Available online at: http://proceeding.rsfpress.com/index.php/ess/index
LPPM UPN "Veteran" Yogyakarta Conference Series
Proceeding on Engineering and Science Series (ESS)
Volume 1 Number 1 (2020): 648-657

Fanaticism Analysis of Social Media Using Machine Learning

Agus Sasmito Aribowo, Nur Heri Cahyana

Universitas Pembangunan Nasional Veteran Yogyakarta
Email address sasmito.skom@upnyk.ac.id

Abstract

Sentiment and emotion analysis on social media is an interesting study because it reveals the emotional state of the public in a domain. The challenges in sentiment analysis research in Indonesian are inefficient preprocessing, inaccurate feature extraction methods, and low classification accuracy by machine learning. One aspect of sentiment analysis is fanaticism. Fanaticism contains an emotional element in sentiment analysis. This article discusses how to detect opinions that contain political fanaticism, then categorize them into several polarities of political fanaticism. Feature extraction is done by processing sentiment, anger, happiness, disgust, surprise, fear, and hate speech analysis. Knowledge for classification is K-NN, Naive Bayes, Random Forest, and Decision Tree. The aim is to find out the best combination of machine learning methods for feature extraction and finally used for fanaticism categorization. The best method is Random Forest with an accuracy of 81% and will be used as a final method for monitoring fanaticism on social media.

Keyword: fanaticism, machine learning, feature extraction, social media

I. INTRODUCTION

This article is the second phase of research on fanaticism categorization in the Indonesian language. Data source from social media. The results of the first study were categorization of fanaticism using several machine learning technique (Ariwibowo, 2020) . Fanaticism categorization is knowledge generated from the machine learning process and in the form of a tree structure. The knowledge categorizes political fanaticism into four levels: Positive fanaticism, Neutral, Negative fanaticism, and Very Negative Fanaticism.

