Pattern-Based Stemmer Analysis and Implementation on Arabic Text

Ananda Wulandari[1], Kemas Rahmat S.W[2], Ade Romadhony[3]
[1,2,3]Fakultas Informatika Institut Teknologi Telkom, Bandung
[1]an_nda_a@yahoo.com, [2]bagindok3m45@gmail.com, [3]ade.romadhony@gmail.com

Abstrak

Pattern-based Stemmer adalah implementasi algoritma pencarian untuk menemukan akar kata dari kata bahasa Arab yang mengimplementasikan morphological anlysis technique dan affix removal technique. Pada riset ini, jika proses stemming telah dilakukan, proses penentuan kelas kata akan dilakukan sebagai berikut: Pertama, sistem akan mencocokkan antara kata yang diinputkan dengan kata yang disimpan di sistem. Jika kata tidak ditemukan, aturan penentuan kelas kata akan dilakukan berdasarkan prefix, suffix, dan infix. Jika sistem masih tidak bisa menemukan kelas kata dari langkah kedua, maka kelas kata akan ditentukan berdasarkan posisi kata pada kalimat.

Pengujian dilakukan untuk mengetahui pengaruh dari jumlah token, pola dan rules terhadap performansi sistem. Data yang digunakan pada pengujian ini adalah 37 surat di juz ke-30 dari Al-Qur`an. Ke-37 surat tersebut akan dikelompokkan ke dalam tiga kategori berdasarkan jumlah baris yang dimiliki tiap surat : surat panjang, surat menengah, dan surat pendek. Berdasarkan hasil pengujian, performansi terbaik diperoleh dengan menyimpan lebih banyak pola bebas imbuhan, kelas kata yang menentukan rule dan menambahkan proses pemeriksaan affix elimination pada sistem

***Kata kunci**: teks Arab, stemming, stem, kelas kata.*

Abstract

Pattern-based Stemmer is an implementation of searching algorithm to find stem from an Arabic word that implement morphological anlysis technique and affix removal technique. In this research, if stemming process has been done, word class determination process will be conducted according this way: First, system would match between word which is entered with the fix word that is stored in the system. If the word was not found, word class determination rules will be conducted based on prefix, suffix, and infix. If this system could not figure out the word class of the word from the second step, then word class would be determined based on the word position in a sentence.

Testing is commited in order to know the influences of the number of token, pattern and rule in the system to the system's performance. Data that used in this testing are 37 surat in juz 30th from Al-Qur`an. They will be put into three categories, based on the number of rows of each surah : long surah, medium surah, and short surah. Based on the testing results, the best performance gained by storing more free-affix pattern, storing more word class determining rule, and adding affix elimination checking process into the system.

***Keywords**: Arabic text, stemming, stem, word class.*

1. Introduction

Indonesia is a country with the most Moslem population in the wold. Result of census in 2000 indicated almost 86,1% of 240.271.522 Indonesian are adherent of Islam [6]. In Islam, Moslems are commanded to understand Al-Qur'an in order to make it guidance in life. Al-Qur'an is kalamullah (words of Allah) descended in Arabic language. For many Indonesian which use Bahasa Indonesia (Indonesian Language) as their native languange, they sometimes get difficulty in understanding Al-Qur'an. One of the ways to understand the meaning of Arabic words is by learning Arabic languange first. Another way to understand the real meaning in Al-Qur'an is through tafseer's (explanation's) books of olama (the scholars).

Nahwu is basic knowledge for learning Arabic language. In nahwu, people learn about the position of word in the sentence and the end of harakat (vowel) of each word. Nahwu recites three subjects. They are letter, word and sentence. There are kinds of word classes that are commonly known such as letter, noun, and verb [16]. Advantages in

studying nahwu are keeping oral from making mistakes in Arabic, and understanding Al-Qur`an and Prophet's hadith with an appropriate understanding [10]. Nahwu has relevance with Sharaf, knowledge studying about word form and it changes by adding or substracting letter [3]. Different of a word's meaning can be occured when that word is given affixes. Therefore, it would be easier to understand the meaning of Arabic words if we know the stem, affixes, and word class of it. For instance : if we find the word يَذْهَبُ (yadzhabo– he is going or will go – verb), then we can figure out that the stem of يَذْهَبُ is the word ذَهَبَ (dzahaba – he has gone) based on pattern below :

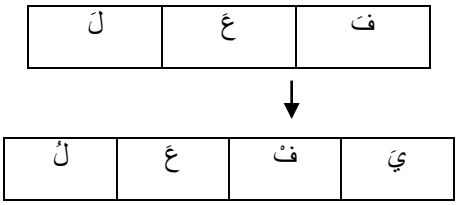

Figure 0. Pattern example of changing word in Arabic langunge

Stemming is a process of finding stem (root) of a word by stripping away the affixes whether it is prefix, infix, suffix or combination of prefix and suffix[8]. Stemming is used to replace word form to become stem according to proper and appropriate morphology [11]. There are several techniques that can be used in stemming process of Arabic texts, for instance dictionary technique, affix removal technique, and morphological analysis technique. Dictionary technique refers to find words into dictionary that stores many words (stem and its affixes). Affix removal technique is a technique that looking for stem by eliminating all affixes of the word. While morphological analysis technique is a technique that looking for stem by examining the structure of word formation.

