Twitter Sentiment Analysis using Naïve Bayes Classifier with Mutual Information Feature Selection

Maria Arista Ulfa, Budi Irmawati, and Ario Yudo Husodo
Prodi. Teknik Informatika, Fakultas Teknik, Universitas Mataram
Jl. Majapahit 62, Mataram, Lombok, NTB-INDONESIA
Email: maria.arista95@gmail.com, budi-i@unram.ac.id, ario@ti.ftunram.ac.id

***Abstract*—Sentiment analysis is an identification technique of emotion expressed in texts. The sentiment analysis goal is to determine a negative or positive opinion within a sentence or a document. Twitter is one of social medias to convey an opinion. The Twitter allows its users to write opinions related to a specific topic in a tweet. The Twitter data used in this research was downloaded using the Twitter Application Programming Interface (API). It consists 500 tweets about Lombok tourism that contained #lombok and #woderfullombok hashtags. We selected the features extracted from the Twitter data using the *Mutual Information* (MI) method and analyzed them using the *Naïve Bayes Classifier* model. The evaluation of sentiment analysis on the Lombok tourism Twitter data in a 10-folds cross validation resulted 97.9% accuracy.**

Keywords: Mutual Information, Naïve Bayes Classifier, Sentiment Analysis, Twitter.

I. Introduction

Sentiment analysis is a computational study where the data are taken from opinions and emotions expressed in texts. The main task of a sentiment analysis is to classify the texts' polarity. The polarity relates to a positive or a negative perception. Therefore, the sentiment analysis is mostly used to grasp opinion about a product, a service, or a political issue [1].

One of social medias to express people's opinions is Twitter. It is possible for the users to share their expressions publicly in a short tweet. However, Twitter data cannot be analyzed rawly because the Twitter users type shorten words to maintain the word length limitation (e.g. 140 character maximum) in a tweet. In words, the Twitter are required to be cleaned before the analysis processes. This research aim was to justify the polarity of a tweet using the *Naïve Bayes* Classifier (NBC). The classifier is simple and fast in training a model yet it results a high accuracy prediction. It is also benefited on an easy implementation [2]. Relying on classification model, the features extracted from the data are required to be selected to shorten the training and prediction calculation time. Therefore, we used *Mutual Information* (MI) [3] to select the extracted features.

This paper describes previous works on some sentiment analyses in Section II. In this section, we describe the use of some classification model to predict a text polarity. This section is followed by our research methods in Section III. We then explained our results and discuss it in Section IV. The conclusion is then reported in Sec V.

II. Previous Works

Utami [4] used *Support Vector Machine* (SVM) and *K-Nearest Neighbour* (K-NN) with *Particle Swarm Optimization* (PSO) feature selection to decide public sentiment on forest fire news. She used 360 data and got 86.11% accuracy using SVM while 73.06% using K-NN. Ernawati [5] used NBC with PSO feature selection to decide the sentiment on selling reviews of a restaurant. She worked on 400 restaurant reviews and got 86.88% accuracy. Darma et al. [6] used SVM to analyze a television show with *Genetic Algorithm* (GA) as their feature selection method. Working on 160 tweets, where 80 tweets were positive and 80 tweets were negative, they got 90.50% accuracy. Gupta and Parveen [7] used NBC with *Term Frequency-Inverse Document Frequency* (TF-IDF) and *Gain Ratio* feature optimalization in centiment analysis of movie review. They obtained 94% accuracy with 500 data.

Based on the literature reviews, the accuracies depends on the classification model and feature selections. Gupta and Parveen who used NBC and TF-IDF feature selection only got 78% accuracy. They then got significantly high improvement compared to others after they applied the *Gain Ratio* optimization.

Nevertheless, feature selection improved the performance without optimization as in Darma et al. [6], SVM generally works well on binary classification and sufficiently be used on high dimension dataset. However, SVM takes longer computation time and is sensitive to the unbalance data [8]. The decreasing of SVM performance on unbalance data is caused by the bias when it meets the hyperplane with soft margin that will force the prediction to the majority class [9]. Aside of SVM model, even though the NBC model is simple, it is effectively fast, and results high accuracy on sentiment analysis. NBC also works well on unbalanced data [10]. Considering those reviews, we used NBC as the classification model in this research.

The feature selection process in the previous research used PSO, GA, and TF-IDF [5]–[7]. These three models only considered the existent of a feature in one class without considering theirs in other classes. Therefore, we use MI to calculate the probability of the features' existent to involve appropriate features in a class.