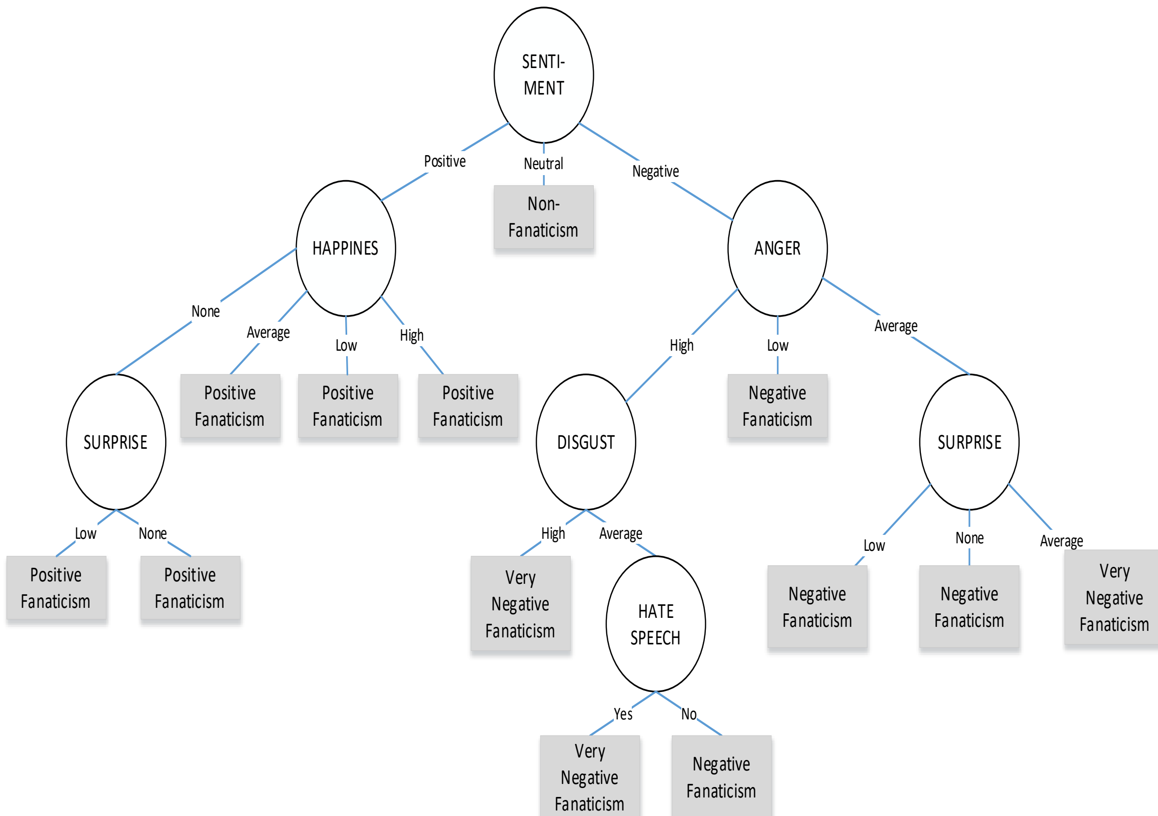

Figure 1. Fanaticism Categorizati

Information on Figure 1.

Each node of the tree structure represents an emotional attribute. Each branch of the tree structure represents the value. The leaves represent the fanaticism category (class). The root is on the top node of the tree structure. As root node (level 0) is sentiment analysis. This means that to check the fanaticism category in an opinion text, the first step is to compute the polarity of the sentiments. Sentiment polarity is divided into three levels: positive, neutral, and negative. If the result is negative, tracking trees shifts to level 1, namely the anger node. To classify the category of fanaticism must know Anger's polarity on that opinion. There are three polarities for Anger, namely high, average, and low. If the polarity is high, tracking the decision tree shifts to level 2, namely a disgust node. To classify the category of fanaticism must know the polarity of disgust. There are two polarities for disgust, namely high and average. If the polarity is average, then tree tracking shifts to level 3, namely hate speech, to detect the polarity of hate speeches in that opinion. The polarity of hate speech has two kinds, namely yes or no. Yes, means hate speech in the text. No means there are no hate speeches. If the polarity of hate speech is yes, then sentences will have total polarity:

Example 1 : “Ayo dukung Y… semoga Tuhan melindungi anda pak Y… saya suka ide anda!!”
Feature :sentiment = positive and happiness = high
Result: fanaticism category: positive

Example 2 : “Saya benci Pak X…, sungguh sungguh suka inkar janji, dasar pengecut!!”
Feature :sentiment = negative and anger = high and disgust = average and hate speech = yes
Result: fanaticism category: very negative

II. LITERATURE REVIEW

II.1. Fanaticism

The terms fanaticism and fanatic has been widely known in the political world, come from the Latin adverb fānāticē (frenziedly, ragingly) and the adjective fānāticus (enthusiastic, ecstatic, furious)(Marima,2011) .Webster's dictionary explains that the term “fanatic” as “a person with an extreme and uncritical enthusiasm or zeal, as in religion or politics”. Fanaticism originates in the thoughts of the perpetrator and is then manifested into actions both words and deeds (Marima, 2011). Fanatics will think of what they believe to be the ultimate truth. Examples are political and religious choices. They cannot accept beliefs, criticism, or differing opinions. We can see many forms of fanaticism around us the majority related to religion and politics. From the last Presidential election in Indonesia, there were many people devoted themselves to one of the candidates and they could do anything (such as in social media) to show their support (Dewi and Aminulloh, 2016).

