Stemming techniques of Arabic Language: Comparative Study from the Information Retrieval Perspective

Mohamed I. Eldesouki
Institute of Statistical Studies and Research
Computer Science Department
Cairo University
5 Dr. Ahmed Zwel Street, Orman, Giza, Egypt
disooqi@ieee.org

Waleed M. Arafa
Institute of Statistical Studies and Research
Computer Science Department
Cairo University
5 Dr. Ahmed Zwel Street, Orman, Giza, Egypt
waleed_arafa@hotmail.com

Kareem M. Darwish
Microsoft
Microsoft Innovation Center
Smart Village –Building B115
Kilo 28, Cairo/Alex. Desert Road
Abou Rawash, Egypt
kareemd@microsoft.com

Abstract.

Arabic language considered one of the most challenging languages for solving the problem of matching in information retrieval, since it depends on both inflectional and derivational morphology, and it has a templatic morphology. Scientists found in their resent studies that using stems as index terms outperform roots. The most popular and successful technique used for producing stems of words is the light stemming techniques. Many studies have been conducted in light stemming since TREC 2002 Cross-language track. This paper aims to compare the most of the existing light stemmers in terms of main ideas, affixes lists, algorithms, and information retrieval performance. The results shows that the light10 stemmer outperformed the other stemmers in non-expanded experiments for the stemmers and Aljlayl-3 outperform them in case of expansion.

1. Introduction

The most challenging problem that faces information retrieval (IR) field is the matching problem. There are many cases when two words are not quite the same but you would like a match to occur. Different techniques have been developed to handle this problem depends on the nature of the language. Arabic language considered one of the most challenging languages for solving this problem. The Arabic language depends on both inflectional and derivational morphology to produces the various forms of the language words. It has a templatic morphology; that interweaves the roots to the patterns. The Arabic consists of a few thousands of roots.

Many techniques have been used for beating this problem for the Arabic language. At the very beginning, researchers tried to use dictionaries of roots and stems, built manually, for each word to be indexed. The roots and stems extracted from a very small collection of text (Al-Kharashi & Evens, 1994). This method is not suitable especially when the collection is very big. People tried to use Arabic morphological Analyzers to obtain the roots of the words automatically to be indexed. A lot of analyzers exist in that time have been used and evaluated; Khoja Morphological Analyzer (Khoja, 1999), Tim Buckwalter morphological analyzer 1.0 (LDC, 2002), ALPNET morphological analyzer (Beesley, 1996), and Sebawai (Darwish, 2002a).

A controversial issue at that time was whether to use roots or stems as terms for indexing. And a lot of studies have claimed that roots outperform stems (Al-Kharashi & Evens, 1994), (Abu-Salem, 1999) and (Darwish, 2001). However, most of the resent studies found that using stems as index terms outperform roots; (Aljlayl, 2002), (Larkey, 2002), (Darwish, 2002b), (Larkey, 2005), (Taghva, 2005), (Darwish, Hassan & Emam, 2005). The reason that the former researchers, that found the root better than stems for IR tasks, have done their experiment on small collections of text which is not enough for judgment.

TREC 2001 and TREC 2002* Conferences help a lot for improving the performance of Arabic information retrieval systems. They also helped in evaluating the different techniques for handling Arabic language. They provided, with help from Linguistic Data Consortium LDC**, a potentially large text collection to be used in evaluation. This helped in deciding which is more appropriate for use as index term in Arabic information retrieval systems.

Using the TREC-2001 Arabic corpus (LDC, 2001), they found that roots are not suitable because Arabic consists of a few thousands of roots. Analyzing each word to its root would conflate many words of different meaning to the same class. For example, the Arabic words for *office*, *book*, *Library*, *writer*, and *letter* have same root.

After TREC Arabic cross-language Information retrieval tracks (CLIR) (Gay & Oard, 2002), researchers have directed their research to use stems as index terms. They developed a lot of stemmers to handle Arabic Language in IR context. Many studies have been conducted in stemming techniques; (Darwish, 2002b), (Aljlayl, 2002), (Larkey, 2002), (Chen & Gay, 2002), (Larkey, 2005), (Al Ameed et al., 2005), (Nwesri, 2005), (Kadri & Nie, 2006), (Nwesri, 2007), and (El-Beltagy & Rafea, 2009). However, most of the recent stemming techniques haven't received a real information retrieval evaluation.

This paper is a comparative study for the most of the existing light stemmers. It compares stemmers in terms of the main ideas behind the development of the stemmers, the prefixes and suffixes they remove, and the algorithms they use to remove the affixes. The stemmers also compared in terms of their IR performance; precision and recall. Some of the stemmers are built for the evaluation purpose.

The rest of the paper is organized as follows: section 2 presents the various definitions of stemming and classifications of the existing stemmers; section 3 presents the criteria the comparison based upon and the stemmers which going to be compared; section 4 compares between the different stemmers, section 5 describes the experiment carried out to evaluate the stemmers. Results and discussions are provided in section 6 and conclusion and future work is derived in section 7.

2. Stemming Definition and Stemmers Classification

Stemming has multiple definitions. Shereen Khoja's definition (Khoja, 2001) limits stemming for Arabic language to the root extraction process. She has defined the stemming process as "…Stemming is the process of removing all of a word's affixes to produce the stem or root. In Arabic this means the removal of prefixes, suffixes and infixes. The stemming component is the

*http://trec.nist.gov/

** http://www.ldc.upenn.edu/

rule-based part…”. However, Leah Larkey (Larkey, 2002) was more general in her definition. She could fetch more techniques under the stemming umbrella. She defined stemming processes as “…we use the term *stemming* to refer to any process which conflates related forms or groups forms into equivalence classes, including but not restricted to suffix stripping ….”. This definition considers more stemmers than Khoja definition. For example, light stemmers and statistical n-gram methods to conflate words to same class are considered stemmers as well as stemmers that extract roots. A close definition to Larkey’s one is (Al-Sughaiyer and Al-Kharashi, 2004) definition. They defined the stemming as, ”… *Stemming* is a method of word standardization used to match some morphologically related words. The *stemming algorithm* is a computational process that gathers all words that share the same stem and have some semantic relation …” The adopted definition for this paper is the Larkey’s definition.

