ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse193942020.pdf
https://doi.org/10.30534/ijatcse/2020/193942020

Classification of Indonesian Presidential Campaign on Twitter Using Word2Vec

Vion Age Tricahyo[1], Sani Muhamad Isa[2]
[1] Computer Science Department, BINUS Graduate Program, Bina Nusantara University, Jakarta, Indonesia 11480, vion.tricahyo@binus.ac.id
[2] Computer Science Department, BINUS Graduate Program, Bina Nusantara University, Jakarta, Indonesia 11480

ABSTRACT

Indonesia is a country that use democracy system and one of the largest in the world. The process of democracy continues every 5 years and always raises the pros and cons in the community especially on social media. This research explains the process of presidential election in Indonesia using Word2Vec as an extraction feature. Some classifications used are K-Nearest Neighbors (K-NN), Naïve Bayes, Decision Tree, Random Forest, and Support Vector Machine (SVM) to decide sentiment. The purpose of this research is to compare several classifications as well as to know the highest accuracy of some classification use. Post data about presidential candidates through the crawling process on social media Twitter. Data originating directly from the community and national media produce variations of various responses. The data processed is as much as 640 in Bahasa Indonesia with the keywords Prabowo, Sandi, Jokowi, and Ma'ruf. The results showed the highest accuracy gained when using the Random Forest Classification method, with the highest accuracy reaching 98.33% and lowest accuracy reaching 81.96% using Support Vector Machine.

Key words : Sentiment Analysis, Word2Vec, Machine Learning, Indonesian Election

1. INTRODUCTION

Political activities are not separated from a country. Especially countries that embrace the democracy system as in Indonesia. With this democratic system, the Indonesian people have the opportunity to choose the country leaders of their choice. So political activities such as campaigns are hard to avoid, because they play an important role to reach the public vote. In this era, political activity in fact is far from 10 years ago that is far from social media. Based on the latest data 2019 the number of active internet users in Indonesia reached 150 million from the total population of 268,2 million. So the information obtained by the community is very fast from various sources [1]. The use of social media such as Facebook, Twitter and Youtube makes political candidates continue to interact with supporters and receive support such as donations and volunteers [2]. Campaign efforts can be done by a person or a group of organized people to perform the achievement of a decision making process within a group, a regular campaign is also done to influence, inhibition, access defect [3].

With existing political candidate, the base of each campaign team formed the public opinion. This opinion split into a pro for supporting prospective and counter candidate who support prospective competitors. With the pros and cons response, community sentiment is very influential for every candidate. Many television media, statistics agencies and governments make it benchmark as an evaluation especially social media twitter. Twitter is the most hot social media in preaching campaign information. Twitter is very fast and responsive compared to other media and makes trending topic news. Another fact is the use of Twitter is very challenging, ranging from the tweet size limitation that only 140 character, the very diverse slang words usage, Twitter let the usage of hashtags, client references, URL links in Twitter features, and user variety such as the use of mixed language from Indonesia and English, until the use of symbols that considered strange[4]. Knowing this, people are very actively commenting on social media and want to compare survey results from a lot of sources. So with such circumstances, data retrieval that represents source content retrieved and processed for research [5].

The process of retrieving information data (text mining through various sources starting from the steps proceeded with pre-processing [6]. Pre-processing step divided into (1) case folding, to change the word to lowercase (2) tokenization, to the cleansing username, URL, and the retweet sign, (3) stopword, to remove the unnecessary word and (4) stemming, to cut the word base [7]. After that the determination phase of text categorization, text clustering, concept/entity extraction, production granular taxonomy, sentiment analysis, document highlighting, and entity relationship modeling to cut high-quality information data [8]. Through this analysis sentiment grouping a positive, negative or neutral orientation can be obtained based on text polarity as the object [9][10]. With a diverse data derived from public opinion such as reviews, forum discussions, blogs, micro-blogs, comments and posts on social network

sites can help decision-making and data ready processed using sentiment analysis[11][12] . From the above process, word mapping converted into vector from using Word2vec. These vectors used a range of tasks Natural Language Processing using Python language. This is because Python comes with a large and comprehensive standard library functionality that makes it easy for machine learning users over 20 million worldwide [13][14]. To find the value of Word2Vec, use the following formula :

$$D = \{w_1, w_2, w_3, \dots, w_N\} \quad (1)$$

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N}\sum_{-t\le j\le t\neq 0} \log p\,(w_{t+j}|w_t) \quad (2)$$

$$p\,(w_j|w_i) = \frac{\exp\,(v'^{T}_{wj}\, v_{wi})}{\sum_{j=1}^{V}\exp\,(v'^{T}_{wj}\, v_{wi})} \quad (3)$$

