Comparison of the Effect of Word Normalization on Naïve Bayes Classifier and K-Nearest Neighbor Methods for Sentiment Analysis

Novrido Charibaldi[1], Atania Harfiani[2], Oliver Samuel Simanjuntak[3]
[1,2,3]*Informatics Department, Universitas Pembangunan Nasional Veteran Yogyakarta, Indonesia*
[1]novrido@upnyk.ac.id, [2]ataniaharfiani2@gmail.com(*), [3]oliver.samuel@upnyk.ac.id

Received: 2023-10-12; Accepted: 2023-11-19; Published: 2023-12-03

Abstract— In the pre-processing stage of sentiment analysis, there are several essential steps, one of which is word normalization, which is converting non-standard words into standard words. However, some research on sentiment analysis generally does not go through the word normalization stage, which can affect accuracy. This study aims to compare the effect of word normalization on the Naive Bayes Classifier and K-Nearest Neighbor methods for sentiment analysis of public opinion on the Agency Social Security Administrator for Health (*BPJS Kesehatan*). Gathering the data, labeling it, pre-processing it with two different scenarios, word weighting it with TF-IDF, classifying it using Naive Bayes Classifier and K-Nearest Neighbor, and lastly computing the accuracy of the Confusion Matrix are the steps that are involved. As a result of these discovered fact, the most superior accuracy results are obtained by the Naive Bayes Classifier method 1st scenario, namely by using word normalization at the pre-processing stage and getting an accuracy of 87.14%. This research shows that the Naive Bayes Classifier method with word normalization produces better accuracy, precision, recall, and F1-score.

Keywords— Sentiment Analysis; Word Normalization; Naïve Bayes Classifier; K-Nearest Neighbor; *BPJS Kesehatan*.

I. Introduction

The pre-processing stage is an essential step in sentiment analysis, the translation of raw, unstructured data into a format that is easier to work with and for processing is the responsibility of this component. Earlier than the processing, there are several essential steps. One is word normalization, which is converting non-standard words into standard words according to the Kamus Besar Bahasa Indonesia [1]. However, some research on sentiment analysis generally does not go through the word normalization stage, so ambiguity occurs, and the system cannot classify the class correctly, affecting accuracy [2]. Data past the pre-processing stage will then be classified using machine learning methods.

Sentiment analysis can utilize a range of machine-learning techniques, including research-based methodologies [3], which use the Affective model method and a lexicon dictionary base, namely the Russell Circumplex Model, conducted two kinds of experiments, namely the manual process resulting in an accuracy of 81.05%. The Affective Models method has succeeded in achieving 83.4% accuracy. However, this research system still needs to add a word normalization stage so that there are no ambiguities in the sentence that result in misclassification.

Furthermore, [4] researched sentiment analysis by comparing Naïve Bayes and Naïve Bayes + SMOTE + Adaboost methods on Twitter social media. They resulted in Naïve Bayes method accuracy of 71.68% and Naïve Bayes + Adaboost+SMOTE method accuracy of 69.11%. Nevertheless, this research has to be enhanced with other techniques as the categorization outcomes remain subpar. Other pre-processing needs to be added, such as word normalization. The analysis results match the class, and accuracy improves.

Research on sentiment analysis with other machine learning methods conducted by [5] using Naïve Bayes Classifier with TF-IDF weighting resulted in an accuracy rate of 62%. However, this research needs to be optimized by comparing other classification methods because the accuracy is still low, as well as adding other pre-processing such as word normalization to improve the accuracy results.

Another study by [6] also compared three methods, namely, K-Nearest Neighbor, Naïve Bayes, C 4.5, and combined the three methods with the Ensemble Vote algorithm method. The Naïve Bayes method has an accuracy result of 81.63%, K-Nearest Neighbor of 71.83%, and C 4.5 of 65.37%. However, this research needs to add other pre-processing, such as word normalization, to correctly classify tweets according to their class.

Because of this, the focus of the research will be on contrasting the K-Nearest Neighbor algorithm with the Naive Bayes Classifier, according to [7], they are simple and easy to implement. Another reason is that the Naive Bayes Classifier method, according to research [8], can produce relatively high accuracy, and the K-Nearest Neighbor method, according to study [9], also has relatively high accuracy. The influence of word normalization is needed because, in research [10], it is known that adding the word normalization stage to pre-processing can affect accuracy, so accuracy has increased from 86.67% to 91.67%. This research aims to compare the effect of word normalization on the Naive Bayes Classifier and K-Nearest Neighbor methods.