There were also a lot of attempts to classify the existing stemmers. Abdusalam Nwesri (Nwesri, 2005) has classified the stemmer in to heavy stemming, or root-based stemmer, and light stemming. Heavy stemming usually starts by removing well-known prefixes and suffixes. It aims to return the actual root of a word. Light stemming stops after removing prefixes and suffixes, and does not attempt to identify the actual root. And he further categorizes the light stemmers into three categories according to the way in which existing stemmers deal with particles; conjunctions and prepositions. He also mentioned that a stemmer can combine between any of these approaches:

- Match and Truncate (MT): the beginning of a word is removed if a match happens and the remaining words more than 3 letters length.
- Remove and Check (RC): the beginning of a word is removed if a match happens and the remaining word exists in the document collection.
- Remove With Other Letters (RW): removing a combination of particles and the definite article ال like فال , وال, كال, and بال

Larkey (Larkey, 2002) divided the Arabic stemmers into four classes:

- Manually constructed dictionaries
- Algorithmic light stemmers; which remove prefixes and suffixes
- Morphological analyses which attempt to find roots
- Statistical stemmers, which group word variants using clustering techniques.
    - new statistical methods involving parallel corpora

Other classification is done by (Al-Sughaiyer and Al-Kharashi, 2004) and (Al-Hajjar, 2009). However, in this paper we are going to adopt the Larkey’s classification since it considers wider range of stemmers.

3. Scope and Criteria of Comparison

Due to the excel of the light stemmers when compared with other techniques for stemming for the purpose of information retrieval, the scope of this study is the light stemming techniques. This paper is going to consider most of the studies in light stemming. It will compare between

the following stemmers: Al-Stem for Kareem Darwish (Darwish, 2002b), (Aljlayl, 2002), Light8 for Leah Larkey (Larkey, 2002), Berkeley Light Stemmer (Chen & Gay, 2002), Light10 (Larkey, 2005), SP_WOAL Light Stemmer (Al Ameed et al., 2005), Restrict Stemmer (Nwesri, 2005), (Nwesri, 2007), linguistic-based stemmer (Kadri & Nie, 2006), and Domain-Specific Stemmer (El-Beltagy & Rafea, 2009).
This paper is going to compare the stemmers in terms of:

- The main idea behind the stemmer built,
- The prefixes and suffixes they remove, and
- The basis of choosing the affixes
- The algorithm they use to remove the affixes.
- IR performance; precision and recall.
- Limitation of the stemmer

4. The Stemming Techniques Comparison

4.1. Al-Stem Stemmer; (Darwish, 2002b)

A stemmer developed by Kareem Darwish and modified by Leah Larkey from University of Massachusetts and further modified later by David Graff form LDC. It is intended for research purposes only. The original stemmer of Kareem Darwish removes the following prefixes: (وال، فال، بال، بت، يت، لت، مت، وت، ست، نت، بم، لم، وم، كم، فم، ال، لل، وي، لي، في، وا، فا، لا، با) and removes the following suffixes: (ات، وا، ون، وه، ان، تي، ته، تم، كم، هم، هن، ها، ية، تك، نا، ين‘ يه، ة، ه،ي). In David Graff version of the stemmer the stemmer two additional prefixes are removed (تت، سي) and two additional suffixes are removed; (تا، ا). Kareem Darwish claims that the Al-Stem is more Aggressive than Light10 stemmer. Graff version of the stemmer has two modes for normalization light and aggressive.
The stemmer works as follows:

- Remove the following prefixes if exist from beginning of the word in the next order from right to left: (وال، فال، بال، بت، يت، لت، مت، تت، وت، ست، نت، بم، لم، وم، كم، فم، ال، لل، وي، لي، سي، في، وا، فا، لا، با)
- Remove the following suffixes if exist from the end of the word in the next order from right to left: (ات، وا، تا، ون، وه، ان، تي، ته، تم، كم، هن، هم، ها، ية، تك، نا، ين، يه، ة، ه،ي، ا)

4.2. Aljlayl Stemmer (Aljlayl, 2002)

Mohammed Aljlayl developed a light stemmer used for his own information retrieval researches in TREC cross-language track. Aljlayl didn’t mention the prefix or suffix list going to be removed from word rather he mentioned only that “… to remove the most frequent suffixes and prefixes..,” then he said “The most common suffixation includes duals and plurals for masculine and feminine, possessive forms, and pronoun forms,” and “The definite articles and prefixes that can be attached to the head of the definite article are considered the most common prefixes. In addition, the letter (و) is a commonly used letter to start the sentences within the Arabic language,” The algorithm of stripping the affixes is as follow:

- If word length is greater than or equal 3 characters, then remove the prefix و
- Remove the article from the beginning of the word if exist then normalize أ، إ، آ from the beginning of the word to ا
- If the length of the remaining stem is greater than or equal 3 characters, then remove the suffixes form the stem using longest first strategy (remove the longest suffix first) only if remaining part of the stem is greater than or equal 3 characters.
- **While** length of the remaining stem is greater than 3 characters **do**,
    - Remove the prefixes form the stem only if remaining part of the stem is greater than 3 characters.
- Return the stem

4.3. Light8 Stemmer; (Larkey, 2002)

It is a light stemmer developed by Leah Larkey for the purpose of researching. The construction of the stemmer based on heuristics; try to remove strings which would be found as affixes far more often than they would be found as the beginning or end of an Arabic word without affixes. The stemmer removes the following prefixes: (ال، وال، بال، كال، فال، و) and the following suffixes: (ها، ان، ات، ون، ين، يه، ية، ه، ة، ي)

The stemmer works as follows:

- Remove و if the remainder of the word is 3 or more characters long.
- Remove any of the definite articles if this leaves 2 or more characters.
- Remove any of the following suffixes in order form right to left (ها، ان، ات، ون، ين، يه، ية، ه، ة، ي) if this leaves 2 or more characters.

4.4. Light10 Stemmer; (Larkey, 2005)

Light8, which has been developed by Leah Larkey, becomes Light10 after some modifications. Light10 was designed to strip off strings that were frequently found as prefixes or suffixes, but infrequently found at the beginning or ending of stems *without intended to be exhaustive*, as light8 did before. Light10 tries to improve the Information Retrieval (IR) performance. Larkey used heuristic as a strategy for developing here stemmer. And it did it. It outperforms most of the morphological analyzers in that time; Amira 1.0, Khoja, Buckwalter morphological analyzer, etc. This is important because some researcher claims that light10 is not good because it doesn't return the right form of the word.

The stemmer removes the following prefixes: (ال، وال، بال، كال، فال، لل، و) and it removes the following suffixes: (ها، ان، ات، ون، ين، يه، ية، ه، ة، ي). The Light10 stemmer removes the same set of suffixes as Light8. However, Light10 add (لل) to the prefix list to be removed. This addition made Light10 outperform Light8.

