The Implementation of Indonesian Stemming System for Indonesian Translation of the Quran

1st Syopiansyah Jaya Putra[1], 2nd Novi Dwi Cahyanti[1] , 3rd Suci Ratnawati[1], 4th Muhamad Nur Gunawan[1], 5th Dwi Puspita Sari[2]
{syopian@uinjkt.ac.id[1], novi.dwicahyanti14@mhs.uinjkt.ac.id[1], suci.ratnawati@uinjkt.ac.id[1], nur.gunawan@uinjkt.ac.id[2], dsardiyo@bu.edu[2]}

UIN Syarif Hidayatullah, Departmen of Informatio System, Jakarta, Indonesia[1], Boston University, United States of America[2]

Abstract. Indonesian translation of the Quran (ITQ) is a translator documentation of Quran to Indonesian language containing Muslim guidance, so it is necessary to apply Indonesian Stemming System (ISS) to support learning and understanding ITQ contents. In the implementation of ITQ documentation stemming process, there were some errors including a little number of beheadings (understemming), non-beheading (overstemming), and that produces an incorrect base word. This paper focuses on the Indonesian stemming system implementation for ITQ to produce data training that can minimize errors in the application of the steaming process. The system development consists of 5 steps, including Empiricism in Algorithm Design, Implementation, Experimental Setup, Measure, and Conclusion. Porter algorithm is used in applying Indonesian Steaming System with the rule-based approach. The application of the text processing process starts from tokenization, stop word removal, stemming result token. The study and system development produce application system that optimizes the application of the tokenization process, stopword removal, and stemming ITQ. The result of this study enhances the application system that can support the researcher in understanding ITQ in developing advanced research in the field of text mining for ITQ and hadith.

Keywords: MSMEs; Seller Loyalty; E-Marketplace Indonesia; Metode SMART; Women

1 Introduction

Al-Qur'an is a holy book and a guide for Muslims daily life. Understanding the verses in the Qur'an requires a deep understanding of knowledge in Arabic, however, the interpreters and language experts have translated into various languages including Indonesian. To understand the translation of the Indonesian Qur'an (ITQ), understanding the basic words in each of the verses is necessary [1]. The stemming process in Natural Language Processing helps to transform words in documents into basic words. The process of stemming Indonesian text in the Al-Qur'an translation is more complex because there are some additions that need to be discarded to get the basic words. There have been many studies that apply the process of stemming on Indonesian documents, such as news, articles and several other Indonesian documents [2], but the fact is that no one has applied the stemming process to ITQ so that it

ICONQUHAS 2018, October 02-04, Bandung, Indonesia

DOI 10.4108/eai.2-10-2018.2295507

requires study and development of the Indonesian Stemming System (ISS) on ITQ to facilitate learning and understand the contents of ITQ.

In applying the stemming process in Indonesian documents, there are three types of errors, namely understemming, overstemming, and unchanged. Understemming is a lack of beheading, overstemming is over beheading, while unchanged reminds the same word without any changes [3].

The purpose of this study is to optimize the application of ISS to ITQ by using the Porter algorithm and generate training data that can minimize errors in the application of the process.

The system development methodology consists of 5 stages, including Empiricism in Algorithm Design, Implementation, Experimental Setup, Measure, and Conclusion. Indonesian stemmer application uses the rule-based stemming algorithm approach, by applying the Porter algorithm for Indonesian [4].

The application of the text processing process starts from sentence boundary detection [5], tokenization and stopword removal, and tokens after the results of stemming [6] and the process of error identification analysis [3].

The system development uses the add-on rule-based stemming algorithm, the Porter algorithm for Indonesian [4].

This research and development produces an application system that optimizes the application of the Indonesian stemming process to ITQ including sentence boundary detection, tokenization, stop word removal, stemming and training data.

In evaluating the application of this system there are evaluations of 3 types of errors in the application of stemming the translation of the Indonesian Qur'an namely understemming, overstemming and unchanged. Most of the errors occur in the type of Overstemming error. The level of accuracy of the application of the algorithm in the system is 86.74% [3].