II. Research Methodology

The purpose of this study was to investigate the effect that word normalization has on the accuracy of the Naive Bayes Classifier and the K-Nearest Neighbor algorithms. The study consisted of four different test scenarios. For the purpose of

DOI : https://doi.org/10.25139/inform.v9i1.7111

conducting the sentiment analysis experiments, a total of ten hundred fifty pieces of data were extracted from comments made on Instagram and Twitter. Following that, the data was divided into two parts: eighty percent for training and twenty percent for testing. There were 840 training data points and 210 test data points that were produced by this technique.

The first seven stages of the research method consist of the following: data collection, labeling, pre-processing scenarios, TF-IDF weighting, Naïve Bayes Classifier sentiment classification, K-Nearest Neighbor sentiment classification, and testing using confusion matrix to produce accuracy, precision, recall, and F-1 score information. During the pre-processing step, there are two distinct outcomes that might occur, namely 1st scenario pre-processing includes case folding, deleting emojis, cleaning data, deleting repetitive words, tokenization, word normalization, stemming, stopwords removal, 2nd scenario pre-processing includes case folding, deleting emojis, cleaning data, deleting repetitive words, tokenization, stemming, stopwords removal and 2nd scenario pre-processing includes case folding, deleting emojis, cleaning data, deleting repetitive words, tokenization, stemming, stopwords removal.

The research methodology flows in Figure 1. This approach is comprised of seven steps, which are as follows: web scraping, labeling, pre-processing, TF-IDF, Naïve Bayes Classifier, K-Nearest Neighbor, and Confusion Matrix.

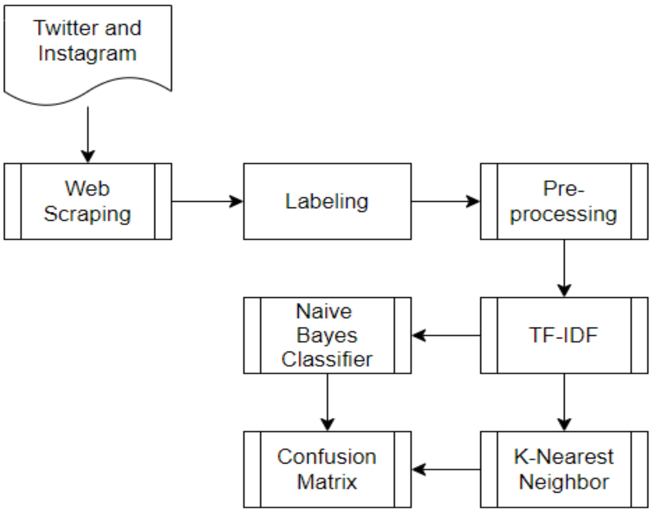

Figure 1. Research Methodology

A. *Data Collection*

Data collection in this study comes from Twitter and Instagram using the automatic scrapping method using the snscrape library. The snscrape library was available in Python. The data taken are tweets that use the keyword "*BPJS Kesehatan*" taken from January 2021. At this time, many people were pro and contra to new policies or information about *BPJS Kesehatan* conveyed by the government, so many people expressed their complaints about the performance of *BPJS Kesehatan* from Twitter with the amount of data taken as much as 616 data and Instagram as much as 434 data. The total data from Twitter and Instagram is 1050 data. The data consists of positive, negative, and neutral comments, as seen in Table I. Two distinct sets of data have been created: 840 for training purposes and 210 for testing purposes.

TABLE I
DATA COLLECTION

Category	Quantity
Positive	350
Neutral	350
Negative	350

B. *Labeling*

The labeling was done manually by humans and checked for validity by an expert, an Indonesian Language Teacher at SMK Negeri 1 Jember, Achmad Zaenul Ulum, SPd. Gr whether the data belongs to the positive, negative, or neutral class. The purpose of labeling by Indonesian language teachers is so that the results of sentiment class classification are more valid or correct in classifying the class because experts do it.

In Table II, samples of manual data labeling are presented for examination. In the process of data labeling, there were three occurrences that were either good, neutral, or negative that were noted.

