 ISSN: 1857-7881 (Print) e - ISSN 1857-7431

**ESJ Natural/Life/Medical Sciences**

**Yulius Denny Prabowo**
Informatics Departement Institut Teknologi dan Bisnis Kalbis, Indonesia

**Harya Bima Dirgantara**
Informatics Departement Institut Teknologi dan Bisnis Kalbis, Indonesia

**Larasati**
Informatics Departement Institut Teknologi dan Bisnis Kalbis, Indonesia

Submitted: 06 April 2020
Accepted: 30 September
Published: 31 October 2020

Corresponding author:
*Yulius Denny Prabowo*

DOI: 10.19044/esj.2020.v16n30p135



# Sentiment Analysis Models for Mapping Public Engagement on Twitter Data

**Abstract**
Unstructured data in the form of text, which is widely distributed on the internet, often has valuable information. Due to its unstructured form, an effort is needed to extract that information. Twitter is a microblogging social media platform used by many people to express their opinions or thoughts. Sentiment analysis is a way to map a sentence whether the value is positive or not. Sentiment analysis is a series of processes used to classify text documents into two classes, namely positive sentiment class and negative sentiment class. The dataset is obtained from sentiment 140 as training data to build the sentiment analysis model. To test the model, the data used by the crawler algorithm were extracted using the Twitter API. This study focuses on determining public sentiment based on their writing on Twitter. The classification model used in the study is multiclass naive Bayes. The TF-IDF method was also used to weigh the selected feature. The experimental results show that the resulting model has an accuracy of 74.16% with an average precision of 74%, a recall of 74%, and an f-measure of 74%.

**Subject:** Informatics / Computer Science

**Keywords:** Sentiment Analysis, Maximum Entropy, Tf-Idf, Twitter, Bahasa Indonesia

 ISSN: 1857-7881 (Print) e - ISSN 1857-7431

## Introduction

Sentiment analysis is one of the areas in the study of text analysis. The term “sentiment analysis” is often used to replace the term “Opinion Mining” by researchers. Most researchers even consider that these two terms can replace one another. However, in this study, researchers used the term “Sentiment analysis”. Sentiment analysis is a series of processes used to understand, extract, and process text data to get the sentiment contained in a sentence.

The tendency which is often referred to as sentiment in question can be a binary tendency such as good and bad, right and wrong, positive and negative as well as non-binary tendencies such as positive-neutral-negative. Content on social media such as emotions, behaviour, opinions, and sentiments can be used to analyze a particular topic or trend that occurs on social media or an object (product).

Research in this field began to develop following the research conducted by Pang and Lee (2002). In 2002, Turney et al. (2002) conducted a study with consumer product review data. This study used the Semantic Orientation (Semantic Orientation) method by means of Pointwise Mutual Information (SO-PMI). The best results achieved were 84% of its accuracy in motor vehicle review data and 66% in film review data. The approach from the field of Natural Language Processing is also used to solve problems in the study of sentiment analysis, as conducted by Femphy Pisceldo et al. (2009).

Research on sentiment analysis is mostly done by foreign researchers (Ashraf et al., 2004; Fink, 2011; Pax & Paroubex, 2011; Kouloumpis et al., 2014; Jansen et al., 2009; Read, 2005; Wilson et al., 2005; Yang et al., 2007). Some studies on sentiment analysis were also conducted by some researchers from Indonesia (Winarko et al., 2011; Widyantoro et al., 2012; Fauzi et al., 2017). According to the results, sentiment analysis can be divided into two broad categories. The first category is Coarse-grained sentiment analysis, which is processsed at the document level. In this category, the researcher classifies the orientation of a document as a whole. There are three clear orientations: positive, neutral, and negative. However, this orientation value can be made continuous/not discrete. The second category is Fined-grained sentiment analysis. In this category, the object to be classified is not at the document level but rather a sentence in a document. In general, two approaches are used to conduct sentiment analysis, namely Machine Learning and Knowledge-Based approach.

Knowledge-based approach is the sentiment analysis approach at the word level, where the entity being processed is the word. The methods included in this approach are Lexicon Based and Pointwise Mutual Information. Lexicon-based approach uses a dictionary or lexicon dictionary to evaluate words. In a dictionary, words are paired with their polarity values. It is important to

determine the words to be analyzed from the corpus before using a lexicon. The choice of words can be done by doing Part-of-Speech Tagging and then searching for words such as adjective and adverb. The meaning of words can change depending on the context of the sentence. Therefore, the lexicon-based approach sometimes cannot capture the true meaning of the word it is processing. However, lexicon-based approach has excellent classification performance in cross-domain cases, and knowledge can be added at any time to the dictionary. Some dictionaries used in research include SentiWordNet and AFINN-111. The Pointwise Mutual Information (PMI) method can be used as a semi-unsupervised method approach, where the selected seed words are 'excellent' and 'poor'. With PMI, the sentence is calculated using semantic orientation of the selected seed words. Another hybrid method is done by combining lexicon-based and machine-learning methods. Merging is done because the lexicon-based method is often not appropriate in detecting polarity. This is because words can have different polarities depending on the context of the sentence (Low Recall).

