Data Set
for Designing and Testing an Arabic Stemmer

Ibrahim A. Al kharashi
Tel: 481-3273, fax: 481-3764
Kharashi@kacst.edu.sa

Imad A. Al sughaiyer
Tel: 481-3217, fax: 481-3764
imad@kacst.edu.sa

Computer and Electronics Research Institute
King Abdulaziz City for Science and Technology
P. O. Box 6086, Riyadh 11442, Saudi Arabia

ABSTRACT

Arabic language has unique characteristics that greatly affect its automation. Arabic language exhibits a very complex but very regular morphological structure. Different proposed morphological analysis techniques for the Arabic language are based on heavy computational processes and/or the existence of large amount of associated information. Researchers in the field of Arabic computational linguistics faced with some basic technical difficulties including lack of propre evaluation and testing frameworks. Because of that, researchers in their works provided general description for approaches with almost no effectiveness or efficiency measures.

This work proposed an initiative for a framework to be used for testing and evaluating Arabic morphological and stemming techniques. A new Arabic stemmer is proposed where the generated data set were used to construct, test and evaluate the stemmer.

INTRODUCTION

Stemming and morphological analysis techniques are computational processes that analyze natural words by considering their internal structures. Stemming techniques usually deal with languages with simple morphological systems while morphological techniques are widely used in languages with complex morphological systems. Stemming and morphological analysis techniques can be viewed as clustering mechanisms and usually help in resolving the lexical ambiguity. The main objective of the stemming algorithms and one objective of morphological analysis techniques is to remove all possible affixes and thus reduce the word to its stem. Both processes are very useful in many natural language applications such as information retrieval, text classification and categorization, text compression, data encryption, vowelization and spelling aids and automatic translation (Lovins, 1968; Dawson, 1974).

Semitic languages require more complicated systems for processing their morphology. Arabic language, for example, consists of a very complex but very rich and regular morphological structure. English has a simple morphology compared to other languages. European languages involve more complex morphology than does English (Savoy, 1999).

In Arabic, a root is a single morpheme that provides the basic meaning of an Arabic word. Arabic root is the word's origin before any transformation process. A stem, on the other hand is a morpheme or a set of concatenated morphemes that refers to some central idea while a word is the single isolated lexeme that represents certain meaning.

An affix is a morpheme that can be added before, after or inserted inside a root or a stem as a prefix, suffix or infix respectively to derive new words or meaning. Arabic prefixes are derived from small set of letters and articles, while suffixes are derived from small set of letters, articles and pronouns. Removal of prefixes in Arabic is not harmful process most of the time because, as oppose to English, the process does not reverse the meaning of the word.

A pattern is a model used to study the internal structure of Arabic words. It consists of the three basic Arabic pattern letters that corresponds to the first, second and third letter of the Arabic trilateral root respectively. For the quadrilateral roots, the third letter is duplicated to represent the fourth root letter. In addition, zero or more augmented letters or one or more short vowels are inserted to expand the pattern.

Computational Arabic morphology drew the attention during the last two decades. This, consequently, has led to the emerging of some morphological analysis techniques. Arabic morphological analysis techniques can be categorized into table lookup, linguistic and combinatorial approaches (Ali, 1988; EL-Affendi, 1991; Al-Fedaghi & Al-Anzi, 1989). Some researchers suggested analyzing Arabic words to reach their roots (Ali, 1988) while others suggested analyzing them to their

stems only (Alsuwaynea, 1995; Al-Atram, 1990). Analyzing words to their roots is preferred in linguistic processing-based applications while analyzing words to their stems is more useful in some other applications such as information retrieval-based systems. A simplified system for generating/analyzing Arabic words is shown in Figure 1.

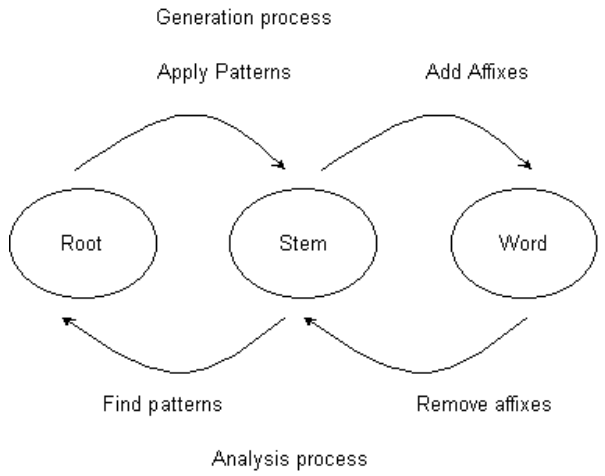

Figure 1. Arabic system for generating/analyzing words

Researchers in the field of Arabic computational linguistics faced with some basic technical difficulties including lack of propre evaluation and testing frameworks. Because of that, researchers in their works provided general description for approaches with almost no effectiveness or efficiency measures.