Something that is notable in Larkey stemmers, Light8 and Light10, that they only remove definite articles. The stemmers don't remove any Arabic prefixes from words.

Light10 stemmer works as follows:

- Remove و if the remainder of the word is 3 or more characters long.
- Remove any of the definite articles if this leaves 2 or more characters.
- Remove any of the following suffixes in order form right to left (ها، ان، ات، ون، ين، يه، ية، ه، ة، ي) if this leaves 2 or more characters.

4.5. SP_WOAL Stemmer; (Al Ameed et al., 2005)

Al Ameed has reviewed multiple stemmers used in TREC 2001 and 2002 cross-language track. He reviewed the Al-Stem, Light8 stemmer and another stemmer he called it TREC-2001 stemmer; which is a modified version of Larkey's Light8 stemmer. Then he decided to enhance the performance of these stemmers in two ways. First enhancement is done by adding new affixes to the existing affixes of the mentioned stemmers. The second way is by modifying the sequence of algorithm components execution.

Although the author was intending to develop a stemmer to improve the performance of information retrieval tasks, he didn't conduct any IR evaluation. He also said that his stemmer is much better than the stemmers that have developed for the TREC cross-language track claiming that his stemmer produces much more correct words than other stemmers and he neglected that it doesn't depend only on the correctness of the words to make an efficient retrieval process.

The enhancement produced in SP_WOAL light stemmer. Although the user mentioned his prefixes list contain 17 two-characters, the list contains only 15. However, it contains 5 single-characters prefixes rather than 3. The stemmer removes the following prefixes (ال، وال، بال، فال، كال، لل، ولل، ب، ل، فا، ست، با، ي، سي، لت، ت، لي، في، وبال، ن، كا، وست، لن، فت، وسن، فن، وسا، ولا، ولي، سا، وسي، ولن، ولت، سن) and removes the following suffixes (ين، ون، ات، ان، ي، ه، ها، هم، ة، يه،كم، نا، وا، تم، ني، يا، ما، تا، كن، ن، و، تن، تك، ته، ك، هن، ت، ا). The stemmer is considered very aggressive stemmer.

The stemmer works as follows:

- Remove the prefix ال from the beginning of the word
- Recursively remove the suffix from the end of the word starting with longest suffixes first.
- Non-recursively removes the prefix form the beginning of the word starting with longest prefixes first.

4.6. Berkeley light stemmer; (Chen & Gay, 2002)

In 2002, University of California at Berkeley participated only in the cross-language track in TREC conference. They developed a light stemmer that made them among the best performers in the track. They used the standard Arabic data collection provided by Linguistic Data Consortium LDC to develop their light stemmer; they chose the affixes with the most frequently occurrence and that give highest performance when practically evaluated using the test collection. They also depended on the on the grammatical functions of the affixes and their English translations to choose their affixes.

The stemmer removes 26 prefixes and 22 suffixes during the stemming processing. The list of the prefixes that should be removed is: (ال، وال، بال، فال، كال، و، لل، ولل، ب، ل، فا، با، سي، وم، وت، وي، وا، لا، وب، ول، وس، كا، مال، اال، سال، لال،) and the list of the suffixes that should be removed is: (ين، ون، ات، ان، ي، ه، ها، هم، ة، ية، كم، نا، وا، تم، ت، هن، تن، كن، ما، يا، ني). The Berkeley light stemmer works as follows:

1. If the word is at least five-character long, remove the first three characters if they are one of the following: (مال، اال، سال، لال، وال، بال، فال، كال، ولل).
2. If the word is at least four-character long, remove the first two characters if they are one of the following: (لل، فا، با، سي، وم، وت، ال، وي، وا، لا، وب، ول، وس، كا،)
3. If the word is at least four-character long and begins with و, remove it.
4. If the word is at least four-character long and begins with either ب or ل, remove ب or ل, only if, after removing the initial character, the resultant word is present in the Arabic document collection.
5. Recursively strips the following two-character suffixes in the order of presentation if the word is at least four-character long before removing a suffix:
   (، ون، ات، ان، ين، تن، تم، كن، كم، هن، يا، ني، يا، وا، ما، نا، هم، ية، ها)
6. Recursively strips the following one-character suffixes in the order of presentation if the character is at least three-character long before removing a suffix: (ت، ي، ه، ة).

4.7. Kadri's linguistic-based stemmer (Kadri & Nie, 2006)

The developing of the Kadri's linguistic-based stemmer depended on idea that the Arabic word consists of five part their order is; antefixes, prefixes, stem, suffixes and postfixes. The first part which is the antefixes is the prepositions and conjunctions. However, the prefixes are the conjugations person of verbs. The suffixes are Termination of conjugation and numbers marks of the nouns. The postfixes are the pronouns that catch up with end of the word.

The list of Antefixes is: وبال، وال، بال، فال، كال، ولل، ال، وب، ول، لل، فس، فب، فل، وس، ك، ف، و، ب، ل and the list of prefixes is: ا، ن، ي، ت. The list of suffixes is: تما، يون، تين، تان، ات، ان، ون، ين، وا، تا، تم، تن، نا، ت، ن، ا، ي، و. And the list of postfixes is: ي، ه، ك، كم، هم، نا، ها، تي، هن، كن، هما، كما.

The linguistic-based stemmer has two phases to work:

1. Training Phase:
   - A list of stems with its frequency occurrence is build for each word using corpus to avoid ambiguity that my happen when removing affixes.
2. The Stemming Phase:
   - The stemmer truncates possible affixes according to the above table.
   - If there an ambiguity raised for the stemmer (more than one combination was available), then stemmer selects the most appropriate candidate; according to corpus statistics computed in the training phase.