The results of the Indonesian stemming for ITQ system development can be useful for users in understanding ITQ as a whole, quickly and accurately. Besides that, it can also be used by researchers in developing advanced research in the field of text mining for ITQ and hadith.

2 Methodology

To apply the ITQ stemming process using the Indonesian version of the ITQ dataset which was downloaded from (http://tanzil.net/trans/) version of the Indonesian Ministry of Religious Affairs. While the Stopword data used is 358 are common words that often appear in Indonesian [4,6].

The research and development of the ITQ stemming system use Algorithm and Experiments Methodology consists of 5 stages (Figure 1), namely: Empiricism in Algorithm Design, Implementation, Experimental Setup, Measure, and Conclusion [7].

Figure 1. The Development of Indonesian Steeming System

During the empirical stage in algorithm design, the data collection and analysis on the algorithm that will be applied to the research. The authors observed literature studies or

references on stemming algorithms, specifically on the Porter algorithm. At this stage, the author produces a path from the Porter algorithm.

In the implementation phase, the stemming for ITQ application system is designed. The author creates several system designs including flowcharts and system algorithms. In the next step, the experimental setup stage is completed by programming or coding activities to build the application. AT this stage, the author produces the application stemming for ITQ.

The fourth stage is measurement where the results of this study are presented starting from the tokenization to the stemming process. The level of accuracy result is calculated by using an accuracy formula to calculate the number of errors generated.

The last stage is the interpretation result that discusses the results of the analysis.

This formula below uses to calculate the Accuracy level that illustrates the level of proximity between the predicted value and the actual value [8-12]:

$$Accuracy = (RW/W) \times 100\%. \quad (1)$$

Where W is the number of words stemmed and RW is the number of words stemming correctly, accuracy is expressed in percent (%).

3 Result And Discussion

The study and development of ISS using the Indonesian-language Porter algorithm illustrates in Algorithm 1 concerning the flow of the Porter algorithm stemming algorithm.

The first step of the stemming process using the Porter algorithm is to enter the word to be searched for by the base word, then erases the particles. The next step is to delete the ownership pronoun or pronoun. The third step deletes the first prefix, if the token does not have the first prefix, it will delete the second prefix and proceed to the suffix deletion. On the other hand, if the token has the first prefix, the next step will be to delete the suffix, if it is not found, the token is assumed to be root word or root word. But if found, it will do the second prefix before the process is finally finished and the token is assumed to be root word.

Algorithm 1 The flow of the porter algorithm stemming process.

INPUT: a non stopword based word

OUTPUT: a base word

1. Input a non stopword based word
2. Delete particle
3. Delete pronouns
4. Check whether token has a first prefix
5. If “yes” then delete the first prefix, if “no” then delete the second prefix
6. Check whether token has a suffix word
7. If “yes” then do point 8, if “no” do point 10
8. Delete the suffix word
9. Delete the second prefix
10. Based word is created
11. Finish

3.1 Text Processing

In the ITQ algorithm implementation of ISS, researchers create some systems design for text processing that consists of tokenization, stopword removal process, the first phase of stemming process, the second phase of stemming process, and data training process.

1. Tokenisation

The purpose of tokenization is to process ITQ document to be a piece of word or token. The user needs to input document for tokenization so the system able to change document's word to lower case. The next step is the system separates the word from space as a result separated word or token shown to the users as the process output.

Based on algorithm 2 regarding tokenization, the first step that will be carried out by the system is to read the document inputted by the user. Then the system changes the capital letters on the document to lowercase. After the process is complete, the system will break down the sentence in the document into pieces of words based on the space, so the system display output in the form of word snippets or tokens.

Algorithm 2 Tokenization.

INPUT: Indonesian Quran Translation Letter

OUTPUT: Token (a piece of word)

1. Read inputted letter
2. Change all the words to lower case character
3. Separate documents to space separated word
4. Become a separate word piece in lower case
5. Finish

2. Stopword removal

The purpose of this phase is to process the resulting document from the tokenization process in the form of a word cut or token into a clean token from stoplist using filtering process.