III. Method

A. Data Collection

We crawled 500 tweets in English using Twiter API. The data were then annotated manually as positive and negative by two English teachers.

B. Pre-processing

As written before, the Twitter data are required to pre-process before they could be analyzed. Consequently, our annotated tweets were then *pre-processed* by seven steps as is drown in Figure 1.

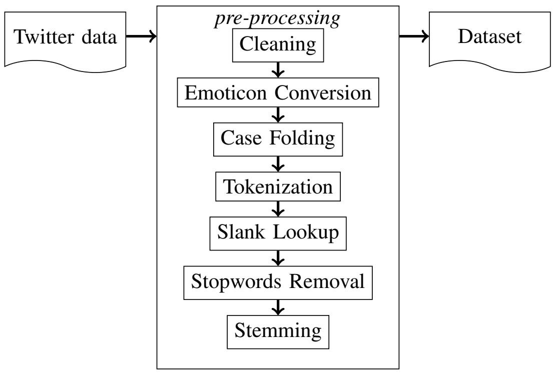


Fig. 1. The pre-processing steps.

For easy illustration, suppose we have a sample tweet in our data,

> OMG!! Lombok is my favourite and most beautiful 6 beaches :* #lombok #beaches #gili #indonesia #travel #ttot https://t.co/xjuRrO5fGK

the pre-processing steps are explained below.

1) **Cleaning**
In this step, we removed unnecessary words. The removed words were: `url`, `hashtag (\#)`, `username (@)`, `retweet (RT)`, and `email`.

> OMG!! Lombok is my favourite and most beautiful 6 beaches :*

2) **Convert Emoticon**
This step converted each emoticon symbol to a word that resembles its meaning. Table I lists the emoticon symbols and their conversion.

TABLE I. Emoticon conversions

Emoticons	Meaning
>:] ;) :-) :) :o) :] :3 :c) :> =] 8) =) :} : ∧) (: ;-) >:D :-D :D 8D x-D xD ;* :* =-D =D =-3 =3 ;D :P :-P ¡3	happy
>:[:-(:(:-c :c :-< :< :-[:[:{ >.> <.< >.< =(>:\ >:/ :-/ :/ :\ =/ = :S :s >:o :O :-O 8-0	bad

Below is the resulted conversion tweet.

> OMG!! Lombok is my favourite and most beautiful 6 beaches happy

3) **Case Folding**
In this step, we converted the tweet to lower case to reduce the vocabulary size.

> omg!! lombok is my favorite and most beautiful 6 beaches happy

4) **Tokenization**
This step splited all the words separated by a white space and removed all the punctuations and numbers. It resulted the tokens below.

omg	lombok	is	my	favorite
and	most	beautiful	beaches	happy

5) **Slang Lookup**
Slang words are terms that are used in the conversation communication, usually by teenagers in the similar age. This step converted slunk words to their corresponding terms.

oh	my	god	lombok	is
my	favourite	and	most	beautiful
beaches	happy			

6) **Stopword Removal**
Stop words are words that have no meaning when they are lonely. They are mostly pronouns or connecting words. Therefore, this step removed the stop words.

god	favourite	beautiful	beaches	happy

7) **Stemming**
This step converted all words to the basic form (root words). The stemming process should follows the morphological rules of a language.

god	favourite	beautiful	beach	happy

C. Mutual Information Feature Selection (MI)

Feature selection aims to reduce the classification computation time by removing uninformative features. MI is one selection method used in machine learning that shows how strong a feature contributes in correct prediction [3]. Figure 2 shows the feature selection process using MI.

The calculation of MI score for each feature is expressed as I in Equation (1).

$$I(T;C) = \sum_{e_c \in C} \sum_{e_t \in T} P(e_t, e_c) log \frac{P(e_t, e_c)}{P(e_t)P(e_c)} \qquad (1)$$

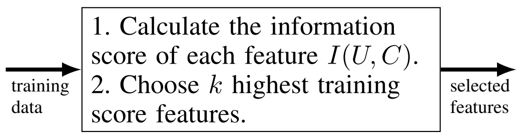

training data
1. Calculate the information score of each feature $I(U, C)$.
2. Choose k highest training score features.
selected features

Fig. 2. Mutual Information feature selection.

TABLE II. EXAMPLE OF ORIGINAL TWEETS AND THEIR EXTRACTED FEATURES.

