Polar Sentiment Analysis for Informal Bahasa Indonesia in Lazada's Product Review Using Naive Bayes

Richard Alison
School of Computer Science
Bina Nusantara University(of Aff.)
Tangerang Selatan, Indonesia
richard.alison@binus.ac.id

Adhella Subalie
School of Computer Science
Bina Nusantara University(of Aff.)
Tangerang Selatan, Indonesia
adhella.subalie@binus.ac.id

Kevin Lotan
School of Computer Science
Bina Nusantara University(of Aff.)
Tangerang Selatan, Indonesia
kevin.lotan@binus.ac.id

Nathanael Santoso
School of Computer Science
Bina Nusantara University(of Aff.)
Tangerang Selatan, Indonesia
nathanael.santoso@binus.ac.id

***Abstract*—Naive Bayes as one of the most popular and reliable machine learning techniques has been used generously to solve problems regarding sentiment analysis. Reviews from popular Indonesian marketplace, Lazada, is tested to analyze its polar sentiment (negative/positive) by using review rating as the scale. The problem was proven quite challenging since the reviews used informal Bahasa Indonesia. After a few training and data adjustment, our model serves a high accuracy with better classification for the 'positive' labeled reviews.**

***Index Terms*—Lazada, Product Reviews, Sentiment Analysis, Naive Bayes, Rating**

I. Introduction

Sentiment Analysis is The common field in Natural Language Processing that has been tried tested over and over again. Naive Bayes for sentiment analysis in particular is one of the most popular machine learning method that has been used numerous times before.

This time, Naive Bayes was tested against a set of customer reviews from one of the biggest marketplace in Indonesia, Lazada. The problem is quite challenging, since the language used is Bahasa Indonesia which is not the most developed language for natural language processing. Moreover, customer reviews tends to be in informal language, causing more unusual data and more work for preprocessing.

In this paper the reviews are used for training and testing for a review-based sentiment analysis model.

II. Data

The data used here were collected by scraping the Lazada's website [1]. There are around 200.000 tuple in the data set and each tuple represents a review. It has a few features, but the main feature is the reviewContent and the rating. rating is an integer ranging from 1-5 and reviewContent is a string. The language mainly used in the data is informal Bahasa Indonesia. Here's an example of how the data looks like: (Table 1)

rating	reviewTitle	reviewContent
5	"ok mantaaapppp barang sesuai pesanan.. good"	"okkkkk mantaaaaaapppp ... goood"
3	-	"Pengiriman super lama.. tapi datang juga sich"
1	-	"baru 10 bulan layarnya dah bergaris"

TABLE I
EXAMPLES OF HOW THE DATA LOOKS LIKE

III. Methodology

A. Preprocessing

The main problems of this data set are as follows:

- Informal language and unstructured grammar
- People who only give ratings (null reviewContent feature)

And so, by considering those problems, a sequence of preprocessing steps were determined

1) Eliminating Tuple:

The data set used was sadly not a perfect one. A lot of the column has null value. This happens because of people who only give ratings when reviewing products, leaving the reviewContent column empty(null). The solution is to just drop tuples that has null column. The downside was that the tuple number reduced immensely, so much so that only half of the data remained. *:*

2) Case Folding:

Standardizing the data is important to ensure maximum usage of the data. Converting all of the letter in the data set into lowercase is the most standard way to do this/ *:*

3) Noise Removal:

At this part, punctuation and numbers are removed.*:*

4) Eliminating Repeating Letters:

Eliminating repeating letters was meant to tackle one of the main problem, which is the informal language used in the data set. There are quite a number of reviews that uses word that has repeating letters, either to emphasize the emotion of customer or to make the review longer. e.g: ”mantaaaaap” to ”mantap”.:

5) Stemming:

Indonesian language also has prefixes and suffixes, these prefixes and suffixes needs to be removed from the data. This is meant to reduce computing new words because of the difference in prefix and or suffixes. Stemming in Bahasa Indonesia can be done with the help of using PySastrawi library. This library has the main function to stem Indonesian words.:

6) Tokenization:

To use the data, tokenization is important. Tokenization separates a string into its word component. The tokenization process was done using NLTK library.:

7) Remove stop words:

Removing stop words like ”the”, ”a”, ”an” can potentially improve the performance as there are fewer and only meaningful tokens left in the data. This process is also done by using NLTK library. :

B. TF-IDF

TF-IDF (Term Frequency–Inverse Document Frequency) is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. TF-IDF has been used in many NLP tasks, e.g: information retrieval, keyword extraction, etc. TF-IDF calculates the frequency of appearance in the document and the inverse document frequency of the word in a set of documents. In changing the words after preprocessing to a binary matrix, we used TF-IDF Vectorizer from sklearn to calculate the score matrix of TF-IDF features.

C. Naive Bayes

In this comparison, we used Naive Bayes classifier. In machine learning, Naive Bayes classifier are a family of ”probabilistic classifier”. Naive Bayes is a simple technique for constructing classifier, it is simplified from Bayesian probability model. The Naive Bayes classifier is operating on independence assumption, means that the probability of one attribute does not affect the probability of the other. If the data has n attributes that means the Naive Bayes creates 2n! independent assumptions [2].

The equation in English to calculate the Bayesian probability is:

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}} \tag{1}$$

The type of Naive Bayes that we used is Bernoulli Naive Bayes. The data is in a binary matrix vector after using TF-IDF Vectorizer. We used Bernoulli Naive Bayes because the data is in Bernoulli distribution. The equation for Bernoulli Naive Bayes will be as follows:

$$p(\mathbf{x} \mid C_k) = \prod_{i=1}^{n} p_{ki}^{x_i}(1 - p_{ki})^{(1-x_i)} \tag{2}$$

IV. Experiment Results

The results after the first training attempt were very poor. Only 61 of the ’negative’ labeled data were classified as true even though that’s not the case.

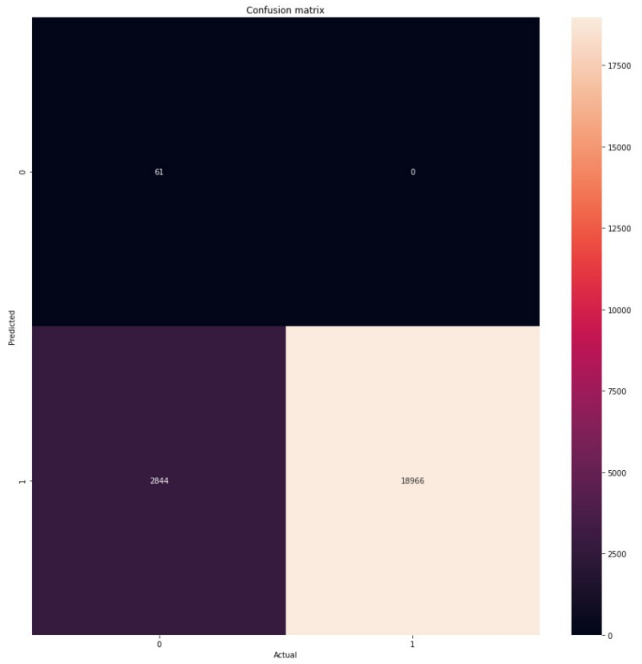

Fig. 1. Confusion Matrix on the first training

We suspected that the aforementioned results were caused by an imbalance in the training data. There were significantly more positive tuples than there were negative ones. To improve our results, we decided to use only 18.000 data tuples out of the total amount of roughly 100.000 tuples. This number was picked based on the amount of available negative tuples in order to balance the distribution of training data more evenly.

The suspicion turned out to be true, as the second training yield satisfactory result, reaching up to 93% accuracy, 94% precision, 93% recall and 93% F-1 Score.

