Automatically Building a Corpus for Sentiment Analysis on Indonesian Tweets

Alfan Farizki Wicaksono, Clara Vania, Bayu Distiawan T., Mirna Adriani
Information Retrieval Lab.
Faculty of Computer Science, University of Indonesia
Depok, Republic of Indonesia
{alfan, c.vania, b.distiawan, mirna}@cs.ui.ac.id

Abstract

The popularity of the user generated content, such as Twitter, has made it a rich source for the sentiment analysis and opinion mining tasks. This paper presents our study in automatically building a training corpus for the sentiment analysis on Indonesian tweets. We start with a set of seed sentiment corpus and subsequently expand them using a classifier model whose parameters are estimated using the Expectation and Maximization (EM) framework. We apply our automatically built corpus to perform two tasks, namely opinion tweet extraction and tweet polarity classification using various machine learning approaches. Experiment result shows that a classifier model trained on our data, which is automatically constructed using our proposed method, outperforms the baseline system in terms of opinion tweet extraction and tweet polarity classification.

1 Introduction

There are millions of textual messages or posts generated by internet users everyday on various user generated content platfroms, such as microblogs (e.g. Twitter[1]), review websites, and internet forums. They post about their stories, experiences, current events that are happening, as well as opinions about products. As a result, the user generated content has become a rich source for mining useful information about various topics.

Twitter, one the popular microblogging platforms, is currently getting a lot of attention from internet users because it allows users to easily and instantly post their thoughts of various topics. Twitter currently has over 200 million active users and produce 400 million posts each day [2]. The posts, known as tweets, often contain useful knowledge so that many researchers focus on Twitter for conducting NLP-related research. McMinn et al. (2014) harnessed millions of tweets to develop an application for detecting, tracking, and visualizing events in real-time. Previously, Sakaki et al. (2013) also used twitter as a sensor for earthquake reporting system. They claimed that the system can detect an earthquake with high probability merely by monitoring tweets and the notification can be delivered faster than Japan Meteorology Agency announcements. Moreover, Tumasjan et al. (2010) demostrated that Twitter can also be used as a resource for political forecasting.

Due to the nature of Twitter, tweets usually express peoples personal thoughts or feelings. Therefore, tweets serve as good resources for sentiment analysis and opinion mining tasks. Many companies can benefit from tweets to know how many positive responses and/or negative responses towards their products as well as the reasons why consumers like/dislike their products. They can also leverage tweets to gain a lot of insight about their competitors. Consumers can also use information from tweets regarding the quality of a certain product. They commonly learn from peoples past experiences who have already used the product before they decide to purchase it. To realize the aforementioned

[1] http://twitter.com

[2] https://blog.twitter.com/2013/celebrating-twitter7

Copyright 2014 by Alfan Farizki Wicaksono, Clara Vania, Bayu Distiawan T., and Mirna Adriani
28th Pacific Asia Conference on Language, Information and Computation pages 185–194

ideas, many researchers have put a lot of effort to tackle one of the important tasks on Twitter sentiment analysis, that is, tweet polarity classification (Nakov et al., 2013; Hu et al., 2013; Kouloumpis et al., 2011; Agarwal et al., 2011; Pak and Paroubek, 2010). They proposed various approaches to determine whether a given tweet expresses positive or negative sentiment.

In this paper, we address the problem of sentiment analysis on Indonesian tweets. Indonesian language itself currently has more than 240 millions of speakers spread in mostly areas of south-east asia. In addition, Semiocast, a company who provides data intelligence and research on social media, has revealed that Indonesia ranked 5th in terms of Twitter accounts in July 2012 and users from Jakarta city (i.e. capital city of Indonesia) were the most active compared to the users from other big cities, such as Tokyo, London, and New York [3]. Therefore, there is absolutely a great need for natural language processing research on Indonesian tweets, especially sentiment analysis, since there would be a lot of information which is worth obtaining for many purposes. Unfortunately, Indonesian language is categorized as an under-resourced language because it still suffers from a lack of basic resources (especially labeled dataset) needed for a various language technologies.

There are two tasks addressed in this paper, namely opinion tweet extraction and tweet polarity classification. The former task is aimed at selecting all tweets comprising users' opinion towards something and the latter task is to determine the polarity type of an opinionated tweet (i.e., positive or negative tweet). To tackle the aforementioned tasks, we employ machine learning approach using training data and word features. However, a problem then appears when we do not have annotated data to train our models. Asking people to manually annotate thousands, even millions of tweets with high quality is not our choice since it is very expensive and time-consuming due to the massive scale and rapid growth of Twitter.