TABLE II
LABELING

Comment	Label
Alhamdulillahhhh bpjs sgt mmbantu. Lahiran berkali pke sc. Anak dirwt pun cover bpjs.	Positive
Untuk saat ini yg utama, pelayanan kepada pengguna bpjs yg ke rumah sakit.	Neutral
Ribet, kenapa sudah dapat resep haruske loket BPJS, nya ga berfungsi percuma	Negative

C. *Pre-processing*

Pre-processing is the initial process of text mining, which converts unstructured data into structured data following the required format, namely by exploring, processing, organizing information, and analyzing data [11], which aims for uniformity and ease of reading [12]. This study's pre-processing stage consists of case folding [13], cleaning data, word normalization, tokenizing, stopword removal, and stemming [1].

a) Case Folding: Case folding aims to convert all letters in a text document into lowercase letters [13] to homogenize the characters in the data. Information that contains characters other than letters and numbers will be removed from the database [14].

b) Deleting Emojis: Deleting emojis removes or deletes emoticon symbols in text documents [15]. When writing tweets or comments, people are sometimes inappropriate in using emoticons, for example, when commenting on funny things but using emoticons:" (crying), so emoticons will

DOI : https://doi.org/10.25139/inform.v9i1.7111

interfere in the sentiment analysis process [16]. So that various forms of emoticons are deleted or ignored.

c) Cleaning Data: Cleaning data is a process to remove punctuation (period, comma, exclamation, question, fence, etc.), duplicate data, and special symbols such as username, re-tweet symbol (RT), and URL [1]. Cleaning data aims to make the data processed cleaner and does not contain much noise that can affect classification results.

d) Deleting Repetitive Words: Deleting repetitive words is a process to correct words with repetitive characters caused by errors in writing that are often found in comments or tweets. The words written are not standardized, so the purpose of deleting repetitive words is to remove or delete characters in a word that are repeated so that it is easier to find the base word [17].

e) Tokenization: Tokenization separates a string of characters or spaces into a list of words [13]. Tokenization aims to collect the number of words that make up the sentences in the dataset. Only one is used if two or more words are the same [11].

f) Word Normalization: Word normalization converts non-standard words into standard words or forms according to the Kamus Besar Bahasa Indonesia (KBBI) [1]. Word normalization in this study converts slang or slang words, word abbreviations, and word spelling errors into standard ordinary words [13]. Word normalization is an important stage, considering that Indonesian Instagram tweets and comments have a lot of non-standard words, so the system cannot detect these words for the classification process [18]. Examples of non-standard word repair or word normalization in tweets and comments that are captured by non-standard words, such as slang or slang writing (ex: gue = saya), writing with abbreviations (ex: bgs = *bagus*), and writing with spelling errors (ex: *jelsk* = *jelek*) [10].

g) Stemming: Stemming is the process of removing affixes and getting the root word from the original word in a sentence [4]. Stemming is used to improve the quality of information from data, for example, to get the relationship between one-word variant and another. Stemming also affects reducing an index file size [19]. In this research, the stemming used by the Indonesian-language Sastrawi library. For example, there are variants of the words "*membuatkan*", "*dibuatkan*", "*membuat*" and "*dibuat*" which only have the root word (stem), namely "buat".

h) Stopword Removal: Stopwords are high-frequency words in a document, both in terms of time and space complexity, with very low informative value [19]. According to [12], examples of stopword lists in Indonesian such as "*yang*", "*di*", "*untuk*", "*dan*", "*ke*", "*dari*", and others. Stopwords removal is a stage to take important words from the tokenizing results by discarding less important words (stoplist) and keeping important words (wordlist). In this research, the stopwords list is from the Indonesian-language Sastrawi library.

This study will compare two pre-processing scenarios, 1st scenario with word normalization and 2nd scenario without word normalization. As shown in Figure 2, it includes case folding, emoji removal, cleaning data, deleting repetitive words, tokenization, word normalization, stemming, and stopwords removal. After the data is pre-processed in the first scenario, the results are in Table III.