Where D as the document and each word as w_N with N is the number of words in the document. On Neural Network Training objects, the variables v_w and v'_w are two representations of the word w. Variable v_w comes from rows of W, which is the input→hidden weight matrix, and v'_wcomes from columns of W' , which is the hidden→output matrix. In subsequent analysis, we call v_w as the "input vector", and v'_w as the "output vector" of the word w.

$$V_{embedding} = v_w{}^T * M_{emb} \quad (4)$$

$$emb_i = \frac{\sum_{j=i}^{n} v_j^i}{n} \quad (5)$$

The result of the training is M_{emb} matrix of *m x d* where *d* is the vector dimension of Word2Vec word. To get the document vector value, the emb_i variable represents the N value as the word count and v_j^i as the *i* element of the *j* vector. So that with $v_{doc} = [emb_1, emb_2, emb_3, \dots, emb_d]$ obtained a vector from a news document. After obtaining vector values, the next process is to compare the classification of K-Nearest Neighbors (K-NN), Naïve Bayes, Decision Tree, Random Forest and Support Vector Machine (SVM). The value of each classification result shows different results and can show the lowest value and highest value of each classification implementation. Selection of classification models through several note that have similarities in big data processing. In various literature found reviews that models of classification above is the most commonly used. Besides that, the focus on finding the highest percentage results is also an important factor in supporting this research.

2. LITERATURE REVIEW

Literature research related to Word2Vec conducted by Parikh about determining sentiment analysis in the case of commercial movie with data obtained from the Internet Movie Database (IMDB). To increase the commercial value, researchers compare the prerelease of review from the audience about their opinion of the film and aim to help producers choose a strategy in film release. In this study using feature extraction such as Term Frequency - Inverse Document Frequency (TF-IDF) and Word2Vec as well as classification using Decision Tree, Random Forest, Naïve Bayes, Logistic Regression and Support Vector Machine [15]. Implementation of Word2Vec conducted by Djaballah *et al*, which researched sentiment analysis on Twitter relating to the radical content in Arabic tweet. The main challenge in applying Word2Vec to sentiment analysis generally results in a high vector dimension context. The highest achieved weighted are 0.64 to 0.76 with global average category using Support Vector Machine and Random Forest Classifiers [16].

The research conducted by Farhan and Khodra determines the Indonesian sentiment analysis with specific word as research data. Data is retrieved from the *TripAdvisor* platform with positive and negative sentiment labels [17]. Data compared to several methods using Word2vec. The determination of the sentiment label is similar to that done by Buntoro with the topic about the election head of DKI Jakarta area 2017. In this study used as many as 100 data results from public opinion on Twitter. Determination of sentiment using Naïve Bayes classification and SVM. From this study produced 3 categories based on the number of candidates in the elections. The accuracy value obtained is between 84-90% [18]

The next research is from Fauzi and Yuniarti who researched about hate speech detection on Twitter. Starting from the difficulty of identifying a lot of hate speech, this research is expected to facilitate the determination of such identification [19]. Another study covering hate speech was also done by Andana et al which focuses on detecting negative content on Twitter. This research uses Naive Bayes and Support Vector Machines with varying K-Fold Cross Validation ranging from k=2 to K=10 which generates accuracy between 72,72% to 94.49% [20]. Research approached by Tian & Wu with topics also about emotional analysis on Twitter. Data obtained via Twitter using the bag of Word and classification feature using the Support Vector Machine. Future research recommendations to implementation the Paragraf2Vec or Word2Vec method as feature extraction [21]. From several sentiments analysis above, Buladaco *et al* research is most varied due to the use of many machine learning such as Support Vector Machine, Random Forest and Naive Bayes. With a topic focused on land transport infrastructure generated an accuracy range of 68 to 76.12% [22].

As far as papers written, research on the use of Word2Vec on election topics in Indonesia has not been found. Making standards / protocols by combining correspondence, linguistic rules, and Sastrawi libraries in data processing also does not yet exist. Moreover, addition various classifications adds variation in research. Besides that, it known the comparison of each classification. From the results of the classification comparison it seen the best classification and can recommended to various survey institutions and considerations in future research.

3. METHODOLOGY

This research explores the use of raw data obtained on Twitter into data that is ready to processed. The data processed has the same amount and time of collection. Starting with how to find the positive and negative labels of existing data. Because the label determination technique is very influential in this study to support conformity (ground truth data), even before the pre-processing process begins. For that reason, making standard / protocol labels in this research needs more attention. The process of making labels starts with the method of correspondence with participants who meet the terms and conditions. After determining the label through correspondence continued to determination of the word class based on linguistic rules through adjectives, adverb, nouns and verbs. To strengthen the protocol, the data can use the Sastrawi library which has groupings of words that contain positive and negative Indonesian Language.