#	Tweet	Kelas	Extracted Features
T1	OMG!! Lombok is my favorite and most beautiful 6 beaches :* #lombok #beaches #gili #indonesia #travel #ttot https://t.co/xjuRrO5fGK	positive	god[1], favorite [1], beautiful[1], beach[1], happy[1]
T2	All you need is love, happy and beach! :* #gilitrawangan #lombok #island	positive	love[1], beach[1], happy[2]
T3	Love this place #gilitrawangan #lombok #ntb https://t.co/l9Xn5Nkv3x	positive	love[1]
T4	so bad !! I found a lot of rubbish on #senggigi #savesenggigi #beach #dirty	negative	bad[1], rubbish[1]
T5	Over Troubled Water :/ #bnw #bw #slowshutter #slowshutterspeed https://t.co/0IBayn	negative	trouble[1], water[1], bad[1]

where $P(e_t, e_c)$ is the joint probability function of T and C, and $P(e_t)$ and $P(e_c)$ are the marginal probability distribution functions of T and C respectively. It could also be expressed as Equation 2.

$$\begin{aligned} I(T;C) =& \frac{N_{11}}{N} log_2 \frac{N \cdot N_{11}}{N_1.N_{.1}} + \frac{N_{01}}{N} log_2 \frac{N \cdot N_{01}}{N_0.N_{.1}} \\ &+ \frac{N_{10}}{N} log_2 \frac{N \cdot N_{10}}{N_1.N_{.0}} + \frac{N_{00}}{N} log_2 \frac{N \cdot N_{00}}{N_0.N_{.0}} \qquad (2) \end{aligned}$$

where:
N: number of documents ($N_{00} + N_{01} + N_{10} + N_{11}$).
$N_1.$: number of documents having e_t ($N_{10} + N_{11}$).
$N_{.1}$: number of documents having e_c ($N_{01} + N_{11}$).
$N_0.$: number of documents not having e_t ($N_{01} + N_{00}$).
$N_{.0}$: number of documents not having e_c ($N_{10} + N_{00}$).

Table II shows the examples of original tweets and the features extracted from each related tweet. From the feature '*happy*' in Table II where e_t='*happy*' and e_c='*positive*', we obtained the ***contingency*** **table** below.

	$e_{positive} = 1$	$e_{positive} = 0$
$e_{happy} = 1$	2	0
$e_{happy} = 0$	1	2

Therefore,

$$\begin{aligned} I_{happy}(T,C) =& \frac{2}{5} log_2 \frac{5 \cdot 2}{(2+0)\cdot(2+1)} \\ &+ \frac{1}{5} log_2 \frac{5 \cdot 1}{(1+2)\cdot(2+1)} \\ &+ \frac{0}{5} log_2 \frac{5 \cdot 0}{(2+0)\cdot(0+2)} \\ &+ \frac{2}{5} log_2 \frac{5 \cdot 2}{(1+2)\cdot(0+2)} \\ &+ 0.419 \end{aligned}$$

Table III shows the calculation for all features. We selected only the first five features in the left side of the table that are written bold.

TABLE III. THE MI SCORE CALCULATION RESULTS.

Features	MI score	Features	MI scores
bad	0.971	*trouble*	0.312
beach	0.419	*water*	0.312
happy	0.419	*god*	0.171
love	0.419	*favourite*	0.171
rubbish	0.322	*beautiful*	0.171

D. Naïve Bayes Classification

The Naïve Bayes classification has two steps: training and classification, where each step consists of two processes, as shown in Figure 3. The explanation of each process is described below.

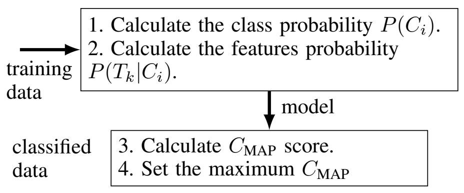

1. Calculate the class probability $P(C_i)$.
2. Calculate the features probability $P(T_k|C_i)$.
training data
model
classified data
3. Calculate C_{MAP} score.
4. Set the maximum C_{MAP}

Fig. 3. The Naïve Bayes classification steps.

1) **Class Probability Calculation**
The class probability is the number of tweets in class C_i divided by the number of training data D expressed in Equation 3. Table IV shows an example to calculate a **class probability**.

$$P(C_i) = \frac{f(C_i)}{|D|} \qquad (3)$$

TABLE IV. THE EXAMPLE OF CLASS PROBABILITY FROM THE EXAMPLES IN TABLE II.