To overcome the aforementioned problem, we propose a method that can automatically develop training data from a pool of millions of tweets. First, we automatically construct a small set of labeled seed corpus (i.e. small collection of positive and negative tweets) that will be used for expanding the training data in the next step. Next, we expand the training data using the previously constructed seed corpus. To do that, we use the rationale that sentiment can be propagated from the labeled seed tweets to the other unlabeled tweets when they share similar word features, which means that the sentiment type of an unlabeled tweet can be revealed based on its closeness to the labeled tweets. Based on that idea, we employ a classifier model whose parameters are estimated using labeled and unlabeled tweets via Expectation and Maximization (EM) framework. In this method, we incorporate two types of dataset: the first dataset is a small set of labeled seed tweets and the second dataset is a huge set of unlabeled tweets that serve as a source for expanding the training data. Intuitively, this method allows us to propagate sentiment from labeled tweets to unlabeled tweets. Later, we show that the training data automatically constructed by our method can be used by the classifiers to effectively tackle the problem of opinion tweet extraction and tweet polarity classification.

In summary, the main contributions of this paper is two-folds: first, we present a method to automatically construct training instances for sentiment analysis on Indonesian tweets. Second, we show some significant works for sentiment analysis on Indonesian tweets which were rarely addressed before.

2 Related Works

There have been extensive works on opinion mining and sentiment analysis as described in (Pang and Lee, 2008). They presented various approaches and general challenges to develop applications that can retrieve opinion-oriented information. Moreover, Liu (2007) clearly mentions the definition of opinionated sentence as well as describes two sub-tasks required to perform sentence-level sentiment analysis, namely, subjectivity classification and sentence-level sentiment classification. However, previous researchers primarily focused on performing sentiment analysis on review data. The trends has shifted recently when social networking platform, such as Facebook and Twitter, has been growing rapidly. As

[3] http://semiocast.com/en/publications/2012_07_30_Twitter_reaches_half_a_billion_accounts_140m_in_the_US

a result, many researchers has now started to perform sentiment analysis on microblogging platform, such as twitter (Hu et al., 2013; Nakov et al., 2013; Kouloumpis et al., 2011; Pak and Paroubek, 2010). In our work, we perform two-level sentiment analysis, similar to that described in (Liu, 2007). In addition, we also perform sentiment analysis on tweets (i.e. Indonesian tweets), instead of general sentences.

Current sentiment analysis research mostly relies on manually annotated training data (Nakov et al., 2013; Agarwal et al., 2011; Jiang et al., 2011; Bermingham and Smeaton, 2010). However, employing humans for manually annotating thousands, even millions of tweets is absolutely labor-intensive, time-consuming, and very expensive due to the massive scale and rapid growth of Twitter. This becomes a significant obstacle for researchers who want to perform sentiment analysis on tweets posted in under-resourced language, such as Indonesian tweets. Limited works have been done previously on automatically collecting training data (Pak and Paroubek, 2010; Bifet and Frank, 2010; Davidov et al., 2010). Some researchers harnessed happy emoticons and sad emoticons to automatically collect training data (Pak and Paroubek, 2010; Bifet and Frank, 2010). They assumed that tweets containing happy emoticons (e.g. ":)", ":-)") have positive sentiment, and tweets containing sad emoticons (e.g. ":(", ":-(") have negative sentiment. Unfortunately, their method clearly cannot get the coverage to reach sentiment-bearing tweets as many as possible since not all sentiment-bearing tweets contain emoticons.

Limited attempts have been made to perform sentiment analysis on Indonesian tweets. Calvin and Setiawan (2014) performed tweet polarity classification limited to the tweets talking about telephone provider companies in Indonesia. Their classification method relies on a small set of domain-dependent opinionated words. Before that, Aliandu (2014) conducted research on classifying an Indonesian tweet into three classes: positive, negative, and neutral. Aliandu (2014) used the method proposed by Pak and Paroubek (2010) to collect training data, that is, emoticons for collecting sentiment-bearing tweets. Even though those researchers performed similar works to us, we have two different points. First, we use different techniques to automatically collect training data. Second, we perform two-level sentiment analysis, namely, opinion tweet extraction and tweet polarity classification. Moreover, in the experiment section, we show that our method to collect training data is better than the one proposed by Pak and Paroubek (2010). Our method also produces much larger data since we do not rely on sheer emoticon-containing tweets to collect training data.