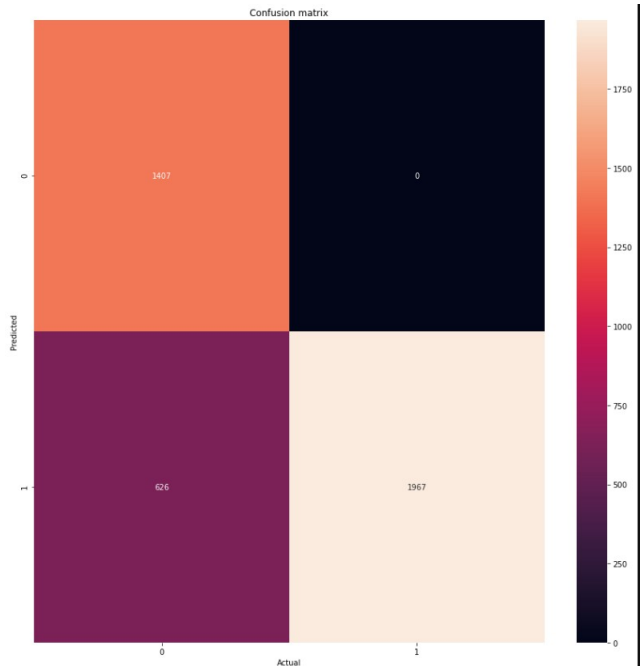

Fig. 2. Confusion Matrix on the second training

Another improvement we made was to adjust the classification of the reviews. We changed the label of each 3/5 star rating to positive (1,2 = 'neg', 3,4,5 = 'pos'). whereas previously it was negative (1,2,3 = 'neg', 4,5 = 'pos').

This last adjustment showed immense improvement, as the training accuracy increased significantly to 100%. We considered the possibility of the model overfits the training data instead of actually improving its performance. However, as shown in figure 3, this concern was unfounded, as the model has also improved its performance on the test sets.

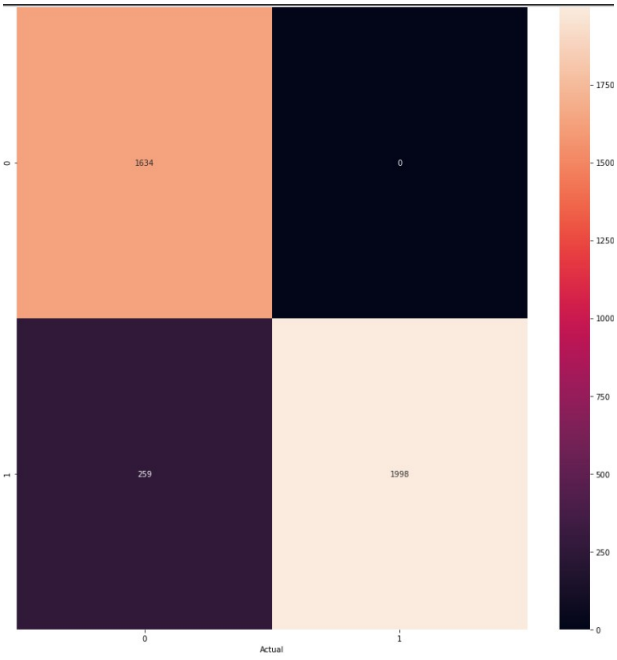

Fig. 3. Confusion Matrix on the second training

V. Conclusion

The experiments presented in this paper shows that our model proves to be extremely accurate in predicting positive sentiment. Figure 3 shows that our model is much better at predicting positive reviews than it is at predicting negative ones.

The reason could be that the rating system is biased towards stronger emotions such as disappointment or satisfactory feelings because people tend to give reviews when they feel stronger emotions. More average reviews could be hiding in the higher rating or are deleted with the rest of the tuple that has no reviewContent. This model could also be prone to overfitting since the accuracy at the final training was suspiciously high.

References

[1] G. Nibras, Lazada Indonesian Review, 2019, Retrived 17 June 2020 from https://www.kaggle.com/grikomsn/lazada-indonesian-reviews?select=20191002-reviews.csv.

[2] S. Mukherjee and N. Sharma, Intrusion Detection using Naive Bayes Classifier with Feature Reduction, 2012.