Class	Tweet					$fd(C_i)$	$p(C_i)$
	1	2	3	4	5		
positive	1	1	1	0	0	3	3/5
negative	0	0	0	1	1	2	2/5

2) **Feature Probability Calculation**
The feature probability given class C_i is the existent of feature T_k in class C_i divided by the sum of the number of features in class C_i and the number of features as expressed in Equation 4. Table V shows the example of feature probability calculation.

$$P(T_k|C_i) = \frac{f(T_k|C_i) + 1}{f(T_k) + |T|} \qquad (4)$$

3) **Classification**
The classification is to maximize production of probability of feature T_k given class C_i times the

probability of class C_i expressed in Equation 5.

$$C_{\text{MAP}} = \arg\max_{C_i \in C} \prod_{k=1}^{n} P(T_k|C_i)P(C_i) \tag{5}$$

Suppose we have a test data:

Rubbish everywhere!! Mataram city Lombok https://www.instagram.com/p/BChXtu

We got four features:

rubbish	mataram	city	lombok

As the only feature that match the training vocabulary is '*rubbish*', the C_{MAP} calculation for the test data is as follow.

$$\begin{aligned} C_{\text{MAP}}(\text{'pos'}) &= P(\text{'pos'}) \cdot P(\text{'}\textit{rubbish}\text{'}|\text{'pos'}) \\ &= \frac{3}{5}3 \times \frac{1}{20} = 0.03 \\ C_{\text{MAP}}(\text{'neg'}) &= P(\text{'neg'}) \cdot P(\text{'}\textit{rubbish}\text{'}|\text{'neg'}) \\ &= \frac{2}{5} \times \frac{2}{15} = 0.05 \end{aligned}$$

where 'pos' and 'neg' for positive and negative classes respectively. In this example, the C_{MAP}('neg') is larger than C_{MAP}('pos') that classifies the test data to negative class.

E. Validation and Evaluation

We validated the experimental results in a k-fold cross validation and we evaluated them using *confusion matrix* to see the performance of the classification model [8]. Table VI is the confusion matrix.

This research evaluated the results using the *accuracy* as expressed in Equation 6.

$$\begin{aligned} accuracy &= \frac{\#\text{ correct classification}}{\#\text{ data}} \\ &= \frac{TP+TN}{TP+FP+FN+TN} \end{aligned} \tag{6}$$

However, in the case of unbalanced data, the accuracy cannot significantly measure the model performance. Therefore, we also used *precision* and *recall* as expressed in Equations 7 and 8.

$$precision = \frac{TP}{TP+FP} \tag{7}$$

$$recall = \frac{TP}{TP+FN} \tag{8}$$

TABLE V. THE EXAMPLES OF CLASS PROBABILITY CALCULATION FROM THE EXAMPLES IN TABLE II.

Data	**Class**	
$f(T_k\|C_i)$	**positive**	**negative**
bad	$\frac{0+1}{10+10} = \frac{1}{20}$	$\frac{2+1}{5+10} = \frac{3}{15}$
beach	$\frac{2+1}{10+10} = \frac{3}{20}$	$\frac{0+1}{5+10} = \frac{1}{15}$
happy	$\frac{3+1}{10+10} = \frac{4}{20}$	$\frac{0+1}{5+10} = \frac{1}{15}$
love	$\frac{2+1}{10+10} = \frac{3}{20}$	$\frac{0+1}{5+10} = \frac{1}{15}$
rubbish	$\frac{0+1}{10+10} = \frac{1}{20}$	$\frac{0+1}{5+10} = \frac{2}{15}$

TABLE VI. THE CONFUSION MATRIX TO EVALUATE MODEL PERFORMANCE.

Prediction	**Class Data**	
	Positive	**Negative**
positive	TP	FP
negative	FN	TN

Note:
True Positive (TP): #correct positive prediction
True Negative (TN): #correct negative prediction
False Positive (FP): #incorrect positive prediction
False Negative (FN): #incorrect negative prediction

IV. RESULTS AND DISCUSSIONS

From 500 crawled Tweeter data, we only used 248 instances contained *#lombok* and *#wonderfullombok* hashtags. This reduction was because there were tweets that only contained hashtags so that they were empty after the pre-processing steps and were excluded. Moreover, we also excluded the retweeted tweets. This section will describe the experimental results as follow.