II.2. Research About Sentiment and Emotion Analysis

Research about sentiment, emotion, and hate speech analysis using machine learning in the Indonesian language is already available such as automatic authority classification for Twitter text in Indonesia as part of the complaint management system (Laksana and Purwarianti, 2014). Research for predicting the Indonesian stock market using simple sentiment analysis using Naïve Bayes and Random Forest algorithm (Cakra and Trisedya, 2015), research about pornography used on text and image uses three machine learning methods namely Decision tree, Naive Bayes and SVM, then comparing which method is the best in the classification process (Barfian, et.al, 2017), explore the use of Random Forest for sentiment classification in the Indonesian language (Fauzi, 2018) , A case study to rank the popularity of online shopping sites in Indonesia uses the Naïve Bayes Classifier (NBC) (Wardani, et al, 2019), sarcasm detection to improve sentiment analysis results in terms of accuracy, precision, and recall. The method for detecting and extracting sarcasm features uses unigram and punctuation-relate features. This research also uses lexical and syntactic features and top word features. The process of sarcasm detection uses the Random Forest algorithm. Feature extraction for sentiment analysis using TF-IDF and its classification using the Naïve Bayes (Yunitasari, et al, 2019). The use of tree-based ensemble machine learning for sentiment analysis reaches an accuracy of 88,8% (Khomsah and Ariwibowo, 2020).

There are studies that categorize fanaticism into 3 levels, namely Code Attitude fanaticism, Code Red Fanaticism, and Non-Fanaticism. This study followed by a study about fanaticism classification using Case-Based Reasoning (CBR) and Naïve Bayes Classification. Accuracy reached 77% (Almonayyes, 2006). This research was followed by a study on fanaticism detection on the text document from the questioner. The accuracy reached 72% (Kléma and Almonayyes, 2006) . This research only categorizes fanaticism in a single polarity that is negative and neutral. This research was followed by a study on fanaticism detection using an in-house collected article from online Arabic newspaper and channel archives. Fanaticism categories are only fanatic and non-fanatic. Classification accuracy reaches 92% (Almonayyes, 2016) . This research continued in 2017 about the detection and classification of fanaticism in texts from the Arabic-language Twitter media. The maximum accuracy obtained is 82,1% (Almonayyes, 2017).

III. RESEARCH METHODOLOGY

There is a set of opinions from social media T that contain many sentences S. Each sentence S has several words W. Formulated that T = {S1, S2, S3, .., Sn} and Sx = {W1, W2, W3 .., Wy }, where n = counts sentences in T and y count words in every sentence Sx. The stages for processing T are: preprocessing stage, classification stage, and accuracy test stage.

III.1. Preprocessing.

Before preprocessing, opinions must detect the existence of fanaticism objects. The object of fanaticism must exist in sentences. The object must be determined and registered as a fanaticism object. In political fanaticism, the object of fanaticism is the name of a political figure and his alias. the list of names of political fanaticism objects and their aliases is in Table 1.

Table 1. Alias Dictionary of Fanaticism Object

No	Aliases	Object
1	Sandi	Sandiaga Uno
2	Sandiwarauno	Sandiaga Uno
3	Prabowo	Prabowo Subianto
4	Joko	Joko Widodo
5	Pakde	Joko Widodo
6	Bapake	Joko Widodo
7	pakde jokowi	Joko Widodo

Data pre-processing tasks are eliminating unstructured text, converting text into words that are easily processed by the system, and deleting the unimportant text data. Pre-processing is crucial in sentiment analysis because social media mostly contains unstructured words. There are several sub-tasks of pre-processing for sentiment analysis:

1. **Tokenizing**. The cutting stage of opinion is based on each word that compiles it.
2. **Remove Punctuation**. Delete all non-alphabetic characters such as symbols, spaces, and others.
3. **Remove Username**. Remove user names that usually start with the “@” symbol.
4. **Remove Hashtag**. Remove symbol “#” that usually used as a topic of conversation.
5. **Clean Number**. Remove unimportant numbers in front of and behind the word.
6. **Clean One Character**. Delete a word because it does not contain a clear meaning.
7. **URL Removal**. URL of the Twitter data has no meaning.
8. **Remove RT**. RT is the symbol “@” before the user name in question. It needs to be deleted.
9. **Convert Number**. The frequent use of slang words such as "s4y", "s4d", "d4d" and others. The number must be converted to letters.
10. **Remove Stop Word**. Stop words are unimportant words such as time, conjunction, and others. The process requires a dictionary.
11. **Convert Non-Standard Word**. Convert non-standard sentences, such as slang words. Requires a slang language dictionary
12. **Convert Emoticon**. Emoticon symbols on Twitter must be converted to meaningful words. Requires a dictionary of emoticons

