Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM)
Vol 2, No.1, March 2019, pp. 37 - 44
p-ISSN: 2614-3372 | e-ISSN: 2614-6150

Spam Classification on 2019 Indonesian President Election Youtube Comments Using Multinomial Naïve Bayes

[1]Jonathan Radot Fernando, [2]Raymond Budiraharjo, [3]Emeraldi Haganusa,
[1,2,3]Departement of Human Computer Interaction, Surya University
Email: [1]jonathan.radot97@gmail.com, [2]vengeancenator@gmail.com, [3]kuuhaku63@gmail.com

Article Info

Article history:
Received Jul 17th, 2018
Revised Sept 18th, 2018
Accepted Jan 06th, 2019

Keyword:
Bag-of-words
Multinomial Naïve-Bayes
Spam
Text classification
Youtube Comments

ABSTRACT

Text classification are used in many aspect of technologies such as spam classification, news categorization, Auto-correct texting. One of the most popular algorithm for text classification nowadays is Multinomial Naïve-Bayes. This paper explained how Naïve-Bayes assumption method works to classify 2019 Indonesian Election Youtube comments. The output prediction of this algorithm is spam or not spam. Spam messages are defined as racist comments, advertising comments, and unsolicited comments. The algorithms text representation method used bag-of-words method. Bag-of-words method defined a text as the multiset of its words. The algorithm then calculate the probability of a word given the class of spam or not spam. The main difference between normal Naïve-Bayes algorithm and Multinomial Naïve-Bayes is the way the algorithm treats the data itself. Multinomial Naïve-Bayes treats data as a frequency data hence it is suitable for text classification task.

Corresponding Author:
Second and Third Author,
Departement of Human Computer Interaction,
Surya University,
Jl. M.H.Thamrin, Tangerang, Banten - Indonesia.
Email: admission@surya.ac.id

DOI: http://dx.doi.org/10.24014/ijaidm.v2i1.6445

1. INTRODUCTION

Indonesia is a large country that holds more than 200 million people and 20 different ethnic groups [1]. The year of 2019 became one of the major events in Indonesia. President election is one of the major events in 2019. People attend to support their soon to be president in their own possible way. Rights became a controversial discussion whether the rights holding the peoples freedom of expression or allowing the situation turned into riot. Many chaotic incident caused by president election. Some of them demanded equity on rights and some of them demanded the action to be supressed [2].

Internet has grown widely in the past 15 years. Many services proceed towards internet for users convenience. One of services in internet that has grown significantly is Youtube. Youtube provides video sharing service in the internet. It attracts 34% of internet users with the the average of 1 billion hours watch time per day [3]. Youtube itself has been giving monetization to content creators based on its views and subscriber. With the rapid growth of Youtube, many media proceed publication through Youtube. CNN Indonesia is one of news media that publicates their contents through Youtube. Youtube not only received positive impact on its growth in the modern era but also negative impact as well. One of the negative impact is spamming habits.

The term spam mainly used to describe Unsolicited Commercial Email (UCE) or Unsoliticed Bulk Email (UBE) [2]. Spamming itself has made negative impact on the internet such as disturbing internet users by invading their privacy information on the internet and using them as a commercial opportunity. Not only by emails, spamming are now common in many internet based platform such as blog, websites, even on Youtube itself. Youtube has dealt with this issue and made a system to detect and delete spam messages. This feature has been used by many youtubers and have prooven to be effective by looking at recent youtube comments with minimal spam occurance. From the main definition, this paper define spam as racist comments and advertising comments in Youtube.

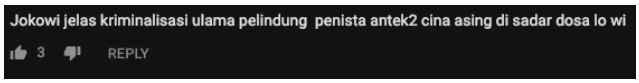


Figure 1. Sample spam messages not filtered by youtube

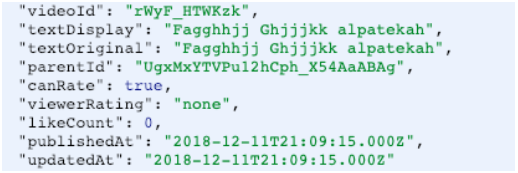


Figure 2. Sample filtered spam messages using youtube api

Youtube spam system has its disadvantages as well. The system sometimes doesn't detect racist comment in indonesian language such as in Figure 1. Youtube still developing its spam filter system to support various language. Youtube feeds its training data from flagged comments. Different language has different grammar and the way of using their own words. Indonesian language have its own slang words such as deh, sih, dong. Indonesian language also have affix and it can cause different meaning with each use [3]. The system itself have to detect those differences to make a accurate predictions. So the input to the system have to be precise about which word and what affix they used.