In this final project, writer will implement Pattern-based Stemmer which is implementation of morphological anlysis technique and affix removal technique in order to find stem of the words in Arabic texts. In addition, system will determine word class of each word in the texts. Writer's motive choosing this method because Arabic language is a language with many patterns. For instance pattern that form a verb, noun, or others that can determine word class of a word. It easier to find stem by examining patterns or structure of a word.

2. Knowledge Base

2.1 Stemming

Stemming is the process of finding the stem (root) of a word, by stripping away the affix attached to the word. In many languages words are often obtained by affixing existing words or roots. 8]. Word's affixes can consist of prefix, infix, suffix, and the combination between prefix and suffix. Stemming is used to replace word form to become stem according to proper and appropriate morphology [11]. Stemming coming in useful for application related with word processing, for instance text classifier, serching tool to find words in dictionary, and information retrieval system[8]. The results of stemming process are stem (root) that is part of words left after stripped all affixes.

2.2 Pattern-based Stemmer

The Arabic languanges is a highly inflected language. This increases the difficulty of the stemming prosess. Pattern-based Stemmer is stemmer that was formed by Riyad Alshalabi by implementing efficient technique in extracting stem for Arabic text. This technique do not depend on word searching in dictionary. This technique depend on affixes elimination of the words. After affixes elimination complete, next step is to find pattern suitable to the free-affixes word, then extract the characters of that word basis.

In Arabic, the word مَسْجِدٌ (masjidun-placed used to bow) has the stem سَجَدَ (sajada-bowed). This stem gained after extracting word مسجد that has pattern مفعل. Meanwhile, most

of Arabic stem are verb which the characters compose is are original, with pattern فعل. In that case, it can be known that stem from word مسجد is سجد.

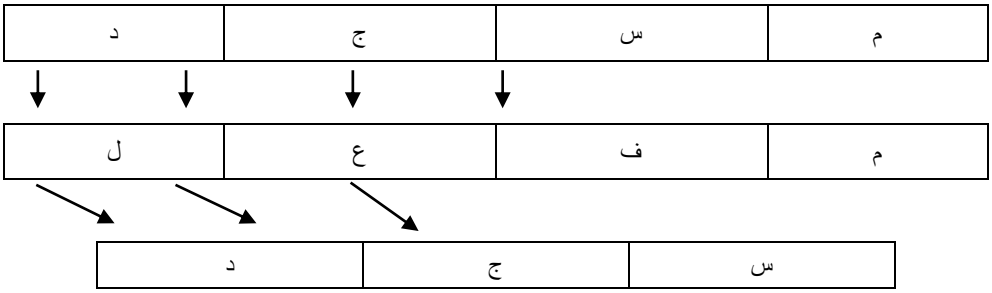

Figure 0. Extracting process

Arabic words demonstrate an intricate morphology. The Arabic language can be said to use root and pattern morphotactics where a pattern can be thought of as a template adhering to established grammatical rules. Such patterns are applied by adding affixes (prefixes, infixes, or suffixes) to roots (which are simple bare verbs that are three letters in length) to form ther parent root. Prefixes and suffixes can be further added to Arabic stems to express common grammatical usages such as the possessives, plurals, definite forms, gender, etc. For example, some of the additional forms of the word كِتَابٌ (kitaabun - book) are shown in Table 1.

Table 01. Some prefixes attached to the word "book"

Word	Meaning
اَلْكِتَابُ	A book
كَالْكِتَبِ	As the book
لِلْكِتَابِ	For the book
بِالْكِتَابِ	With the book
وَكِتَابٌ	And a book

2.3 Tagging Non-Vocalized Arabic Word

Arabic texts could be either a vocalized text – such as the language of the holy Qur`an- or a non vocalized thext –which is used in newspapers, books, and media-. In this system, word class determination process will be conducted if stemming process has been done. Meanwhile affixes of the word will be stored, in the same time affixes removed by stemmer. Therefore, this word class determination process will process non-vocalized (there's no harakat) word in Arabic text. Except in this circumstance, last vowel letter (harakat) of the word will be stored to abridge word class determination. Approximation used in word class determination consist of 3 steps [4]:

1. The Lexicon Analyzer
   The initial tagging (determining word class) level is a lexicon analyzer. The system has a lexicon which stores all Arabic fixed words and particles (prepositions, adverbs, conjunctions, interrogative particles, exceptions, questions and interjections). Each word in the reading non-vocalized text is explored in the lexicon. If it is found, the corresponding tag (word class) is returned. But if it is not found, the word would be transfered to the second level of the system i.e. the morphological analyzer.
2. The Morphological Analyzer
   There are several signs in the Arabic language that indicate whether the word is a noun or a verb. One sign is the pattern of the word. Some of the patterns are used with verbs and others are used with nouns. But when text contains non-vocalized words, many words are ambiguous since their patterns are used with both verbs and nouns. For instance word **كتب** can be read as kataba (he wrote – verb), kutiba (it is written – verb), or kutubun (books – noun). Therefore, another way can be conducted word class determination process of word, by extracting stem, prefix, suffix, and infix of that word. Originally, stem is word basis gained from stemming process. By knowing the stem of

the word, we can determine affixes of that word. In this system, which is consist of stemming process and also word class determination process, Affixes of the word will be stored meanwhile it also removed by stemmer. Next, word class determination rules will be stored based on prefix, suffix, and infix.

- Rule 1:
  The following prefixes (or part of prefix) map
  the word to NOUN class: ال, فل, لل, م, كال, بال, فال , or وال
- Rule 2:
  The following suffixes (or part of suffix) map
  the word to NOUN class: ت, ات, or اء
- Rule 3:
  The following suffixes (or part of suffix) map the word to NOUN class with the condition of not existing of the imperfect tense letters (ا, ن, ي, or ت) : ون, ين, ان, or ي
- Rule 4:
  The following infixes map the word to NOUN class with the condition of satisfying the corresponding position within the stem pattern determined between parentheses : ا, ي, و, او, or اوي (after the ayn of the word/ayn fi'il), وا (after the fa' of the word/fa' fi'il)
- Rule 5:
  The following prefixes (or part of prefix) map
  the word to VERB class : ي, ن, or ا with istiqbal letter (Future س) at the beginning, then the word class is verb
- Rule 6:
  The following prefixes (or part of prefix) map the word to VERB class with the condition of the Rules above did not satisfy: ا
- Rule 7: The following suffixes map the word to VERB class: Opening T (ت)

3. Syntax Analyzer
   If this system can not figure out the word class of the word from the input yet, then word class will be determined based on the word position in a sentence.
   - Rule 1: If that word is after al-jarr letter (prepositions that cause the word after become majrur) or an-nida letter (interjections expletive or caracter to call someone), then the class word is noun
   - Rule 2: If the word class can not be determined yet, then system will determine word class based on sequence position in the sentence. These rules cover the following sequences: verb noun, verb noun noun, verb noun particle noun, verb noun particle noun noun, verb noun noun particle noun, noun noun, noun verb noun, noun verb, particle noun verb noun, noun verb particle noun. If there is more than one rule that matches with the sentence to analyze we ignore the word which being unanalyzed word. [4]

Tabel 2. Document Input per Category

d'Number of Surah	Category		
	Long	Medium	Short
37	4	8	25

3. Stemming and Tagging Performance

3.1 Stemmer Evaluation

Pattern-based Stemmer will be evaluated by calculating the accuracy and Index Compression Factor (ICF) using the following formulas :

$$\text{accuracy} = \frac{\text{correct stem}}{\text{all terms}} \times 100\%$$

Accuracy is ratio between the number of correct stem and the number of terms exist in document. The higher accuracy, then the better stemmer used. ICF (Index Compression Factor) represents the extent to which a collection of unique words is reduced (compressed) by stemming. The higher the ICF values obtained, the better the performance of the stemmer used. This can be calculated by :

$$ICF = \frac{(N - S)}{N}$$

While N is the number of unique words before Stemming and S is the number of unique stems after Stemming

3.2 Tagger/Word Class Determiner Evaluation

Performance of rules that determine word class will be observe, which is relevance with stem resulted by previous process. These rules performance will be observe based on accuracy, precision, and recall using the following formulas :

$$\text{accuracy} = \frac{\text{correct}}{\text{all token}} \times 100\%$$

$$\text{precision} = \frac{\text{correct}}{\text{correct} + \text{incorrect}}$$

$$\text{recall} = \frac{\text{correct}}{\text{correct} + \text{unanalyzed}}$$

Accuracy is ratio between the number of correct word class and the number of tokens exist in document. The higher accuracy, the better tagger used. Precision and recall are ratio between the number of correct, the number of incorrect, and the number of unanalyzed word classes.