This research uses 320 records for each candidate. The ready data continues into pre-processing, feature extraction and classification. The existing vector results can also be shown through the parameters used in the Word2Vec extraction to see the results and visualization of the vector. Further data can continue to incorporated into classification models calculations. The result will pay attention to the weight of each data such as accuracy, precision, recall and confusion matrix value to analyze the results of the amount of positive and negative data of each classification.

4. ANALYST RESULT

4.1 Pre-processing

In this stage, the data used has been tested for truth. Test the truth using a grammar combination that complies with the rules of linguistics, literary libraries, and correspondence. With the linguistic basis, the crawl data presented in the form of a questionnaire with the voting model as the label determination. In its implementation, determining the value of data sentiment using quantitative methods, where neutral class can considered into negative class [23]. The following are the data used in this study:

Table 1: Dataset example used in this research

Jokowi/Prabowo Data	1	2	3	4	5	6	7	8	9	10	*Sentiment*
Debat sesaat lagi dimulai! @jokowi mengaku sudah mantap betul (mantul), sementara Ma'ruf akan menambahi apa yang di… https://t.co/yjT5NeL7BX	+	+	+	-	-	+	+	+	+	+	Positive
@prabowo@sandiunodilaporkan ke Bawaslu karena penyampaian visi-misi yang disiarkan beberapa stasiun TV swasta.… https://t.co/pBtWeMY32G	-	-	-	+	-	-	+	-	+	-	Negative

In the Table 1, sentiment can show positive or negative according to the data obtained by crawling results. Participants in the voting must follow the provisions as they should neutral, only focus on the text presented etc. The next step is to use a corpus or dictionary of Indonesian language to reinforce the label result sentiment data above. The following is an example of the positive and negative word grouping data used :

Table 2: Word dataset label

Word List	Type
abnormal, acak, bencanaalam, cabul, dendam, edan,, fatal, gila, histeris, ilegal, jelek, kebencian, lemot, mabuk, najis, otoriter, pecundang, roboh, sabotase, terjelek, ugal-ugalan, virus, zina	Neg
adil, bonus, cakap, dipercaya, enak, favorit, gigih, harum, idola, juara, kaya, keadilan, layak, manis, nikmat, optimis, pintar, riang, senang, teladan, unik, yeah	Pos

With the word list in Table 2, the Protocol of labeling is increasingly robust and can tested in truth. So the pre-processing process is getting faster and easier. This pre-processing process starts from case folding to stemming. In stemming, mapping between different words formed by a basic word form [24]. The following are examples of stemming as well as the number of tokens per word in a single record :

Table 3: Stemming dataset with tokens

Stemming List	Data
[('jokowi', 4), ('amat', 2), ('nilai', 2), ('rocky', 2), ('debat', 1), ('aku', 1), ('mantap', 1), ('mantul', 1), ('makruf', 1), ('tambah', 1), ('di', 1), ('httpstcoyjt5nel7bx', 1), ('segmen', 1), ('debatkeduapilpres2019', 1), ('pakar', 1), ('politik', 1), ('yunarto', 1), ('wijaya', 1), ('unggul', 1), ('cara', 1), ('httpstcoxqohys86gq', 1), ('gerung', 1), ('capres', 1), ('tahana', 1), ('joko', 1), ('widodo', 1), ('cermat', 1), ('menggarisbawahi', 1), ('jok', 1), ('httpstcozoo7o76vlo', 1), ('perintah', 1), ('australia', 1), ('berat', 1), ('putus', 1), ('bebas', 1), ('syarat', 1), ('abu', 1), ('bakar', 1), ('baasyir', 1), ('ini', 1), ('httpstcoy0ybsgnjgb', 1)]	Jokowi
[('bpn', 3), ('lapor', 2), ('prabowosandi', 2), ('pilih', 2), ('anggap', 1), ('berita', 1), ('kandung', 1), ('fitnah', 1), ('prabowo', 1), ('sandiuno', 1), ('tabloid', 1), ('indonesia', 1), ('barokah', 1), ('httpstcowzgxcht0hk', 1), ('kpu', 1), ('ruang', 1), ('mula', 1), ('capres', 1), ('mu', 1), ('httpstcos57iit001k', 1), ('menang', 1), ('telak', 1), ('debat', 1), ('moderator', 1), ('brilian', 1), ('httpstco2u4hnzcclo', 1), ('prabowosandiuno', 1), ('bawaslu', 1), ('sampai', 1), ('visimisi', 1), ('siar', 1), ('stasiun', 1), ('tv', 1), ('swasta', 1), ('httpstcopbtwemy32g', 1)]	Prabowo

Table 3 summarized the stemming result will be further processed in the extraction feature which will calculate the value of each word. The application of stemming above uses the Sastrawi Library of Indonesian literary language which is also widely applied in related research. The data presentation of each candidate above remains separated also including positive and negative value labels. The value of each word later becomes a vector to processed further into the classification process.