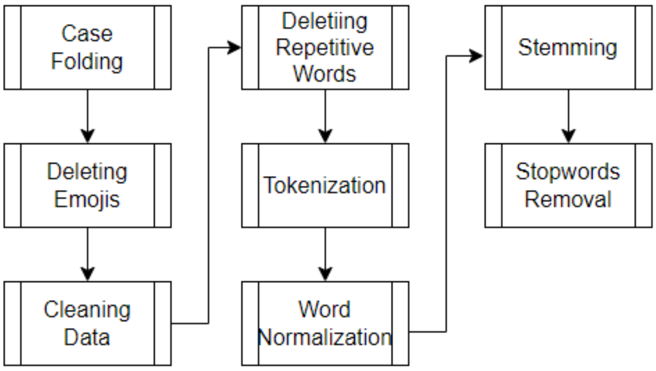

Figure 2. Preprocessing 1st Scenario

TABLE III
PREPROCESSING 1ST SCENARIO

Before	After
@theansm @hynxf Datang sj ke optik terdekat...lbh mdh dan dibantu dgn baikkk😀😀	*datang optik dekat lebih mudah bantu baik*

Figure 3 includes case folding, removal of emoji, cleaning data, deleting repetitive words, tokenization, stemming, and removal of stopwords. After the data is pre-processed in the first scenario, the results are in Table IV.

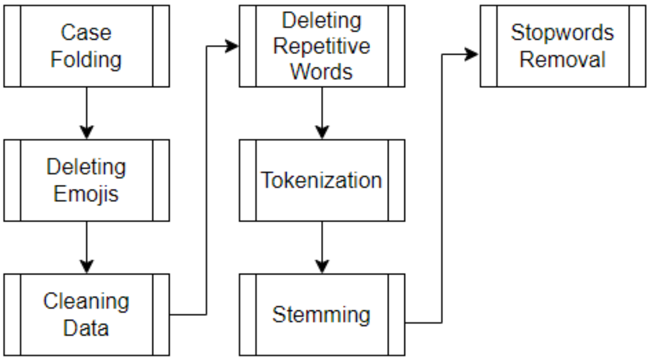

Figure 3. Preprocessing 2nd Scenario

TABLE IV
PREPROCESSING 2ND SCENARIO

Before	After
@theansm @hynxf Datang sj ke optik terdekat...lbh mdh dan dibantu dgn baikkk😀😀	*datang optik dekat lbh mdh bantu baik*

D. TF-IDF (Term Frequency – Inverse Document Frequency)

As mentioned in [1] and [13], the weight of each word in a text is derived by combining the Term Frequency (TF) and the

Inverse text Frequency (IDF) of the document. Term Frequency-Inverse Document Frequency (TF-IDF) is the name given to this method of approaching the problem. The word "term frequency" (TF) refers to the number of times a certain phrase occurs in a particular piece of writing. One may say that the worth of a term is exactly proportional to the number of times it is used [20]. For the purpose of computing the value of the TF-IDF, the equation that is employed is as follows [13]:

$$TF_{(t)} - IDF_{(t)} = tf_{t,d} \times \log\left(\frac{N}{dft}\right) \quad (1)$$

E. Naive Bayes Classifier

The Naive Bayes Classifier algorithm is an algorithm that uses probabilities in making decisions [7]. Naïve Bayes Classifier is a method with simple probability, a derivative of Bayes' theorem. Bayes' theorem is a theory of calculating conditional probability (posterior), namely the calculation of the probability of an event *m* if it is known that there is an event n or *P(m|n)* [22]. According to [7], the classification problem with the Naive Bayes Classifier model can generally be Equation (2).

$$P(m \mid n) = \left(\frac{P(n \mid m) \times P(m)}{P(n)}\right) \quad (2)$$

$$P(m \mid n) = P(n \mid m) \times P(m) \quad (3)$$

Equation (3) or posterior is the probability of class (m) when word (n) occurs. Proseterior is the product of the likelihood and prior, divided by the evidence. Likelihood is the probability of word *(n)* from class *(m)*. Prior is the probability of class (m) before processing the data. Evidence is the probability of the occurrence of the word (n). Calculating evidence or the chance of word occurrence can be omitted because each feature in the dataset is independent or not bound [7] using Equation (3).

F. K-Nearest Neighbor

"nearest neighbor" is one of the various interpretations of the term "k-nearest neighbor," where "k" refers to the total number of neighbors that are closest to you [7]. Due to the fact that it is a supervised learning approach, K-Nearest Neighbor sorts the results of a new query instance in accordance with the majority of the K-Nearest Neighbor categories. In accordance with [23], the outcome of the classification will display the greatest number of classes. When it comes to classifying new objects, the K-Nearest Neighbor algorithm is responsible for locating the training examples that are geographically nearest to the query instance and then using those samples. It is possible to use the Euclidean Distance approach to determine the value of k in accordance with Equation (4) in order to determine whether the distance is close or distant.

$$d_{(x,y)} = \sqrt{\sum\nolimits_{i-1}^{n} (x_i - y_i)^2} \quad (4)$$

G. Confusion Matrix

The last stage tests the classification results using the confusion matrix, which is one of the methods used to measure the performance of a classification method [11]. In measuring the performance of this research using confusion matrix calculation for multi-class and obtained calculations to calculate accuracy, precision, and recall, the confusion matrix table is used in Table V.