So if applied to T, the preprocessing pseudocode is as follows:

```
T = Dataset
S = Sentences in Dataset
1) START
2) T=READ(data testing)
3) N=number Record of T
4) X=0
5) FOR i=1 to N
   a. Subject=GET Subject(S[i])
   b. IF Subject THEN
       i. INC(X)
      ii. SCLean[X] = PREPROCESS(S[i])
     iii. TClean=ADD(SClean[X])
   c. END IF
6) NEXT i
7) RETURN clean
```

The result of the data cleaning stage is clean, which is a net opinion set that has an X-record fanaticism object in which count X <= N.

III.2. Classification Process

Indecision trees (Figure 1), for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node. We continue comparing our record’s attribute values with other internal nodes of the tree until we reach a leaf node with a predicted class value. As we know how the modeled decision tree can be used to predict the target class or the value. Now let's understand how we can create the decision tree model. The classification process is described in the following algorithm.

```
 1  START
 2  T=READ (TClean)
 3  N=record number of T
 4  X=0
 5  FOR i=1 to N
 6      S=READ(T[i])
 7      DTREE=READ(FanaticKnowledge)
 8      ThisNode=READ(DTREE[ROOTNODE])
 9      WHILE ThisNode<>LEAF
10          MLType= ThisNode
11          POLARITY=ANALYIS(S, MLType)
12          ThisNode =DTREE[ThisNode.Branch[POLARITY].NextNode]
13      ENDWHILE
14      Category[i]=DTREE.Node
15  NEXT i
16  RETURN Category
```

The algorithm above is explained as follow :

Line 2 is the process of reading a clean dataset from TClean and store to variable T. Line 5 until 15 is the process of repeating a number of N records T. In the looping process (lines 6 until 14), the first step is to read sentences T on the record I and store it to S. The next step is to read the knowledge (decision tree) and store it to DTREE variable.
Followed by reading ROOT Node from DTREE (5c) and stored in this node variable. While looping is used to process the polarity analysis based on the Node. The first step is to determine the type of machine learning (MLType = ThisNode). The machine learning process is in the analysis of sentence S using MLType machine learning. Polarity results are stored in the Polarity variable. The node is shifted towards the next node according to the value of the polarity calculated. While looping will be done until this node is the tip of the tree (leaf). The leaf has no branch anymore and is a category of fanaticism (5f).

III.3. Classification Model

The classification model is combined with several machine learning methods to produce one optimal predictive model. There are several machine learning methods used for the classification of each node in the decision tree. Machine learning tested in this study are:

(1) K-Nearest Neighbor (K-NN)
    K-NN classifying objects based on learning data that is the closest distance to the object.
(2) Naive Bayes (NB)
    The Naive Bayes classifier is a machine learning method that uses probability and statistical calculations to predict the probability of a category based on existing data.

(3) Decision Tree (DT)
A decision tree is a classification method that uses an inverted tree-like structure, where the top node represents the root, each subsequent node represents the attribute and the branch represents the attribute value, and the leaves represent the class.

(4) Random Forest (RF).
Random forest (RF) is an algorithm for data classification by combining trees and training on the data samples that are owned. The use of trees (trees) more and more will affect the accuracy for the better. Determination of the classification using random forest is taken based on the voting results from the small trees formed.

Testing on all four machine learning is on the accuracy of the classification results on T that have been manually labeled by experts. So for documentation during testing, the classification results are done by completing Table 2.