4.2 Extraction Feature

A feature is a unique characteristic of each object. According to some studies, the types of features can be divided into natural and artificial features. So we need a process to get features that differentiate each object feature [25] . In this study features implementation to used is Word2Vec. The extraction of the Word2Vec feature can represent the word input via the vector fix-length feature by providing two main architectural models for continuous vector computing. The vector representation is the Continuous Bag-Of-Words (CBOW) model and the Continuous Skip-Gram model.

Implementing this feature will count words in vectors. In presenting values, vector data remains separated between positive data and negative data. It aims to find the comparison of vector values of each candidate. The use of the Word2Vec formula in the previous explanation generates the document vector value V_{doc} = [$emb_1, emb_2, emb_3 \ldots emb_n$] as a representation model. The following is the result of Word2Vec vector implementation form of document pre-processing results each candidate. For Jokowi candidate dataset with word 'jokowi' = (0.10933, -0.0948, -0.0811, emb_n), 'https' = (0.08412, 0.02427, -0.057, emb_n), 'yang' = (0.07107, 0.06424, 0.04294, emb_n), 'makruf' = (-0.0525, -0.013, -0.1094, emb_n), 'di' = (-0.0586, -0.018, -0.119, emb_n), 'sebut' = (-0.0066, 0.05306, -0.0124, emb_n), 'soal' = (-0.1205, 0.0085, 0.10484 , emb_n). And for Prabowo candidate dataset with word 'prabowo' = (0.084119, 0.024265, -0.05702, emb_n), 'https' = (0.109329, -0.09475, -0.08109, emb_n), 'yang' = (0.071075, 0.064239, 0.042938, emb_n), 'di' = (-0.0525, -0.01297, -0.10939, emb_n), 'saya' = (-0.05858, -0.01803, -0.11897, emb_n), 'tidak' = (-0.00662, 0.053064, -0.01236, emb_n), 'dan' = (-0.12045, 0.0085, 0.104836, emb_n) , 'sandiaga' = (-0.07676, -0.06461, -0.05808, emb_n).

Vector data above can also be known the size of value. Therefore, the implementation of the parameters is also important to know the vector weights and vector visualizations. The parameters used are *most similar* to the use of the word **"president"** to find out how close the word is to each candidate. Previously parameters *size*, *min_count* and *SG* were applied to figure out the dimensions and frequencies in the specified output result of the parameters model used. The following are Figure 1 and Figure 2 sample result of the implementation of the parameters and their visualizations. For Jokowi candidate dataset obtained parameters :

[('**jokowi', 0.2905200123786926**), ('dari', 0.28112560510635376), ('bpn', 0.2692156136035919), ('di', 0.2592431306838989), ('yang', 0.2442457377910614), ('dengan', 0.23718471825122833), ('presiden', 0.22448422014713287), ('akan', 0.22430789470672607), ('itu', 0.22243788838386536), ('tkn', 0.21347741782665253)].

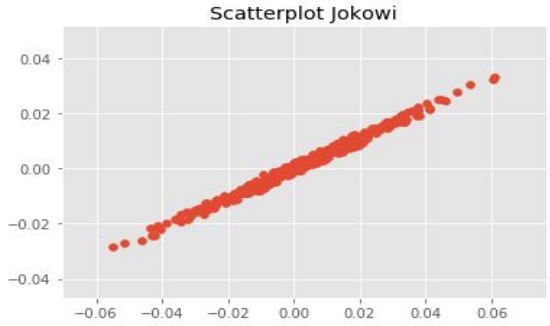

Figure 1: Scatterplot of Jokowi result

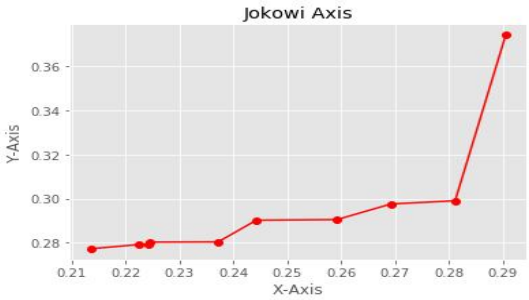

Figure 2: Visualization of Jokowi result example

As for the Prabowo candidate datasets will be shown in Figure 3 and Figure 4. The following parameters result :

[('**prabowo', 0.24484783411026**), ('uno', 0.22321300208568573), ('di', 0.22255849838256836), ('saat', 0.20595777034759521), ('menyebut', 0.19772271811962128), ('itu', 0.1956208050251007), ('bpn', 0.1925741583108902), ('video', 0.1919659674167633), ('pilpr', 0.1903660148382187), ('subianto', 0.1828940212726593)].