TABLE V
CONFUSION MATRIX

		Predicted Class		
		Negative	**Neutral**	**Positive**
Actual Class	**Negative**	TN (*True Negative*)	FL2 (*False Neutral2*)	FP (*False Positive*)
	Neutral	FN2 (*False Negative2*)	TL (*True Neutral*)	FP2 (*False Positive2*)
	Positive	FN (*False Negative*)	FL (*False Neutral*)	TP (*True Positive*)

III. Result and Discussion

In this part, we evaluate the results using various classifiers by employing the Naïve Bayes Classifier and the K-Nearest Neighbor algorithms. The pre-processing scenario is comprised of two different scenarios, namely 1st and 2nd, which are described in this section. The 1st scenario with pre-processing stages consists of case folding, deleting emojis, cleaning data, deleting repetitive words, tokenization, word normalization, stemming, and removing stopwords. The 2nd scenario pre-processing stage consists of case folding, deleting emojis, cleaning data, deleting repetitive words, tokenization, stemming, and stopwords removal. The data used in this study were taken from Twitter and Instagram and included topics related to *BPJS Kesehatan*.

After the data is collected, it will be labeled into positive, neutral, and negative classes using manual labeling. There are 1050 data used in this study with a training data division of 80%, namely 840 data, and test data of 20%, namely 210 data. The word weighting process will be done with TF-IDF after the data-sharing stage. The classification process will be carried out in both the first and second occurrences by using the Naïve Bayes Classifier and the K-Nearest Neighbor methods. A sentiment analysis model will be constructed using two different approaches and two different scenarios, with the k value of the K-Nearest Neighbor method being set to 3.

a) In the first place, assessing the Anomaly Detection Protocol Pre-processing is the first scenario. The Naïve Bayes Classifier method is used specifically for the purpose of pre-processing in the first instance. In the first scenario, the pre-processing stage also includes the application of word normalization. As part of the pre-processing, the following

DOI : https://doi.org/10.25139/inform.v9i1.7111

operations are performed: case folding, emoji removal, data cleaning, removal of repeated words, tokenization, word normalization, stemming, and removal of stopwords. The results of the implementation of the Naïve Bayes Classifier 1st scenario confusion matrix model are shown in Table VI.

TABLE VI
CONFUSION MATRIX MODEL NAÏVE BAYES 1ST SCENARIO

		Predicted Class			
		Negative	**Neutral**	**Positive**	**Total**
Actual	**Negative**	TN=50	FL2=2	FP=10	62
	Neutral	FN2=2	TL=69	FP2=1	72
	Positive	FN=4	FL=8	TP=64	76
	Total	56	79	75	210

b) Testing The Naïve Bayes Classifier Method Pre-processing 2nd scenario: The second test uses the Naïve Bayes Classifier method pre-processing 2nd scenario. The pre-processing stage of 2nd scenario is without using word normalization. Pre-processing consists of case folding, deleting emojis, cleaning data, deleting repetitive words, tokenization, stemming, and stopwords removal. Table VII shows the result of the confusion matrix model Naïve Bayes Classifier 2nd scenario.

TABEL VII
CONFUSION MATRIX MODEL NAÏVE BAYES 2ND SCENARIO

		Predicted Class			
		Negative	**Neutral**	**Positive**	**Total**
Actual	**Negative**	TN=51	FL2=3	FP=8	62
	Neutral	FN2=2	TL=68	FP2=2	72
	Positive	FN=6	FL=7	TP=63	76
	Total	59	78	73	210

c) Testing The K-Nearest Neighbor Method Pre-processing 1st Scenario: The third test uses the K-Nearest Neighbor method pre-processing 1st scenario. The pre-processing stage of 1st scenario is by using word normalization. Pre-processing consists of case folding, deleting emojis, cleaning data, deleting repetitive words, tokenization, word normalization, stemming, and stopwords. Table VIII shows the result of the confusion matrix model K-Nearest Neighbor 1st scenario.