Text mining is defined as a process of extracting information where the user (can be human/machine) interacts with a group of documents using analytical tools. The purpose of text mining is to get useful information from a collection of documents. Data sources used in text mining are a collection of texts that have an unstructured or semi-structured format. Specific tasks from text mining include categorizing text and grouping texts. Also, there are stages in text mining, namely text preprocessing and feature selection.

Text preprocessing is one of the stages in text mining that aims to convert the raw data obtained into data that is ready to be analyzed. Preprocessing reduces the number of noisy features that do not affect the classification process. In other words, the process of digging, processing, and organizing information is done by analyzing the relationship. These rules exist in semi-structured or unstructured textual data. Text preprocessing changes the form of unstructured text data into structured data so that the program can recognize and process the data. In addition, there are several stages in preprocessing text, including Tokenizing, case folding, cleaning, stopword removal, and stemming.

Feature selection stage (feature selection) aims to reduce the dimensions of a collection of text or delete words that are considered not necessary or do not describe the contents of the document. This is done so that the classification process is more effective and accurate. Feature selection is an essential part of optimizing the performance of the classifier. This process will produce any terms or words that can be used as an outstanding representative for a set of documents to be analyzed. Simplifying these features is done for several reasons: (1) Simplifying data/models makes them easier to analyze, (2) To reduce training time (reduce complexity), (3) Avoid the curse of

 ISSN: 1857-7881 (Print) e - ISSN 1857-7431

dimensionality, (4) Remove non-informative features, and (5) Increase generalization by reducing overfitting. There are several methods in feature selection, including Mutual Information (MI), Chi-Square, frequency-based, expert knowledge, information gain (IG), backward elimination, and forward selection.

Thus, this study aims to build a sentiment analysis model with the training dataset provided so that it can be applied to tweets collected via Twitter API. After the model is built, tweets from different cities will be collected. Sentiment analysis will be carried out on the data collected and will be compared with each other.

**Method**

The initial stage in the development of this system is to collect a dataset in the form of tweets (cuitan) that will be used as learning data, test data, and case data on the program to be made. Data collection is done by crawling tweets using the Twitter API. To be able to crawl, researchers must have key credentials to be able to access the Twitter API. After getting credential keys, the researchers then conduct a search according to the specified parameters.

Data that has been given a class is then preprocessed to clean up tweets from unneeded features. In this study, there are several stages of preprocessing. Preprocessing consists of six(6) stages. The first stage is the normalization of tweets by making all letters lowercase; deleting website addresses, usernames, punctuation marks, hashtags, numbers and words less than three characters; and repeating characters. The second stage is word normalization, which replaces words with misspellings and slang languages. The third stage is bigram made up of compound words, which consists of two words with one meaning. Therefore, in the normalization of a bigram, the two words are joined, for example, 'green table' becomes 'mejahijau'. The fourth stage is stemming, which changes a word into its basic words. The fifth stage is negation, which gives additional words to the previous word or to a word that has the word negation (no, no, less). The sixth stage is stopword removal, which eliminates common words that often appear in sentences.

After going through preprocessing, the data is then converted into a matrix by weighing each word using Term Frequency-Inverse Document Frequency. The weight matrix is then entered into the algorithm as learning, and it produces an algorithm model based on the weight matrix.

After learning is done, the test data is then used to ensure that the model created has a good level of accuracy and classification. The test data also goes through a preprocessing process and is converted into a weight matrix that will be inputed into the algorithm to be predicted. The output will be issued in the form of algorithmic accuracy, recall, precision, and f-measure of the model.

The final step is to ensure that the model can work well using new data without test data and training data, i.e., with case data. Case data through preprocessing is converted into a weight matrix and predictions are made again on the data. The output that will be issued in the form of class predictions on each case data is entered.

**Result and Discussion**

The main thing to do when crawling is to access the Twitter API to get data following the research conducted. Twitter API is an application or program created by Twitter to make it easier for developers to access information on Twitter.

After getting the token and key to access the API, the next thing to do is to make the code to crawl. In this study, the code is created using python. It also uses the search API function to get the tweets needed by the research. In order to make the code to crawl, the program is created using the python tweepy library. Tweepy is a library that is used to access the Twitter API by using the credential keys that have been obtained.