4. Testing and Analysis

Data that used in this testing are some surah in Al-Qur`anul Kareem, specially in Juz 30$^{th}$. They will be put into three categories, based on the number of rows of each surah : long surah (consists of 19 rows or approximately one-half page of mushaf Al-Qur`an), medium surah (consists of 13 rows or approximately one page of mushaf Al-Qur`an), and short surah (consists of 6 rows or approximately a half page of mushaf Al-Qur`an).

4.1 System Testing

This testing is used to analyze and evaluate pattern-based stemmer algorithm in producing stem, and some rules that have been determined to produce class word from every term. We evaluated the stemmer performance by analyzing accuration level and ICF value that obtained based on stem that produced by stemmer. Whereas we evaluated rules that determined class by analyzing accuration level, precision, and recall that obtained based on word class that produced by system.

4.2 The analysis of testing result

4.2.1 The analysis of pattern influence to stemmer performance

Data that used in this testing are Arabic texts that included in short surah. In this testing, there is variation of pattern number that saved inside the system. This testing is used to know the influence of pattern number to stemmer performance. Table 3 shows the testing result and analysis of stemmer performance that obtained by implementing pattern-based stemmer.

Table 0. The pattern influence to stemmer performance

Surah Name	Pattern Number			
	A (128 – asli)		B (93 – acak)	
	Accuracy	ICF	Accuracy	ICF
An Naas	85%	0.17	75%	0.11
Al Falaq	94.73%	0.12	73.68%	0
Al Ikhlaash	85.71%	0.09	52.91%	0.21
Al Kaafiruun	100%	0.42	74.19%	0.45
At Takaatsur	83.33%	0.18	63.33%	0.13
Al Qaari'ah	67.74%	0.03	45.16%	0
Al Zalzalah	80%	0.13	65.71%	0.03
Al Qadr	89.28%	0.08	78.57%	0.04
At Tin	79.41%	0.03	61.76%	0
Al Lail	61.90%	0.03	57.14%	0.03

Surah Name	Pattern Number			
	C (58 - asli)		D (58 plus)	
	Accuracy	ICF	Accuracy	ICF
An Naas	85%	0.17	90%	0.23
Al Falaq	94.73%	0.12	100%	0.12
Al Ikhlaash	85.71%	0.09	85.71%	0.09
Al Kaafiruun	100%	0.42	100%	0.42
At Takaatsur	83.33%	0.18	83.33%	0.18
Al Qaari'ah	67.74%	0.03	70.96%	0.03
Al Zalzalah	80%	0.13	80%	0.13
Al Qadr	89.28%	0.08	89.28%	0.08
At Tin	79.41%	0.03	85.29%	0.09
Al Lail	61.90%	0.03	76.19%	0.07

Type A testing is a testing done by saving all patterns that have been explained before (total number of patterns is 128), where these patterns saved in stemmer that purely built using pattern-based stemmer. In type B, testing done by deleting patterns randomly that previously used in type A testing. Total number of patterns in type B is 93.

The result above shows that with the increasing of pattern number that saved in system, the accuration level of stemmer will show better result, therefore stem that produced is also becomes more unique. This proved by the increasing of accuration level and ICF value. From type A and type B testing, we can see that the number of patterns that saved in system give influence to stemmer performance. We got these results because the references that become a basis of stemming process in pattern-based stemmer algorithm is pattern matching, after previously we deleted some affix in words. Therefore, patterns that have influence to stemmer performance are patterns that free from affix that previously deleted by doing step 1 until step 6 in this algorithm. This can be proved by doing type C testing that only saved 58 patterns, where these patterns are patterns that free from affix that previously deleted doing step 1 until step 6 in this algorithm. In type C testing, we can see that the result of stemmer performance produced by conducting this testing is same with type A testing's result. Therefore, pattern-based stemmer is algorithm that efficient enough to in determining stem from Arabic texts, without saving all original patterns that arrange words.

