Plagiasi Prosiding Improving Text Preprocessing For Student

by 39 Perpustakaan UMSIDA

Submission date: 23-Jan-2024 04:23PM (UTC+0700)
Submission ID: 2276590869
File name: 2._Prosiding_Improving_Text_Preprocessing_For_Student.pdf (770.01K)
Word count: 2520
Character count: 13273

Improving Text Preprocessing For Student Complaint Document Classification Using Sastrawi

To cite this article: Mochamad Alfan Rosid *et al* 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **874** 012017

Improving Text Preprocessing For Student Complaint Document Classification Using Sastrawi

Mochamad Alfan Rosid*, Arif Senja Fitrani, Ika Ratna Indra Astutik, Nasrudin Iqrok Mulloh, Haris Ahmad Gozali

Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Jl. Raya Gelam 250 Candi Sidoarjo, Jawa Timur, Indonesia

*alfanrosid@umsida.ac.id

Abstract. In the text mining there are stages that must be passed namely the text preprocessing stage. Text preprocessing is the stage to do the data selection process in each document, including case folding, tokenizing, filtering, and stemming. The results of the preprocessing process can affect the accuracy of document classification. In documents Bahasa Indonesia, there are still often over-stemming and under-stemming, so improvements are needed in the stemming process. In this study, it is proposed to use sastrawi libraries to improve the results of previous studies that are still not optimal in the results of preprocessing, especially in the filtering and stemming process. From the results of the study, the sastrawi library is able to reduce over stemming and under stemming and a faster processing time compared to using a Tala stemmer.

1. Introduction

Text Mining is a method of mining unstructured data in the form of text located at sources such as documents, attachments or sentences that will later find the words of the document so that analysis can be done related to documents[1]. Text mining is usually used to retrieve important words that will be classified which will later be analyzed the relationship between documents. With text mining we can find out the information contained in a document that we did not know before.

The process of text mining is more or less the same as data mining, but there are different, data mining is designed to handle structured data while text mining can handle unstructured or semi-structured datasets such as HTML email files and text documents. Structured data is data that is in a fixed field in a record or file. This data is contained in relational databases and spreadsheets. Unstructured data usually refers to information that is not in a traditional row column database and that is the opposite of structured data. Semi-structured data is data that is not raw data, or data that is typed in a conventional database system[2].

Before carrying out operations on text documents, both for classification purposes, looking for similarities, sentiment analysis etc. then the text document must go through preprocessing[3], This stage aims to structure the text and extract features[4], in general there are four steps such as tokenization, stop-word removal and lower case conversion (case foding) and stemming[5]. The results of this preprocessing affect the level of classification accuracy and computational time required.

In a study conducted by Mochamad Alfan Rosid et al in 2015, produced a machine to classify student complaints using the centroid base classifier method and the tf-idf-icf feature with an accuracy rate of 79% [6], but in this study found weaknesses in the stemming process using Tala porters to process student complaint documents Bahasa Indonesia, there are terms that experience under stemming and over stemming, so this in my opinion greatly affects the level of accuracy. For this reason, another stemming method is needed to solve the problem, in this study we propose the use of sastrawi libraries, in addition to the stemming process we also apply to the stopword remover process.

Sastrawi library is a stemmer library that is used to overcome the problem of changing words with words into basic words[7]. Sastrawi stemmer applies an algorithm based on Nazief and Adriani, then enhanced with the CS (Confix Stripping) algorithm, the ECS Algorithm (Enhanced Confix Stripping), further enhanced with Modified ECS. It is expected that by applying this sastrawi library to the preprocessing process of student complaints documents, especially the stemming and stopword remover stages, can make significant non-important words so as to improve the accuracy of the classification of student complaints.

2. Literature Review

Porter's algorithm was first created by Martin Porter (1980), used for the English stemming process. Then this algorithm was developed by W.B. Frakes in 1992 to be implemented in Indonesian. Then Fadilah Z Tala develops again using five steps by simulating the inflection and derivation of words. At each step, a certain ending is deleted by substitution. The design of Porter Stemmer for Indonesian can be seen in Figure 1[8].