	Main Idea	Removed Affixes (in general)	Reason of choosing these affixes	Limitations
Al-Stem	Blindly remove affixes from the beginning and end of the words	prefixes: (وال، فال، بال، بت، يت، لت، مت، وت، ست، نت، بم، لم، وم، كم، فم، ال، لل، وي، لي، في، وا، فا، لا، با) suffixes: (ات، وا، ون، وه، ان، تي، ته، تم، كم، هم، هن، ها، ية، تك، نا، ين‘ يه، ة، ه،ي).	Firstly obtained from the training step of Sebawai. Then the list is refined manually	Remove affixes without any prior knowledge (linguistic rules). Second it very aggressive (which could remove a lot wrong strings from the words beginnings and ends). Does not handle irregular plural
Light8	The stemmers remove only prefixes and suffixes. It tries to remove strings which would be found as affixes far more often than they would be found as the beginning or end of an Arabic word without affixes.	prefixes: (ال، وال، بال، كال، فال، و) suffixes: (ها، ان، ات، ون، ين، يه، ية، ه، ة، ي)	The construction of the stemmers based on heuristics.	Remove affixes without any prior knowledge (linguistic rules) Does not handle irregular plural
Light10		prefixes: (ال، وال، بال، كال، فال، لل، و) suffixes: (ها، ان، ات، ون، ين، يه، ية، ه، ة، ي).		
Aljlayl	The stemmer removes only prefixes and suffixes. It removes the most frequent suffixes and prefixes	didn't mention the Affixes. However he said, "includes duals and plurals for masculine and feminine, possessive forms, and pronoun forms" "The definite articles and prefixes that can be attached to the head of the definite article"	–	Does not handle irregular plural
SP_WOAL Light Stemmer	He tried to be exhaustive; by removing everything could appear as a prefix or suffix.	prefixes (ال، وال، بال، فال، كال، لل، ولل، ب، ل، فا، ست، با، ي، سي، لت، ت، لي، في، وبال، ن، كا، وست، لن، فت، وسن، فن، وسا، ولا، ولي، سا، وسي، ولن، ولت، سن) suffixes (ين، ون، ات، ان، ي، ه، ها، هم، ة، يه،كم، نا، وا، تم، ا، ت، هن، ك، ته، تك، تن، و، ن، كن، تا، ني، يا، ما).		The stemmer hasn't evaluated against IR tasks. Does not handle irregular plural
Berkeley light stemmer	The stemmer is constructed using the Arabic Corpus statistics. It removes only prefixes and suffixes. The suffixes removed recursively while the prefixes are not. He Also check if the remaining word is exist in the corpus for some words only.	prefixes: (ال، وال، بال، فال، كال، و، لل، ولل، ب، ل، فا، با، سي، وم، وت، وي، وا، لا، وب، ول، لال، سال، اال، مال، كا، وس) suffixes: (ين، ون، ات، ان، ي، ه، ها، هم، ة، ية، كم، نا، وا، تم، ت، هن، تن، كن، ما، يا، ني).	Frequently occurrence and give highest performance grammatical functions of the affixes and	Does not handle irregular plural

			English translations	
Kadri's linguistic-based stemmer	Arabic word consists of five parts; antefixes, prefixes, stem, suffixes and postfixes.	Prefixes: (ال، وال، بال، فال، كال، و، لل، ولل، ب، ل، ي، ت، وبالن، وب، ول، ا، ك، وس، فل، فب، ف، فس) Suffixes: (ين، ون، ات، ان، ي، ه، ها، هم، كم، نا، وا، تم، ا، ت، هن، تي، ك، تن، و، ـن، كن، تا، هما، تين، يون، تان، كما، تما)	enumerate all the affixes that could appear as each kind of affixes	Does not handle irregular plural
Restrict Stemmer	His main idea is to retain valid Arabic core words using a large lexicon that contains all the forms of the Arabic language and using simple rules or heuristics exist in Arabic language	Prefixes: (لل، و، ف، ك، ل، ب، ال، ا، ن، س، ي، ت) Suffixes (ة، ه, يه, ية, هن، ان، ات، ين، ون، وا, هما, هم, ها، ي)		rules don't guarantee correctness of hundred percent. He needs a lexicon contains all the forms of all the words in Arabic language which is very difficult to obtain. Doesn't handle irregular plural
El-Beltagy Stemmer	A domain specific stemmer, After removing the affixes from the words the remaining word is looked up in the corpus text and user stems list.	Prefixes: (ال, وال, بال, كال, فال, لل, وبال, ولل, وكال, وفال, و, ك, ف, ل, ب, لا) Suffixes: (يات, ات, اته, ها, ون, وا , ين , ان, ية , يه, هم, ي, ه , ة)		First, the stemmer has to be used with a specific domain. The second limitation is the stem list which is built during the construction phase has to have the user intervention to edit the mistakes done by the stemmer.

Table 1: a summary of differences and similarities between various stemmers

4.8. Restrict Stemmer; (Nwesri, 2005; Nwesri 2007)

(Nwesri, 2005) focused on removing conjunctions, و, and ف, and prepositions, ل, و, ك, and ب, that come as prefixes in the beginning of the words. He (only in 2005) didn't mention other affixes such as articles and suffixes in general. He just tried to find a way to recognize the two types of affixes; the conjunctions and the prepositions.

In (Nwesri, 2007) he developed an Arabic stemmer called *Restrict*. Its main idea is to retaining valid Arabic core words. This because he claimed that removing wrong affixes sometimes results in incorrect stem and in most cases reduces retrieval precision by conflating different words to the same class.

Nwesri used in his proposed technique two things to improve his performance. The first was the Microsoft Office 2003 Arabic spellchecker to ensure that he extracted only correct words. The second was simple rules or heuristics exist in Arabic language to guarantee the correctness of the affixes removal. Although these rules don't guarantee correctness of hundred percent, they improve the information retrieval performance. The rules are removing a prefix keep the remaining word correct, adding the *waw* or *faa* conjunction keeps the modified word correct, altering the a prefix with *waw* and *faa* keeps the modified word correct, and duplicating a particle result in a wrong word except for the *lam* case.

He has a good justification for depending on correctness of words for improving the IR performance as follows, "Although correct words are not the main target of stemming, an incorrect stem can have a completely different meaning and correspond to a wrong index cluster."

The algorithm work as follows:

i. Dealing with لل, prefix:
   a. If the word is correct after removing the prefix لل, then remove it.
   b. Otherwise, we add the letter ا, before the word, if the new word is correct we drop one lam from the original word.

ii. Dealing with ل, particle when precedes definite article ال:
   a. We replace the first lam with the letter ا, if the word exists in the lexicon remove the prefix without check lexicon (he depends on that the words that start with lam cannot preceded by Alef)
   b. We remove the first letter and check to see whether we can drop the first lam.

iii. (Here could be one of the three algorithms suggested by Nwesri)

iv. If a word starting with either waw or faa and after stemming has three or more characters that has either waw, kaf, baa, or lam as its first character

He suggested three algorithms to handle the conjunctions and the prepositions. All the algorithms depend on checking the words in the lexicon after removing the first letter as follows:

- Remove and Check in Lexicon (RCL)
  - The prefix of a word is removed if the remaining word exists in the Arabic lexicon.
- Replace and Remove (RR)
  - Remove the prefix and check the remaining word in lexicon

- If exist, produce to instances of the remaining word by appending waw and faa to the beginning of the word and check them if they correct words
- If both of the new instances are correct then the prefix is removed
- Otherwise the original word is returned

- Replicate and Remove (RPR)
  - Remove the prefix and check the remaining word in the lexicon
    - If the word not exist go to the duplicate step
    - If the word exist return the original word
  - Duplicate the initial letter except the lam and check the new word in the lexicon
    - If the word exist, return the original word
    - Else, remove the prefix
  - For the words start with the letter lam, we add both *baa* and *kaf* instead of replicating them
    - If both new instances are incorrect, we remove the first lam.
    - Else keep the original word.