This work proposed an initiative for a framework to be used for testing and evaluating Arabic morphological and stemming techniques.

ARABIC DATA SET

The framework is based on a data set initially used to design and evaluate a proposed Arabic stemmer. The data set is a collection of about 23,000 Arabic words extracted from 100 short Arabic articles collected randomly from the internet. Extracted words were normalized by removing vowels and then stored in a binary file in the same order as the original natural text. Since word order was preserved, it is very easy to deduce the contextual meaning of any word by listing few words before and after the current word.

Structure of the word record is shown in Figure 2. Each word in the data set was manually investigated to produce morphological components including stem and affixes. In this work, the stem is defined as a singular, masculine and past tense Arabic word without affixes. To guarantee an adequate level of accuracy, an Arabic linguist has been consulted during this stage.

Figure 2. Data structure used in storing words.

If expanded, this data set can be used for other linguistic studies and researches such as morphological analysis techniques, affixation compatibility and different frequency analysis.

Usually, gathered natural text used in modern Arabic is full of spelling errors and spelling variations. Errors corrected partially during the manual processing stage and then completed semi-automatically. Table 1. Lists some examples of errors and spelling variations.

Following is some statistical characteristics of the data set. Figure 3. shows the length distribution of words. Most of the words with length of two letters and some of those with length of three letters are stop words. Furthermore, most of words with the highest lengths are foreign words. Figures 4 and 5 show frequency distribution for word and stem respectively. Figures show normal distribution over the collection.

Spelling mistake	Arabic terminology	Example
Using different spelling variations of foreign words	تهجئة الكلمات الأجنبية	انترنت و انترنيت
Compound nouns. With and without space between parts.	الأسماء المركبة	عبد الرحمن و عبدالرحمن
Confusing between Arabic letters	ه and ة ا and أ ى and ي	هرة – هره استنجار – أنباء علي – على

Table 1. Spelling errors and variations

Table 2. and Table 3. give the frequency of prefixes and suffixes. Such statistics are very useful in different computational and linguistic studies.

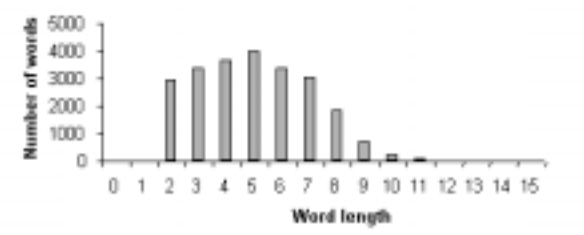

Figure 3. Word length distribution.

Figure 4. Word frequency distribution.

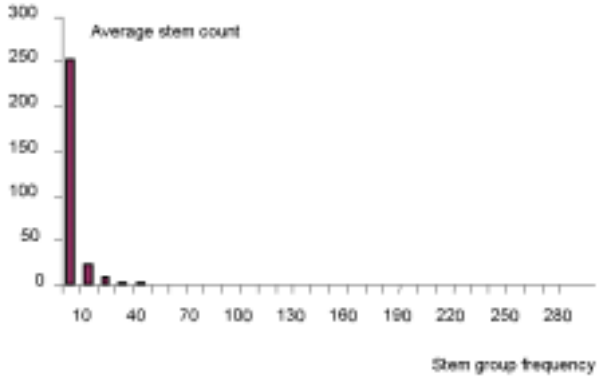

Figure 5. Stem frequency distribution.

Prefix	freq	Prefix	freq	Prefix	freq	Prefix	freq
ال	6890	لل	368	وس	32	ولل	5
و	1484	بال	260	ول	18	كال	5
ب	620	ف	111	وب	16	فال	5
ل	575	س	100	وبال	15	فل	2
وال	476	ك	82	وت	11	فس	2

Table 2. Prefix List Derived from data set

Suffix	freq	Suffix	freq	Suffix	freq	Suffix	freq
ت/ة	1312	وا	44	اتهم	7	وه	2
ات	1136	اتها	34	يون	6	تم	2
ية	1060	ما	32	هما	6	يه	1
ي	698	يات	23	اتية	6	يتها	1
ها	570	تها	21	كم	5	يتنا	1
ه	494	ته	15	ونه	4	وهم	1
ا	445	اته	15	يها	3	وننا	1
ين	256	ان	14	وها	3	تين	1
ون	141	يين	11	هن	3	تموها	1
هم	134	و	9	تهم	3	تان	1
يا	68	ك	9	تنا	3	اها	1
نا	61	ني	8	اتنا	3	اتكم	1