3 Automatically Building Training Data

3.1 Data Collection

Our corpus consists of 5.3 million tweets which were collected using Twitter Streaming API between May 16th, 2013 and June 26th, 2013. As we wanted to build Indonesian sentiment corpus, we used tweet's geo-location to filter tweets posted in the area of Indonesia. We applied language filtering because based on our observation, Indonesian Twitter users also like to use English or local language in their tweets. We then divided our corpus into four disjoint datasets. Table 1 shows the overall statistics of our Twitter corpus.

Dataset	Label	#Tweets
DATASET1	Unlabeled	4,291,063
DATASET2	Unlabeled	1,000,000
DATASET3	Neutral	12,614
DATASET4	Pos, Neg, Neutral	637
Total		5,304,314

Table 1: The statistics of our Tweet collection

To collect DATASET3 (i.e. neutral or non-opinion tweets), we used the same approach as in (Pak and Paroubek, 2010). First, we selected some popular Indonesian news portal accounts from the overall corpus and then labeled them as objective. Here, we assume that tweets from news portal accounts are neutral as it usually comes from headline news. This method was actually proposed by (Pak and Paroubek, 2010). But, we also did some empirical observation and acknowledged that this method performs quite well to collect neutral tweets.

The remaining corpus which is not published by news portal accounts is then used to build seed corpus (DATASET2), development corpus (DATASET1), and gold-standard testing data

(DATASET4). In this study, DATASET2 is used to construct labeled seed corpus. The seed corpus itself contains initial data that is believed to have opinion as well as sentiment. On the other side, development corpus DATASET1 contains unlabeled tweets used to expand our seed corpus. Our testing data (DATASET4) consists of 637 tweets which were tagged manually by the human annotators. These tweets were collected using some topic words which have tendency to be discussed by a lot of people. Two annotators were asked to independently classify each tweet into three classes: positive, negative, and neutral. The agreement of the annotators reached the level of Kappa value 0.95, which is considered as a satisfactory agreement. The label of each tweet in DATASET4 is the label agreed by the two annotators. But, when they did not agree, we asked the third annotators to decide the label. It is also worth to note that our testing data comes from various domains, such as telephone operator, public transportation, famous people, technology, and films. Table 2 and 3 shows the details of DATASET4.

Sentiment Type	#Tweets
Positive	202
Negative	132
Neutral	303
Total	637

Table 2: The statistics of DATASET4

Domain	#Tweets
Telephone operators	94
Public transportations	53
Government companies	11
Figures/People	61
Technologies	12
Sports and Athletes	41
Actress	29
Films	67
Food and Restaurants	34
News	214
Others	21
Total	637

Table 3: The domains in DATASET4

We also show some examples of Tweets found in DATASET4 as follows:

- " *Telkomsel memang wokeeehhh (free internet) :)*" (Telkomsel is nice (free internet) :))
- "*Kecewa sama trans Jakarta. Manajemen blm bagus. Masa hrs nunggu lbh dr 30 menit utk naek busway.*" (really dissapointed in transjakarta. The management is not good. We waited for more than 30 minutes to get the bus on)
- "*man of steel keren bangeeeettttt :D*" (Man of steel is really cool :D)
- "*RT @detikcom: Lalin Macet, Pohon Tumbang di Perempatan Cilandak-TB Simatupang*" (RT @detikcom: Traffic jam, a tree tumbled down in the Cilandak-TB Simatupang intersection)

3.2 Building Seed Training Instances

As we explained before, our seed corpus contains initial data used for expanding the training corpus. We propose two automatic techniques to constuct the seed corpus from DATASET2:

3.2.1 Opinion Lexicon based Technique

In the first technique, we use Indonesian opinion lexicon (Vania et al., 2014) to construct our seed corpus. A tweet will be classified as positive if it contains more positive words then negative words and vice versa. If a tweet contains word with a particular sentiment but the word is preceded by a negation, the polarity of the tweet will be shifted to its opposite sentiment. Moreover, we did not consider the tweets that do not contain any words from the opinion lexicon. In total, we have collected 135,490 positive seed tweets and 99,979 negative seed tweets.