Then we conducted type D testing to see the influence of suffix and affix deleting rule to stemmer performance. In type D, there are some additional checking rules that will be checked before we deleted the affix. The result shows that beside influenced by patterns saved by system, stemmer performance also influenced by additional rules that used before suffix and prefix deleting. If stemmer wrongly recognized affix, then automatically stemmer will get wrong stem, that obtained when matching the pattern of word in the next step (after affix deleting). In type D testing, when input is An-Naas, we got 2 wrong stems, that is الوسواس and يوسوس . In step 6 pattern-based stemmer algorithm, output that produced from step 6 using input from both those words is وسوس . Then, in step 7, word وسوس that has 4 characters will be matched with patterns saved in system. The patterns that match with word وسوس are فعول and فعلل , but because pattern فعول is saved formerly in system, system will extract stem based on pattern فعول , therefore stem that produced is وسس , whereas these two words should have had pattern فعلل . System is wrongly determined stem, where the right stem from words الوسواس and يوسوس is وسوس. Although system is wrong in

determining stem, but because there is additional affix checking process, then ICF value that obtained will be bigger. It's different from type C testing that doesn't use additional prefix deleting rule و in pattern فعلال . Therefore character و in word وسواس will be considered as prefix, whereas actually it is a part of stem. Stem from word الوسواس and يوسوس that produced from type C testing are سوس and وسس . This stem result shows that system fail to condense 2 words that have same stem. Therefore, affix checking rule will influence in accuration and ICF value.

From 4 conducted testing, we can see that although there are many affix-free patterns saved in system, system can't differentiate which right pattern for all ambiguous words (match with 2 patterns or more). Therefore, system will only extract stem based on matching patterns, which formerly saved in system than other matching patterns.

4.2.2 The analysis of pattern influence to word class performance

Data that used in this testing are Arabic texts that included in short surah. In this testing, there is variation of pattern number that saved inside the system. This testing is used to know the influence of pattern number to class word performance. Here is the result and the analysis.

Table 0. The pattern influence to word class determining

Surah Name	Pattern Number					
	A (128 – asli)			B (93 – acak)		
	Accu	Prec	Rec	Accu	Prec	Rec
An Naas	95.8%	0.95	1	91.6%	0.91	1
Al Falaq	77.77%	0.84	0.91	74.07%	0.8	0.9
Al Ikhlaash	94.73%	0.94	1	94.73%	0.94	1
Al Kaafiruun	87.09%	0.9	0.96	90.32%	0.9	1
At Takaatsur	84.37%	0.84	1	84.37%	0.84	1
Al Qaari'ah	72.5%	0.78	0.90	62.5%	0.75	0.78
Al Zalzalah	65%	0.70	0.89	65%	0.78	0.78
Al Qadr	76.47%	0.86	0.86	73.52%	0.86	0.83
At Tin	71.05%	0.75	0.93	68.42%	0.72	0.92
Al Lail	69.33%	0.74	0.91	66.67%	0.72	0.89

Surah Name	Pattern Number					
	C (58 – asli)			D (58– plus)		
	Accu	Pec	Rec	Accu	Prec	Rec
An Naas	95.8%	0.95	1	95.8%	0.95	1
Al Falaq	77.77%	0.84	0.91	77.77%	0.84	0.91
Al Ikhlaash	94.73%	0.94	1	94.73%	0.94	1
Al Kaafiruun	87.09%	0.9	0.96	87.09%	0.9	0.96
At Takaatsur	84.37%	0.84	1	84.37%	0.84	1
Al Qaari'ah	72.5%	0.78	0.9	72.5%	0.78	0.9
Al Zalzalah	65%	0.70	0.89	65%	0.70	0.89
Al Qadr	76.47%	0.86	0.86	76.47%	0.86	0.86
At Tin	71.05%	0.75	0.93	71.05%	0.75	0.93
Al Lail	69.33%	0.74	0.91	72%	0.80	0.87

Type A testing is a testing conducted by saving all patterns that have explained before (the total patterns are 128), where these patterns have been saved in stemmer purely built using pattern-based stemmer. In type B, testing conducted by erasing patterns randomly that previously used in type A. Total number of patterns in type B is 93.

From the result above, we can see that with the increasing of pattern number that saved in system, the accuration level, precision, and recall of stemmer will show better result and the decreasing of wrong number of word class and unanalyzed word. In word class determining process, if a word can't be stemmed in lexicon analyzer step, then we will see stem from word that will be processed and affix in it. Then, system will do morphology analyzer, that is word class determining based on affix in that word. There are some prefix, infix, and certain suffix that form prefix, infix, and suffix from certain word class. Therefore, the number of pattern that saved in system gives influence to word class determining process. But in 1 surah in type B testing, there is word class performance that has higher accuration, precision, and recall. System can't determine word class in 2 first

steps (lexicon analyzer and morphological analyzer), therefore system will do syntax analyzer. Because there are some wrong word classes (where system determined a word from verb to noun), this will influence system in determining word class from word that near the wrong word. For example, word أَعْبُدُ (I pray) is a word that has pattern افعل . But, because in type B system doesn't save pattern افعل , stem that produced by system is اعبد , whereas it should have عبد . The mistake in process of determining stem will influence word class determining process. Therefore, word اعبد that should categorized as verb will be recognized as noun. This 1 mistake will influence word class that placed after word اعبد .