A. Feature Selection

Each feature extracted from the training data are expected to highly contributes to the prediction but they are not contribute evenly. Therefore, MI helps to select the highly contribution features. We calculated their MI score for each feature from each category following Equation 2. We got about 610 features from the extracted data.

To get the most informative features, we selected n highest score features. To do this, we sorted the features descendingly and we experimented n on 10, 20, 30, 40, and 50 features. Our experiments showed the results were optimum when we selected 10 features. Table VII shows the MI score for ten selected features.

TABLE VII. TEN SELECTED FEATURES AND THEIR MI SCORE.

Features	$[N_{00}, N_{01}, N_{10}, N_{11}]$	**MI Score**
paradise	[4.0, 223.0, 2.0, 19.0]	0.009
sunset	[12.0, 215.0, 0.0, 21.0]	0.006
happy	[12.0, 215.0, 0.0, 21.0]	0.006
beauty	[11.0, 216.0, 0.0, 21.0]	0.006
enjoy	[11.0, 216.0, 0.0, 21.0]	0.006
dirty	[15.0, 6.0, 0.0, 227.0]	0.256
rubbish	[4.0, 17.0, 0.0, 227.0]	0.059
bad	[3.0, 18.0, 1.0, 226.0]	0.032
garbage	[2.0, 19.0, 0.0, 227.0]	0.029
sad	[2.0, 19.0, 0.0, 227.0]	0.029

B. Classification

As explained in Subsection III-D, the NB classification has two steps: training and prediction. In the first step, we calculated the class and selected features probabilities. Figures 4 and 5 show the classes and selected features probabilities.

In this classification step, the model looked for words that the same as ones in training vocabulary to obtain their probability scores. Then each score was multiplied to each class (positive and negative) probability score.

```
Number of positive class = 227.0
Number of data = 248.0
Probability of positive class :
227.0/248.0=0.9153225806451613
===================================
Number of negative class = 21.0
Number of  data = 248.0
Probability of negative class :
21.0/248.0=0.0846774193548387
```

Fig. 4. Examples of negative and positive V_{MAP} score of a feature.

```
Positive feature probability :
============================
happy (14.0+1)/(972.0+610.0)=0.009481668773704172
beauty (11.0+1)/(972.0+610.0)=0.007585335018963337
sunset (12.0+1)/(972.0+610.0)=0.008217446270543615
dirty (0.0+1)/(972.0+610.0)=6.321112515802782E-4
rubbish (0.0+1)/(972.0+610.0)=6.321112515802782E–4
...
Negative feature probability :
=============================
happy (0.0+1)/(108.0+610.0)=0.001392757660167131
beauty (0.0+1)/(108.0+610.0)=0.001392757660167131
sunset (0.0+1)/(108.0+610.0)=0.001392757660167131
dirty (14.0+1)/(108.0+610.0)=0.020891364902506964
rubbish (5.0+1)/(108.0+610.0)=0.008356545961002786
...
```

Fig. 5. Feature's score of a tweet example.

The multiplied scores were compared so that the pair with the highest probability was chosen as the class prediction following Equation 5. Figure 6 shows the calculation for the prediction of the examples in Figures 4 and 5. In this figure, the corresponding scores are written in the same colour.

```
monday happy bless god amazing thing week
Words in the vocabulary =
happy
Positive = 0.9153225806451613 ∗ 0.009369144284821987
         = 0.008575789325220124
Negative = 0.0846774193548387 ∗ 0.0013850415512465374
         = 1.1728174425877937E–4
Result = POSITIVE
=========================================================
loang baloq bad kinda dirty
Words in the vocabulary =
dirty
Positive = 0.9153225806451613 ∗ 0.0006321112515802 78
         = 5.78585702051303E–4
Negative = 0.0846774193548387 ∗ 0.0316515530288977
         = 0.002680172
Result = NEGATIVE
=========================================================.
```

Fig. 6. The class prediction calculation.

C. Evaluation

The accuracy variation of using and not using MI feature selection are expressed in Table VIII. The table shows that, mostly, the accuracy, precision, and recall of using MI feature improves compared to ones when the feature is absent. The average improvements are shown in Figure 7. This table also indicates that the recall is significantly higher for the prediction using MI feature.