Table 2. List of Testing Machine Learning Methods for Political Fanaticism Detection

No. Test	Sentiment	Anger	Disgust	Happiness	Surprise	Hate speech
1.	K-NN	K-NN	K-NN	K-NN	K-NN	K-NN
2.	NB	NB	NB	NB	NB	NB
3.	DT	DT	DT	DT	DT	DT
4.	RF	RF	RF	RF	RF	RF

Table 2 explains that each testing phase involves several machine learning methods. In one testing phase, not all machine learning will be done depending on the decision tree in Figure 1.

III.4.Classification Accuracy

III.4.1.Confusion Matrix

In evaluating the algorithm performance of Machine Learning (ML), this study uses the Confusion Matrix as a reference. The Confusion Matrix represents predictions and actual (actual) conditions of the data generated by machine learning algorithms. A confusion matrix is often used to describe the performance of a classification model. This method can be used to visualize the performance of an algorithm. The confusion matrix contains a summary of the predicted results on the classification. The number of predictions of true and false classifications is summarized by adding the value distributed by each class. The confusion matrix for testing is as in Figure 2

Table 3. Confusion Matrix for Each Experiment Based On Table 2.

Parallel Machine Learning Model Number: 111111

		Correct Labels			
		Positive Fanaticism	**Neutral**	**Negative Fanaticism**	**Very Negative Fanaticism**
Classified Result (Prediction)	**Positive Fanaticism**	TP	FP	FP	FP
	Neutral	FNe	TNe	Fne	Fne
	Negative Fanaticism	FN	FN	TN	FN
	Very Negative Fanaticism	FVN	FVN	FVN	TVN

Information of the terms:

TP: True Positive. The real label is positive and the classification result is also positive.
TNe: True Neutral. The real label is neutral and the classification result is also neutral.
TN: True Negative. The real label is negative and the classification result is also negative.
TVN: True Very Negative. The real label is very negative and the classification result is also very negative.
FP: False Positive. The real label is positive but the classification result is not positive.
FNe: False Neutral. The real label is neutral but the classification result is not neutral.
FN: False Negative. The real label is negative but the classification result is not negative.
FVN: False Very Negative. The real label is very negative and the classification result is not very negative.

III.4.2. Classification Rate/Accuracy
To measure the classification rate, we use formula (1) :

$$Accuracy = \frac{TP+TNe+TN+TVN}{TP+TNe+TN+TVN+FP+FNe+FN+FVN} \quad .....(1$$

The classification model that has the highest accuracy value will be used as a final classification model and is used for the classification of testing data.

IV. FINDING AND DISCUSSION

The experiment was carried out on opinion data obtained from video comments on the presidential candidate debate on Youtube. There are 5 debates for the presidential candidates used. Each presidential candidate debate is carried out by data crawling, preprocessing data, labeling, detection of subjects or objects of fanaticism, then feature extraction. The number of testing data tested is 5000 records for each presidential candidate debate. The dataset is selected randomly.

The machine learning process is calculated using the four methods above (K-Nearest Neighbor, Naive Bayes, Decision Tree, and Random Forest). The most optimal accuracy calculation results are using the Random Forest method with the accuracy in Table 4.

Table 4. Accuracy Result of Four Machine Learning

Dataset Number	Data Set Source	Number of Comments	Accuracy			
			K-NN	**Naïve Bayes**	**Decision Tree**	**Random Forest**
1	Debate I	5000	74,4%	78,85%	79,23%	80,22%
2	Debate II	5000	73,4%	77,23%	78,56%	80,09%
3	Debate III	5000	68,6%	75,32%	76,45%	79,85%
4	Debate IV	5000	78,3%	76,33%	77,33%	81,00%
5	Debate V	5000	78,5%	74,22%	75,33%	80,09%

Based on Table 5 above, it can be concluded that the Random Forest method provides the best accuracy in fanaticism classification. The best accuracy on the Debate IV dataset is 81.00%. For other datasets, Random Forest also provides better accuracy results than other machine learning methods. The next best machine learning methods are the Decision Tree, Naïve Bayes, and K-NN. The K-NN and Naïve Bayes methods have less than optimal performance due to an imbalance of labels in the dataset.