TABLE VIII
CONFUSION MATRIX MODEL K-NEAREST NEIGHBOR 1ST SCENARIO

		Predicted Class			
		Negative	**Neutral**	**Positive**	**Total**
Actual	**Negative**	TN=49	FL2=7	FP=6	62
	Neutral	FN2=6	TL=63	FP2=3	72
	Positive	FN=7	FL=12	TP=57	76
	Total	62	82	66	210

d) Testing The K-Nearest Neighbor Method Pre-processing 2nd Scenario: The fourth test uses the K-Nearest Neighbor method pre-processing 2nd scenario. The pre-processing stage of 2nd scenario is without using word normalization. Pre-processing consists of case folding, deleting emojis, cleaning data, deleting repetitive words, tokenization, stemming, and stopwords. The result of the confusion matrix model K-Nearest Neighbor 2nd scenario is in Table IX.

TABLE IX
CONFUSION MATRIX MODEL K-NEAREST NEIGHBOR 2ND SCENARIO

		Predicted Class			
		Negative	**Neutral**	**Positive**	**Total**
Actual	**Negative**	TN=48	FL2=7	FP=7	62
	Neutral	FN2=10	TL=61	FP2=1	72
	Positive	FN=11	FL=12	TP=53	76
	Total	69	80	61	210

Based on the values in Tables VI, VII, VIII, and Table IX, calculations are made to determine accuracy, precision, recall, and f1-score. Table X presents a comparison of the test results obtained from the model by employing the Naïve Bayes Classifier technique in the first scenario, the Naïve Bayes Classifier method in the second scenario, the K-Nearest Neighbor method in the first scenario, and the K-Nearest Neighbor method in the second scenario methods.

TABEL X
MODEL TESTING RESULT

Model	Accuracy	Precision	Recall
NBC 1st Scenario	87,14%	87,18%	87,14%
NBC 2nd Scenario	86,67%	86,64%	86,67%
KNN 1st Scenario	80,48%	80,93%	80,47%
KNN 2nd Scenario	77,14%	78,12%	77,14%

Based on Table X, the model with the Naïve Bayes Classifier method with 1st scenario is superior to the other three models with 87.14% accuracy, 87.18% precision, and 87.14% recall. In comparison, the Naïve Bayes Classifier with 2nd scenario gets 86.67% accuracy, 86.64% precision, and 86.67% recall. When applied to the first scenario, the K-Nearest Neighbor method achieves an accuracy of 80.48%, a precision of 80.93%, and a recall of 80.47%, respectively. By using the second scenario as an example, we are able to observe that the K-Nearest Neighbor method reaches a level of accuracy, precision, and recall that is 77.14 percent. The created system presents the results of the model tests that are shown in Table X in a graphical format. The graph that compares the test results of the two models is shown in Figure 4.

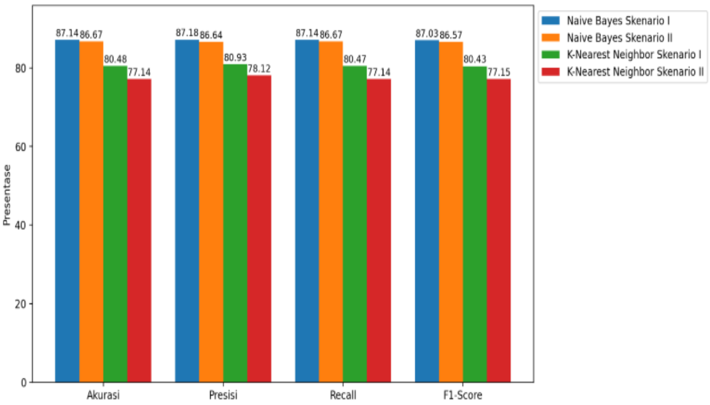

Figure 4: Graph of Model Testing Results

DOI : https://doi.org/10.25139/inform.v9i1.7111

In comparison to the model that employs the Naïve Bayes Classifier technique with the second scenario, which attains an accuracy of 86.67%, precision of 86.64%, and recall of 86.67%, the sentiment analysis model that employs the first scenario achieves an accuracy of 87.14%, precision of 87.18%, and recall of 87.14%. The K-Nearest Neighbor method with 1st scenario obtained an accuracy of 80.48%, precision of 80.93%, and recall of 80.47%. It is also superior to the model using the K-Nearest Neighbor method with 2nd scenario, which obtained an accuracy of 77.14%, precision of 78.12%, and recall of 77.14%. Based on the research that has been done, the model with the Naïve Bayes Classifier method with 1st scenario is a model that has the most superior accuracy, precision, and recall compared to the other three models.

