Atlantis Highlights in Engineering (AHE), volume 2
International Conference on Industrial Enterprise and System Engineering (IcoIESE 2018)

Analysis on Opinion Mining Using Combining Lexicon-Based Method and Multinomial Naive Bayes

Geriska Isabelle
Computer Science
School Of Computing Telkom University
Bandung, Indonesia
geriska.isabelle@gmail.com

Warih Maharani
Computer Science
School Of Computing Telkom University
Bandung, Indonesia
wmaharani@telkomuniversity.ac.id

Ibnu Asror
Computer Science
School Of Computing Telkom University
Bandung, Indonesia
iasror@telkomuniversity.ac.id

Abstract—**Opinion mining is the analysis on opinions which is done by looking at the sentiments, behaviors, or emotions contained in a product. Some of the opinion mining methods are using the lexicon-based and supervised learning. Lexicon-based method has a low recall, while supervised learning has good accuracy but requires a long training period. Therefore this paper will discuss lexicon-based method with one of the supervised learning methods namely Multinomial Naïve Bayes for the English language. These methods are used to classify opinions based on the sentiments, i.e., positive and negative. This research employed the feature extractions: unigram, POS-Tagging, and score-based feature on lexicon. The output of the system is the polarity of each document and the performance will be calculated using Precision, Recall, and F-measure. By implementing the opinion mining using the combining lexicon-based method and Multinomial Naive Bayes, the accuracy obtained was 0.637.**

Keywords—opinion mining; multinomial naive bayes; lexicon-based method

I. INTRODUCTION

In this digital era, writing opinion or online review becomes a trend. According to a survey conducted by BrightLocal, 85% consumers read online review, and 67% consumers will consider the online review result before buying or assess a product [1]. From that number, we know that public opinions have a big impact on product value, whether from the marketing side or opinion that can develop a product. But with the increasing number of the review, assessing product polarity manually, especially for online review, becomes difficult. Opinion mining is one of the answers for classifying opinion based on the sentiments and reviews automatically based on the review polarity.

The appearance of a word that has a sentiment in an opinion will affect the classification of opinion polarity. Research about opinion mining has several approach when determining the polarity. Some of them use unsupervised learning approach, and the rest use supervised learning. Each approach has its advantages and disadvantages. Unsupervised learning, i.e., lexicon-based has a high classification speed but low recall [2]. Meanwhile, supervised learning approach has a high accuracy, but it requires a long running time and a lot of data for its training process. One of the supervised learning approaches that is frequently used for text classification is Multinomial Naive Bayes. Multinomial Naive Bayes uses a probabilistic method suitable for classifying uncertain polarity of opinion. Due to the advantages and the disadvantages of each approach, the hybrid approach is the answer in this research to maximize the advantages and minimize the disadvantages. The combination of both approach is proven to be able to hide the disadvantages of each approach. This can be seen from the increase of the accuracy of the results [3].

The problem occurs on how to combine the two approaches: lexicon-based and supervised learning--Multinomial Naive Bayes when deciding opinion polarity and finding the highest accuracy. Preprocessing being implemented are POS Tagging, lemmatization, stopword removal. The feature extraction used in this research is n-gram for the lexical feature and POS Tagging for the syntactic. The features are selected based on the compability of its features using Multinomial Naive Bayes and lexicon-based method for classifying.

II. THEORETICAL BACKGROUND

A. Opinion Mining

Sentiment according to cambridge dictionary is "a thought, opinion, or idea based on a feeling about a situation" [4]. Opinion mining is an analysis of opinions by looking at the sentiments, behaviors, or emotions contained in a product [5].

The opinions are then grouped according to their polarity, i.e., whether the opinion is positive or negative.

Besides its polarity, opinion mining is also divided into three levels: document-level sentiment analysis, sentence-level sentiment analysis, and feature or aspect-level sentiment analysis. This research will include sentence-level sentiment analysis and document-level sentiment analysis. Opinion extracted from the review will be classified by its polarity. Document-level classification is suitable for data that is written by reviewers and contained an opinion or sentiment.