3.2.2 Clustering based Technique

The second technique was implemented by using clustering (Li and Liu, 2012). This technique has several advantages, such as we do not need to provide any resources, such as lexicon or dictionary for a particular language. Each tweet from DATASET2 will be put into three clusters, namely positive tweets, negative tweets, or neutral tweets. We use all terms and POS tags from the tweet and each term is weighted using the TF-IDF as a features. Using this approach, 194 tweets were grouped

into negative tweets, 325 tweets were grouped into positive tweets, and the rest was left out.

3.3 Adding New Training Instances

After we automatically construct labeled seed corpus from DATASET2, we are now ready to obtain more training instances. We use DATASET1, which is much bigger than DATASET2, as a source for expanding training data. The idea is that sentiment scores of all unlabeled tweets in DATASET2 can be revealed using propagation from labeled seed corpus. To realize that idea, we employ a classifier model whose parameters are estimated using labeled and unlabeled tweets via Expectation and Maximization (EM) framework. The well-known research done by (Nigam et al., 2000) have shown that Expectation and Maximization framework works well for expanding training data to tackle the document-level text classification problem. In our work, we also show that this framework works quite well for tweets.

EM algorithm is an iterative algorithm for finding maximum likelihood estimates or maximum a posteori estimates for models when the data is incomplete (Dempster et al., 1977). Here, our data is incomplete since the sentiment scores of unlabeled tweets are unknown. To reveal the sentiment scores of unlabeled tweets using EM algorithm, we perform several iterations. First, we train the classifier with just the labeled seed corpus. Second, we use the trained classifier to assign probabilistically-weighted labels or sentiment scores (i.e. the probability of being a positive and negative tweet) to each unlabeled tweets. Third, we trained once again the model using all tweets (i.e. both the originally and newly labeled tweets). These last two steps are iterated until the parameters of the model do not change. At each iteration, the sentiment scores of each unlabeled tweets are improved as the likelihood of the parameters is guaranteed to improve until there is no more change (Dempster et al., 1977). In addition, only tweets whose sentiment scores surpass a certain threshold will be considered as our new training instances.

Formally, we have a set of tweets $\mathcal{T}$ divided into two disjoint partitions: a set of labeled seed tweets $\mathcal{T}_l$ and a set of unlabeled tweets $\mathcal{T}_u$, such that $\mathcal{T} = \mathcal{T}_l \cup \mathcal{T}_u$. In this case, $\mathcal{T}_l$ represents seed tweets which are selected from DATASET2 and automatically labeled using the method described in the previous section and $\mathcal{T}_u$ represents a set of all tweets in DATASET1. Each tweet $t_i \in \mathcal{T}$, that has length $|t_i|$, is defined as an ordered list of words $(w_1, w_2, ..., w_{|V|})$ and each word w_k is an element of the vocabulary set $V = \{w_1, w_2, ..., w_{|V|}\}$.

For the classifier in the iteration, we employ Naive Bayes classifier model. In our case, given a tweet t_i and two class label C_j, where $j \in S$ and $S = \{pos, neg\}$, the probability that each of the two component classes generated the tweet is determined using the following equation:

$$P(C_j|t_i) = \frac{P(C_j)\prod_{k=1}^{|t_i|} P(w_k|C_j)}{\sum_{j \in S} P(C_j)\prod_{k=1}^{|t_i|} P(w_k|C_j)} \quad (1)$$

The above equation holds since we assume that the probability of a word occuring within a tweet is independent of its position. Here, the collection of models parameters, denoted as θ, is the collection of word probabilities $P(w_k|C_j)$ and the class prior probabilities $P(C_j)$. Given a set of tweet data, $\mathcal{T} = \{t_1, t_2, ..., t_{|T|}\}$, the Naive Bayes uses the maximum a posteori (MAP) estimation to determine the point estimate of θ, denoted by $\widehat{\theta}$. This can be done by finding θ that maximize $P(\theta|\mathcal{T}) \propto P(\mathcal{T}|\theta)P(\theta)$. This yields the following estimation formulas for each component of the parameter.