The count parameter is used to retrieve data as the value assigned to a single page, the lang parameter is utilized for the language used and to determine the snooze timeout, and the tweet mode parameter is used to retrieve the snipe altogether. The API will perform a snapshot search according to the parameters entered. The details are then stored in a CSV format and are processed in the next step.

Preprocessing is done through several stages such as normalizing tweets (cleaning); normalizing words by replacing slang words and misspellings with dictionaries that have been made; normalizing bigram by combining compound words into one with dictionaries that have been made; stemming by changing words into the original form; the negation of words; and the elimination of the phrase stopword (stopword removal).

Cleaning includes removing URLs, usernames, punctuation marks, hashtags, unique character numbers using regex, and changing shapes into the lowercase using the python function. Standard words are normalized using slang words, word lists, and abbreviations that are often used in txt format. Word normalization is done by matching each word in the tweet in the dictionary that has been made and doing the word replacement following the dictionary. Word normalization process is achieved using 7045 words of data in the dictionary. This dictionary refers to the repository Github riochr17/Analysis-Sentiment-ID, nasalsabila/dictionary-alay, and the dictionary made by researchers following a dataset collected which contains over 240 words. Bigram normalization checks every two words in the tweet and combines the words according to the dictionary made.

 ISSN: 1857-7881 (Print) e - ISSN 1857-7431

The bigram normalization is done by using a dictionary made by researchers who have up to 611 compound words with txt format. Stopword removal is done by using a stopword list in Indonesian, amounting to 778 words in txt format. Stopword data uses data in the Github nolimitid/nolimit-dictionary repository and an additional six words based on the dataset. Preprocessing results is saved into a file using CSV format. The tweets are then converted into a weight matrix by the Term Frequency-Inverse Document Frequency (TF-IDF) method using the Scikit-learn library. The results of this weight matrix is then used as input for the Naïve Bayes algorithm in learning to form a classification model.

The normalization code uses the regex and string libraries. The regex library is used to recognize patterns created by the developer to identify each character in a given string and to replace the characters or words that match the pattern. In the initial process of preprocessing, all characters in the string are converted into lowercase or lowercase letters. This is done to ensure uniformity of the letters in the data to be processed. The next step is to delete some features that have no meaning such as username, website address, and hashtag. Although hashtags sometimes have the meaning of some tweets, in this study, hashtags are not used because of the use of words that are combined.

The function of the delete_data_baca (tweet) is to erase punctuation characters contained in the string punctuation method, which is special characters (! "# $% & '() * +, -. / :; <=>? @ [\] ^ _ `{|} ~). Regex is used to recognize words other than letters and numbers to eliminate symbols and foreign language characters. It eliminates numbers in strings because they are judged to have no sentiment effect and also eliminates features that have fewer than three characters in each word. Furthermore, it removes excess space characters to clean the tweet. The last step in normalization is to eliminate the writing of excessive characters (e.g., u character on 'batuuuu').

The preprocessing data is then defined as tweet frame data and combined with frame label data. The frame data is then stored in the cleaned dataset sentiment file in CSV format. Thereafter, TF-IDF weighing and Multinomial Naïve Bayes classification were carried out. In carrying out the weighing and classification, researchers use the python Scikit-learn (sklearn) library. In addition to weighing and classification, researchers also use the function in the screen to share data and provide performance reports from models that have been made from training data. Pandas library is used to read data. The matplotlib library is used to show visualizations of confusion matrix.

In this study, training data used about 80% of all datasets. The random state parameter provides initial state values when sharing data. The shuffle parameter aims to randomize the dataset to divide. In this study, the shuffle

 ISSN: 1857-7881 (Print) e - ISSN 1857-7431

parameter is true. This means that the dataset to be shared will be randomized first.

## Conclusion

The research is calculated by using real data (case data). Case data is unseen data or data that did not participate in testing in the previous process. It was used to test over thirty (30) tweets. Before testing the data, preprocessing is also done in case data. After preprocessing, the test data is converted into a weight matrix for input to the Multinomial Naïve Bayes algorithm. The results of weighing each word can be seen in Figure 1.

```
(0, 1276)     6.482720089545816
(0, 759)      5.78957290898587
(0, 444)      1.230446661499186
(1, 1141)     5.78957290898587
(1, 929)      2.7570266623091633
(1, 444)      1.230446661499186
(1, 359)      6.482720089545816
(2, 1143)     5.566429357671661
(2, 949)      5.78957290898587
(2, 463)      4.536809940490503
(2, 444)      1.230446661499186
(2, 221)      2.8191584434161694
(3, 1155)     6.482720089545816
(3, 748)      5.566429357671661
(3, 585)      5.78957290898587
(3, 450)      5.566429357671661
(3, 226)      4.873282177111715
(3, 175)      3.880030404101432
(3, 146)      6.482720089545816
(4, 1259)     5.566429357671661
```