Algorithm 3 Stopword Removal

INPUT: Token (a piece of word)

OUTPUT: Non stopword

1. Read token
2. Check whether words is in a stoplist
3. If "yes" remove the words, if "no" kerjakan point 4
4. Become a collection of words that are not in the stoplist
5. Finish

Based on algorithm 3 about stopword removal, the first step that the user will do is to check whether the word or token in the document is the same as the word in the stoplist. If "yes" the system filters the word, if "no" the system will skip and check the next word. So finally got a word or token that is clean from the words stopword.

3. Stemming

The purpose of stemming is to process the output document from stopword removal process into a basic word. Stemming stage is divided into two parts, namely the stemming process of stage 1 (Algorithm 3) and the stemming process of stage 2 (Algorithm 4)

a. Stage 1 Steeming Process

Algorithm 4 Stage 1 Steeming Process

INPUT: Non stopword token

OUTPUT: A piece of affix word

1. Read non stopword
2. Delete particle
3. Delete pronouns
4. Check whether token has a first prefix
5. If “Yes” delete the first prefix, if “No” delete second prefix
6. Check whether token has a suffix
7. If “Yes” then do point 8, if “No” then do point 10
8. Delete suffix
9. Delete second prefix
10. Finish

Based on algorithm 4 on stage 1 stemming process, there will be some remittance cuts such as pronouns, first prefixes, second prefixes and suffixes. After the process is completed, the place of the based word on the process of remedied is known.

b. Stage 2 Stemming Process

Algorithm 5 Stage 2 Stemming Process

INPUT: Piece of word remedial

OUTPUT: Word base

1. Read the number of letters in the word
2. Detect the location of the first letter and the last letter of the base word
3. Check whether the base word is the same as the word in the data training list
4. If “Yes” replace the word based with data training list, if “No” do point 5
5. Become a collection of underlying word token
6. Finish

After retrieving base word, the next step is to check whether the basic words obtained are the same as the words in the training data list. If "yes" then the basic word is replaced with the basic data in the training data list. And if "no" then the data replacement process is not carried out. This aims to minimize errors in the stemming process. Where the errors that have been detected before will not reappear in the next process

Based on the algorithm on stage stemming process for stage 2 stemming process, it can be concluded that the system will read the amount of word to detect the location of the first word and the last word from the base word based on the stemming output process stage 1. Once the system obtains the basic word, the next step is to check whether the base word retrieved is the same as the word in the data training list. If "yes" the basic word is replaced with the basic data contained in the data training list, however if "no" then the process of replacing the data is not done. It aims to minimize the existing errors in the stemming process. Where errors that have been detected before will not reappear in the next process

4. The Error Data Training

Based on algorithm 6 about the error training data, there are some data that must be inputted including words stemming errors, basic words, and types of errors. After the three data above is inputted when the user clicks the "save" button and the system will save the data and make it a list of error training data.

The purpose of error data training is to identify and classify the errors in the results of the stemming process so that in the stemming process the subsequent error will not reappear. This phase builds a training data that can help optimize the stemming process in ITQ.

Based on algorithm 6 about error training data, there are some data that must be entered into such as word stemming error, basic word, and error type. After the three data above is

inputted when the user clicks the "save" button and the system will store the data and make it list the data training error.

Algorithm 6 Error Data Training
INPUT: Stemming error
OUTPUT: Data Training

1. Input stemming error word
2. Input word base
3. Select a type of error
4. Click save button
5. Receive input
6. Finish

3.2 Indonesian Steming System (ISS) for ITQ

The first process in applying ISS is using Porter Algorithm for ITQ before applying stemming process namely dataset (Figure 2). In this application, the dataset used contains the translation of Indonesian language Qur'an texts from the Tanzil web page (http://tanzil/net/trans/) Indonesian language version, the Indonesian Ministry of Religious Affairs. This dataset is used as the input for tokenisation process.