B. Lexicon-Based Approach

A lexicon-based approach is an approach that uses a dictionary and contains polarity of the word in it. If a word appears in a text, it will be compared with a word in the dictionary, and the sentiment score will be added. The determination of the sentiment is using the lexicon-based approach, and then it is calculated by the total of the polarity contained in a text.

Lexicon-based approach tends to be fast when viewed in terms of computing, since it does not require training on its data, however this approach have disadvantage on its recall, and weak in accuracy [3]. Lexicon-based calculation is divided into three steps, word-level calculation, sentence-level calculation, and document-level calculation [6]. Word-level calculation or lexical compares the word that appears in a review with the word in lexicon dictionary based on its part of speech. The sentence-level will be calculated by using the following formula :

1. Positive sentence score :
$$\frac{1}{nk}\sum_{i=1}^{nk} PositiveSentence(i) \quad (1)$$

2. Negative sentence score :
$$\frac{1}{nk}\sum_{i=1}^{nk} NegativeSentence(i) \quad (2)$$

The formula above shows where Positive Sentence(i) and Negative Sentence(i) are positive scores and negative scores of word-ith based on a lexicon dictionary, and nk is the total number of words in a sentence. For step two, the calculation of the document score will determine the document polarity. The calculation of the document polarity uses the average score of sentences in each polarity. The document will be calculated by using the following formula :

1. Positive sentence score :
$$\frac{1}{ns}\sum_{i=1}^{ns} PositiveDocument(i) \quad (3)$$

2. Negative sentence score :
$$\frac{1}{ns}\sum_{i=1}^{ns} NegativeDocument(i) \quad (4)$$

The formula above shows where Positive Document(i) and Negative Document(i) are positive scores and negative scores of sentence-ith. The sentence score is summed until nsth index. Ns is the total number of sentences in a document. The polarity of the document is determined by comparing the positive and negative document scores. If the positive document scores are higher than the negative ones, then the polarity assigned to the document is positive, and vice versa.

C. AAAVC Algorithm

AAAVC algorithm or Adverb+Adjective and Adverb+Verb is an algorithm using linguistic feature word pair adverb and adjective or adverb and verb. Adjective and verb according to the linguistic experts are parts of speech that usually have sentiment score [7]. Adverb usually acts as a modifier for the word following by it. For example, if there is a sentence "The actor is very attractive" it expresses more positive opinion than "The actor is attractive". Word 'very' here is an adverb and will modify the positive score of word 'attractive' as adjective. Scalling factor used for the calculation is 0.35 according to the best accuracy that stated in [8]. Here is the AAAVC algorithm [9] using SentiWordNet :

SWN (AAAVC) Algorithm

Extract word pair adverb+adjective and adverb+verb for every sentence in document.
For every word pair adverb+adjective do :
if *score* (*adjective*) = 0 then
    ignore it.
end if
if *score* (*adverb*) > 0 then
    if *score*(*adjective*) > 0 then
        f (*adverb, adjective*) = *min* (1, *score* (*adjective*) + *scalling factor* * *score* (*adverb*))
    end if
    if *score*(*adjective*) < 0 then
        f (*adverb, adjective*) = *min* (1, *score* (*adjective*) – *scalling factor* * *score* (*adverb*))
    end if
end if
if *score* (*adverb*) < 0 then
    if *score*(*adjective*) > 0 then
        f (*adverb, adjective*) = *max* (-1, *score* (*adjective*) + *scalling factor* * *score* (*adverb*))
    end if
    if *score*(*adjective*) < 0 then
        f (*adverb, adjective*) = *min* (-1, *score* (*adjective*) – *scalling factor* * *score* (*adverb*))
    end if
end if

For every word pair adverb+verb do:
if *score* (*verb*) = 0 then
ignore it.
end if
if *score* (*adverb*) > 0 then
if *score*(*verb*) > 0 then
f (*adverb, verb*) = *min* (1, *score* (*verb*) + *scalling factor* * *score* (*adverb*))
end if
if *score*(verb) < 0 then
f (*adverb, verb*) = *min* (1, *score* (*verb*) – *scalling factor* * *score* (*adverb*))
end if
end if
if *score* (*adverb*) < 0 then
if *score*(*verb*) > 0 then
f (*adverb, verb*) = *max* (-1, *score* (*verb*) + *scalling factor* * *score* (*adverb*))
end if
if *score*(*verb*) < 0 then
f (*adverb, verb*) = *min* (-1, *score* (*verb*) – *scalling factor* * *score* (*adverb*))
end if
end if
Add polarity score (positive and negative) to respective group