**Figure 1.** Test Result

According to the tests performed, the accuracy of the model in the classification of the test data was 77.17%. This test also used 120 test data. Used test data is a set of sentiment data that is used to test how well the model performs. The test data used in this study is different from the training data used to form the classification model. The result of the classification of 120 ticks in the test data resulted in 34.16% positively sentimental tweets, 36.66% neutral sentiment tweets, and 29.16% negative sentiment tweets.

Based on the results of the experiment during the time of the study and the analysis of the results of the experiment, the researcher formulated several research conclusions as follows:

1. The combination of the TF-IDF weighing and the Naïve Bayes Multinomial can be used to classify the quotations based on the sentiment.
2. The algorithm produced from this study can classify the data into three classifications, namely positive, neutral, and negative.

3. With testing data of 120 tweets, a classification accuracy of 74.16% was generated. 34.16% tweets tested positive, 36.66% tested neutral, and 29.16% tested negative.
4. Thirty (30) tweets in the case data obtained 40% tweets with positive sentiment, 23.333% tweets with neutral sentiment, and 36.667% with negative specimens.
5. The results of the model classification on the test data have an average precision of 74%, 74% recall, and f1 score (f-measure) 74%.

**References:**

1. Ashraf, K. M., Eibe, F., Fahringer, B. P., & Holmes, G. (2004). "Multinomial Naïve Bayes for Text Categorization Revisited," Australian joint conference on artificial intelligence No 17.
2. Fink, C. R. (2011). "Coarse- and Fine-Grained Sentiment Analysis of Social Media Text," Johns Hopkins APL Technical Digest, Vol. 30 No. 1.
3. Fauzi, R. S. P. & Ali, M. (2017). "Analisis Sentimen Tentang Opini Pilkada DKI Jakarta 2017 Pada Dokumen Twitter Berbahasa Indonesia Menggunakan Näive Bayes dan Pembobotan Emoji," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 1, No. 12, pp. pages. 1718-1724.
4. Jansen, J. B., Zhang, M., Sobel, K., & Chowdury, A. (2009). "Micro-blogging as online word of mouth branding," Proceedings of the 27th international conference extended abstracts on Human factors in computing systems. New York, NY, USA. ACM, p. pages 3859–3864.
5. Kouloumpis, E., Wilson, T., & Moore, J. (2014). "Twitter Sentiment Analysis: The Good the Bad and the OMG!," Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media.
6. Pang, B., Lee, L., & Vaithyanathan, S. (2002). "Thumbs up?: sentiment classification using machine learning techniques," Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, pp. pp. 79-86.
7. Pisceldo, F., Manurung, R., & Adriani, M. (2009). "Probabilistic Part-of-Speech Tagging for Bahasa Indonesia," Third International MALINDO Workshop, colocated event ACLIJCNLP.
8. Pax, A. & Paroubek, P. (2011). "Twitter as a Corpus for Sentiment Analysis and Opinion Mining".
9. Read, J. (2005). "Using emoticons to reduce dependency in machine learning techniques for sentiment classification," The Association for Computer Linguistics.

10. Turney, P. D. (2002). "Thumbs Up or Thumbs Down? Semantic orientation Applied to Unsupervised Classification of Reviews," Association for Computational Linguistics 40 Anniversary Meeting, New Brunswick, N.J.
11. Wilson, T., Wiebe, J., & Hoffman, P. (2005). "Recognizing contextual polarity in phrase-level sentiment analysis," Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Morristown, NJ, USA. Association for Computational Linguistics, p. pages 347– 354
12. Winarko, Putranti & Edi (2011). "Analisis Sentimen Twitter untuk Teks Berbahasa Indonesia dengan Maximum Entropy dan Support Vector Machine," Indonesian Journal of Computing and Cybernetics System.
13. Widyantoro, I. S., & Hendratmo, D. (2012). "Analisis Sentimen dan Ekstraksi Topik Penentu Sentimen pada Opini Terhadap Tokoh Publik," Jurnal Sarjana Institut Teknologi Bandung Bidang Teknik Elektro dan Informatika Volume 1, Number 2.
14. Yang, C., Lin, K. H. Y., & Chen, H. H. (2007). "Emotion classification using web blog corpora," Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, p. pages 275–278.