Stemming algorithm:

- Dealing with لل, prefix:
  - Replace the prefix لل, with ل, if the new word is in the lexicon remove the first ل
  - Else return the original word
- Use the RPR method to remove the conjunctions and prepositions.
- Remove the definite article ال, from the beginning of the word
- If the word starts with س, ي, ت, ن, and ا, generate two instances of the word by adding ك, and ال, to the beginning of the word, if either of the new words exist in the lexicon
  - Return original word
  - Else, remove the starting letter
- Repeat the previous step until the condition is not longer exist
- Remove the suffixes ها, هم, هما, هن
- If the word ends with ان, replace it with ين, and remove it only if the word exists in lexicon.
- If the word ends with ات, replace the suffix with ة only if:
  - Removing the suffix produces a word that exists in the lexicon, or
  - Replacing the suffix with ة, produces a word that exists in the lexicon
- If the word ends with ين, remove the suffix only if:
  - Replacing the suffix with ان, produces a word that exist in the lexicon, or
  - Replacing the suffix with ون, produces a word that exist in the lexicon
- If the word ends with ون, remove the suffix only if replacing the suffix with ين, produces a word that exists in the lexicon,
- Else, remove it if the word start with ي, or سي,
- Remove the suffixes وا, ية, يه, ه, ة
- If the word ends with ي, remove the suffix only if:

- Removing the suffix produces a word that exists in the lexicon, or
- Replacing the suffix with ها, produces a word that exist in the lexicon, or
- Replacing the suffix with ه, produces a word that exist in the lexicon

One limitation of his method is that it needs a lexicon contains all the forms of all the words in Arabic language which is very difficult to obtain. He used Microsoft office 2003 proof kit as resource of Arabic words. It contains about 15,500,000 Arabic words.

4.9. Beltagy Stemmer (El-Beltagy & Rafea, 2009)

Samhaa El-Beltagy and Ahmed Rafea (El-Beltagy & Rafea, 2009) have proposed a stemming technique that not only removes prefixes and suffixes from the beginning and the end of the word, but also converts the irregular plural form of the word to its singular form.

The stemmer also is a domain specific stemmer which conducts stemming according to the domain of the collection of text to be indexed. The domain specific idea is implemented using a stem list that contains the words and their stems. So, before accepting a stem that produce from a word using stem-based stemmer, the system check whether the produced stem exist in the list or not.

This idea is helpful since words could have different stems on different domain. For example, the word "دقيقة" could have the following different meanings: minute and very fine. In the first sense, stemming is going to harm the word. However, stemming will be good choice for the second meaning.

For simplicity, we refer to the stemmer as *Beltagy Stemmer*. The stemmer first has to be built or trained then it can be used for stemming. The construction of the stemmer is done using a subset of the documents to be stemmed. The construction done as follow:

- A subset of documents from the corpus to be stemmed is selected for the stem list building.
- Through a user interface, the user checks the stem list to verify its correctness.
- The user can provides stems he wanted for specific words that he wants to stem them in a particular way.

In the second phase (operational phase),

- Stemming can be done by checking whether the possible stem exists in the stem list or not.
- The stemmer has two modes; restrict mode (the original word is returned if the word does not exist in the stem list or the corpus) and light stemming mode (the stem rather than the original word is returned if the word I not exist in the stem list or the corpus).

Stemming algorithm is done in the two phases. In the training phase the stemmed word checked only in the corpus, but in the stemming phase the word checked in the stem list and the in the document:

- Remove the following prefixes if the length of the remaining word is greater than or equal 2: ال, وال, بال, كال, فال, لل, وبال, ولل, وكال, وفال
- Remove the prefixes و, ك, ف, ل, ب, only if the remaining word exists in the stem dictionary or in the input document collection.
- Remove the prefixes لا, if the length of the remaining word is greater than or equal 2
- Remove the suffixes يات, ات, اته, only if
    - The remaining word exists in the stem dictionary or in the input document collection, or,

  - Adding ة, to the remaining word and the modified word exist in the stem dictionary or in the input document collection,
- Remove the suffixes ها, ون, وا , ين , ان, ية , يه , هم, ي, ه , ة only if
  - The remaining word exists in the stem dictionary or in the input document collection, or,
  - If the remaining word ends with ت, replace the ت, with ة and check if it exists in the stem dictionary or in the input document collection,

There are two limitations for this stemming technique. First, the stemmer has to be used with a specific domain. It cannot be used as a general stemmer. This means that there is no sense to be compared with the Light10 stemmer. The second limitation is the stem list which is built during the construction phase has to have the user intervention to edit the mistakes done by the stemmer.
The table 1 summarizes the similarities, differences and limitations between all the stemmers compared in this study.

5. Experiments and IR performance Comparison of the stemmers

The TREC-2001 Arabic corpus, also called the AFP_ARB corpus, consists of 383,872 newspaper articles in Arabic from Agence France Presse. This fills up almost a gigabyte in UTF-8 encoding as distributed by the Linguistic Data Consortium. There were 25 and 50 topics used in 2001 and 2002 respectively with relevance judgments, available in Arabic, French, and English, with Title, Description, and Narrative fields. We used the Arabic titles and descriptions as queries of the 75 topic in the experiments.
For all the experiments, we used the Lemur language modeling toolkit*, which was configured to use Okapi BM-25 term weighting with default parameters and with and without blind relevance feedback (the top 50 terms from the top 10 retrieved documents were used for blind relevance feedback). To observe the effect of alternate indexing terms, mean average precision was used as the measure of retrieval effectiveness. To determine if the difference between results was statistically significant, a paired t-test (Hull, 1993) and Wilcoxon sign test (Wonnacott, 1990) have been used with p values less than 0.05 as indication for significance.
As a requirement for Arabic text to be indexed with Lemur toolkit, corpus and topics have been first converted to CP1256 encoding. Then a normalization step was performed. The encoding conversion and normalization steps were conducted on both text collection and the topic where queries were extracted.
A normalization step is important for Arabic scripts. Normalization used as a complement task for stemming. They both are used as techniques for helping matching desired words. While stemming is used as a technique of grouping words by considering the different morphology of the words, normalization looks for different writing habits of people. For example, normalization task is taking into consideration whether or not people are neglecting Hamza in writing Alef due to speed, drawing diacritics in seeking meaning or uttering accuracy, using Kashida for decoration, interchanging in using ى and ي, and interchanging in using ة and ه. The normalization steps have to be consistent with the stemming technique so they complement each other. For

*http://www.lemurproject.org/

example, if a stemmer removes only letter ة from end of the word, the normalization has to replace ه with ة so the stemmer removes both.

