Detection of Hoax Content on Social Media in Indonesia using a *Levenshtein Distance* Method

Frista Gifti Weddiningrum[1], Anang Kunaefi[1] and Noor Wahyudi[1]
Information System, Islamic State University of Sunan Ampel Surabaya
Ahmad Yani 117, Surabaya, Indonesia

Keywords: Hoax, Levenshtein Distance, Tf-Idf, Pre-processing, Detection System, Social Media.

Abstract: Social media is a very supportive means to strengthen communication among fellow human beings. However, not all information disseminated through social media is a fact. There have been various cases of dissemination of news which are not facts or often called hoaxes. The development of anti-hoax technology has sprung up, but anti-hoax technology applied in the hoax detection system is still rarely found. In this study, Tf-Idf calculations were used to measure the weight of a word in a hoax document and the *Levenshtein Distance* (LD) method was used to measure the distance between words in a document. The application of the *Levenshtein Distance* Method in the Hoax Detection System has several steps, started with word pre-processing, followed by the Tf-Idf calculation phase, and then the calculation phase of the minimum inter-word distance using *the Levenshtein Distance* method. The results of the 0.0014 limit in the testing scenario have training data as many as 100 news indicated as hoaxes and 40 news as test data. The 0.0014 limit has a consistent value of *Precision, Recall,* and *Accuracy.*

1. INTRODUCTION

Social media is a very supportive means to strengthen communication among fellow human beings. Distance and time do not become a barrier to communicate with one another. Not only does social media act as a medium of communication, but it also functions as a medium for disseminating information. Information spread through social media will be quickly consumed by every account owned by people.

Sharing information with others is a positive thing, but not all information disseminated through social media is a fact. There have been various cases of dissemination of news which are not facts or often called hoaxes.

As reported by the CNN Indonesia website, the data presented by the Ministry of Communication and Information stated that there were 800 thousand websites in Indonesia indicated as disseminators of fake news and hate speech. In fact, the Indonesian Government has issued a regulation in Article 28 paragraph 1 of Law No. 11 of 2008 on Information and Electronic Transactions or the ITE Law.

As the growing trend of hoaxes that poisons the news, especially on social media, there are also emerging thoughts to take precautions against the spread of fake news. There have been many tips to avoid getting caught up in fake news, and many social media platforms provide additional services to report contents that are thought to contain elements of hoaxes and SARA – racist or sectarian sentiment. For the development of anti-hoax technology, there are also some that have emerged, but anti-hoax technology applied in hoax detection systems is still rarely found. Some systems use artificial intelligence to determine whether news contains hoaxes or not, and some use text comparison algorithms.

There was a previous study that used the self-organizing map and Feed Forward Neural Network methods to detect English hoax content (Vuković, Pripužić, & Belani, 2009). The system created in the study can distinguish new fake emails and classify them by comparing their pattern with the same stored pattern. However, if there is an e-mail with a new pattern, the system cannot distinguish it because the

Weddiningrum, F., Kunaefi, A. and Wahyudi, N.
Detection of Hoax Content on Social Media in Indonesia using a Levenshtein Distance Method.
DOI: 10.5220/0008905600002481
In *Proceedings of the Built Environment, Science and Technology International Conference (BEST ICON 2018)*, pages 255-261
ISBN: 978-989-758-414-5

pattern is not previously in storage. In another study, *Levenshtein Distance* was used to detect hoax content in English emails. The study showed a positive predictive value of 0.96. Nonetheless, the system has not been able to identify the original email. All hoax contents in the e-mails will be measured (Ishak, Chen, & Yong, 2012).

In this study, the *Levenshtein Distance* (LD) method will be used to measure the amount of difference in each document being processed, so that the final result will be a limit that can classify the news whether it is a hoax or not. Based on the previous study, there are several stages to detect hoax content in an article. The first stage is word pre-processing to filter important and influential words in an article, then the feature extraction stage is used to give weight to each filtered word in order to know the word that has a big influence on an article and the last one is the classification stage (Rasywir & Purwarianti, 2015). This study also applies word pre-processing which begins with a stemming process, then removal of stop words, and the last is a lexical analysis. For the feature extraction stage, this study uses Tf-Idf as a weighting calculation for each word, and for classification, this study uses the Levenshtein Distance algorithm.