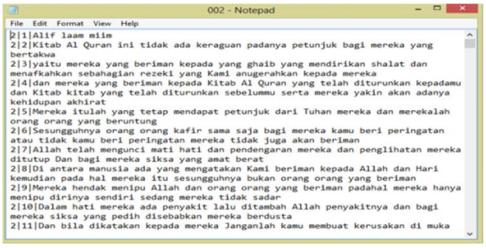

Figure 2. Dataset system

Figure 2 shows dataset that contains the letter that will be processed. This dataset shows the details of letters as well as sentence with separated symbol "|". For example the sentence "2 | 1 | Alif laam miim" that shows "Alif laam miim" contains in letter 2 of the first sentence.

In application home page (Figure 3), the user needs to click button "Login" to be able to enter to the main menu.

Figure 3. Home Page

After login to the main menu, there are 5 features in this application including tokenisation, stopword removal, stage 1 stemming, stage 2 stemming, and data training.

a. Tokenization

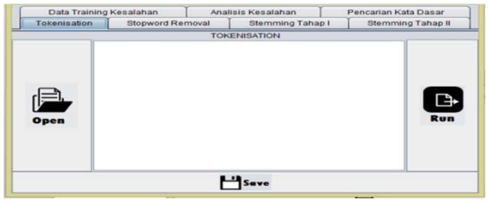

Figure 4. Tokenisation Menu Interface

In tokenisation feature (Figure 4), users can input the dataset by clicking button menu “open”. Furthermore, the process of cutting the word by clicking “run” button so the system can produce the result of tokenization. The input included in the sentence "God will not deny his promise" after the tokenisation process, the word spacing is based on 5 words: "God", "no", "will", "deny", "promise”

b. Stopword removal

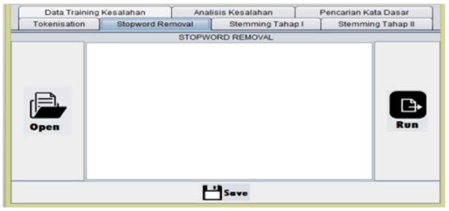

Figure 5. Stopword Removal Menu Interface

In the stopword removal form (Figure 5), the tokenisation processing is completed by clicking the “open” button/ Furthermore, the fitering process is completed by clicking the “run” button. The purpose of filtering process is to eliminate meaningless words. In this process it is necessary to list a meaningless list of words (stopword list) to facilitate the filtering process. The examples of filtering on the stopword removal process are shown in Table 1. The input entered has 5 pieces of word (token) after the process of stopword removal or filtering from a word that is considered meaningless, the resulting output becomes 3 pieces of word (token) such as "god", "deny", "promise". Where 2 tokens token are "no" and "will" be deleted because it is considered a word that has no meaning (stoplist).

Table 1. The Example of Cutting Stopword Removal

Input	god	not	will	deny	promise
Output	god	deny	promise		

c. Stemming

There are 2 steps in stemming, including stage 1 stemming (Figure 6) and stage 2 stemming (Figure 7).

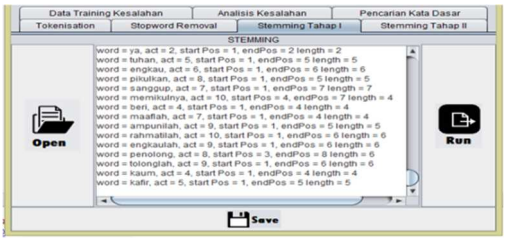

Figure 6. Stage 1 Stemming Menu Interface

1. Stage 1 Stemming Process

In stemming process form (Figure 6), the ourput of stopword removal process is inputted into the application by clicking button “open” then the process of removing the affix can be completed by clicking button “run”.

After that, the system will cut the affix based on the Porter algorithm path. The system will display output that describes the word in the process (word), many letters (act), the location of the base word (start Post and endPos), number of letters in the basic word (length). Figure 6. shows the output of the stemming process for stage 1, for example "word = pikulkan, act = 8, start Pos = 1, endPos = 5, length = 5". Based on the example above, it can be seen that the word "pikulkan" has 8 letters, the basic word produced starts in the first letter to the fifth letter and the number of letters of the basic word is 5. So that it can be seen that the basic word produced is " pikul ". However, this process has not directly displayed the basic words obtained, so it is necessary to stage 2 stemming (Figure 7) to be able to display these basic words.