In our experiments, if we would like to evaluate the stemming techniques we have to unify the environment of the experiments and change only the stemming technique to be used. The normalization step is one the factors that has to be unified for all the experiments. However, for most of the stemmers in this comparison, each stemmer has its own normalization steps. The question is which normalization steps to consider. Unfortunately, some normalization steps of one stemmer could not be suitable for another stemmer; due to the *conflict* which may happen between the normalization steps and the stemming mechanism. For example, some normalization replaces ة with ه other do the opposite which make this conflict. A normalization conflict happens when a normalization step affects the working mechanism of the stemming algorithm. For example, if a normalization step affects the removal of an affix by stemmer.

To unify the normalization steps we have consider the following method: First, normalization steps that are shared among all the stemmers are chosen. Then, for each remaining normalization step check whether it makes any conflict with the stemmers that don't use them. If the no confliction is happen with all the stemmers consider it, otherwise neglect the step. Table 2 shows the stemmers and the normalization steps they perform and status of every normalization step in terms of confliction.

According to the previous analysis we have unified the normalization steps for all experiments and for all stemmers. The normalization step goes as follows (hint: the order of the normalization steps has to be considered):

- Text was broken up into words at any white space or punctuation characters,
- Remove punctuation, diacritics and kashida
- Remove non letters (consider only alphanumeric letters)
- Replace أ، إ، آ with ا
- Replace final ى with ي
- Replace final ة with ه, the step made some changes in the stemming algorithms of some stemmers; Aljlayl stemmer is modified to remove ه instead of ة and Berkeley stemmer is modified to remove يه instead of ية and stop removing ة from the end of the words. Beltagy stemmer also has some minor modifications.
- Replace the sequence ى and ء from the end of the word to ئ.

Criteria	Al-Stem	Light8	Light10	Aljlayl	SP_WOAL	Berkeley	Kadri	Restrict	Beltagy
Replace أ by ا	☑ Light & Aggressive	☑	☑	☑	Not Available	☑	☑	☑	Not Available
Replace إ by ا	☑ Light & Aggressive	☑	☑	☑		☑	☑	☑	
Replace آ by ا	☑ Light & Aggressive	☑	☑	☑		☑	☑	☑	
Replace ء by ا	☑ Aggressive	No conflict	No conflict	Conflict (هدوء)		conflict	conflict (مضيء)	conflict	
Replace ؤ by ا	☑ Aggressive	Conflict (المؤن)	Conflict (المؤن)	conflict		conflict	Conflict (تبرؤ)	conflict	
Replace ئ by ا	☑ Aggressive	Conflict (زبائن)	conflict	conflict		conflict	Conflict (مبادئ)	conflict	
remove diacritics	☑	☑	☑	☑ Except Shaddah		☑	☑	☑	
remove kashida	☑	No conflict	No conflict	No conflict		No conflict	☑	No conflict	
remove punctuation	☑	☑	☑	No conflict		☑	No conflict	☑	
Replace ى by ي in the end	☑	☑	☑	☑		☑	☑	☑	
Remove Non letters	☑	☑	☑	No conflict		☑	No conflict	☑	
Replace ة by ه in the end	No conflict, No effect	☑	☑	Conflict		Conflict	☑	☑	
Replace ه by ة in the end	No conflict, No effect	No conflict	No conflict	☑		☑	Conflict	Conflict	
Replace the sequence ىء by ئ	Conflict, Beneficial	No conflict	No conflict	☑		No conflict	☑	☑	
Replace the sequence يء by ئ	Conflict, Harm (مضيء)	No conflict	No conflict	☑		No conflict	No conflict	☑	
Replace the sequence وء by ؤ	Conflict, Harm (سوء)	No conflict	No conflict	No conflict depends		No conflict	No conflict	☑	
Replace the sequence اا by ا	No conflict, Beneficial	No conflict	No conflict	N/A		Conflict (باآخ)	No conflict	☑	
Separate the word into two words if letter ة exist in the middle	Conflict, Beneficial	Conflict	Conflict	Conflict		Conflict	Conflict	☑	

Table 2: Summary of Normalization steps for each stemmer

6. Results and Discussions

The following table shows the mean average precision of the stemmers. The stemmers are in descending order according to their mean average precisions (MAP) for expanded experiments:

Stemmer	**Unexpanded**	**Expanded**
Aljlayl-3	0.3332	**0.4003**
Light10	**0.3490**	0.3982
Light8	0.3375	0.3930
Aljlayl-1	0.3425	0.3923
Aljlayl-2	0.3411	0.3881
Beltagy_R	0.3320	0.3841
Beltagy_LR	0.3311	0.3830
Restrict	0.3119	0.3774
Al-Stem	0.3188	0.3715
Berkeley	0.3262	0.3656
Kadri	0.3078	0.3594
SP_WOAL	0.2843	0.3549
normalized	0.2478	0.3057
raw	0.2056	0.2645

Table 3: Mean Average precisions for each stemmer with and without relevance feedback (the best performances are shown in bold)

The first experiment was conducted on the Arabic News corpus without performing any normalization steps or stop words removal and was called *raw*. After performing normalization steps and stop word removal we have conduct an experiment called *normalized*. The *normalized* experiment was used as a baseline experiment for all other stemming techniques' experiments. We have stemmed the corpus using each stemmer described above. The stemmers used to stem both the corpus and the topics. For Samhaa El-Beltagy stemming experiments, we have used the stemmer as a general-purpose stemmer. We haven't generated any stem list. Two experiments have been conducted; first one, Beltagy_R, used the restricted version of the stemmer by check the existence of the possible stem in the collection and considers the one which exists only and the second experiment, Beltagy_LR, used the less restriction version of the stemmer which considers the stem whether or not it exists in the collection. We have used Aljlayl stemming algorithm (Aljlayl, 2002) in three experiments using different affixes list. We have used light10 definite articles for all the experiments. For the first experiment, *Aljlayl-1*, we have used Al-Stem prefixes list and Light10 suffixes list. For experiment *Aljlayl-2*, we have used Al-Stem prefixes and suffixes lists. *Aljlayl-3* experiment have used SP_WOAL prefixes and suffixes lists. Although Nwesri (Nwesri, 2005, 2007) have used Microsoft Office 2003 proof toolkit in his research, we have used in our experiment for his stemmer *Restrict*, the Microsoft Office 2007 proof toolkit. The rest of the stemmers have been developed as they mentioned in their studies.