Many news media on Youtube covered topics about 2019 Indonesia President Election. One of them is CNN Indonesia. The comments itself was promotions about the candidates. Sometimes people promote their favorite candidate by disrespecting other candidate. Sometime they use harsh words even racist comments. This phenomena can create conflict between two sides from the internet and taken to real life. The problem lead to a solution and that is spam classifier.

Spam classifier use text classification method. Many text classification has been implemented such as news categorization, auto-correction, even spam classification. text classification represents data using text representation method. When using probabilistic model, there are two method of representation bag-of-words and N-gram method. The difference is when using bag-of-words, each word represents one unit. When using N-grams each word will consider the words that occured before and after the word itself. Bag-of-of words model treats a word as independen vice versa N-gram model treats a word as dependen [4].

Probabilistic model commonly used in this type of case. This paper use probabilistic model called Multinomial Naïve-Bayes. Based on naïve-bayes algorithm, Multinomial naïve bayes count the occurance of a word being a spam or not and make a probability whether the word is spam or not. This algorithm is easy to use and one of the best algorithm for this case [5].

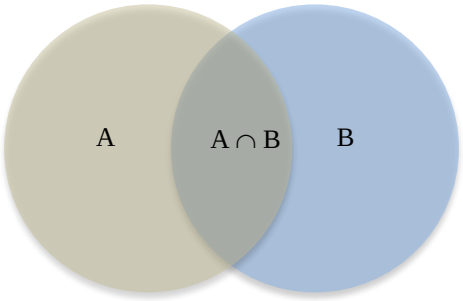


Figure 3. Probabilistic model

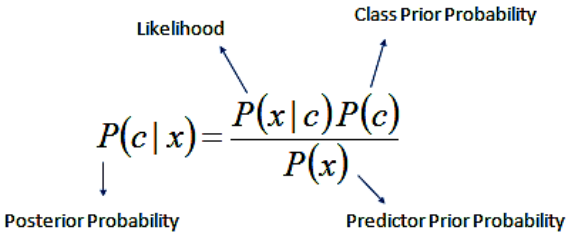
Probabilistic model theory based on Figure 3 which is venn diagram. The problem that appeared in the Figure 3 for probabilistic model is value of probability A in class B and probability B in class A. Probability of A ∩ B can be called as probability of A in B or B in A. Given the probability of two condition in each area can create equation in Equation 1. From Equation 2, can be derived into new equation in Equation 3. Since the probability of (A ∩ B) is the same with (B ∩ A) we can nominator of Equation 1 and create a functional equation of Equation 3 called Naïve Bayes Algorithm.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (1)$$

$$P(A \cap B) = P(A|B).P(B) = P(B \cap A) = P(B|A).P(A) \quad (2)$$

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \quad (3)$$

Naïve-bayes algorithm uses probability to calculate predictions. Naïve-bayes is one of *supervised learning* method. This method use training data with defined label to identify key value or variable for prediction [6]. Naïve bayes identify key value or value by calculating the variable or value occurance in data resulting probability of spesific element. To calculate probabilities (posterior probability) in naïve-bayes, the algorithm must define prior probability, likelihood, predictor prior probability. Equation 3 describes that if the data is multi-feature, naïve-bayes algorithm can create chain rule of probabilities of each input in likelihood and balanced by creating chain rule in predictor prior probability.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure 4. Chain rule Naïve Bayes algorithm

Naïve-bayes algorithm can predict nominal type values with multiple features, but the case in this study doesnt match the input type of naïve-bayes. The data in Table 1 considers one feature (content) that contains many feature (words). This case study needs an algorithm that can calculate features based on its frequencies of occurance in dataset. The algorithm that is suitable in this case is multinomial naïve-bayes is one of the most popular algorithm to be used in text classification. Each feature in multinomial naïve-bayes ia words from dataset [7].

$$P(W_I|c) = \frac{count\ (w_{i,c})}{\sum_{w \in V} count(w,c)} \quad (4)$$

$$P(W_I|c) = \frac{count\ (w_{i,c}) + 1}{(\sum_{w \in V} count(w,c) + 1)} = \frac{count\ (w_{i,c})}{(\sum_{w \in V} count(w,c)) = |V|} \quad (5)$$

Equation 4 is the likelihood equation in multinomial naïve-bayes. Where count(w_i,c) is the occurance of word i in class c. The denominator is the sum of words that is known by the vocabulary in class c. Multinomial naïve-bayes implements bayes theorem which calculates the probability of a word given a class as their likelihood. There is a problem when implementing Equation 4. The problem is if the system received new words, it will create a probability of 0 and affected the calculation. Laplacian smoothing helps the algorithm to not be affected by 0 probabilities. Laplacian smoothing in multinomial naïve-bayes adds 1 as a constant to nominator. By that, the algorithm will provide probability value in likelihoods and not damaging the result value for predictions [8]. In Equation 5 used laplacian smoothing and in denominator added by the number of elements in vocabulary.