1. Stage 2 Stemming Process

Stage 2 stemming menu interface illustrates in the Figure 7.

Fig.7. Stage 2 Stemming Menu Interface

In the result stemming form (Figure7), user can input the output of stemming process result to the application by clicking “open” button before cutting the affix process is started by clicking button “run”. The last step is to determine the based word by reading the output of stemming process.

Figure 7 shows the output of stage 2 stemming process. In this process, letter, sentence, word, and base word is produced. For example the word of “keraguan” with a base word of “ragu” on Al-Baqarah letter and second sentence.

After completing all of these process, user will know the total number of token output (Table 2) from each process.

Table 2. The Number of Token Output

	Surat 003: Al-Imran	Surat 005: Al-Maidah	Surat 018: Al-Kahfi	Total
Tokenisation	7.468	5.869	3.526	16.863
Stopword removal	4.039	3.195	1.796	9.030
Stemming				
Stemming Token	4.039	3.195	1.796	9.030
Inappropriate	328	233	125	686

token				
Appropriate token	3.711	2.441	1.605	7.757

Based on Table 2, the total number of token from each process is 16.863 token for tokenization process, after the stopword removal the number of token decrease to 9.030 token and 7.757 token from stemming process that exactly accordance with the base word.

Based on the token result above, the researchers do the measurement and analysis including the stemming error analysis and accuracy analysis (Table 3).

2. Stemming Error Type Analysis

After applying the stemming process using the Porter algorithm, three types of errors were found, including understemming, overstemming and unchanged. The highest number of errors in overstemming errors is 490 errors. The second most error is the type of understemming error with 181 errors. And lastly unchanged with 15 errors.

Overstemming is a condition where the beheading of an additive exceeds the one that is supposed to be. The researchers claim that this happened because the word resembled the prefix. For example, the word "*suami*" becomes "*suam*" because the algorithm process detects the "i" suffix is considered as an affix that is removed so as to produce an inappropriate word that is "*suam*" who should be "*suami*".

Understemming is a condition where the beheading of an affix has too little than it should. The researcher felt that this happened because of a lack of the existing pattern of additions. For example, the word "mengetahui" to "etahu" should "tahu".

Unchange is a word that does not change after the stemming process, but the word produced is not the right basic word. The researchers claim that this happened because of the lack of cutting rules on the Porter algorithm. An example of the error that entered the category of the Unchange error is the word "seibu" to remain "seibu" even though the word should produce the word "ibu". Due to the Porter algorithm process there are no rules for deleting "se" add-ins so that the system does not detect these additions to be removed.

Based on the stemming errors, it can be explained again the errors of the stemming process in each letter.

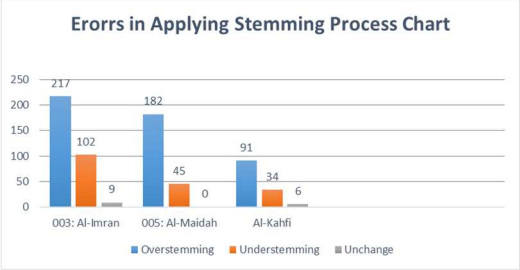

Fig.8. Error in applying stemming process chart

Based on Figure 8, the most common error is found in letter 003: Al-Imran with a total error of 328 errors. The second largest is letter 005: Al-Maidah with a total error of 233 errors followed by the letter 018: Al-Kahfi with a total of 125 errors.

The biggest mistake is in Letter 003: Al-Imran which is 328 errors. Researchers state that this is not an oddity, because the results are consistent with the recapitulation of the number of tokens of each letter that is used as a sample of the study. The recapitulation shows that the number of tokens in 003: Al-Imran is indeed more than other letters, which is 3,429 tokens.

3. Accuracy Analysis

The level of accuracy of stemming on each letter can be seen in Table 3.