The Mean Average precision measure is sometimes not enough to measure the stemmers' performance. Statistical measures are used to show the possibilities that the differences between stemmers performance occurred by chance. In our study we have used two type of statistical measures paired t-test and Wilcoxon sign test. Table 4 and Table 5 show both measures to stemmers for non-expanded and expanded experiments respectively. Each cell in the two tables has two values; the upper one reflects the t-test probability and the lower is the Wilcoxon test probability. The Dark cells indicate significant differences between stemmers' results according to t-test measure.

Normalized	SP_WOAL	Kadri	Restrict	Al-Stem	Berkeley	Beltagy_LR	Beltagy_R	Aljlayl-3	Light8	Aljlayl-2	Aljlayl-1	Light10	
0.000 0.000	0.003 0.002	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	**Raw**
	0.073 0.052	0.002 0.001	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.001	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	**Normalized**
		0.059 0.052	0.032 0.018	0.013 0.082	0.024 0.032	0.002 0.082	0.003 0.032	0.001 0.032	0.001 0.018	0.001 0.032	0.001 0.018	0.000 0.002	**SP_WOAL**
			0.348 0.322	0.167 0.032	0.106 0.052	0.009 0.082	0.009 0.082	0.005 0.082	0.007 0.002	0.001 0.010	0.001 0.018	0.000 0.001	**Kadri**
				0.220 0.082	0.134 0.124	0.020 0.244	0.024 0.322	0.011 0.244	0.007 0.322	0.002 0.082	0.002 0.052	0.000 0.005	**Restrict**
					0.307 0.322	0.104 0.408	0.098 0.244	0.096 0.408	0.041 0.591	0.017 0.408	0.012 0.322	0.001 0.018	**Al-Stem**
						0.351 0.917	0.327 0.947	0.231 0.677	0.161 0.408	0.043 0.322	0.031 0.408	0.017 0.082	**Berkeley**
							0.324 0.952	0.386 0.408	0.215 0.032	0.093 0.322	0.064 0.244	0.004 0.005	**Beltagy_LR**
								0.436 0.677	0.252 0.052	0.114 0.052	0.077 0.052	0.005 0.005	**Beltagy_R**
									0.281 0.177	0.040 0.023	0.035 0.082	0.003 0.018	**Aljlayl-3**
										0.268 0.591	0.194 0.244	0.014 0.040	**Light8**
											0.241 0.032	0.010 0.000	**Aljlayl-2**
												0.010 0.052	**Aljlayl-1**

Table 4: t-test and Wilcoxon statistical measures for non-expanded experiments

Normalized	Kadri	SP_WOAL	Al-Stem	Berkeley	Restrict	Beltagy_LR	Beltagy_R	Aljlayl-2	Light8	Aljlayl-1	Light10	Aljlayl-3	
0.002 0.000	0.000 0.000	0.001 0.003	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	**Raw**
	0.014 0.010	0.021 0.007	0.003 0.000	0.000 0.000	0.003 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	**Normalized**
		0.605 0.852	0.166 0.208	0.361 0.244	0.138 0.363	0.022 0.147	0.022 0.100	0.008 0.023	0.003 0.003	0.003 0.001	0.000 0.003	0.006 0.023	**Kadri**
			0.122 0.208	0.300 0.322	0.109 0.147	0.035 0.065	0.037 0.208	0.080 0.453	0.012 0.013	0.011 0.065	0.022 0.147	0.010 0.007	**SP_WOAL**
				0.633 0.177	0.331 0.852	0.118 0.852	0.087 0.719	0.072 0.322	0.020 0.280	0.030 0.280	0.005 0.363	0.047 0.280	**Al-Stem**
					0.291 0.591	0.136 0.500	0.123 0.719	0.252 0.792	0.039 0.124	0.035 0.082	0.019 0.052	0.036 0.124	**Berkeley**
						0.356 0.453	0.330 0.280	0.252 0.792	0.163 0.065	0.187 0.147	0.099 0.065	0.047 0.208	**Restrict**
							0.384 0.636	0.270 0.546	0.113 0.040	0.156 0.100	0.026 0.013	0.125 0.280	**Beltagy_L R**
								0.325 0.363	0.129 0.065	0.188 0.065	0.035 0.000	0.151 0.363	**Beltagy_R**
									0.224 0.363	0.210 0.407	0.047 0.174	0.149 0.546	**Aljlayl-2**
										0.554 0.453	0.081 0.453	0.312 0.852	**Light8**
											0.088 0.079	0.279 0.792	**Aljlayl-1**
												0.441 0.792	**Light10**

Table 5: t-test and Wilcoxon statistical measures for expanded experiments

From Experiments we have found that affixes lists to be removed from the words could affect significantly the performance of one stemmer. In our experiments Aljlayl-1, Aljlayl-2 and Aljlayl-3, in each of them we have tried different affixes list for the same algorithm and found the performance of the stemmer has improved significantly when using Al-Stem affixes lists, 03411, against SP_WOAL affixes lists, 0.3332, and the difference is statistically significant with p values of 0.040 and 0.023 for t-test and sign test respectively for the case without relevance feedback. However, in the case of relevance feedback the performance of the stemmer when using SP_WOAL affixes lists outperforms its performance when using Al-Stem affixes lists and the difference is not statistically significant.

7. Conclusion

The results shows that the light10 stemmer outperformed the other stemmers in non-expanded experiments and Aljlayl-3 outperform them in case of expansion.

Aljlayl-1, Aljlayl-2 and Aljlayl-3 experiments shows that different affixes lists could affect significantly the performance of one stemmer.

Aljlayl-2 and Al-Stem experiments shows that using different stemming algorithm for removing affixes even with the same affixes list produce different results.