2. LITERATURE STUDY

There are several literature study in this research.

2.1 Text Preprocessing

Text Preprocessing has several stages (Katariya & Chaudhari, 2015) :

1. Lexical Text Analysis
   It is the process of converting a text or sentence into words, aiming to identify words in a text.
2. Removal of Stop Words
   A stop word is a common word that is often used in a text and usually has no use when used for search purposes. One of its examples is conjunctions, such as 'and', 'or', and 'but'. The removal of stop words has an important benefit – to reduce the size of the index used later.
3. *Stemming*
   Stemming is the process of separating a word that contains a prefix or suffix to produce its basic form. This is useful for improving word retrieval performance because it will reduce the same word variant in general concepts. In addition, the stemming process is also useful to reduce the size of the indexing structure because the number of different index terms dwindles.

2.2 TF-IDF

The *Term Frequency Inverse Document Frequency* or commonly referred to as TF-IDF is an algorithm used to measure the weight of each word in a document or even a set of documents. The weight will represent the importance of a word in a document. The greater the weight value gets, the more important the role of the word has in forming a document. TF (Term Frequency) will calculate the frequency of occurrence of a word, and compare it with the number of all words in the document. The following is the equation used to calculate TF (Saadah, Atmagi, Rahayu, & Arifin, 2013).

$$tf(i) = \frac{freq\ (t_1)}{\sum freq\ (t)} \quad (1)$$

Details:

tf(i) : The *Term Frequency* value of a word in a document.

freq (t_i) : The occurrence frequency of a word in a document.

$\sum freq\ (t)$: The total number of words in the document.

Meanwhile, IDF (*Inverse Document Frequency*) calculates the logarithm of the total number of documents and compares them with the number of documents in which the intended word (*t*) occurs. The following equation is used to calculate IDF (Saadah et al., 2013).

$$idf(i) = log \frac{|D|}{|\{d: t_i \in d\}|} \quad (2)$$

Details:

idf(i) : The value of *Inverse Document Frequency* of a word in the entire document.

|D| : The total number of documents.

|(d: $t_i \in$ d}| : The number of documents containing the word (t).

2.3 Levenshtein Distance

Levenshtein Distance is a matrix to measure the number of differences between 2 strings. The distance between strings is measured by the number of added letters, letter deletion, or letter replacement needed to change the source string into a target string (Ishak et

al., 2012). The following is the matrix of Levenshtein Distance (Afriansyah, & Puspitaningrum, 2015).

$$leva,b(i,j) = \begin{cases} \max(i,j)\ if\ \min(i,j) = 0 \\ min\begin{cases} leva,b(i-1,j)+1 \\ leva,b(i,j-1)+1 \\ leva,b(i-1,j-1)+1(ai \neq bj) \end{cases} \end{cases} \quad (4)$$

Details:

lev a,b is the *levenstein distance* matrix;

i is a matrix line;

j is a matrix column.

After the results from the Levenshtein matrix above are obtained, the calculation of the number of the similarity values is performed by comparing strings using the following formula (Afriansyah et al., 2015)

$$Similarity = \left\{1 - \frac{editdistance}{maxLength(stra,strb)}\right\} \quad (5)$$

Details:

edit distance is the result of the comparison that has been done or Levenstein Distance.

maxLength is the number of strings from the longest word between stra and strb.

stra is the first-string length.

strb is the second-string length.

Similarity is the similarity value between the two strings.

2.4 Performance Measure

There is a set of formulas that can be used as a measurement medium that is in accordance with the study being carried out, namely *Precision, Recall* and *Accuracy*. *Precision and Recall* are calculation matrices used to measure the effectiveness of information retrieval (Manning, Raghavan, & Schütze, 2008).