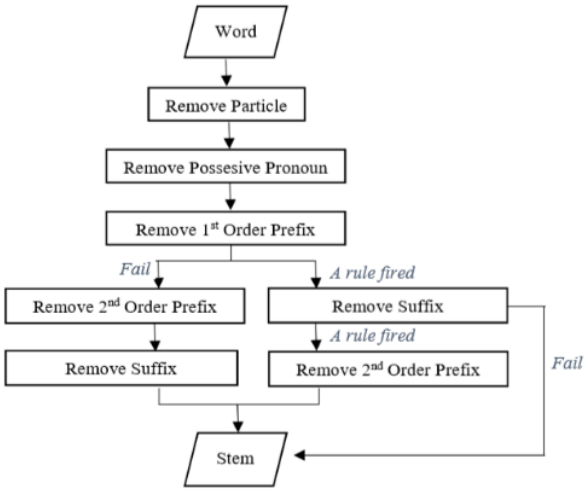

Figure 1: The Algorithm of a Porter stemmer for Bahasa Indonesia.

In this stemmer porter there is a shortage of high levels of over stemming, this is because the stemming process removes the longest first string in each step, potter stemming also produces many ambiguous words[9]. Besides these shortcomings, the porter algorithm also has a problem of dependence on the basic word dictionary in the stemming process.

The Nazief-Adriani algorithm was first developed by Bobby Nazief and Mirna Adriani. This algorithm is based on broad Indonesian morphological rules that are collected into a group encapsulated in the affixes that are allowed and the prefixes that are not allowed. This algorithm uses basic words to support recoding, which is rearranging words that get excess stemming. Compared to porters, this algorithm has a lower level of over stemming[7]. The Nazief and Adriani algorithms were further developed into sastrawi libraries with several improvements. In basic word formation, the sastrawi

library does not need a base word database like porters. Sastrawi libraries provides for the programming languages PHP, Python, Java, C, Go and Ruby. The steps in this algorithm are as follows:

1. The first step is to check whether the word is the root word (root) contained in the list of root words (root). We say the word is the root word, then the process is stopped at this first stage.
2. Eliminating Inflection Suffixes ("-lah", "-lah", "-ku", "-mu", or "-nya"). If the word is in the form of particles ("-lah", "-lah", "-tah" or "-pun") then this step is repeated again to remove the Positive Pronouns ("-ku", "-mu", or "-nya").
3. Eliminating the derivational suffix. Eliminate the affixes -i, -kan, -an.
4. Remove derivational prefixes. Eliminate the prefix be-, di-, ke-, me-, pe-, and te-.
5. If from step 4 above you haven't found it yet. Then analyze whether the word is included in the last column's diambiguity table or not.
6. If from step 4 above you haven't found it yet. Then analyze whether the word is included in the last column's diambiguity table or not.
7. If all the above processes fail, then the algorithm returns the original word.

3. Methods

Preprocessing method is a very important stage in the technique and application of text mining[10][11]. This is the first step in the text mining process. In this paper, we discuss the four main preprocessing steps, namely, Case Folding, Tokenizing, Stop-word removal and stemming (Figure 2). In this study the stop-word removal and stemming stages use sastrawi libraries.

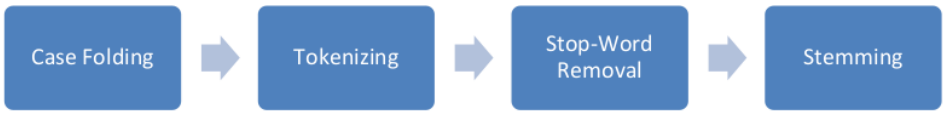

Figure 2: Text Mining Preprocessing Step

3.1. Case Folding

In a document that uses capital letters or the like sometimes it does not have in common, this can be due to writing errors. In text preprocessing the case folding process aims to change all the letters in a text document into lowercase letters, for example the word "Wifi" becomes "wifi".

3.2. Tokenizing

A text document consists of a set of sentences, the tokenization process breaks the document into parts of words called tokens[12].