TABEL III
The Accuracy of Stemming

No	Surah	Number of Token			Accuracy
		Token	Correct Token	Incorrect Token	
1	Surat 003: Al-Imran	4.039	3.429	328	84,89%
2	Surat 005: Al-Maidah	3.195	2.674	233	83,69%
3	Surat 018: Al-Kahfi	1.796	1.730	125	96,32%
The total number of		9.030	7.833	686	Average of Accuracy 86,74%

Based on Table 3, it can be seen that the highest accuracy is found in Surah 018: Al-Kahfi with accuracy of 96.32%, then the second highest Surah 003: Al-Imran with accuracy of 84.89%, and finally Surah 005: Al-Maidah with 83, 69%. The final result of the accuracy analysis has shown that the accuracy of Stemming Porter in the translation of the Qur'an is 86.74%.

It can be interpreted that the correctness of the application of the Porter algorithm is 86.74% and the error rate is 13.26%.

Stemming Algorithm Porter is categorized into a heavy Stemming algorithm because it produced 490 inversion errors in this study, the results were more than the Understemming error which only contained 181 errors and 15 Unchange errors.

It has lower accuracy compared to Nazief and Adriani algorithm [9]. The Nazief & Adriani algorithm [9] has an accuracy percentage of 95.26%, however the training data in this application helps the Porter algorithm performs better.

Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersted’s. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.

4 Conclusion

This study is successfully optimizing the application of thePorter algorithm stemming process by applying training data on the stemming application to Indonesian translation of the Qur'an specifically to the context of the letter on Trade and Industry.

The calculation of accuracy process obtained the percentage of accuracy of the stemming process using the Porter algorithm that is equal to 86.74% of the 9,033 words tested. There are 686 words whose output does not match the expected results. This is because there are words that resemble the additive word, causing an inversion, the lack of an affix pattern so that it causes Understemming and Unchange.

The ISS system developed for ITQ is to make it easier to get basic words, analyze errors and optimize the application of the stemming process to ITQ.

The further development can develop this system to the search engine system so that it can be easier to review and understand the contents of ITQ and the hadith..

References

[1] M. Q. Shihab. Kaidah Tafsir. Tangerang: Lentera Hati. , (2013)

[2] B. Nazief, (2000, October). Development of computational Linguistics research: A challenge for Indonesia. In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (pp. 1-2). Association for Computational Linguistics.

[3] C. D. Paice, an evaluation method for Stemming algoritms springer-verlag. 1994.

[4] F. Z. Tala, A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. In Institute for Logic, Language and Computation Universeit Van Amsterdam. 2003.

[5] S. J.Putra, , , M. N.Gunawan, , I. Khalil., & T. Mantoro,. Sentence boundary disambiguation for Indonesian language. In Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services (pp. 587-590). ACM. (2017, December)

[6] S.J.Putra, and , M.N Gunawan., S.R. Agung. Tokenization and N-gram for Indexing Indonesia Translation of the Quran. Ciputat: UIN Jakarta. 2018.

[7] C.C. McGeach. Experimental Analysis of Algorithms. Volume 48. Number 3. 2001

[8] A. Ledy,. Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief & Andriana. 2009.

[9] S. J., Putra, T.Mantoro, , & M. N. Gunawan, Text mining for Indonesian translation of the Quran: A systematic review. In Computing, Engineering, and Design (ICCED), 2017 International Conference on (pp. 1-5). IEEE. (2017, November).

[10] D.Wahyudi, , T.Susyanto, , and D.Nugroho,. Implementasi dan Analisis Algoritma Stemming Nazief & Adriani dan Porter pada Dokumen Berbahasa Indonesia. Jurnal Ilmiah Sinus. 2017

[11] G. S., Ingersol, T. S Morton, and A. L. Faris,. Taming Text. United States of America. 2013

[12] K.Sumathy, , and M.Chidambaram,. Text Mining: Concepts, Applications, Tools and Issues-An Overview. International Journal of Computer Applications. Vol. 80 No. 4. 2013