In type C testing we saved 58 patterns, where these patterns are affix-free patterns. The result shows that by saving 58 affix-free patterns, the performance of word class determining is same with system that saved 128 patterns. Therefore, pattern-based stemmer is algorithm that efficient enough in word class determining process. In type D testing, where there are additional rules to erase suffix and prefix, the result shows that rules to erase prefix and suffix influence word class determining performance, because if system can't determine word class in step lexicon analyzer then system will determine word class based on affix in that word. So, if system is wrongly recognized affix (there in no checking process before erasing suffix and prefix), then system will wrong in determining word class.

4.2.3 The analysis of word class determining rule to word class performance

In this testing, to get the best word class, then we used 58 patterns, with additional rules to erase affix (based on previous testing). But, there will be some changes in word class determining rule to know the influence of rule to performance of word class determining process. Here are the results.

Table 05. The rule influence to word class determining

Nama Surah	Jumlah Pattern					
	A			B		
	Akurasi	prec	rec	Akurasi	prec	rec
An Naas	91.66%	0.91	1	95.83%	0.95	1
Al Falaq	62.96%	0.85	0.70	77.77%	0.84	0.91
Al Ikhlaash	63.15%	0.8	0.75	94.73%	0.94	1
Al Kaafiruun	87.09%	0.9	0.96	87.09%	0.9	0.96
At Takaatsur	84.37%	0.84	1	84.37%	0.84	1
Al Qaari'ah	60%	0.77	0.72	72.5%	0.78	0.90
Al Zalzalah	60%	0.68	0.82	65%	0.70	0.89
Al Qadr	73.52%	0.83	0.86	76.47%	0.86	0.86
At Tin	71.05%	0.77	0.9	71.05%	0.75	0.93
Al Lail	68%	0.77	0.85	72%	0.80	0.87

Nama Surah	Jumlah Pattern					
	C			D		
	Akurasi	prec	rec	Akurasi	prec	rec
An Naas	58.33%	0.63	0.87	67.74	0.91	0.72
Al Falaq	40.74%	0.91	0.42	77.77%	0.84	0.91
Al Ikhlaash	63.15%	0.85	0.70	89.47%	0.94	0.94
Al Kaafiruun	67.74%	0.72	0.91	83.87%	0.86	0.96
At Takaatsur	65.62%	0.80	0.77	75%	0.85	0.85
Al Qaari'ah	40%	0.69	0.48	72.5%	0.80	0.87
Al Zalzalah	37.5%	0.6	0.5	62.5%	0.67	0.89
Al Qadr	47.05%	0.76	0.55	76.47%	0.86	0.86
At Tin	36.84%	0.6	0.48	71.05%	0.75	0.93
Al Lail	46.66%	0.85	0.50	70.66%	0.79	0.86

Type A testing is a testing conducted by implementing rule that has been explained in chapter 2, that is A Rule-based Approach for Tagging Non-Vocalized Arabic Words. Therefore, in this testing, harakat in Arabic texts will be ignored. Type B testing is a testing that conducted by giving additional rule in rule that previously applied in type A testing. Type C testing is a testing conducted by erasing some rules randomly.

In type B, harakat will not always be ignored. There are additional rules as explained in chapter 3. From table above, we can see that accuration level, precision, and

recall will increase as the increasing of rule number. The result obtained from type B testing gives the best performance among the 3 other testing, that have less rules. It means that the increasing number of applied rule will increase the probability of right stem, and decrease wrong stem and unanalyzed. Therefore, applied rule indeed has influence to the performance of word class determining.

Type C testing conducted by erasing word class determining rules, except the forth rulein morphological analyzer step. So, in this step, system will not recognize a word as a noun, except if the system considers infix in that word. The result obtained from this testing shows that word class determining rule influences system performance. The increasing number of noun determining rule will increase the produce right stem. Then, accuration level, precision, and recall will be better. In testing result, there are some surah that have high precision and low recall. It shows that the decreasing number of noun determining rule will make unanalyzed words increase. The increasing number of unanalyzed words produce low recall. Whereas, high precision and low accuration because of wrong word class determined by system is less than unanalyzed words.

Type D testing is almost similar with type B testing, but in this testing we erased verb determining rules, except the fifth rule in morphological analyzer step and additional rule or word class determining in type B testing. Table above shows that type D's result is lower that type B. It means that the increasing number of word class determining rule will increase the right word that stemmed by system, therefore the accuration level, precision, and recall is also increased.

4.2.4 The analysis of token number to system performance

In this testing, there are no changes in patterns or rules that saved in system. This testing scenario only uses 58 patterns with addition of prefix and suffix deleting rules and word class determining rule with the biggest number. Here are the results.