The word probabilities $P(w_k|C_j)$ are estimated using the following formula:

$$P(w_k|C_j) = \frac{1+\sum_{i=1}^{|\mathcal{T}|} N(w_k,t_i).P(C_j|t_i)}{|V|+\sum_{n=1}^{|V|}\sum_{i=1}^{|\mathcal{T}|} N(w_n,t_i).P(C_j|t_i)} \quad (2)$$

where $N(w_k, t_i)$ is the number of occurences of word w_k in tweet t_i. Similarly, the class prior probabilities $P(C_j)$ are also estimated using the same fashion.

$$P(C_j) = \frac{1 + \sum_{i=1}^{|\mathcal{T}|} P(C_j|t_i)}{|S| + |\mathcal{T}|} \quad (3)$$

In the above equation, $P(C_j|t_i), j \in \{pos, neg\}$, are sentiment scores associated with each tweet $t_i \in \mathcal{T}$, where $\sum_j P(C_j|t_i) = 1$. For the labeled seed tweets, $P(C_j|t_i)$ are rigidly assigned since the label is already known in advance:

$$P(C_j|t_i) = \begin{cases} 1 & \text{if } t_i \text{ belongs to class } C_j \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Meanwhile, for the set of unlabeled tweets T_u, $P(C_j|t_i)$ are probabilistically assigned in each iteration, so that $0 \leq P(C_j|t_i) \leq 1$. Thus, the probability of all the tweet data given the parameters, $P(\mathcal{T}|\theta)$, is determined as follows:

$$P(\mathcal{T}|\theta) = \prod_{t_i \in \mathcal{T}} \sum_j P(t_i|C_j)P(C_j) \quad (5)$$

Finally, we can compute the log-likelihood of the parameters, $logL(\theta|\mathcal{T})$, using the following equation:

$$\begin{aligned} logL(\theta|\mathcal{T}) &\approx \log P(\mathcal{T}|\theta) \\ &= \textstyle\sum_{t_i \in \mathcal{T}} \log \sum_j P(t_i|C_j)P(C_j) \end{aligned} \quad (6)$$

The last equation contains "log of sums", which is difficult for maximization process. Nigam et al. (2000) shows that the lower bound of the last equation can be found using Jensen's inequality. As a result, we can express the complete log-likelihood of the parameters, $logL_c(\theta|\mathcal{T})$, as follows:

$$\begin{aligned} &logL(\theta|\mathcal{T}) \\ &\geq logL_c(\theta|\mathcal{T}) \\ &\approx \textstyle\sum_{t_i \in \mathcal{T}} \sum_j P(C_j|t_i) \log(P(t_i|C_j)P(C_j)) \end{aligned} \quad (7)$$

The last equation is used in each iteration to check whether or not the parameters have converged. When the EM iterative procedure ends due to the convergence of the parameters, we then need to select several tweets from the set of unlabeled tweets $\mathcal{T}_u$, which are eligible for our new training instances. The criteria of selecting new training instances, denoted by $\mathcal{T}_n$, is as follows:

$$\mathcal{T}_n = \{t \in T_u | \ |P(C_{pos}|t) - P(C_{neg}|t)| \geq \epsilon\} \quad (8)$$

where ϵ is an empirical value, $0 \leq \epsilon \leq 1$. In our experiment, we set ϵ to 0.98 since we want to obtain very polarized tweets in terms of sentiment as our new training instances. In summary, the EM algorithm for expanding training data is described as follows:

- **Input:** A set of labeled seed tweets $\mathcal{T}_l$, and a large set of unlabeled tweets $\mathcal{T}_u$
- Train a Naive Bayes classifier using only the labeled seed teets $\mathcal{T}_l$. The estimated parameters, $\widehat{\theta}$, are obtained using equation 2 and 3.
- Repeat until $logL_c(\theta|\mathcal{T})$ does not change (i.e. the parameters do not change):
  - **[E-Step]** Use the current classifier, $\widehat{\theta}$, to probabilistically label all unlabeled tweets in $\mathcal{T}_u$, i.e. we use equation 1 to obtain $P(C_j|t_i)$ for all $t_i \in \mathcal{T}_u$.
  - **[M-Step]** Re-estimate the parameters of current classifier using all tweet data $\mathcal{T}_u \cup \mathcal{T}_l$ (i.e. both the originally and newly labeled tweets). Here, we once again use the equation 2 and 3.
- Select the additional training instances, $\mathcal{T}_n$, using the criteria mentioned in formula 8.
- **Output:** The expanded training data $\mathcal{T}_n \cup \mathcal{T}_l$

4 Experiments and Evaluations

4.1 Training Data Construction

After we applied our training data construction method, we collected around 2.8 millions of opinion tweets when we used opinion lexicon based technique to automatically construct labeled seed corpus. Meanwhile, when we used clustering based technique to construct labeled seed corpus, we collected around 2.4 millions of opinion tweets. We refer to the former yielded training dataset as `LEX-DATA` and the latter as `CLS-DATA`. Table 4 and 5 show the statistics of `LEX-DATA` and `CLS-DATA`, respectively.