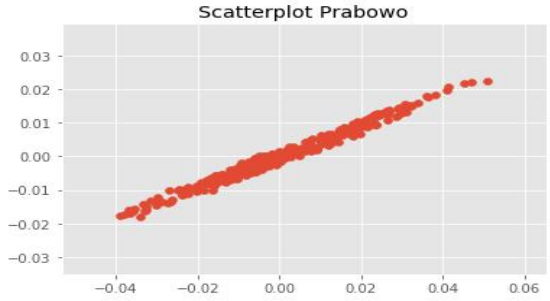

Figure 3: Scatterplot of Prabowo result

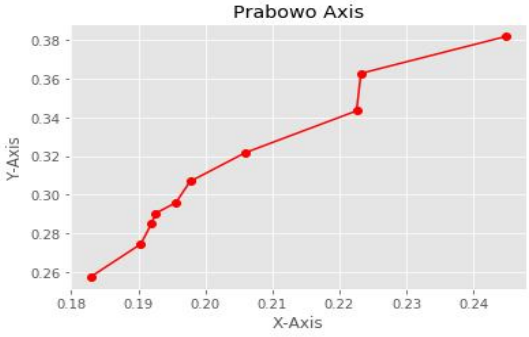

Figure 4: Visualization of Prabowo candidate result example

In the data above, each data mentions Jokowi and Prabowo has the highest value for most similar parameters. In addition to data presentation, presentation through visualization can also be applied. The data above is a step to prepare the training process / train the model. Training using the Gensim Word2Vec library. The process done by entering a list of tokens that have been ready. Data that has the same vocabulary counted as one. Below is the code for the training process :

```
  model = genism.models.Word2Vec (document, size = 1200,
window = 5, min_count = 2, workers = 8)
  model.train (documents, total_examples = len (documents),
epochs = 10)
```

Data processed in memory and log data will process training results to completion. This training process uses 8 threads on the laptop's CUDA GPU by producing a list of 303173090 words and processing time of approximately 12540.3 seconds. Testing done by entering one word into the model and see the same words. Testing done directly in python using the word "presiden".

```
  w1 = "presiden"
  model.wv.most_similar (positive = w1)
```

The code produces the output that has been used in the visualization in Picture 1 and 2 above. The above result devoted to visualizing the high dimension Word2Vec embedded Word using libraries in Python. Visualizations can be useful for understanding how Word2Vec works and how to interpret the link between captured vectors from text before using them in a neural network or other learning machine algorithms. After testing is complete and gets a grade, do to machine learning technique, which is classification using several models.

4.3 Classification

In this research to process vector results from the extraction process of the previously acquired features. At this stage tried various classifications to determine the difference in classification result and determining which classification value is the highest. Measurement of performance using confusion matrix, there are 4 terms as a representation of classification process results. The four terms are True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) [26]. In this process should be prepared data in the form of Excel, CSV or other formats supported before the determination of replace missing values, set role, nominal to text, process document and cross validation are linked to each other. In the cross-validation process, at this stage will be carried out test validation result of classification that has been generated using K-Fold test data sharing method with k = 10. The following are the Figure 5 and Figure 6 that show the process and design of the cross validation. Followed by several classifications used.

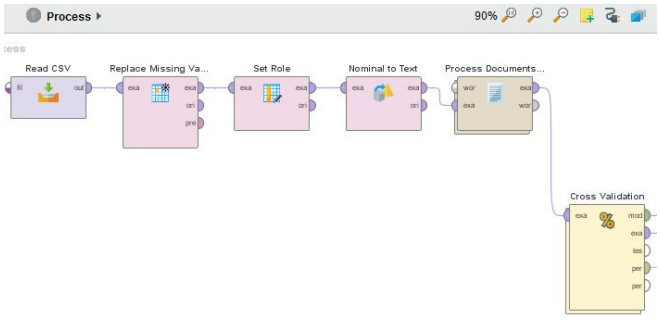

Figure 5: Process Design

Figure 6: Cross Validation Design

4.3.1 Naïve Bayes.

Algorithms using the Bayes theorems assume all independent or interdependent attributes are given values on class variables [27]. At the Bayes theorem, if there are two separate events (e.g. X and H), then Bayes theorem formulated as follows [28] :

$$P(H|X) = \frac{P(X|H)}{P(X)}.P(H) \quad (6)$$

The advantage of using Naïve Bayes is that this method requires only a small amount of training facts to decide the necessary the parameter estimate in the classifications process. Naive Bayes often works much better in most complex real-world situations than expected [29]. Here is Table 4 result of Naïve Bayes value of vector implementation.