Fig. 1. AAAVC Algorithm

D. Multinomial Naive Bayes

Multinomial Naive Bayes is a supervised learning approach using a probabilistic method. As a supervised learning, the process is divided into two: training and testing. For the training, the probability of each word in a class is calculated by using the following formula [10]:

$$P(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}} \quad (5)$$

Where T_{ct} is a total number of appearance of word t in training document class c, and $\sum_{t' \in V} T_{ct'}$ is the total number of attribute in class c. The attribute are total number of word in class c and the total number of word in vocabulary. However in the formula above, a problem rises when T_{ct} equal to zero, or there are words that did not appear in the training data. To eliminate zero value, laplace smoothing is used for adding 1 to every equation [10]:

$$P(t|c) = \frac{T_{ct}+1}{\sum_{t' \in V}(T_{ct'}+1)} = \frac{T_{ct}+1}{(\sum_{t' \in V} T_{ct'})+B'} \quad (6)$$

The Maximum a Posteriori (MAP) formula is used to avoid the underflow in the testing process to decide the best class for a document :

$$c_{map} = argmax_{c \in C}[\log P(c) + \sum_{1 \le k \le n_r} \log P(t|c)] \quad (7)$$

Prior *p(c)* calculated by using following formula :

$$P(c) = \frac{N_c}{N} \quad (8)$$

Where N_c is total of document in class *c* and *N* is total document in dataset.

E. Lexicon Pooled

Lexicon pooled is an equation used for the probability score aggregation between the lexicon-based method and Multinomial Naive Bayes. The lexicon is pooled using the linear pooling with the following formula [7]:

$$P(t_i|c_p) = \propto_{MNB} P_{MNB}(t_i|c_p) + \propto_{LB} P_{LB}(t_i|c_p) \quad (9)$$

MNB (Multinomial Naive Bayes) and LB (Lexicon Based) are used as the methods and $P(t_i|c_p)$ is probability of term i in class c_p. As for $\propto_{MNB}$ and $\propto_{LB}$ are weight assigned for lexicon-based method and Multinomial Naive Bayes method. To calculate the weight of each method, we use the following formula:

$$\propto_{MNB} = \log(\frac{acc_{MNB}}{1 - acc_{MNB}}) \quad (10)$$

$$\propto_{LB} = \log(\frac{acc_{LB}}{1 - acc_{LB}}) \quad (11)$$

III. IMPLEMENTATION

A. Proposed Method

Figure 2 shows a flowchart diagram for lexicon pooled (combining the lexicon-based method and Multinomial Naive Bayes).

B. Preprocessing

Figure 3 is a step by step of preprocessing that will be used for this system :