V. CONCLUSION

This study aims to know how to develop an effective and efficient model for classifying fanaticism using knowledge from a previous study. The best method for the classification process is the Random Forest method. The use of decision trees as an appropriate knowledge and search strategy will provide process efficiency because not all types of attributes (sentiments, emotions and hate speech) must be tested. This will speed up the classification process for many sentence opinions. The methods chosen for classification are the best method for each attribute of fanaticism, which is the highest accuracy.

VI. REFERENCES

A. S. Aribowo, H. Basiron, N. S. Herman, and S. Khomsah, "Fanaticism category generation using tree-based machine learning method," *Journal of Physics: Conference Series*, vol. 1501, no. 1, 2020, DOI: 10.1088/1742-6596/1501/1/012021.

K. Marimaa, "The Many Faces of Fanaticism," in *ENDC Proceedings*, 2011, vol. 14, pp. 29–55.

S. I. Dewi and A. Aminulloh, "Social Media : Democracy in the Shadow of Fanaticism," in *The 3rd Conference on Communication, Culture and Media Studies*, 2016, pp. 79–88.

J. Laksana and A. Purwarianti, "Indonesian Twitter Text Authority Classification For

Government in Bandung," *International Conference of Advanced Informatics: Concept, Theory, and Application (ICAICTA) Indonesian*, pp. 129–134, 2014.

Y. E. Cakra and B. D. Trisedya, "Stock Price Prediction using Linear Regression based on Sentiment Analysis," in *ICAICS*, 2015, pp. 147–154.

E. Barfian, B. H. Iswanto, and S. M. Isa, "Twitter Pornography Multilingual Content Identification Based on Machine Learning," in *International Conference on Computer Science and Computational Intelligence*, 2017, vol. 116, pp. 129–136, DOI: 10.1016/j.procs.2017.10.024.

M. A. Fauzi, "Random Forest Approach for Sentiment Analysis in Indonesian Language," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 1, pp. 46–50, 2018, DOI: 10.11591/ijeecs.v12.i1.pp46-50.

F. Kusuma Wardani, R. Valentinus Hananto, and V. Nurcahyawati, "Analisis Sentimen Untuk Pemeringkatan Popularitas Situs Belanja Online Di Indonesia Menggunakan Metode Naive Bayes," *JSIKA*, vol. 08, no. 01, pp. 1–9, 2019.

Y. Yunitasari, A. Musdholifah, and A. K. Sari, "Sarcasm Detection For Sentiment Analysis in Indonesian Tweets," *Indonesian Journal of Computing and Cybernetics Systems*, vol. 13, no. 1, pp. 53–62, 2019.

S. Khomsah and A. S. Aribowo, "Model text-preprocessing komentar Youtube dalam bahasa Indonesia," *Rekayasa Sistem dan Teknologi Informasi, RESTI*, vol. 4, no. 4, pp. 648–654, 2020.

A. Almonayyes, "Multiple Explanations Driven Naive Bayes Classifier.," *Journal of Universal Computer Science*, vol. 12, no. 2, pp. 127–139, 2006.

J. Kléma and A. Almonayyes, "Automatic Categorization of Fanatic Text Using Random Forests," *Kuwait Journal of Science and Engineering*, vol. 33, no. 2, pp. 1–18, 2006.

A. Almonayyes, "Classifying Documents By Integrating Contextual Knowledge With Boosting," in *International Conference on Artificial Intelligence and Computer Science*, 2016, no. November, pp. 28–29.

[A. Almonayyes, "Tweets Classification Using Contextual Knowledge And Boosting," *International Journal of Advances in Electronics and Computer Science*, no. 4, pp. 87–92, 2017.