Table 4: Classification of Naïve Bayes (NB) each candidate

Accuracy : 96.90% +/- 6.55% (micro average : 96.72%)			
NB Jokowi	True Neg	True Pos	Class Precision
Pred. Neg	7	0	100.00%
Pred. Pos	2	52	96.30%
Recall	77.78%	100.00%	
Accuracy : 82.50% +/- 15.44% (micro average : 82.86%)			
NB Prabowo	True Neg	True Pos	Class Precision
Pred. Neg	1	1	50%
Pred. Pos	5	28	84.85%
Recall	16.67%	96.55%	

4.3.2 Decision Tree

In building a top-down Decision Tree, the first stage is to check all existing attributes using a statistical measure (which is widely used is information gain) to measure the effectiveness of an attribute in classifying a sample set of data.

The attributes placed on the root node are attributes that have the largest information gain. All attributes are categories of discrete value. The attributes with the continuous value must be discredited [30]. As for finding such value is :

$$I(S_1, S_2, \ldots, S_n) = -\sum_{i=1}^{m} P_i \log_2(P_i) \qquad (7)$$

S for the Space (data) sample used for training and P (+) for a positive resolution amount. Meanwhile, P (-) for a solution that is negative (does not support) on the sample data for certain criteria. Table 5 below is the result of implementing the Decision Tree vector .

Table 5: Classification of Decision Tree (DT) each candidate

Accuracy : 97.50% +/- 7.91% (micro average : 97.78%)			
DT Jokowi	True Neg	True Pos	Class Precision
Pred. Neg	9	1	90%
Pred. Pos	0	51	100.00%
Recall	100.00%	98.08%	
Accuracy : 93.33% +/- 14.05% (micro average : 94.29%)			
DT Prabowo	True Neg	True Pos	Class Precision
Pred. Neg	5	1	83.33%
Pred. Pos	1	28	96.55%
Recall	83.33%	96.55%	

4.3.3 Random Forest

Random Forest is a classifier consisting of a tree-shaped classifier {h (x, θ*k*), k = 1,...} where θ*k* is a random vector that is independently divined and each tree in a unit will select the most popular class in input x. Random Forest relies on a random vector value with the same distribution on all trees that each Decision Tree has a maximum depth [31]. The implementation result of the Random Forest vector in the following Table 6.

Table 6: Classification of Random Forest (RF) each candidate

Accuracy : 98.33% +/- 5.27% (micro average : 98.36%)			
RF Jokowi	True Neg	True Pos	Class Precision
Pred. Neg	9	1	90%
Pred. Pos	0	51	100.00%
Recall	100.00%	98.08%	
Accuracy : 93.33% +/- 14.05% (micro average : 94.29%)			
RF Prabowo	True Neg	True Pos	Class Precision
Pred. Neg	5	1	83.33%
Pred. Pos	1	28	96.55%
Recall	83.33%	96.55%	

4.3.4 K-Nearst Neighbours (K-NN)

K-Nearest Neighbor is a method that uses the supervised algorithm, where the results of a new query classified based on majority of the categories on the K-NN [32].

$$d = \sqrt{(x_2 - x_1)^2} + \sqrt{(y_2 - y_1)^2} \qquad (8)$$

The accuracy of the K-Nearest Neighbor algorithm determined by presence and absence of irrelevant data, or if the weight of the feature is equal to its relevance to the classification. Nearby neighboring search techniques are common using Euclidean distance formulas. The Euclidean distance is a formula for finding distances between 2 points in a two-dimensional space. Table 7 Below is the result of implementing the K-Nearest Neighbor vector.

Table 7: Classification of K-Nearest Neighbor (K-NN) each candidate

Accuracy : 97.50% +/- 7.91% (micro average : 97.78%)			
K-NN Jokowi	True Neg	True Pos	Class Precision
Pred. Neg	5	1	83.33%
Pred. Pos	0	39	100.00%
Recall	100.00%	97.50%	
Accuracy : 93.33% +/- 14.05% (micro average : 94.29%)			
K-NN Prabowo	True Neg	True Pos	Class Precision
Pred. Neg	5	1	83.33%
Pred. Pos	1	28	96.55%
Recall	83.33%	96.55%	

4.3.5 Support Vector Machine

Support Vector Machine (SVM) is a set of guided learning methods that analyzes data and recognizes patterns, used for the classification and analysis of regression [33].

$$K(x, xi) = x \cdot x^T \qquad (9)$$

The original SVM algorithm is a derivative of the current standard (soft margins) proposed by Corinna Cortes and Vapnik Vladimir [34]**.** Table 8 is implementation SVM result.