For prediction the system calculate the product of likelihood using chain rule in Equation 4 and multiply by the prior probability in Equation 6. To calculate the prior probability the algorithm needs the number of elements in class c divided by the number of elements in the dataset. Then the algorithm predicts the outcome by choosing the max probability between spam probability and not spam probability of the sentece in each respective classes. The prediction equation is on Equation 4 where the algorithm choose the max value of the product of likelihood multiplied by the prior probability. The algorithm doesn't include the predictor prior probability in Figure 4 because the prediction is derived without needing the predictor prior probability. Since the prediction is a comparison between two classes that has the same predictor prior probability.

$$P(c) = \frac{N_c}{N_{doc}} \tag{6}$$

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(d|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}} \tag{7}$$

2. RESEARCH METHOD

The Data used in this paper is from CNN Indonesia YouTube video titled "Jokowi vs Prabowo Episode 2 di Pilpres 2019". Since YouTube has implemented spam filtering in their system, this paper use YouTube API to retrieve some spam comment and use data dummies to teach the system whether the word is spam or not. This paper use 265 training data and 65 testing data to calculate accuracy, precision, recall, and f-measure. Data labels are defined by the word that determine advertisement or racism such as "subscribe", and "cina". Affix has been splitted for data preprocessing. The reason the affix being splitted is because it can change the words meaning [3] and can affect its probabilities in the system.

Table 1. Sample dataset

Content	Label
jokowi 2 periode yang setuju like	1
ke lihat an mem bangun nya ke lihat an juga hutang nya negara cebong ga mikir pikiran nya cetek	0
like kalau setuju udah lama berita nya	1
subscribe and like video ini ya	1
salam dari borneo kalah menang aku dukung prabowo	0

In code, first the inputs need to split the data content in Table 1 into words [7]. In this process the system also calculate the number of data in spam, the number of data not in spam, the vocabulary, and the total data. The words then inserted into *python* dictionary in element spam or not spam. The number of spam is calculated when the conditional requirements of spam is met. Same goes to not spam class, the number of not spam elements is calculated when the conditional requirements of not spam is met. The vocabulary is calculated when the words is not faund in the dictionary. And the sum data is added by 1 each iteration. This data is needed for future process in the algorithm to calculate prior probability in predict function Figure 8 and likelihood in training function Figure 7.

```
# split_text_train(dataset):
##    for data in each row dataset:
      split data into words
      splitted.append(words)
      calculate occurance of spam (1) from splitted
      calculate occurance of not spam(0) from splitted
      insert data dictionary spam
      insert data dictionary not spam
##      if data not in data dictionary:
          vocabulary += 1
      sum data += 1
##    return splitted, data dictionary, occurance of spam, occurance of not spam, vocabulary, sum data
```

Figure 5. Pseudo code: splitting the input data

Then each data in the main dictionary is transformed into feature. The system create two new *python* dictionary consists of spam dictionary and not spam dictionary. Each word declared as a feature and given 0 value as start value. The return for this function is spam dictionary and not spam dictionary for later processing. Later in training process each word value will be updated by word occurance in each respective classes.

```
# count_class_freq(data dictionary):
##    for word in data dictionary[spam]:
##       if word not in word dictionary spam:
             add to word dict spam[spam]

##    for word in data dictionary[not spam]:
##       if word not in word dictionary not spam:
             add to word dict not spam[not spam]
##     return word dict spam, word dict not spam
```