Table 3. Suffix List Derived from data set

PATTERN-BASED ARABIC STEMMER

In this section, a new approach that utilizes the apparent symmetry of generated natural Arabic words is introduced. In this approach, a unique regular expression-based rule is generated for group of similar Arabic words. Rules are used to describe the internal morphological structure of Arabic words and guide the decomposition process of a given word to its basic units i.e. stem, prefix and suffix. A very simple rule parser was developed to perform the analysis to process and extract word morphological components.

Created rules are written from right to left to match script writing direction of Arabic language. Rule pattern may contain up to three distinct parts. The first and last parts describe affixation properties of the word while the middle part controls the stem extraction process. Pairs of angle brackets surround affixation parts. Absence of prefix or suffix in the rule patterns is sometimes denoted by empty angle brackets. This is necessary in order to distinguish them from an angle-bracketed part of the stem.

The complexity of rules varies from very simple passive ones to very complicated rules that deal with complex morphological behaviors. Set of passive rules is created to handle words already in stem forms, isolated articles, proper names and foreign words.

A rule will be fired if it has the same length as the length of the inspected word. A match is achieved if and only if a fired rule produces the correct prefix, stem and suffix. A given word should fire at least one rule and match only one rule.

EXPERIMENTATION

Created data set has been used in the design and implementation stages of the stemmer. The first part of the experiment was designed to study rule growth in a natural text. In this part each word passed to the parser for analysis. The parser has access to list of accumulated rules. The parser tries to fire rules in sequence. On match, the word structure will be updated with number of fired rules, the id of matched rule and its sequence. On mismatch, a new rule should be created and appended to the rule list.

Figure 6 shows the growth of rules. It shows very rapid growth at lower number of words and a tendency to

be stabilized as more words introduced. Figure 7 depicts number of generated rules for every thousand words. It clearly shows that number of generated rules decreases as number of words increases.

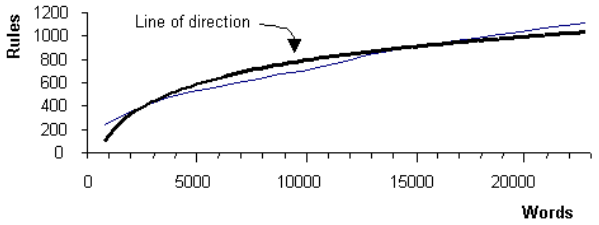

Figure 6. Rules growth per 1000 words

Figure 8 shows the length distribution of words and created rules for the test collection. It can be deduced that majority of rules were generated by words of length 5, 6, and 7 letters. This is a normal phenomenon because words of such lengths are more likely to have diverse kind of affixes. Existence of affixes, consequently, produces more rules. For words with shorter lengths, number of introduced rules were low due to the fact that shorter words are most likely to be particles or words already in stem forms. Fewer rules were introduced for words with longer lengths because most words are either proper names or foreign words. Such type of words is less likely to have affixes.

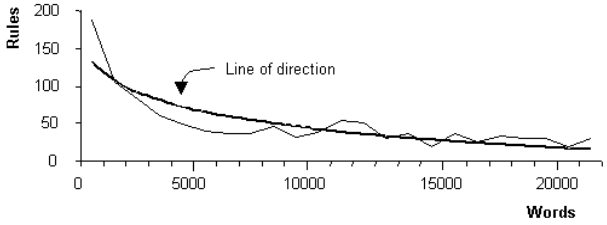

Figure 7. Number of rules generated per 1000 words.

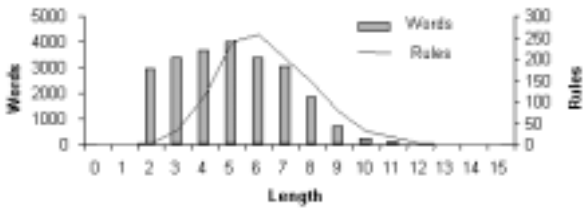

Figure 8. Distribution of rule and word lengths

The order of rule firing plays an important role in the efficiency of the analyzer. For a given word, it is desirable to fire less number of rules and to maintain firing order in such a way that first fired rule is the matched one. Figures 9 and 10 show the relationship between matched and total firings per rule. Having different rule orders will produce different plots. In order to achieve optimized performance the curve of Figure 9. should follow the horizontal line or the scattered points in Figure 10. aligned with the diagonal line. Although it is impractical to achieve such optimum state, it is possible to have certain rule ordering that produces the best performance for such rule set.