Sentiment Type	Pos	Neg
#Seed Tweets	135,490	99,797
#Added Tweets	1,180,506	1,419,438
Total	1,315,996	1,519,235

Table 4: The statistics of `LEX-DATA`

We also automatically collected training data using the method proposed by Pak and Paroubek (2010). We used the well-known positive/negative

Sentiment Type	Pos	Neg
#Seed Tweets	325	194
#Added Tweets	1,332,741	1,160,387
Total	1,333,066	1,160,581

Table 5: The statistics of CLS-DATA

emoticons in Indonesian tweets, such as ":)", ":-)", ":(", ":-(", to capture the opinion tweets from DATASET1 and DATASET2. We refer to this training dataset as EMOTDATA, and we used it for comparison to our proposed method. Table 6 shows the detail of EMOTDATA.

Sentiment Type	Pos	Neg
#Tweets	276,970	103,740

Table 6: The statistics of EMOTDATA

4.2 Evaluation Methodology

To evaluate our automatic corpus construction method, we performed two tasks, namely opinion tweet extraction and tweet polarity classification, harnessing our constructed training data. In other words, we see whether or not a classifier model trained on our constructed training data is able to peform both the aforementioned tasks with high performance.

Task 1 - Opinion Tweet Extraction: Given a collection of tweets **T**, the task is to discover all opinion tweets in **T**. Liu (2011) defined an opinion as a positive or negative view, attitude, emotion, or appraisal about an entity or an aspect of the entity. Thus, we adapt the aforementioned definition for the opinion tweet.

Task 2 - Tweet Polarity Classification: The task is to determine whether each opinion tweet extracted from the first task is positive or negative.

To measure the performance of the classifier, we tested the classifier on our gold-standard set, i.e. DATASET4, which was manually annotated by two people. In addition, we also compared our method against the method proposed by Pak and Paroubek (2010). For the classifier, we employ two well-known classifier algorithms, namely the Naive Bayes classifier and the Maximum Entropy model (Berger et al., 1996). We use the unigrams as our features, i.e. the presence of a word and its frequency in a tweet, since unigrams provide a good coverage of the data and most likely do not suffer from the sparsity problem. Morever, Pang et al. (2002) previously had shown that unigrams serve as good features for sentiment analysis.

Before we train our classifier models, we apply data preprocessing process to all datasets. This is done because tweets usually contain many informal forms of text that can be difficult to be recognized by our classifiers. We use the following data preprocessing steps to our training data:

- Filtering: we remove URL links, Twitter user accounts (started with '@'), retweet (RT) information, and punctuation marks. All tweets are normalized to lower case and repeated characters are replaced by a single character.
- Tokenization: we split each tweet based on whitespaces.
- Normalization: we replace each abbreviation found in each tweet with its actual meaning.
- Handling negation: each negation term is attached to a word that follows it.

4.3 Evaluations on Opinion Tweet Extraction

As we mentioned previously, we see the problem of opinion tweet extraction as a binary classification problem. Thus, we assume that a tweet can be classified into two categories: an opinion tweet and non-opinion tweet. For the testing data, we use DATASET4 that consists of 303 neutral/non-opinion tweets and 334 opinion tweets (i.e. the combination of positive and negative tweets). For the training data, we only have 12,614 non-opinion tweets from DATASET3. But, we have a larger set of opinion tweets either from LEX-DATA, CLS-DATA, or EMOTDATA depending on the method we apply. To cope with this problem, we randomly selected 12,614 opinion tweets either from LEX-DATA, CLS-DATA, or EMOTDATA so that the training data is balanced. Moreover, we use the precision, recall, and F1-score as our evaluation metrics.

First, we measured the performance of the classifiers trained on the data constructed by the method proposed by Pak and Paroubek (2010). We refer to this method as BASELINE. Furthermore, the non-opinion training data consists of all tweets in DATASET3 and the opinion training data consists of 12,614 tweets randomly selected from EMOTDATA. Second, we evaluated the classifiers trained on the data constructed using our proposed method. In this case, we run experiment using the two different seed corpus construction techniques. We refer to the method that use clustering based technique (for constructing seed corpus) as CLS-METHOD and the method that use opinion lexicon as LEX-METHOD. The opinion training data was constructed in the same manner as before. This time, we used LEX-DATA and CLS-DATA to randomly select 12,614 opinion tweets for LEX-METHOD and CLS-METHOD, respectively.

