLS**. In summary, we can conclude that the proposed **LMS** method outperforms all the baseline methods by large margins in identifying opinions on entities.

7 Conclusions

The unique characteristics of Twitter data pose new problems for current lexicon-based and learning-based sentiment analysis approaches. In this paper, we proposed a novel method to deal with the problems. An augmented lexicon-based method specific to the Twitter data was first applied to perform sentiment analysis. Through Chi-square test on its output, additional opinionated tweets could be identified. A binary sentiment classifier is then trained to assign sentiment polarities to the newly-identified opinionated tweets, whose training data is provided by the lexicon-based method. Empirical experiments show the proposed method is highly effective and promising.

References

1. Aue, A. and Gamon, M. 2005. Customizing Sentiment Classifiers to New Domains: a Case Study. RANLP 2005.

2. Barbosa, L. and Feng, J. 2010. Robust Sentiment Detection on Twitter from Biased and Noisy Data. COLING 2010.

3. Blitzer, J., Dredze, M., and Pereira, F. 2007 Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. ACL 2007.

4. Davidov, D., Tsur, Or., and Rappoport, A. 2010. Enhanced Sentiment Learning using Twitter Hashtags and Smileys. COLING 2010.

5. Ding, X., Liu, B., and Yu, P. 2008. A Holistic Lexicon-based Approach to Opinion Mining. WSDM 2008.

6. Hu, M and Liu, B. Mining and summarizing customer reviews. KDD'04, 2004.

7. Kim, S and Hovy, E. Determining the Sentiment of Opinions. COLING'04, 2004.

8. Pan, S, J., Ni, X., Sun, J., Yang, Qiang., and Chen, Z. 2010. Cross-Domain Sentiment Classification via Spectral Feature Alignment. WWW 2010.

9. Pang B and Lee L, Opinion mining and sentiment analysis. Foundations and Trends in IR. 2008. 1-135.

10. Park, A. and Paroubek, P. 2010. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. LREC 2010.

11. Tan, S., Wang, Y. and Cheng, X. 2008. Combing Learn-based and Lexicon-based Techniques for Sentiment Detection without Using Labeled Examples. SIGIR 2008.

12. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. 2010. Lexicon-based Methods for Sentiment Analysis. Journal of Computational Linguists 2010.

13. Wiebe, J. and Riloff, E. 2005. Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. CICLing 2005.