Figure 6. Pseudo code: Transform each data into features

In training function we calculate the likelihood of each word given spesific class in this case spam or not spam. the system first count the occurance of spesific word in each classes. Training function needs test data to add probability of that word or learn new words by giving laplace smoothing to the algorithm. Then calculate the likelihood for each word in spam and not spam. Alpha in the pesudo code represents laplace smoothing so new data can produce non-zero probabilities [8]. The likelihood then stored in two new directories based on spam or not spam classes.

```
# training(dataset, test data, data dictionary, word dict spam, word dict not spam, vocabulary, alpha =
 1):
  _alpha = laplace smoothing (prevents probabilities being 0)_

##    for word in dataset:
        check if word exist in word dict spam (yes = spam word[word] + 1 | no = spam word [word] +0)
        check if word exist in word dict not spam (yes = not spam word[word] + 1 | no = not spam word [wo
rd] +0)

##    for word in test data:
        check if word exist in word dict spam (yes = spam word[word] + 1 | no = spam word [word] +0)
        check if word exist in word dict not spam (yes = not spam word[word] + 1 | no = not spam word [wo
rd] +0)

##    for word in spam_words:
        prob_spam_words[word] = (spam words[word] + alpha)/(num of word in spam + vocabulary)

##    for word in not_spam_words:
        prob_not_spam_words[word] = (not spam words[word] + alpha)/(num of word in not spam + vocabulary)

##    return prob_spam_words, prob_not_spam_words
```

Figure 7. Pseudo code: Training

In predict function the system called the value inside likelihood dictionary from training process. First the system implements the chain rule to calculate the product of likelihood that wants to be tested. The product of likelihood then multiplied by prior probability of spesific class uses equation in Equation 6. Then the value of probability is stored in spam meter and not spam meter. Finally find the max value between spam meter and not spam meter using argmax function.

```
# predict(test data, prob_spam_words, prob_not_spam_words, occurance of spam, occurance of not spam, sum
 data):

##    for word in test data:
        check if word exist in word dict spam (yes = prob spam word[word] | else = prob spam word [word])
        check if word exist in word dict not spam (yes = prob not spam word[word] | else = prob not spam
 word [word])

##    for value in probability spam:
        spam meter = spam meter * value

      spam meter = spam meter * (occurance of spam/sum data)


##    for value in probability not spam:
        not spam meter = not spam meter * value

      not spam meter = not spam meter * (occurance of not spam/sum data)

      if spam meter > not spam meter= prediction(spam)
      if not spam meter > spam meter= prediction(not spam)

      prediction value = argmax(spam meter, not spam meter)

##    return prediction, prediiction value
```

Figure 8. Pseudo code: Predict

The method used to calculate accuracy, precision, recall, and f-measure is confusion matrix. In confusion matrix the system calculate the difference of label before prediction and after prediction. There are four variables need to be calculated, *true positive* (TP) is when before prediction label is spam and after prediction still spam, *true negative* (TN) is when before prediction label is not spam and after prediction still not spam, *false positive* (FP) is when before prediction label is not spam and after prediction shows spam, *false negative* (FN) is when before prediction label is spam and after prediction shows not spam[9].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F\_Measure = \frac{2TP}{2TP + FN + FP}$$

3. RESULTS AND ANALYSIS

After running the program with the multinomial naïve-bayes algorithm with 265 training data and 65 testing data, result and analysis were calculated using performance evaluation method. Accuracy is over 80% meaning that the prediction success rate is over 4 out of 5 which is decent. Accuracy sometimes not the most dependable way to calculate the success rate of an algorithm because if there is an imbalance between class the accuracy takes the major number resulting good accuracy with zero predictive power, This phenomena is called *accuracy paradox* [10]. The precision is 80% meaning that the positive predictive value is 4 out 5. Based on the precision, the predictive power of multinomial naïve-bayes algorithm using this spesific dataset is decent. Recall value is 97% which is high meaning that this algorithm is strict when predicting spam data. The downside is sometimes data prediction can be false as a spam data while the original data doesn't have spam elements. The F-measure shows the balance value between precision and recall, it can detect the imbalance of prediction. The result of F-measure is 88% which is balance.

Table 2. Performance evaluation result

Performance Evaluation	Result
Accuracy	85.2459016393%
Precision	80.487804878%
Recall	97.0588235294%
F-Measure	88.0%

For further analysis ranked value of spam and not spam word is needed to show how spesific word will affect the prediction result. The ranked words will be sorted and taken the top 10 highest probability value in each respective classes. Some of the top 10 highest probability words has the same probability and from the graph it can derived comparison value whether the probability is higher in one of the class. The probability will be presented in table form.