Model	Prec(%)	Rec(%)	F1(%)
BASELINE			
Naive Bayes	75.47	58.98	66.21
Maxent	78.36	74.85	76.56
LEX-METHOD			
Naive Bayes	76.24	64.37	**69.80**
Maxent	81.90	79.94	**80.91**
CLS-METHOD			
Naive Bayes	73.11	46.40	56.77
Maxent	80.00	63.47	70.78

Table 7: The evaluation results for opinion Tweet extraction task

Table 7 shows the results of the experiment. We can see that the classifiers trained on EMOTDATA, which was constructed using BASELINE, actually perform quite well. Maximum Entropy model achived 76,56% in terms of F1-score, which is far from the performance score resulting from Naive Bayes model. It is worth to note that the classifiers trained on LEX-DATA outperform those trained on EMOTDATA by over 3% and 4% for Naive Bayes and Maximum Entropy model, respectively, which means that LEX-METHOD is better than BASELINE. But, the situation is different for CLS-METHOD. This is actually no surprise since LEX-METHOD uses a good prior knowledge obtained from opinion lexicon. This might also suggest that the seed corpus construction is an important aspect in our method.

4.4 Evaluations on Tweet Polarity Classification

After we extract the opinion tweets, we then classify the sentiment type of the opinion tweets into two classes: positive and negative. In the first scenario, we evaluated the classifiers trained on both positive and negative tweets from EMOTDATA since we aimed at comparing BASELINE against our proposed method. In the second scenario, we then measured the performance of the classifiers when they were trained on the data constructed by our method (i.e. LEX-METHOD and CLS-METHOD). For the testing data, both scenarios use DATASET4 that consists of 202 positive tweets and 132 negative tweets. We left the neutral/non-opinion tweets. For the training data, the first scenario uses all tweets in EMOTDATA as the training data. But, we cannot directly use all tweets in LEX-DATA or CLS-DATA for the second scenario since the size of LEX-DATA and CLS-DATA, respectively, is much bigger than EMOTDATA. As a result, due to fairness, we randomly selected 276,970 positive tweets and 103,740 negative tweets from LEX-DATA and CLS-DATA, respectively, and subsequently use them for the second scenario. Moreover, we use a classification accuracy as our metric in this experiment.

Model	Accuracy(%)
BASELINE	
Naive Bayes	74.85
Maxent	73.35
LEX-METHOD	
Naive Bayes	**81.13**
Maxent	**86.82**
CLS-METHOD	
Naive Bayes	42.81
Maxent	45.80

Table 8: The evaluation results for Tweet polarity classification task

Table 8 shows the results. We can see that the classifiers trained on LEX-DATA significantly outperform those trained on EMOTDATA by over 7% and 13% for Naive Bayes and Maximum Entropy model, respectively. Just like the previ-

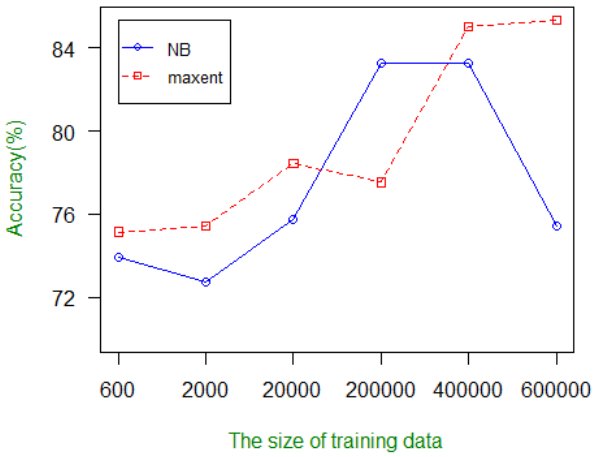

Figure 1: The effect of training data size. Here, we used the training data constructed using LEX-METHOD

ous experiment, CLS-METHOD is no better than LEX-METHOD and BASELINE. We also suggest that Maximum Entropy model is a good model for our sentiment analysis task since the results show that this model is mostly superior to Naive Bayes model.