Table 8: Classification of Support Vector Machine (SVM) each candidate

Accuracy : 86.67% +/- 15.32% (micro average : 86.89%)			
SVM Jokowi	True Neg	True Pos	Class Precision
Pred. Neg	7	6	53.85%
Pred. Pos	2	46	95.83%
Recall	77.78%	88.46%	
Accuracy : 81.96% +/- 9.39% (micro average : 81.94%)			
SVM Prabowo	True Neg	True Pos	Class Precision
Pred. Neg	1	6	14.29%
Pred. Pos	7	58	89.23%
Recall	12.50%	90.62%	

In the results of table 4-8 above the highest classification obtained through the Random Forest classification which reaches 98.33% for the Jokowi's pair. While the lowest accuracy value is 81.96% using the Support Vector Machine classification for Prabowo's pair.

5. RESULT

The label determination protocols made very influential in strengthening basic word justification as data that is ready before applied the extraction feature. Through Word2Vec obtained a valid vector and can used to generate classification values. The classification value obtained by various algorithms is very high and varied. Table 9 below is the result of comparative view of machine learning models.

Table 9: Model Accuracy Comparison

Classification	Accuracy of Jokowi	Accuracy of Prabowo
K-NN	97.5 %	93.33 %
Naïve Bayes	96.9 %	82.5 %
Random Forest	98.33 %	93.33 %
Decision Tree	97.5 %	93.33 %
SVM	86.67 %	81.96 %

The highest value of Jokowi pair accuracy is 98.33% through Random Forest classification. Followed by K-Nearest Neighbor and Decision Tree with a value of 97.5%. Then 96.9% through the Naive Bayes classification and the lowest with a value of 86.67% through Support Vector Machine. As for Prabowo pair, the highest score is obtained through K-Nearest Neighbor, Random Forest and Decision Tree with an accuracy value of 93.33%. Followed by Naive Bayes with a value of 82.5% and the lowest via Support Vector Machine with a score of 81.96%. Data used as much as 640 and using K-fold with a value of k = 10. With this value the accuracy is higher compared to the previous research values as well as the use of various methods add to the variation of results.

6. CONCLUSION

The Word2Vec feature extraction method can implemented and looks compatible with a range of classifications. In addition, the parameters features on Word2Vec are very diverse and make it easy to get varied data results. An important key both are pre-processing processes that produce high quality data. The need for implementation in various institutions or statistical agencies can help their research. In the future, research can improved through addition of the Indonesian language library with the word not raw and slang words that are often used on social media. As well as addition and modification of extraction features.

REFERENCES

[1] DataReportal, **Q2 Global Digital Statshot**, 2019, Available: datareportal.com. [Accessed: 02-Dec-2019].

[2] B. Ardha, "**Social Media Sebagai Media Kampanye Partai Politik 2014 di Indonesia**", *J. Visi Komun.*, vol. 13, no. 1, pp. 105–120, 2014.

[3] P. Laure, **Political Strategy and Tactics : A Practical Guide.** Lakehead University in Thunder Bay, Ontario, Canada: Nova Science Pub Inc, 2006.

[4] J. P. Pinto, V. T Maurari, and S. Kelur, "**Twitter Sentiment Analysis : A Political View**", *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 1, pp. 723–729, 2020. https://doi.org/10.30534/ijatcse/2020/103912020

[5] M. Harlian, **Machine Learning Text Categorization.** Austin: The University of Texas, 2006.

[6] H. Marti, **What is Text Mining?**, *SIMS, UC Berkeley*, 2003, Available: http://people.ischool.berkeley.edu/~hearst/text-mining.html. [Accessed: 01-Dec-2019].

[7] N. Rustiana, Deden Rahayu, "**Analisis Sentimen Pasar Otomotif Mobil : Tweet Twitter Menggunakan Naive Bayes**", *J. Simetris*, vol. 8, no. 1, pp. 113–120, 2017.

[8] N. W. S. Saraswati, "**Naive Bayes Classifier dan Support Vector Machines untuk Sentiment Analysis**", *SESINDO*, pp. 585–591, 2013.

[9] D. M D, S. C, and A. Ganesh, "**Sentiment Analysis : A Comparative Study on Different Approaches**", *ELSEVIER Procedia Comput. Sci.*, vol. 87, pp. 44–49, 2016. https://doi.org/10.1016/j.procs.2016.05.124

[10] D. Michelle, **Sentiment Analysis, Hard But Worth It!** 2010, Available: customerthink.com. [Accessed: 02-Dec-2019].

[11] J. Jotheeswaran and S. D. Koteeswaran, "**Sentiment Analysis : A Survey of Current Research and Techniques**", *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 3, no. 5, pp. 3749–3757, 2015.

[12] B. Liu, **Sentiment Analysis and Opinion Mining.** Chicago, USA: Morgan & Claypool Publishers, 2012.