- *Precision* (P) is document fragments from which relevant things are taken.

$$Precision = \frac{\#(hoax\ documents\ classified\ as\ hoaxes)}{\#(the\ number\ of\ documents\ classified\ as\ hoaxes)}$$

- *Recall* (R) is parts of relevant documents taken.

$$Recall = \frac{\#(hoax\ documents\ classified\ as\ hoaxes)}{\#(the\ number\ of\ hoax\ documents\ tested)}$$

The idea can be made clearer through the following Table 2.1:

Table 2.1 *Confussion Matrix*
(Source: Manning, Raghavan, & Schütze, 2008)

	Relevant		**Irrelevant**	
Taken	*true* (tp)	*positive*	*false* (fp)	*positive*
Not taken	*false* (fn)	*negative*	*true* (tn)	*negative*

Based on Table 2.1, the following formula can be written to calculate the accuracy of a system using the calculation of *Precision and Recall* (Manning et al., 2008):

$$P = tp / (tp + fp) \quad (6)$$
$$R = tp / (tp + fn) \quad (7)$$

In addition to *Precision and Recall*, the calculation of system performance also requires the calculation of system accuracy to ascertain how the system can be used to accurately detect hoaxes in the news content. The accuracy of a system can be calculated using the following equation (Syafitri, 2010).

$$ac = \frac{\sum match}{\sum tp} \times 100\% \quad (8)$$

Details:
ac : The level of accuracy (%)
$\sum match$: The number of correct detections
$\sum tp$: The amount of data tested

3. METODHOLOGY

To conduct the research, firstly we had to assure that our datasets are valid, in which the news are provenly hoaxes. Therefore, we harnessed news from Indonesia Anti-hoax Society's database through their website (https://turnbackhoax.id/). They have been collecting and clarifying hoaxes news since late of November 2016.

From the website, we grabbed 100 fake news during the years of 2017 as our sample datasets. There is no limitation on topics from the selected news. Additionally, we utilized another 40 news randomly selected from various social media in 2018 as testing datasets. The reason behind these numbers is limited

clarified news available in the website from the society.

We then performed the computation on the sample data based on our proposed method, which will be described in the following sub section, to get the bound number, which will be used as a classifier to assess whether the test data is hoax or not.

3.1 System Architecture

In the system flow (Figure 3.1), the text pre-processing for the new news text input is performed. This system uses the results of the Tf-Idf calculation to calculate the word weight and uses *Levenshtein Distance* calculation to calculate the distance between words compared. Tf-Idf calculation results are obtained through target data. The results of word distance calculation are acquired by comparing two words. The first word is obtained through the news entered in the system, referred to as the source word. The second word is obtained from the target data, hereafter referred to as the target word.

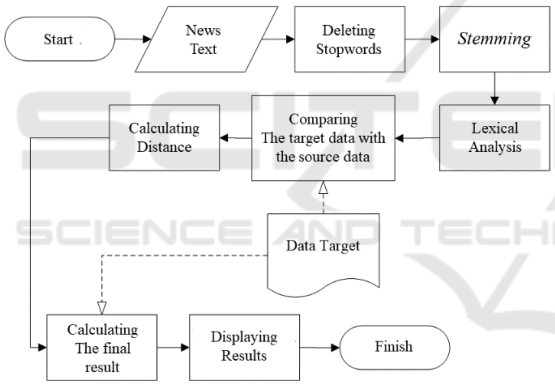

Figure 3.1 The Flow Chart of A Hoax Detection System

The comparison of the two words uses *Levenshtein Distance* calculation by calculating the number of attempts made to change a source word into a target word. The intended attempts are in the form of letter deletion, letter addition, and letter substitution. If there are 2 letters deleted in the source word, and 1 letter is changed to another letter so that it becomes the same as the target word, the *Levenshtein Distance* calculation value is 3 because there are 3 attempts to change the source word into the target word. When the distance between the two words has been found, the system will calculate their similarity value using the *similarity* formula. The similarity value of the words will then be multiplied by the weight value (Tf-Idf) and the final result obtained from a source word will be processed. The last result of this system is the result of calculation of all the average final results of each source word compared.

3.2 System Testing And Analysis

In the analysis stage, the system will be assessed by measuring the accuracy of the system using *Precision and Recall* and Measurement Accuracy. The precision and accuracy will be seen in the system through the calculation. Based on the formula specified in section 2, the formula can be further clarified by entering the element of calculation based on the data produced in the test. Here is a further explanation when the formula includes elements that need to be calculated based on the test data.