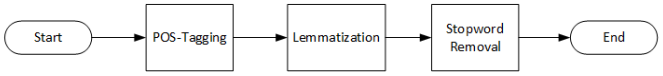

Fig. 3. Preprocessing

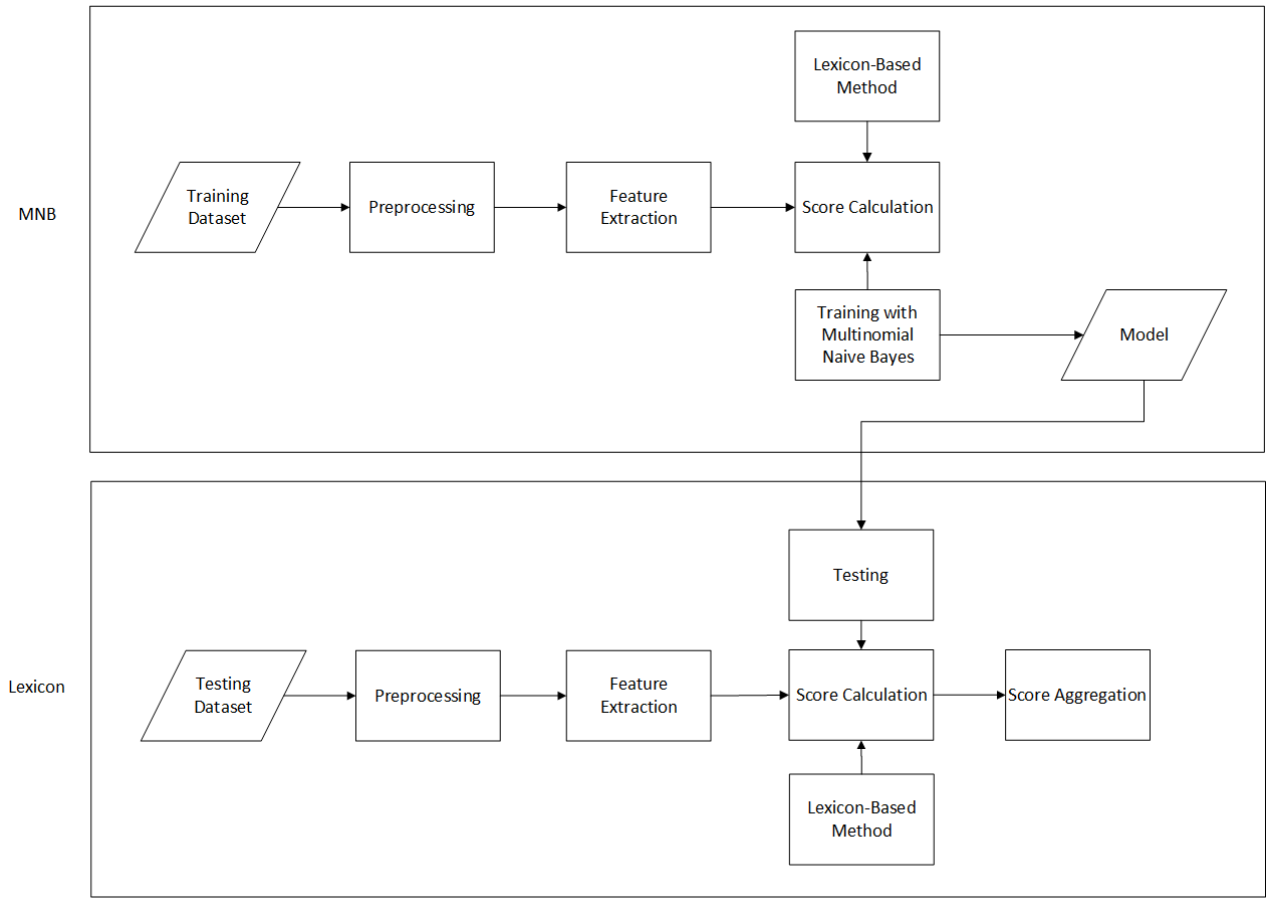

Fig. 2. Flowchart

Preprocessing is used to make a cleaner dataset for the input in the extraction and classification. Preprocessing for the training and testing data has the same stages. Here is the description for each preprocessing step :

1. POS-Tagging
   POS-tagger that is being used is Penn-Treebank-Tag [11]. The input for POS-tagging is movie review dataset. And the output is word with its part of speech label. For example : (films, NNS).
2. Lemmatization
   Lemmatization is used to remove suffix in word. For the example : lemmatization for word 'films' is 'film'.
3. Stopword Removal
   It is used to remove all words containing no meanings, such as : it, them, etc.