Table 6. the influence of token number to stemmer performance

Kategori	Nama Surah	Jumlah Token	Akurasi	ICF
Surah Panjang	An Nabaa`	178	78.62%	0.08
	An Naazi'aat	183	79.87%	0.11
	Al Muthaffifiin	173	75.55%	0.16
	Al Fajr	143	88.49%	0.14
	At Takwiir	108	86.07%	0.10
Surah Menengah	Al Insyiqaaq	112	79.31%	0.12
	Al Buruuj	113	79.31%	0.13
	Al A'laa	76	85.93%%	0.09
	Al Ghaasyiyah	96	77.33%	0.10
	Al Lail	75	76.19%	0.07
	At Tiin	38	85.29%	0.09
	Al Qadr	34	89.28%	0.08
	Al Zalzalah	40	80%	0.13
Surah Pendek	Al Qaari'ah	40	70.96%	0.03
	At Takaatsur	32	83.33%	0.18
	Al Kaafiruun	31	100%	0.42
	Al Ikhlaash	19	85.71%	0.09
	Al Falaq	27	100%	0.12
	An Naas	24	90%	0.23

Table 7. The influence of token number to word class performance

Kategori	Nama Surah	Jumlah Token	Akurasi	Precision	Recall
Surah Panjang	An Nabaa`	178	81.46%	0.83	0.96
	An Naazi'aat	183	72.67%	0.83	0.84
	Al Muthaffifiin	173	75.14%	0.84	0.86
	Al Fajr	143	84.61%	0.90	0.92
	At Takwiir	108	90.74%	0.95	0.95
	Al Insyiqaaq	112	80.35%	0.90	0.87
Surah Menengah	Al Buruuj	113	78.76%	0.85	0.90
	Al A'laa	76	76.31%	0.78	0.96
	Al Ghaasyiyah	96	88.54%	0.89	0.98
	Al Lail	75	72%	0.8	0.87
	At Tiin	38	71.05%	0.75	0.93
	Al Qadr	34	76.47%	0.86	0.86
	Al Zalzalah	40	65%	0.7	0.89

Surah Pendek	Al Qaari'ah	40	72.50%	0.78	0.9
	At Takaatsur	32	84.37%	0.84	1
	Al Kaafiruun	31	87.09%	0.9	0.96
	Al Ikhlaash	19	94.73%	0.94	1
	Al Falaq	27	77.77%	0.84	0.91
	An Naas	24	95.80%	0.95	1

From table above we can see that token number has big influence to system performance. Some surah that have more tokens have better performance. For example, the comparison between An-Naazi'aat and Al-Qaari'ah. When system processing An-Naazi'aat that has 183 tokens, then stemmer accuration, ICF, word class accuration, and precision that produced is bigger than Al-Qaari'ah that has 40 tokens. This is happened because terms in this surah have similar patterns. For example the first 4 ayat in surah, pattern that used are فاعلات and فعلا , that can be stemmed directly. For example, word نازعات will be erased its suffix ات in step 6. Then, in step 7, the rest of the word نازع will be matched with pattern saved in system. The matching pattern for this word is pattern فاعل so that the stem is نزع . The recall value that obtained from Al-Qaari'ah is higher than An-Naazi'at. This is because the increasing of term number will make higher probability of pattern that used, so if system is wrong in determining word class of a word, then it will influence system in determining word class from other terms. As explained in chapter 2, if system can't determine word class of a term in the forst two steps, system will do syntax analyzer, where in this step word class sequence will have big influence.

From the table above we also can see some tokens have less numbers that have better performance. For example, the comparation between Al-Kaafiruun and Al-Qaari'ah. Al-Kaafiruun has 31 tokens, stemmer accuration reaches 100%. With less token numbers, then the pattern probability of the right term produced is bigger. Beside that, in this surah there are 5 unique terms, اَعْبُدُ (I pray), تَعْبُدُوْنَ (you pray), عَابِدُوْنَ (prayer-plural), عَابِدٌ (prayer-singular), and عَبَدْتُمْ (you prayed to), where all words coming from the same stem عبد (pray). Therefore, the more condense of stem that condensed by stemmer, then the ICF value that produced will be bigger.

5 Conclusions and Future Works

5.1 Conclusions

1. Pattern-based stemmer is algorithm that efficient enough to determine stem from Arabic texts.
2. Prefix-free and suffix-free patterns that saved in system have influence in accuration level and ICF. The increasing number of affix-free pattern that saved in system, then stemmer performance will have better result.
3. Affix-free patterns that saved in system have influence in word class accuracy, precision, and recall. The increasing number of affix-free pattern that saved in system, then word class performance will have better result.
4. Word class determiner rules have influence in performance of word class determining process. The increasing number of applied rule, then the performance will have better result.
5. The number of tokens don't associate directly with system performance.