The 1st scenario model using the word normalization stage, namely changing non-standard words to standard words in pre-processing, it is possible to increase recall, accuracy, and precision by refining the comment classifications that are used by the sentiment analysis system. Within the framework of the Naive Bayes Classifier method, every single word is considered to be an independent feature. Word normalization can affect how the Naive Bayes Classifier interprets the relationship between words, affecting the probability calculations used by this algorithm. If normalization changes the meaning or frequency of words, it can also affect the probability calculation and classification process of words in the Naive Bayes Classifier model. Meanwhile, the K-Nearest Neighbor method uses the distance or similarity between data to classify new data. Word normalization affects the calculation of distance or similarity between words in the feature space. Suppose normalization changes the representation of words that were originally different to be more uniform. In that case, it can affect the distance measurement between data and how K-Nearest Neighbor classifies new data.

The Naïve Bayes Classifier technique is superior than the K-nearest neighbor method in terms of accuracy, precision, and recall values. This is because the Naïve Bayes Classifier approach is more successful on relatively small datasets. The K-nearest neighbor algorithm, on the other hand, is effective when used to large datasets.

IV. CONCLUSION

The research that performed the word normalization stage in the Naïve Bayes Classifier and K-Nearest Neighbor methods produced higher accuracy than the word normalization stage. The Naïve Bayes Classifier method, both with 1st scenario and 2nd scenario, has superior accuracy compared to the K-Nearest Neighbor methods in both the 1st and 2nd scenarios. In model testing with 1050 data, the accuracy obtained by the Naïve Bayes Classifier method with 1st scenario is 87.14%. The accuracy obtained by the Naïve Bayes Classifier method with 2nd scenario is 86.67%. Meanwhile, the accuracy obtained from the K-Nearest Neighbor method in 1st scenario was 80.48%, and 2nd scenario resulted in an accuracy of 77.14%.

ACKNOWLEDGMENT

The number of datasets should be included, according to the study's findings, which may be used as input or suggestions, among other things. In addition, it is recommended to compare the results with those obtained from other machine learning or deep learning approaches and incorporate additional pre-processing stages to achieve more accurate results.

REFERENCES

[1] Hana, K. M., Adiwijaya, Al Faraby, S., & Bramantoro, A. (2020). Multi-label Classification of Indonesian Hate Speech on Twitter Using Support Vector Machines. August. ResearchGate. https://www.researchgate.net/publication/347154120

[2] Aribowo, A. S. (2018). Analisis Sentimen Publik pada Program Kesehatan Masyarakat Menggunakan Twitter Opinion Mining. Seminar Nasional Informatika Medis (Snimed), 17–23.

[3] Rasyada, I., Setyowati, Y., Barakbah, A., & Tafaqquh Fiddin, M. (2020). Sentiment Analysis of BPJS Kesehatan's Services Based on Affective Models. *IEEE Xplore*, 549–556.

[4] Kusumawati, N., Maspupah, U., F, D. S. R., & Hamzah, A. (2022). Comparing Algorithm for Sentiment Analysis in Healthcare and Social Security Agency (BPJS Kesehatan). Techno Nusa Mandiri: Journal of Computing and Information Technology, 19(1), 31–37. https://doi.org/10.33480/techno.v19i1.3167.

[5] Karim, A. (2021). Analisis Sentimen pada Komentar Sosial Media Instagram Layanan Kesehatan BPJS Menggunakan Naïve Bayes Classifier.

[6] Fahlapi, R., & Rianto, Y. (2020). Twitter Comment Predictions on Dues Changes BPJS Health in 2020. Jurnal dan Penelitian Teknik Informatika, 5(1), 170–183. https://doi.org/10.33395/sinkron.v5i1.10588

[7] Saputra, Irwansyah & Kristiyanti, Dinar Ajeng. (2022). Machine Learning untuk Pemula. Penerbit Informatika.