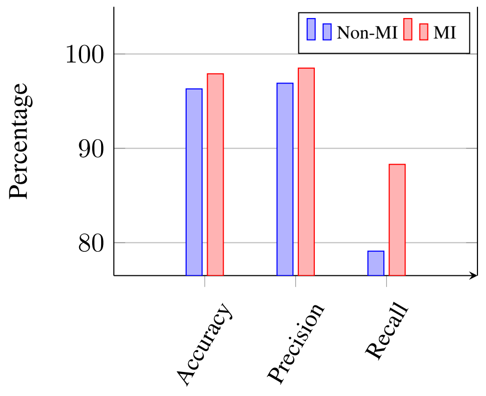


Fig. 7. The average accuracy, precision, and recall of the MI feature existent.

TABLE VIII.
ACCURACY (ACC), PRECISION(PREC), AND RECALL(REC) FOR MI AND NON-MI FEATURES IN PERCENTAGE.

Fold	Data		Non MI			MI		
	Training	Test	Acc	Prec	Rec	Acc	Prec	Rec
1	223	25	100	100	100	100	100	100
2	223	25	96	97	75	96	97	75
3	224	24	95	97	75	95	97	75
4	223	25	96	97	75	96	97	75
5	224	24	96	97	75	100	100	100
6	223	25	96	97	75	100	100	100
7	223	25	92	92	50	96	97	75
8	223	25	100	100	100	100	100	100
9	223	25	100	100	100	100	100	100
10	223	25	92	92	66	96	97	83

D. Classification Time

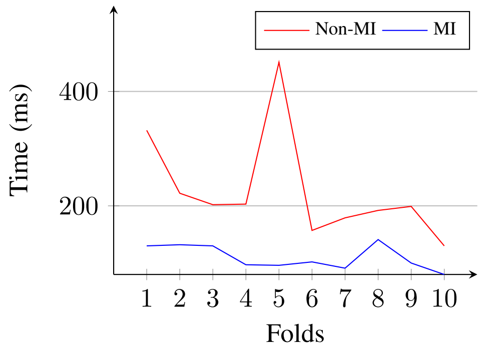


Fig. 8. The classification time variation of the MI feature existent.

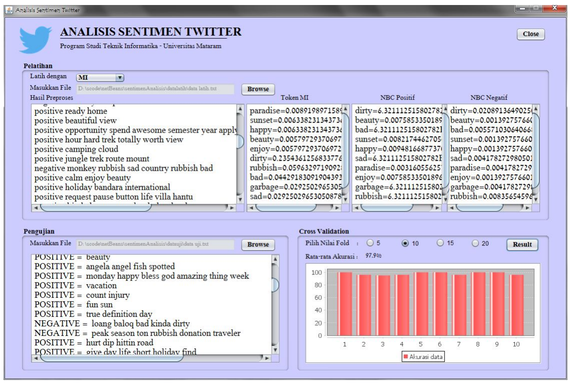
TABLE IX.
THE AVERAGE CLASSIFICATION TIME AND ONE TWEET CLASSIFICATION TIME IN MILI SECONDS.

Item	Non-MI	MI
Classification time average	226.7 ms	109.9 ms
Classification time for 1 tweet	90.68	4.39

Figure 8 shows that the training time of using MI feature is shorten than one without MI feature. To be precise, we provide additional information shown in Table IX. This table reports the average classification time for all the ten folds. Therefore, the time difference of using and not using MI feature for one tweet is 86.28 mili seconds.

E. User Interface

To show the classification results easily, we built a graphical user interface as shown in Figure 9. From this interface, we may see from the left to right, the original features extracted from each tweet before the pre-processing steps, the selected features' MI score, and their probabilities for positive and negative classes in the half top of the screen. The left bottom corner shows the features of evaluated tweets. The system then calculated the probability of these tweets before they were compared to select the highest score. The highest score were used to predict the class. Then the cross validation calculation is given in the right bottom corner.