5.2 Future Works

1. Input that used in the system are all kinds of Arabic texts, not only surah in Al Qur'an
2. Analyze and evaluate the stemming influence to text categorization
3. Tag/word class becomes more detail. For example : fa'il (noun subject), adverb place (zharaf makan), etc

References

[1] Agusta, Ledy. 2009. Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief & Adriani untuk Stemming Dokumen Teks Bahasa Indonesia. Konferensi Nasional Sistem dan Informatika 2009. Tersedia di : http://yudiagusta.files.wordpress.com/2009/11/196-201-knsi09-036-perbandingan-algoritma-stemming-porter-dengan-algoritma-nazief-adriani-untuk-stemming-dokumen-teks-bahasa-indonesia.pdf diunduh pada tanggal 26 Maret 2010.

[2] Alshalabi, Riyad. 2005. Pattern-based Stemmer for Finding Arabic Roots. Information Technology Journal 4 (1): 38-43, 2005 ISSN 1812-5638. Tersedia di : http://198.170.104.138/itj/2005/38-43.pdf diunduh pada tanggal 26 Maret 2010
[3] Al-Atsary, Abu Hamzah Yusuf. 2007. Pengantar Mudah Belajar Bahasa Arab. Bandung : Pustaka Adhwa.
[4] Al-Taani, Ahmad & Al-Rub, Salah. 2009. A Rule-Based Approach for Tagging Non-Vocalized Arabic Words. The International Arab Journal of Information Technology, Vol. 6, No. 3, July 2009. Tersedia di : http://www.ccis2k.org/iajit/PDF/vol.6,no.3/17.pdf diunduh pada tanggal 30 April 2011.
[5] BBC - اكتشاف بروتين يساعد في التنبوء بالزهايمر. 2011. Tersedia di : http://www.bbc.co.uk/arabic/scienceandtech/2011/06/110623_alzheimer_research.shtml diakses pada tanggal 24 Juni 2011.
[6] CIA - The World Factbook – Indonesia. 2010. Tersedia di: https://www.cia.gov/library/publications/the-world-factbook/geos/id.htm diakses pada tanggal 26 Maret 2010.
[7] Dwiswistyan, Fiqi. 2009. Pengaruh Affix Removal dengan Porter Stemmer dan Krovetz Stemmer dalam Kategorisasi Berita Berbahasa Indonesia. Tugas Akhir Teknik Informatika Institut Teknologi Telkom.
[8] Indradjaja, Lily Suryana. & Bressan, Stephane. Automatic Learning of Stemming Rules for the Indonesian Language. Tersedia di : http://www.aclweb.org/anthology-new/Y/Y03/Y03-1007.pdf diunduh pada tanggal 26 Maret 2010.
[9] Larkey, Leah S., Esteros, Lisa Ba. & Conne, Margaret E. Improving Stemming for Arabic Information Retrieval :Light Stemming and Co-occurrence Analysis.
[10] Muhyidin, Muhammad. 2007. Tuhfatus Saniyah. Tegal : Ash-Shaf Media.
[11] Putri, Amelia Yosi. 2009. Stemming untuk Teks Berbahasa Indonesia dan Pengaruhnya dalam Kategorisasi. Tugas Akhir Teknik Informatika Institut Teknologi Telkom.
[12] Purwantiningsih, Oky. 2005. Perangkat Lunak Kamus Berintelegensia untuk Bahasa Indonesia untuk Menentukan Kelas Kata Berdasarkan Kelas Akar Kata dan Imbuhan. Tugas Akhir Teknik Informatika Institut Teknologi Telkom
[13] Waiyamai, Kitsana. Introduction to Text Mining. Dept of Computer Engineering, Faculty of Engineering, Kasetsart University, Bangkok, Thailand.
[14] School of Computing & Communications. Stemming Performance. Tersedia di : http://www.comp.lancs.ac.uk/computing/research/stemming/general/perfomance.htm
[15] Zaid, Bakr bin ‘Abdillah Abu. 1995. Hilyatu Thalibil ‘Ilmi. Arab Saudi : Darul ‘Ashimah.
[16] Zakaria, Aceng. 2009. Al-Muyassar fii ‘Ilmi An-Nahwi. Garut : Ibnu Azka Press.
[17] Zakaria, Aceng. 1996. Al-Kafi fii ‘Ilmi Ash-Sharfi I.
[18] Zakaria, Aceng. 1997. Al-Kafi fii ‘Ilmi Ash-Sharfi II.