C. The Experiment

The test is done in the document level using four parameters [3] : accuracy, precision, recall, and F-measure where

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

$$Precision = \frac{TP}{TP+FP} \quad (13)$$

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

$$F - measure = 2\,x\,\frac{Precision\ x\ Recall}{Precision+Recall} \quad (15)$$

For this experiment, we used Pang and Lee's movie review dataset [12]. The reviews are 2000 movie reviews (1000 positive reviews and 1000 negative reviews). The previous research in [7] and [9] used 811 and 1000 as the total data for the experiment. We used Pang and Lee's movie review dataset because they have sufficient dataset, based on the total dataset from the previous research. 2000 movie reviews are divided into training and testing with 80:20 Pareto principle and the test results are divided into eight types : (i) Multinomial Naive Bayes with lemmatization, (ii) Multinomial Naive Bayes with no lemmatization, (iii) Lexicon-based with lemmatization and

AAAVC algorithm, (iv) Lexicon-based with AAAVC algorithm without lemmatization, (v) Lexicon-based with lemmatization and without AAAVC algorithm, (vi) Lexicon based without lemmatization and AAAVC algorithm, (vii) Linear pooling with lemmatization and AAAVC algorithm, (viii) Linear pooling with AAAVC without lemmatization. The result shown in Table 1.

TABLE I. EXPERIMENT RESULT

Method	Accuracy	*Precision*	*Recall*	*F-measure*
Multinomial Naive Bayes+ *Lemmatization*	0.62089	0.710801	0.643533	0.675497
Multinomial Naive Bayes without *Lemmatization*	0.599609	0.689046	0.625	0.655462
Lexicon-based+ *Lemmatization*+ AAAVC	0.700556	0.704403	0.856415	0.773006
Lexicon-based+ AAAVC without *lemmatization*	0.70273	0.705718	0.859083	0.774885
Lexicon-based+ *Lemmatization* without AAAVC	0.707792	0.71833	0.844426	0.776291
Lexicon-based without *Lemmatization* and AAAVC	0.699093	0.706836	0.847128	0.770649
Lexicon pooled+ *Lemmatization*+ AAAVC	0.637081	0.699029	0.703583	0.701299
Lexicon Pooled+ AAAVC without *Lemmatization*	0.61	0.672131	0.683333	0.677686

D. Result and Analysis

It can be seen from Table 1, that in this experiment, the accuracy of the Multinomial Naive Bayes is smaller than the lexicon-based accuracy. The Multinomial Naive Bayes usually shows a better accuracy than the lexicon-based method. This is because the n-gram used for this system was only unigram. Many possibilities of word pairs for example, bigram such as 'VERY_GOOD', are not calculated.

Multinomial Naive Bayes with lemmatization accuracy has better accuracy than Multinomial Naive Bayes without lemmatization. This is due to higher number of extracted words in Multinomial Naive Bayes without lemmatization compared to Multinomial Naive Bayes with lemmatization. Without lemmatization, the word 'film' and 'films' are distinguished even the two words have same meaning so when the words are calculated in the classification process, the system will compute the probability of every word which has the same meaning as to cause a decrease in accuracy.

The lexicon-based result has better accuracy than the Multinomial Naive Bayes. The accuracy is more than 0.70, except for the lexicon-based without lemmatization and AAAVC algorithm. This is because there is a multiplier or word polarity with part of speech 'adverb' which does not classify word meaning correctly to its polarity. For example, there is a pair of adverb+adjective word 'plenty good', plenty in SentiWordNet dictionary have 0 positive score and 0.375 negative score. Because the multiplier or adverb is negative, then the next word, 'good' positive score will decrease. It should be seen in terms of meaning, that 'plenty good' meaning same as 'very good' or 'really good', so that the word 'good' should have more positive score. The total words in all document is 820,465 words, and AAAVC algorithm only affects 149,641 words or 18.23% of the total word. As for the affected word, there are possibilities for the miss-classification of adverbs as stated from the example 'plenty good' before.

The lowest accuracy score for lexicon-based method is the lexicon-based without lemmatization and AAAVC algorithm which has 0.699093. This is because without the lemmatization, there are many words which are not found in the SentiWordNet dictionary due to the unnormalized words for the words without lemmatization. The SentiWordNet only accepts the normalized words, for example, the word 'adapted' after lemmatization is changed into 'adapt' so that the score of each polarity can be found in the SentiWordNet dictionary. When the word cannot be found, it will returned to zero value and affects the calculation of document polarity.