3.3. Stop-Word Removal

The purpose of erasing words is to eliminate words that are not important, this process can reduce the dimensions of space that looks heavy. Common words in text documents such as clothing, prepositions and nouns etc., which do not give meaning to the document. Examples of the words "which", "and", "you", "until" are deleted from the document because they are not measured as keywords in a text mining application. In this study using sastrawi libraries. Figure 3 is a stop-word source code snippet.

```
require_once __DIR__ . '/vendor/autoload.php';
$stopwordFactory = new \Sastrawi\StopWordRemover\StopWordRemoverFactory();
$stopword  = $stopwordFactory->createStopWordRemover();
$hasilstop  = $stopword->remove($hasilcasefolding);
```

Figure 3: Source Code Stop-Word Removal Snippet

3.4. Stemming

Stemming is the process of mapping and breaking down the form of a word into its basic word form. Easier to say, the process of changing affixed words into basic words. This stemming process is the most important process in the text mining stage. Good stemming results can affect whether or not the text mining application. Figure 4 is a piece of source code using the sastrawi library. Sastrawi library for the PHP programming language can be installed with the following steps:

1. Open a terminal (command line) and navigate to your project directory
2. Download Composer so that the composer.phar file is in that directory.
3. Add sastrawi to your composer.json file:

php composer.phar require sastrawi/sastrawi:^1

```
require_once __DIR__ . '/vendor/autoload.php';
$stemmerFactory = new \Sastrawi\Stemmer\StemmerFactory();
$stemmer  = $stemmerFactory->createStemmer();
$hasilstem= $stemmer->stem($hasilstop);
```

Figure 4: Source Code Stemming Piece

4. Result and Discussion

The results of this study show a very significant difference between the tala porter algorithm and Nazief and Andriani algorithms that use sastrawi libraries, in terms of time, in our study the porter algorithm takes 241.6 seconds while the sastrawi library only takes 0.6 seconds. This long porter processing time is influenced by the use of a base word data dictionary taken from the base word database, whereas sastrawi does not use a base word database. Figure 5 preprocessing applications using tala stemmer porters, Figure 6 preprocessing applications using sastrawi libraries.

Figure 5: Preprocessing application with Tala Stemmer Porters

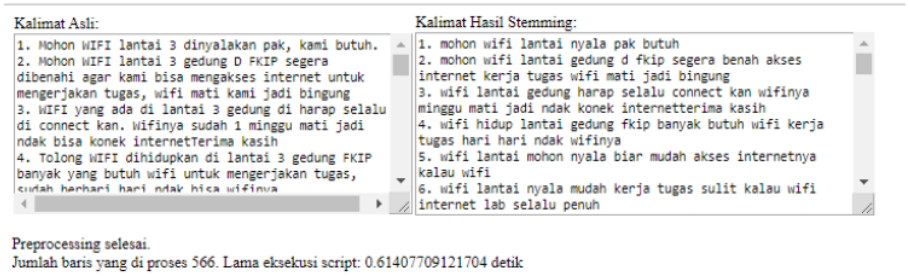

Figure 6: Preprocessing application with Sastrawi Library

From the results shown in Figures 5 and 6 it can also be seen that the results obtained at the porter still appear to be unreadable characters, while the character's sastrawi has been removed. In the research results also found several words that are over stemming in porters such as the word "Allah" into the word "al" while in sastrawi remains "allah" and in porters found under stemming "as soon as" in sastrawi is appropriate to produce the word "fast" etc. Table 1 shows the comparison obtained in 50 affixed words that are processed by the porter algorithm and sastrawi library in this study.

Table 1. Comparison of Porter Stemming Processes and Sastrawi Library

Stemmer Results Categories	Porter		Sastrawi	
	Total	Percentage (%)	Total	Percentage (%)
Exact Match	41	82	46	92
Unchange	2	4	3	6
Over Stemming	5	10	0	0
Under Stemming	2	4	1	2
Total	**50**	**100**	**50**	**100**

5. Conclusion

From the results of trials conducted with 50 samples of Indonesian text word-influenced text documents that have been selected by the author, resulting in the category of Exact Match stemmer results of 82% for Porter stemmer and 92% for Sastrawi, Unchange of 4% for Porter stemmer and 6% for Sastrawi, Under Stemming by 4% for Porter stemmer and 2% for sastrawi, and Overstemming by 10% for Porter stemmer and 0% for Sastrawi. Based on the processing time, sastrawi libraries are faster which is an average of 0.6 seconds while porters require an average time of 241.6 seconds. So it can be concluded that the use of sastrawi libraries in the preprocessing stage of student complaint documents produces better results compared to previous studies using the porter algorithm.