We further investigated the effect of increasing the size of training dataset on the accuracy of the classifiers. In this case, we only examined LEX-DATA since LEX-METHOD yielded the best result before. Figures 1 shows the results. Training data of size N means that we use N/2 positive tweets and N/2 negative tweets as the training instances. As we can see, learning from large training data plays an important role in tweet polarity classification task. But, we also notice a strange case. When the size of training data is increased at the last point, the performance of Naive Bayes significantly drops. This should not be the case for Naive Bayes. We admit that the quality of our training data set is far away from perfect since it is automatically constructed. As a result, our training data set is still prone to noise disturbance and we guess that this is why the performance of Naive Bayes drops at the last point.

5 Conclusions and Future Works

We propose a method to automatically construct training instances for sentiment analysis and opinion mining on Indonesian tweets. First, we automatically build a set of labeled seed corpus using opinion lexicon based technique and clustering based technique. Second, we harness the labeled seed corpus to obtain more training instances from a huge set of unlabeled tweets by employing a classifier model whose parameters are estimated using the EM framework. For the evaluation, we test our automatically built corpus on the opinion tweet extraction and tweet polarity classification tasks.

Our experiment shows that our proposed method outperforms the baseline system which merely uses emoticons as the features for automatically building the sentiment corpus. When we tested on the opinion tweet extraction and tweet polarity classification tasks, the classifier models trained on the training data using our proposed method was able to extract opinionated tweets as well as classify tweets polarity with high performance. Moreover, we found that the seed corpus construction technique is an important aspect in our method since the evaluation shows that prior knowledge from the opinion lexicon can help building better training instances than just using clustering based technique.

In the future, this corpus can be used as one of the basic resources for sentiment analysis task, especially for Indonesian language. For the sentiment analysis task itself, it will be interesting to investigate various features beside unigram that may be useful in detecting sentiment on Indonesian Twitter messages.

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38, Stroudsburg, PA, USA. Association for Computational Linguistics.

Paulina Aliandu. 2014. Sentiment analysis on indonesian tweet. In *The Proceedings of The 7th ICTS*.

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, March.

Adam Bermingham and Alan F. Smeaton. 2010. Classifying sentiment in microblogs: Is brevity an advantage? In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1833–1836, New York, NY, USA. ACM.

Albert Bifet and Eibe Frank. 2010. Sentiment knowledge discovery in twitter streaming data. In *Proceedings of the 13th International Conference on Discovery Science*, DS'10, pages 1–15, Berlin, Heidelberg. Springer-Verlag.

Calvin and Johan Setiawan. 2014. Using text mining to analyze mobile phone provider service quality (case study: Social media twitter). *International Journal of Machine Learning and Computing*, 4(1), February.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 241–249, Stroudsburg, PA, USA. Association for Computational Linguistics.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38.

Xia Hu, Lei Tang, Jiliang Tang, and Huan Liu. 2013. Exploiting social relations for sentiment analysis in microblogging. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 537–546, New York, NY, USA. ACM.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 151–160, Stroudsburg, PA, USA. Association for Computational Linguistics.

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! In Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts, editors, *ICWSM*. The AAAI Press.

Gang Li and Fei Liu. 2012. Application of a clustering method on sentiment analysis. *J. Inf. Sci.*, 38(2):127–139, April.

Bing Liu. 2007. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Data-Centric Systems and Applications. Springer.

Andrew J. McMinn, Daniel Tsvetkov, Tsvetan Yordanov, Andrew Patterson, Rrobi Szk, Jesus A. Rodriguez Perez, and Joemon M. Jose. 2014. An interactive interface for visualizing events on twitter. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 1271–1272, New York, NY, USA. ACM.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using em. *Mach. Learn.*, 39(2-3):103–134, May.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May. European Language Resources Association (ELRA).

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

T. Sakaki, M. Okazaki, and Y. Matsuo. 2013. Tweet analysis for real-time event detection and earthquake reporting system development. *Knowledge and Data Engineering, IEEE Transactions on*, 25(4):919–931, April.

A. Tumasjan, T.O. Sprenger, P.G. Sandner, and I.M. Welpe. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 178–185.

Clara Vania, Mohammad Ibrahim, and Mirna Adriani. 2014. Sentiment lexicon generation for an under-resourced language. *International Journal of Computational Linguistics and Applications (IJCLA) (To Appear)*.