- *Precision* (P) measures the precision of the system for classifying hoax and non-hoax document types.

$$Precision = \frac{\#(hoax\ documents\ classified\ as\ hoaxes)}{\#(the\ number\ of\ documents\ classified\ as\ hoaxes)}$$

- *Recall* (R) measures the precision of the system for producing relevant values so that documents can be classified.

$$Recall = \frac{\#(hoax\ documents\ classified\ as\ hoaxes)}{\#(the\ number\ of\ hoax\ documents\ tested}$$

The idea can be clarified through the following Table 3.1:

Table 3.1 Confusion Matrix Based on Test Data

Actual / Prediction	Hoax Document	Non-Hoax Document
Classified as *Hoax*	*true positive* (tp)	*false positive* (fp)
Classified as *Non-Hoax*	*false negative* (fn)	*true negative* (tn)

Then, the following is the accuracy formula:

$$ac = \frac{\sum match\ (tp+tn)}{\sum tp} \times 100\% \qquad (8)$$

Details:

ac : the level of accuracy (%)

$\sum match$: the number of hoax documents classified as hoaxes, and non-hoax documents classified as non-hoaxes.

$\sum tp$: the number of data tested.

4. RESULTS AND DISCUSSION

We will explain about the result in this section.

4.1 Results

The testing scenario uses the test data from 40 news, divided into 20 non-hoax news and 20 hoax news. The following is Table 4.1 which shows the test results from 40 news.

Table 4.1 Test Results from 40 News

Non-Hoax News	Results	Hoax News	Results
doc 1	0.00156626	doc 1	0.001479
doc 2	0.00178153	doc 2	0.002889
doc 3	0.001337517	doc 3	0.001638
doc 4	0.001302774	doc 4	0.001314
doc 5	0.001314324	doc 5	0.001805
doc 6	0.001168763	doc 6	0.001895
doc 7	0.001172135	doc 7	0.001567
doc 8	0.001028519	doc 8	0.001389
doc 9	0.001399543	doc 9	0.001581
doc 10	0.001565115	doc 10	0.001281
doc 11	0.001177235	doc 11	0.001309
doc 12	0.001232402	doc 12	0.001378
doc 13	0.001580199	doc 13	0.00106
doc 14	0.001221855	doc 14	0.001498
doc 15	0.00091315	doc 15	0.001887
doc 16	0.001092715	doc 16	0.001939
doc 17	0.001940282	doc 17	0.001811
doc 18	0.001311824	doc 18	0.00163
doc 19	0.001590698	doc 19	0.001768
doc 20	0.001206414	doc 20	0.001442

In the stage of System Result Analysis, three analyses will be determined to decide the value of *Precision, Recall & Accuracy*, namely 0.0013 limit, 0.0014 limit, 0.0015 limit of the hoax detection system. Each of these limit numbers will produce a different precision and recall value. The following is a description of the classification and component value of the calculation of *Precision, Recall & Accuracy* for each limit number. Table 4.2 depicts the value of the calculation component with a limit of 0.0013.

Table 4.2 the Value of Calculation Components with A Limit of 0.0013.

	0,0013 limit
True Positive	18
False Positive	11
False Negative	2
True Negative	9

The calculation of the above components shows the test results with a limit of 0.0013, hence the values of *Precision*=0.62; *Recall*=0.9; and *Accuracy*=68%. Table 4.3 shows the value of the calculation component with a limit of 0.0014.

Table 4.3 the Value of Calculation Component with A Limit of 0.0014

	0,0014 limit
True Positive	14
False Positive	6
False Negative	6
True Negative	14

The calculation of the above components shows the results with a limit of 0.0014, hence the values of *Precision*=0.7*; Recall*=0.7*;* and *Accuracy*=70%. Table 4.4 describes the value of the calculation component with a limit of 0.0015.

Table 4.4 the Value of Calculation Cmponent with A Limit of 0.0015

	0,0015 limit
True Positive	11

False Positive	6
False Negative	9
True Negative	14

From the results of the calculation above, thus, the values of *Precision*=0.65*; Recall*= 0.55*;* and *Accuracy*=63%.

4.2 Discussion

Based on the results of the testing scenario, a limit of 0.0014 reached highest value for the *Precision* and *Accuracy* which is equal to 70%. Although the result is sufficient, we think that it still needs much improvement.

There are certain conditions that we think greatly influence the result previously discussed. The number of datasets, which were 100 samples of fake news, is one of the factors that might contribute to the result. We believed that the result will be much better if the sample data is greater than 100, at least 500 data samples. This is also become our notable challenge if we want to make a better system.