11
Acknowledgements

We hereby thank you to Universitas Muhammadiyah Sidoarjo for supporting the publication of this research.

References

[1] Jing L P, Huang H K and Shi H B 2002 Improved feature selection approach TFIDF in text mining *Proc. 2002 Int. Conf. Mach. Learn. Cybern.* **2** 944–6
[2] Vijayarani S, Ilamathi M J, Nithya M, Professor A and Research Scholar M P Preprocessing Techniques for Text Mining -An Overview **5** 7–16
[3] Durairaj M and Alagu Karthikeyan A 2018 Modified Porter's Algorithm For Pre -Processing Academic Feedback Data *Int. J. Pure Appl. Math.* **118** 3009–15
[4] Sugumar R and Priya M R 2018 Journal of Global Research in Computer Science Available Online at www.jgrcs.info IMPROVED PERFORMANCE OF STEMMING USING EFFICIENT STEMMER The resultant stemmer to which we will refer as Enhanced Porter Stemmer algorithm includes four **9** 1–5
[5] Rani Spr, Ramesh B and Anusha M 2015 Evaluation of stemming techniques for text classification *J. Comput. ...* **43** 165–71
[6] Rosid M A, Gunawan G and Pramana E 2015 Centroid Based Classifier With TF – IDF – ICF for Classfication of Student's Complaint at Appliation E-Complaint in Muhammadiyah University of Sidoarjo **1**
[7] Purnamasari K K and Suwardi I S 2018 Rule-based Part of Speech Tagger for Indonesian Language *IOP Conf. Ser. Mater. Sci. Eng.* **407**
[8] Tala F Z 2003 A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia *M.Sc. Thesis, Append. D* **pp** 39–46
[9] Magriyanti A A 2018 Analisis Pengembangan Algoritma Porter Stemming Dalam Bahasa Indonesia *Attrib. 4.0 Int.*
[10] Torayev A, Magusin P C M M, Grey C P, Merlet C and Franco A A 2019 Text mining

assisted review of the literature on Li-O 2 batteries *J. Phys. Mater.* **2** 044004
[11] Virmani D and Taneja S 2019 *A text preprocessing approach for efficacious information retrieval* vol 669 (Springer Singapore)
[12] Gupta G 2015 Text Document Tokenization for Word Frequency Count using Rapid Miner (Taking Resume as an Example) *Int. J. Comput. Appl.* 24–6

Plagiasi Prosiding Improving Text Preprocessing For Student

ORIGINALITY REPORT

13%	9%	7%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	trilogi.ac.id Internet Source	2%
2	Submitted to Sriwijaya University Student Paper	1%
3	Evaristus Didik Madyatmadja, Bernardo Nugroho Yahya, Cristofer Wijaya. "Contextual Text Analytics Framework for Citizen Report Classification: A Case Study Using the Indonesian Language", IEEE Access, 2022 Publication	1%
4	journal.portalgaruda.org Internet Source	1%
5	repository.dinamika.ac.id Internet Source	1%
6	Submitted to Universitas Amikom Student Paper	1%
7	nlistsp.inflibnet.ac.in Internet Source	1%
8	Submitted to University of Essex Student Paper	

		1 %
9	Zeying Wang, Shihong Yue, Huaxiang Wang, Yanqiu Wang. "Data preprocessing methods for electrical impedance tomography: a review", Physiological Measurement, 2020 Publication	1 %
10	journal2.um.ac.id Internet Source	1 %
11	eprints.umsida.ac.id Internet Source	1 %
12	journal-isi.org Internet Source	1 %
13	Submitted to SASTRA University Student Paper	1 %
14	Boby Siswanto, Ford Lumban Gaol, Benefano Soewito, Harco Leslie Hendric Spits Warnars. "Sentiment Analysis of Big Cities on The Island of Java in Indonesia from Twitter Data as A Recommender System", 2021 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS, 2021 Publication	1 %

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		