Analisis Sentimen Twitter
ANALISIS SENTIMEN TWITTER
Program Studi Teknik Informatika - Universitas Mataram
Close
Browse
Browse
Result
Akurasi data

Pelatihan

Latih dengan: MI

Masukkan File: D:\scode\netBeans\sentimenAnalisis\datalatih\data latih.txt

Hasil Preproses

```
positive ready home
positive beautiful view
positive opportunity spend awesome semester year apply
positive hour hard trek totally worth view
positive camping cloud
positive jungle trek route mount
negative monkey rubbish sad country rubbish bad
positive calm enjoy beauty
positive holiday bandara international
positive request pause button life villa hantu
```

Token MI

```
paradise=0.00891989715
sunset=0.00633823134373
happy=0.00633823134373
beauty=0.0057972937069
enjoy=0.005797293706972
dirty=0.235436125683377
rubbish=0.059632971909
bad=0.044291830919043
garbage=0.0292502965305
sad=0.029250296530508
```

NBC Positif

```
dirty=6.32111251580278
beauty=0.0075853350189
bad=6.3211125158027821
sunset=0.008217446270
happy=0.009481668773
sad=6.3211125158027821
paradise=0.0031605562
enjoy=0.007585335018
garbage=6.321112515802
rubbish=6.321112515802
```

NBC Negatif

```
dirty=0.020891364902
beauty=0.00139275766
bad=0.0055710306406
sunset=0.0013927576600
happy=0.0013927576600
sad=0.004178272980501
paradise=0.004178272
enjoy=0.0013927576601
garbage=0.004178272
rubbish=0.00835654596
```

Pengujian

Masukkan File: D:\scode\netBeans\sentimenAnalisis\datauji\data uji.txt

```
POSITIVE =  beauty
POSITIVE =  angela angel fish spotted
POSITIVE =  monday happy bless god amazing thing week
POSITIVE =  vacation
POSITIVE =  count injury
POSITIVE =  fun sun
POSITIVE =  true definition day
NEGATIVE =  loang baloq bad kinda dirty
NEGATIVE =  peak season ton rubbish donation traveler
POSITIVE =  hurt dip hittin road
POSITIVE =  give day life short holiday find
```

Cross Validation

Pilih Nilai Fold : 5 [✓ 10] 15 20

Rata-rata Akurasi : 97.9%

Fig. 9. The system user interface.

V. Conclusion and Future Works

Our experiments was about the tweeter sentiment analysis. We evaluated the experiments of using and not using *Mutual Information* (MI) feature selection. The results show that the MI feature selection improves the accuracy from 96.2 % to 97.9%. It also contributes to increase the precision and recall. Moreover, the classification time also reduced by 51.52%.

For the future works, we would like to increase the number of the Twitter data and observe on using different classification and selection features. It is also interesting to work on different language and different topic as the pre-processing steps and important features might be different.

References

[1] B. Liu, *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.

[2] V. Narayanan, I. Arora, and A. Bhatia, "Fast and accurate sentiment classification using an enhanced naive bayes model," in *Proceedings of the 14th International Conference on Intelligent Data Engineering and Automated Learning — IDEAL 2013 - Volume 8206*, ser. IDEAL 2013. New York, NY, USA: Springer-Verlag New York, Inc., 2013, pp. 194–201. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-41278-3_24

[3] C. D. Manning, P. Raghavan, and H. Schutze, *An Introduction to Information Retrieval*. Cambridge University Press, 2019.

[4] L. Utami, "Analisis sentimen opini publik berita kebakaran hutan melalui komparasi algoritma *support vector machine* dan *k-nearest neighbour* berbasis particle swarm optimization," *Jurnal Evolusi*, vol. 13, pp. 2527–6514, 2017.

[5] S. Ernawati, "Penerapan *particle swarm optimization* untuk seleksi fitur pada analisis sentimen," *Jurnal Evolusi*, vol. 4, 2016.

[6] I. Darma, R. Perdana, and Indriati, "Analisis sentimen televisi pada twiter menggunakan *support vector machine* dan algoritma genetika sebagai metode seleksi fitur," *Jurnal Evolusi*, vol. 2, pp. 998 – 1007, 2017.

[7] N. Gupta and S. Parveen, "Efficient sentiment analysis using optimal feature and bayesian classifier," *International Journal of Computer Applications*, vol. 145, 2016.

[8] V. Garcia, J. Sanchez, and R. Mollineda, "An empirical study of the behaviour of classifiers on imbalanced and overlapped data sets," 2007, lecture Notes in Computer Science.

[9] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support machines to imbalanced dataset," in *Proceedings of the 15th European Conference on Machine Learning*, 2004, pp. 39 – 50.

[10] N. Sobran, A. Ahmad, and I. Z., "Classification of imbalanced dataset using conventional naïve bayes classifier," in *Proceeding of the International Conference on Artificial Intelligence in Computer Science and ICT*, 2013, pp. 978 – 967.