Another factor that might significantly contribute to the result is the topics of the news in the datasets. Since there are no limitations in the topics, the words stored in the dictionary are dispersed. It is possible because the hoax news library stores the news with different topics, but it does not store news that is continuously distributed, so that when the news is compared with other news that also has different topics, the effect will not be as great as that of the news continuously used as a hot topic to spread lies. This is also very influential in calculating the weight of words.

The weight for each word then is less significant and becomes useless for the next process. In the Tf-Idf computation, the number of occurrences of words in the stored documents is extremely influential. The large number of words, their occurrence, and the number of documents used greatly affect the weight of calculation of a word compared. Hence, limiting the topics may significantly help the computation of Tf-Idf for each word in the system because the words become sufficiently homogeneous. For example, political news should be separated from entertainment or musical news, and different computation should be performed for each topic in order to obtain the best result.

In hoax news storage documents, there are several topics that are often used as targets for hoax news dissemination. However, the news is very specific to politics, so it greatly influences the existence of documents with hoax news that rarely becomes the reporting target and its number does not dominate. For the test data, not all news in the test data discusses political topics, so there are some *false negative* and *false positive* values that affect the final results of *Precision, Recall* and *Accuracy*.

For future works, we hope that we will be able to collect more fake news with certain topics separated from the others to assure the homogeneous words in the datasets. In addition, we will try different methods to obtain better result.

5. CONCLUSION

Based on the research findings on Hoax Detection on Social Media in Indonesia using the *Levenshtein Distance* Method, it can be concluded that:

1. There are some steps to apply the *Levenshtein* Distance Method in a Hoax Detection System:

   a. The creation of Target Data Documents in which there is a simplified collection of hoax words in pre-processing words and selection of pre-processing words by giving weight to each word using Tf-Idf.

   b. The creation of a Hoax Detection System in which there are several processes to produce classification values – pre-processing the source word, comparing the source word with the target word, calculating the distance (Levenshtein Distance), giving weight (Tf-idf), and calculating the final result with its classification.

2. The application of the Levenshtein Distance method combined with Tf-Idf is proved to be able to distinguish hoax and non-hoax news with a fairly good level of accuracy.

3. The testing scenario with 0.0014 limit, which has training data as many as 100 news indicated as hoaxes and 40 news as test data, was divided into two, that are 20 non-hoax news and 20 hoax news, and had consistent values of *Precision* 0.7, *Recall* 0.7, and *Accuracy* 70%. This means that the more hoax words are used as training data, the more accurate the system performs detection.

REFERENCES

Afriansyah, Z., & Puspitaningrum, D. (2015). MENGGUNAKAN ALGORITMA LEVENSHTEIN DISTANCE (Studi Kasus : DNA Kanker Hati Manusia), *3*(2), 61–67.

Ishak, A., Chen, Y. Y., & Yong, S. P. (2012). Distance-based hoax detection system. *2012 International Conference on Computer and Information Science, ICCIS 2012 - A Conference of World Engineering, Science and Technology Congress, ESTCON 2012 - Conference Proceedings*, *1*, 215–220. https://doi.org/10.1109/ICCISci.2012.6297242

Katariya, N. P., & Chaudhari, M. S. (2015). Text Preprocessing for Text Mining Using Side Information, *3*, 3–7.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. *Computational Linguistics* (Vol. 35). https://doi.org/10.1162/coli.2009.35.2.307

Rasywir, E., & Purwarianti, A. (2015). Eksperimen pada Sistem Klasifikasi Berita Hoax Berbahasa Indonesia Berbasis Pembelajaran Mesin. *Jurnal Cybermatika*, *3*(2), 1–8.

Saadah, M. N., Atmagi, R. W., Rahayu, D. S., & Arifin, A. Z. (2013). Sistem Temu Kembali Dokumen Teks dengan Pembobotan Tf-Idf Dan LCS. *Jurnal Ilmiah Teknologi Informasi (JUTI)*, *11*(1), 17–20. https://doi.org/10.12962/j24068535.v11i1.a16

Vuković, M., Pripužić, K., & Belani, H. (2009). An intelligent automatic hoax detection system. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5711 LNAI*(PART 1), 318–325. https://doi.org/10.1007/978-3-642-04595-0_39