Figure 11. indicates that average fired rules is in-line with the conclusion derived from Figure 8. For optimized analyzer, it is desirable to keep average fired rule for each word length class as low as possible. Also, for optimization, it is needed to keep the sequence of the matched rule at the top of firing sequence.

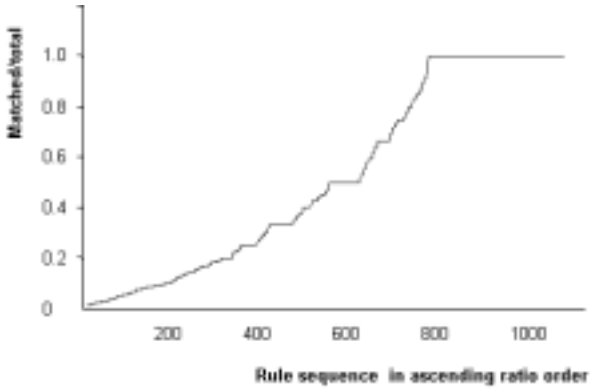

Figure 9. Relation between matched and total firing.

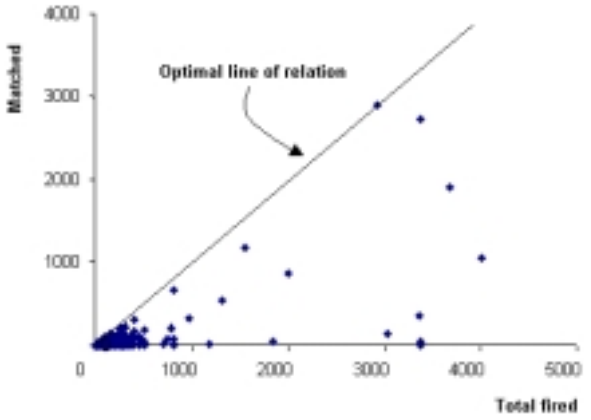

Figure 10. Match vs. total per rule matched and total firing.

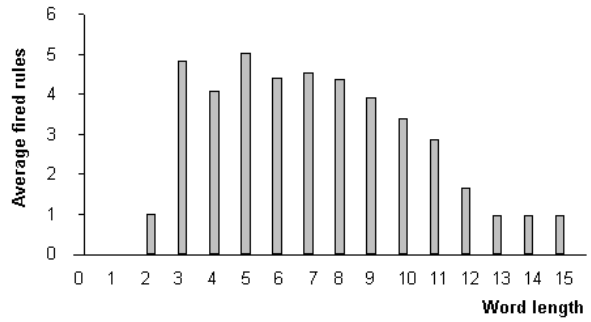

Figure 11. Average fired rules vs. word length.

CONCLUSION

Known Arabic morphological analysis techniques suffer from few problems including the need for testing and evaluating frameworks. This paper proposed an initiative for a framework to be used for testing and evaluating Arabic morphological and stemming techniques. The framework is based on a data set initially used to design and evaluate a proposed Arabic stemmer. The data set is used to construct, test and evaluate the stemmer.

This data set can be used for other linguistic studies and researches such as morphological analysis techniques, affixation compatibility and different frequency analysis.

REFERENCES

Lovins, J. (1968). Development of a stemming algorithm. Mechanical Translation and Computational Linguistics, No. 11, (pp 22--31).

Dawson, J. (1974). Suffix removal and word conflation. *ALLC Bulletin*, 2(3), 33--46.

Savoy, J. (1999). A stemming procedure and stopword list for general French corpora. *Journal of the American Society for Information Sciences*. 50(10), 944--952.

Ali, N. (1988). Arabic Language and Computer. Ta'reeb. (in Arabic)

El-Affendi, M. (1991). An algebraic algorithm for Arabic morphological analysis. The Arabian Journal for Science and Engineering. 16(4B), 605--611.

Al-Fadaghi, S. and Al-Anzi, F. (1989). A new algorithm to generate root-pattern forms. Proceedings of the 11th National Computer Conference, KFUPM. (pp 391--400).

Alsuwaynea, A. (1995). Information Retrieval in Arabic language. King Fahad National Library, (in Arabic).

Al-Atram, M. (1990). Effectiveness of Natural Language in Indexing and Retrieving Arabic Documents. KACST, AR-8-47. (in Arabic).