Table 3. Top 10 highest probability spam words

Words	Probability
like	0.08559153998678123
indonesia	0.04130865829477859
Presiden	0.023463317911434238
jokowi	0.020819563780568408
prabowo	0.017184401850627893
pak	0.012227362855254461
subscribe	0.011235955056179775
rakyat	0.010905485789821546
ini	0.009583608724388631
kami	0.008922670191672175

Table 4. Top 10 highest probability not spam words

Words	Probability
like	0.06815789473684211
indonesia	0.03289473684210526
Presiden	0.01868421052631579
jokowi	0.016578947368421054
prabowo	0.01368421052631579
2019	0.011052631578947368
pak	0.009736842105263158
rakyat	0.008684210526315789
ini	0.007631578947368421
kami	0.007105263157894737

From the ranked probability of spam and not spam words, it is clear why the recall value are high. The probability of spam of the same words is higher than the probability of not spam. The problem that occured might be the cause of not enough training data to define the word itself. The other problem is indonenesian language has broad usage and meaning so it needs more complex system to define each word.

Other problem might be indonesian language is dependent with other words that are connected with the word itself. So it might have better result when using N-gram text representation method rather than bag-of-words text representation method.

```
['Jokowi', 'bantu', 'cina', 'tak', 'pendidik', 'an']

PREDICTION:  [1]
probability measure:  5.64653666801e-18
```

Figure 9. Test data: spam prediction result

```
['politik', 'indonesia', 'perlu', 'benah']

PREDICTION:  [0]
probability measure:  9.72794623904e-12
```

Figure 10. Test data: not spam prediction result

4. CONCLUSION

Indonesian language need much more complex preprocessing to define a better meaning of the words that are used in various sentences. Since the language has some complex system, bag-of-word methods is not a very good choice, while N-gram method is possibly a better choice because of the consideration of dependent words. As for the algorithm, multinomial naïve-bayes is one of the most effective and used algorithm to classify text. This can be seen by the performance evaluation result which is decent. Multinomial naïve-bayes needs a large dataset to define the data so it can be accurate.

Future work will discuss multinomial naïve-bayes using N-gram text representation method and will be compared with bag-of-words text representation method result. Bernoulli naïve-bayes is one of the popular text classification method especially for short text classification. For future works, multinomial naïve-bayes will be compared with bernoulli naïve-bayes method to analyze performance evaluation between the two algorithm.

ACKNOWLEDGEMENTS

The author thank to Surya University especially to Department of Human Computer Interaction (HCI) for supporting author by vigour and prayer in this work.

REFERENCES

[1] Badan Pusat Statistik . 2102032. *Kewarganegaraan, Suku Bangsa, Agama, dan Bahasa Sehari-hari Penduduk Indonesia*. Jakarta. Badan Pusat Statistik Jakarta-Indonesia. 2011.
[2] David S, Craney G. “How Do I Stop Spam”. 2001: 1-5.
[3] Quinn, G. The Learner's Dictionary of Today's Indonesian. PhD Thesis. Sydney: Australian National University; 2001.
[4] George S, Joseph S. Text Classification by Augmenting Bag of Words (BoW) Representation with Co-occurance Feature. *IOSR Journal of ComputerEngineering*. 2014; 16(1): 34-38.
[5] Alberto T C, Lochter J V, Almeida T A. Tubespam: Comment Spam Filtering on YouTube. São Paulo: Federal University of São Carlos; 2015.
[6] Vulandari R. T. Data Mining Teori dan Aplikasi Rapidminer. 1. Yogyakarta: Penerbit Gava Media. 2017: 7.
[7] Kibriya A M, Frank E, Pfahringer B, Holmes G. Multinomial Naïve Bayes for Text Categorization Revisited. *AI 2004 LNAI 3339*. 2004: 488-499
[8] Yuan Q, Cong G, Thalmann N M. Enhancing Naïve Bayes with Various Smoothing Methods for Short Text Classification. Singapore: Nanyang Technological University; 2012
[9] Sokolova M, Japkowicz N, Szpakowicz S. *Beyond Accuracy, F-score and ROC: a Family of Discriminant Measures for Performance Evaluation*. Australian Joint Conference on Artificial Intelligence. Hobart. 2006; 19: 1015-1021.
[10] Akosa J S. Predictive Accuracy: A Misleading Performance Measure for Highly Imbalanced Data. Oklahoma: Oklahoma State University; 2017.

BIBLIOGRAPHY OF AUTHORS

Jonathan Radot Fernando, M, College Student (2015) in Department of Human Computer Interaction, Surya University. Machine Learning and Deep Learning enthusiast.

Raymond Budiraharjo, M, College Student (2015) in Department of Human Computer Interaction, Surya University. Interaction Design and User Experience enthusiast.

Emeraldi Haganusa, M, College Student (2016) in Department of Human Computer Interaction, Surya University. Interaction Design and User Experience enthusiast.