Lastly, the lexicon pooled results proven to have improved the accuracy of the Multinomial Naive Bayes. Lexicon pooled with AAAVC algorithm and lemmatization increased Multinomial Naive Bayes accuracy from 0.62089 to 0.637081. This is because if a word does not exist in training data, it will be handled by lexicon-based method. But the accuracy improvement was only 0.016191 because the lexicon-based accuracy was only 0.70 and the lexicon-based method will not change the document polarity when the gap between positive and negative in Multinomial Naive Bayes probability is too high. The lexicon pooled without lemmatization has a smaller accuracy result because as it has been stated above, the lemmatization has an important role for the accuracy in each method. But it is still proven that there was an improvement in the Multinomial Naive Bayes accuracy, which was from 0.599609 to 0.61.

IV. CONCLUSION AND FUTURE ENHANCEMENT

It can be concluded that the result based on the movie review dataset, the lexicon pooled with lemmatization and AAAVC algorithm successfully increased the accuracy of Multinomial Naive Bayes which was 0.016191, and the lexicon pooled with AAAVC algorithm and without lemmatization increased the accuracy which was 0.010391.

The methods in which the lemmatization was included in the preprocessing, the accuracy was higher than those which did not have the lemmatization. The AAAVC algorithm did not increase the accuracy significantly because it only affected 18.23% of the total words. There were possibilities of different polarity for the adverbs and the meanings of word pair in the review so that they will affect the document polarity and AAAVC algorithm.

Some future enhancements that can be added to improve this research are: consider adding another extraction for the Multinomial Naive Bayes method input like bigram or unigram with POS ; use cross validation for the experiment; and add more rule for the lexicon-based method like intra sentence conjunction rule, intra sentence comma rule, and inter sentence similarity rule to enhance lexicon-based method accuracy.

REFERENCES

[1] BrightLocal, "Local Consumer Review Survey," 2016. [Online]. Available: https://www.brightlocal.com/learn/local-consumer-review-survey/. [Accessed 20 September 2016].

[2] L. Zhang, M. Dekhil, M. Hsu and B. Liu, "Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis," 2011.

[3] G. Vaitheeswaran and D. L. Arockiam, "Combining Lexicon and Machine Learning Method to Enhance the Accuracy of Sentiment Analysis on Big Data," *International Journal of Computer Science and Information Technologies,* vol. 7, no. 1, pp. 306-311, 2016.

[4] Cambridge, "Cambridge Dictionary," [Online]. Available: http://dictionary.cambridge.org/. [Accessed 29 August 2017].

[5] B. Liu, "Sentiment Analysis: A Fascinating Problem," in *Sentiment Analysis and Opinion Mining*, Morgan and Claypool Publishers, 2012, p. 7.

[6] S. Agrawal and T. j. Siddiqui, "Using Syntactic and Contextual Information for Sentiment Polarity Analysis," in *Int. Conf. on Information Systems*, Seoul, 2009.

[7] R. Dalal, I. Safhath, R. Piryani, D. R. Kappara and V. K. Singh, "A Lexicon Pooled Machine Learning Classifier for Opinion Mining from Course Feedbacks," in *Advances in Intelligent Informatics*, Switzerland, Springer, 2015, pp. 419-428.

[8] F. Benamara, C. Cesarano and D. Reforgiato, "Sentiment Analysis : Adjectives and adverbs are better than adjectives alone," *ICWSM,* 2007.

[9] S. V.K., A. Uddin, R. Piryani and P. Waila, "Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification," in *Int. Multi-Conf. on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*, 2013.

[10] C. D. Manning, P. Raghavan and H. Schütze, "Text Classification and Naive Bayes," in *Introduction to Information Retrieval*, Cambridge University Press, 2008, pp. 253-287.

[11] E. Atwell, "The University of Pennysylvania (Penn) Treebank Tag-set," [Online]. Available: http://www.comp.leeds.ac.uk/amalgam/tagsets/upenn.html. [Accessed 30 September 2016].

[12] B. Pang, "Movie Review Data," [Online]. Available: https://www.cs.cornell.edu/people/pabo/movie-review-data/. [Accessed 5 November 2016].