Acknowledgments

We would like to thank Linguistic Data Consortium (LDC), for providing us with the Arabic text collection “Arabic Newswire part 1”; special thanks for Dr. Christopher Cieri the executive director of LDC for helping obtaining the data. Thanks goes to Dr. Ibrahim Al-Kharashi for his invaluable help.

References

(Abu-Salem, 1999) Abu-Salem, H., Al-Omari, M., and Evens, M. Stemming methodologies over individual query words for Arabic information retrieval. JASIS, 50 (6), pp. 524-529, 1999.

(Al-Ameed et al., 2005) Al-Ameed k. Hayder, Al-Ketbi O. Shaikha, Al-Kaabi A. Amna, Al-Shebli S. Khadija, Al-Shamsi F. Naila, Al-Nuaimi H. Noura, Al-Muhairi S. Shaikha, Arabic Light Stemmer: A new Enhanced Approach, The second international conference on innovations technology (IIT’05), 2005.

(Al-Hajjar, 2009) AL-HAJJAR A., HAJJAR M., ZREIK K., Classification of Arabic Information Extraction methods, Proceedings of the Second International Conference on Arabic Language Resources and Tools, 2009.

(Al-Kharashi & Evens, 1994) Al-Kharashi, I. and Evens, M. W. Comparing words, stems, and roots as index terms in an Arabic information retrieval system. JASIS, 45 (8), pp. 548-560, 1994.

(Al-Sughaiyer and Al-Kharashi, 2003) Imad A. Al-Sughaiyer, Ibrahim A. Al-Kharashi, Arabic morphological analysis techniques: a comprehensive survey, Journal of the American Society for Information Science and Technology, v.55 n.3, p.189-213, February 2004

(Aljlayl, 2002) Aljlayl, M., & Frieder, O, On Arabic search: Improving the retrieval effectiveness via light stemming approach. In Proceedings of the 11th ACM International Conference on Information and Knowledge Management, Illinois Institute of Technology (pp. 340–347). New York: ACM Press.2002.

(Beesley, 1996) Beesley, K. R. Arabic finite-state morphological analysis and generation. In COLING-96: Proceedings of the 16th international conference on computational linguistics, vol. 1, pp. 89-94, 1996.

(Chen & Gay, 2002) Chen, A., and Gey, F. Building an Arabic stemmer for information retrieval. In TREC 2002. Gaithersburg: NIST, pp 631-639, 2002.

(Darwish, 2001) Darwish, K., Doermann, D., Jones, R., Oard, D., and Rautiainen, M. TREC-10 experiments at Maryland: CLIR and video. In TREC 2001. Gaithersburg: NIST, 2001.

(Darwish, 2002a) Darwish, K. Building a shallow morphological analyzer in one day. ACL 2002 Workshop on Computational Approaches to Semitic languages, July 11, 2002.

(Darwish, 2002b) Darwish, K. and Oard, D.W. CLIR Experiments at Maryland for TREC-2002: Evidence combination for Arabic-English retrieval. In TREC 2002. Gaithersburg: NIST, pp 703-710, 2002.

(Darwish, Hassan & Emam, 2005) Darwish K., Hassan H., and Emam O., Examining the Effect of Improved Context Sensitive Morphology on Arabic Information Retrieval. Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, pages 25–30, Ann Arbor, June 2005.

(El-Beltagy & Rafea, 2009) El-Beltagy S., Rafea A.. A FRAMEWORK FOR THE RAPID DEVELOPMENT OF LIST BASED DOMAIN SPECIFIC ARABIC STEMMERS, Proceedings of the Second International Conference on Arabic Language Resources and Tools, 2009.

(Gay & Oard, 2002) Gey, F. C. and Oard, D. W. The TREC-2001 cross-language information retrieval track: Searching Arabic using English, French, or Arabic queries. In TREC 2001. Gaithersburg: NIST, 2002.

(Hull, 1993) Hull, D. Using Statistical Testing in the Evaluation of Retrieval Performance. In Proceedings of the 16th ACM/SIGIR Conference, pages 329-338, 1993.

(Khoja, 1999) Khoja, S. and Garside, R. Stemming Arabic text. Computing Department, Lancaster University, Lancaster, 1999.

(Khoja, 2001) Khoja S., APT: Arabic Part-of-speech Tagger, Proceedings of the Student Workshop at the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL2001), Carnegie Mellon University, Pittsburgh, Pennsylvania. June 2001. http://www.comp.lancs.ac.uk/computing/users/khoja/NAACL.pdf

(Kadri & Nie, 2006) Kadri, Y., and Nie, J. Y. (2006), \E_ective stemming for Arabic information retrieval". The challenge of Arabic for NLP/MT Conference, The British Computer Society. London, UK.

(Larkey, 2002) Larkey, Leah S., Ballesteros, Lisa, and Connell, Margaret. (2002) Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis In Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2002), Tampere, Finland, August 11-15, 2002, pp. 275-282.

(Larkey, 2005) Larkey, S. L., Ballesteros, L., and Connell, E. M. (2005), Light stemming for Arabic information retrieval. Arabic Computational Morphology: Knowledge-based and Empirical Methods.

(LDC, 2001) LDC, Linguistic Data Consortium. LDC2001T55, 2001. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2001T55

(LDC, 2002) LDC, Linguistic Data Consortium. Buckwalter Morphological Analyzer Version 1.0, LDC2002L49, 2002. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002L49

(Nwesri, 2005) Nwesri A., S.M.M Tahaghoghi, Falk Scholer, Stemming Arabic Conjunctions and Prepositions, In Mariano Consens and Gonzalo Navarro (eds.), *Lecture Notes in Computer Science - Proceedings of the Twelfth International Symposium on String Processing and Information Retrieval (SPIRE'2005)*, Buenos Aires, Argentina, 3772:206-217, November 2-4,2005.

(Nwesri, 2007) Nwesri A., S.M.M. Tahaghoghi and Falk Scholer, Arabic Text Processing for Indexing and Retrieval, Proceedings of the International Colloquium on Arabic Language Processing, Rabat, Moroc, 18-19 June, 2007.

(proofing, 2003) Microsoft Office 2003 proofing toolkit,http://www.microsoft.com/middleeast/arabicdev/office/office2003/proofing.asp.

(Taghva, 2005) Taghva, K., Elkoury, R., and Coombs, J. Arabic Stemming without a root dictionary. 2005. www.isri.unlv.edu/publications/isripub/Taghva2005b.pdf

(Wonnacott, 1990) Wonnacott, R., Wonnacott, T. Introductory Statistics, John Wiley & Sons, Fourth Edition,1990.