[13] S. Deibel, **Executive Summary : The Python Software Foundation**, 2008. .

[14] Anaconda, **Your Data Science Toolkit - Individual Edition**, 2019, Available: https://www.anaconda.com/products/individual. [Accessed: 03-Dec-2019].

[15] Y. Parikh, A. Palusa, S. Kasthuri, R. Mehta, and D. Rana, "**Efficient Word2Vec Vectors for Sentiment Analysis to Improve Commercial Movie Success**", *Springer Nat. Singapore - Lect. Notes*, pp. 269–279, 2018.

[16] K. A. Djaballah, Boukhalfa, Kamel, and O. Boussaid, "**Sentiment Analysis of Twitter Messages using Word2Vec by Weighted Average**", *IEEE - Sixth Int. Conf. Soc. Netw. Anal. Manag. Secur.*, pp. 223–228, 2019.

[17] A. N. Farhan and M. L. Khodra, "**Sentiment-specific Word Embedding for Indonesian Sentiment Analysis**", *IEEE - Int. Conf. Adv. Informatics, Concepts, Theory, Appl.*, 2017. https://doi.org/10.1109/ICAICTA.2017.8090964

[18] G. A. Buntoro, "**Analisis Sentimen Calon Gubernur DKI Jakarta 2017 di Twitter**", *Integer J.*, vol. 2, no. 1, pp. 32–41, 2017.

[19] M. A. Fauzi and A. Yuniarti, "**Ensemble Method for Indonesian Twitter Hate Speech Detection**", *Indones. J. Electr. Eng. Comput. Sci.*, vol. 11, no. 1, pp. 294–299, 2018.

[20] E. K. Andana, M. Othman, and R. Ibrahim, "**Comparative Analysis of Text Classification Using Naive Bayes and Support Vector Machine in Detecting Negative Content in Indonesian Twitter**", *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 1.3, pp. 356–362, 2019. https://doi.org/10.30534/ijatcse/2019/6481.32019

[21] H. Tian and L. Wu, "**Microblog Emotional Analysis Based on TF-IWF Weighted Word2vec Model**", *IEEE - 9th Int. Conf. Softw. Eng. Serv. Sci.*, pp. 893–896, 2019.

[22] M. V. M. Buladaco, J. S. Buladaco, and L. M. Cantero, "**Sentiment Analysis on Public Land Transport Infrastructure in Davao Region Using Machine Learning Algorithms**", *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 1, pp. 685–690, 2020. https://doi.org/10.30534/ijatcse/2020/97912020

[23] Y. Azhar, A. Z. Arifin, and D. Purwitasari, "**Otomatisasi Perbandingan Produk Berdasarkan Bobot Fitur Pada Teks Opini**", *J. Ilm. Ilmu Komput. Univ. Udayana*, vol. 6, no. 2, pp. 31–34, 2013.

[24] F. Z. Tala, **A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia.** Amsterdam, Netherlands: Universiteit van Amsterdam, 2003.

[25] D. Putra, **Pengolahan Citra Digital.** Yogyakarta: Andi Offset, 2010.

[26] S. Visa, B. Ramsay, A. Ralescu, and E. Knaap, "**Confusion Matrix-based Feature Selection**", *Proc. 22nd Midwest Artif. Intell. Cogn. Sci. Conf.*, 2011.

[27] T. R. Patil and S. S. Sherekar, "**Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification**", *Int. J. Comput. Sci. Appl.*, vol. 6, no. 2, 2013.

[28] Bustami, "**Penerapan Algoritma Naive Bayes Untuk Mengklasifikasi Data Nasabah Asuransi**", *TECHSI J. Penelit. Tek. Inform.*, vol. 8, no. 1, pp. 127–146, 2013.

[29] U. Dulhare, "**Prediction System for Heart Disease Using Naive Bayes**", *Int. J. Adv. Comput. Math. Sci.*, 2018.

[30] A. Basuki and I. Syarif, "**Decision Tree**", *Politeknik Elektronika Negeri Surabaya*, 2003. Available: http://basuki.lecturer.pens.ac.id. [Accessed: 04-Dec-2019].

[31] L. Breimen, **Random Forests Machine Learning.** Berkeley, California: University of California, 2001.

[32] J. Han, M. Kamber, and J. Pei, **Data Mining : Concept and Technique,** 3rd Edition. New York: Morgan Kaufmann, 2012.

[33] A. S. Nugroho, A. B. Witarto, and D. Handoko, "**Application of Support Vector Machine in Bioinformatics**", *Proceeding Indones. Sci. Meet. Cenral Japan*, 2003.

[34] C. Cortes and V. Vapnik, **Support-Vector Networks.** Boston: Kluwer Academic, 1995.