[8] Rish, I. (2014). An Empirical Study of The Naïve Bayes Classifier. T.J Watson Research Center. https://www.researchgate.net/publication/228845263

[9] Permana, T., Siregar, A. M., Masruriyah, A. F. N., & Juwita, A. R. (2020). Perbandingan Hasil Prediksi Kredit Macet pada Koperasi. Conference on Innovation and Application of Science and Technology, 737–746

[10] Antinasari, P., Perdana, R. S., & Fauzi, M. A. (2017). Analisis Sentimen tentang Opini Film pada Dokumen Twitter Berbahasa Indonesia Menggunakan Naive Bayes dengan Perbaikan Kata Tidak Baku. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 1(12), 1733–1741. http://j-ptiik.ub.ac.id

[11] Mega, P., Dharmapatni, N., Luh, N., & Merawati, P. (2020). Penerapan Algoritma Support Vector Machine dalam Sentimen Analisis Terkait Kenaikan Tarif BPJS Kesehatan. Jurnal Bumigora Information Technology, 2(2), 105–112. https://doi.org/10.30812/bite.v2i2.904

[12] Widyawati, & Sutanto. (2019). Perbandingan Algoritma Naïve Bayes dan Support Vector Machine. Jurnal Sains & Teknologi, 3(2), 178–194.

[13] Pradana, A. W., & Hayaty, M. (2019). The Effect of Stemming and Removal of Stopwords on The Accuracy of Sentiment Analysis on Indonesian-Language Texts. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control Journal homepage, 4(3). http://kinetik.umm.ac.id

[14] Jumeilah, F. S. (2017). Penerapan Support Vector Machine (SVM) untuk Pengkategorian Penelitian. Jurnal Rekayasa Sistem dan Teknologi Informasi, 1(1), 19–25. http://jurnal.iaii.or.id

[15] Fitriyah, N., Warsito, B., & Maruddani, D. A. I. (2020). Analisis Sentimen Gojek pada Media Sosial Twitter dengan Klasifikasi Support Vector Machine. Jurnal Gaussian, 9(3), 376–390. https://ejournal3.undip.ac.id/index.php/gaussian/

[16] Najiyah, I., & Haryanti, I. (2021). Sentimen Analisis Covid-19 dengan Metode Probabilistic Neural Network dan TF-IDF. Jurnal Responsif, 3(1), 100–111. http://ejurnal.ars.ac.id/index.php/jti

DOI : https://doi.org/10.25139/inform.v9i1.7111

[17] Anugerah, F., & Djunaidy, A. (2017). Improving The Performance of Repeated Character Pre-processing in Recognizing Words in The Indonesian Sentiment Classification. Journal of Basic and Applied Scientific Research, 7(9), 1–9. www.textroad.com

[18] Putra, M. F., Herdiani, A., & Puspandari, D. (2019). Analisis Pengaruh Normalisasi , TF-IDF , Pemilihan Feature-set terhadap Klasifikasi Sentimen Menggunakan Maximum Entropy (Studi Kasus : Grab dan Gojek). e-Proceeding of Engineering, 6(2), 8520–8529.

[19] Jayashree, R., & Murthy, K. S. (2014). Effect of Stop Word Removal on The Performance of Naïve Bayesian Methods for Text Classification in The Kannada Language. Journal Artificial Intelligence and Soft Computing, 4, 264–282.

[20] Meisya, F. (2013) Perancangan Sistem Temu Balik Informasi dengan Metode Pembobotan Kombinasi TF-IDF untuk Pencarian Dokumen Berbahasa Indonesia. Jurnal Sistem dan Teknologi Informasi, 1(1). https://jurnal.untan.ac.id/index.php/justin/article/view/1319/1288

[21] Septian, J. A., Fahrudin, T. M., & Nugroho, A. (2019). Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor. Journal of Intelligent System and Computation. https://www.researchgate.net/publication/335826349

[22] Nugroho, M. A., & Santoso, H. A., (2016). Klasifikasi Dokumen Komentar pada Situs Youtube Menggunakan Algoritma K-Nearest Neighbor. http://eprints.dinus.ac.id/18746/

[23] Rahman, H. (2021). Klasifikasi Sentimen Masyarakat terhadap Layanan Badan Penyelenggara Jaminan Sosial (BPJS) Kesehatan di Twitter Menggunakan Metode K-Nearest Neighbor.

DOI : https://doi.org/10.25139/inform.v9i1.7111