COMPARATIVE ANALYSIS OF ARABIC STEMMING ALGORITHMS

Dr. Mohammed A. Otair

Department of Computer Information Systems, Amman Arab University, Jordan
Otair@aau.edu.jo

ABSTRACT

In the context of Information Retrieval, Arabic stemming algorithms have become a most research area of information retrieval. Many researchers have developed algorithms to solve the problem of stemming. Each researcher proposed his own methodology and measurements to test the performance and compute the accuracy of his algorithm. Thus, nobody can make accurate comparisons between these algorithms. Many generic conflation techniques and stemming algorithms are theoretically analyzed in this paper. Then, the main Arabic language characteristics that are necessary to be mentioned before discussing Arabic stemmers are summarized. The evaluation of the algorithms in this paper shows that Arabic stemming algorithm is still one of the most information retrieval challenges. This paper aims to compare the most of the commonly used light stemmers in terms of affixes lists, algorithms, main ideas, and information retrieval performance. The results show that the light10 stemmer outperformed the other stemmers. Finally, recommendations for future research regarding the development of a standard Arabic stemmer were presented.

KEYWORDS

Information Retrieval, Arabic stemmers, Morphological analysis, and Computational Linguistics.

1. INTRODUCTION

Information Retrieval is ultimately an issue of determining which documents in a corpus should be retrieved to satisfy a user's information need which is represented by a query, and contains search term(s), in addition to some information such as the relatively importance. Thus, the document will be retrieved if the similarity between the query terms with the index terms appearing in the document is high. The decision of the retrieval process may be taking any shape of the following result: binary: relevant, non-relevant, or partial. In the last case, it may involve rating the degree of important relevancy that the document has to the submitted query. [15]. Unfortunately, the words that appear in the documents and in queries at the same time often have many morphological variations (Morphology means the internal structure of words). For instance, some terms such as "extracting" and "extraction" is not evaluated as similar or tantamount, except it is processed by some similarity calculation steps. Specific Techniques such as conflation are needed to compare word variations that having the identical semantic meanings [15].

Arabic stemming is a technique that aims to find the stem or lexical root for words in Arabic natural language, by eliminating affixes stuck to its root, because an Arabic word can have a more complicated form than any other language with those affixes. Morphological variants of words accept agnate semantic interpretations and can be advised as agnate for the purpose of advice retrieval systems. Hence, a advanced ambit amount stemming Algorithms or stemmers accept been developed to abate a chat to its axis or root. Many researches were conducted to compare

DOI : 10.5121/ijmit.2013.5201

these algorithms. A study by Sawalha [23] gives a comparison between three stemming algorithms: Khoja's stemmer [17], Buckwalter's morphological analyzer [7] and the Tri-literal algorithm which uses root extraction technique [5]. Other stemming algorithm such as Khoja obtains the accomplished accurate-ness followed by the tri-literal basis abstraction algorithm, and assuredly the Buckwalter morphological analyzer. Another study by Darwish [10] found that light stemming is one of the most superior in morphological analysis. Based on morphological analysis a similar study were done by Larkey [20] to compare some stemming algorithms with the Light Stemmer, the study used many criteria such as stems and roots. Their results showed that the light stemmer passed the other algorithms in terms of performance (precision and recall) [12].

As a summary, stemming algorithms in Arabic language is categorized based on the eligible analysis degree: stem-based approach [20] and root-based approach [17]. Root-Based approach uses morphological analysis to find the Arabic word root. Many algorithms have been proposed for this approach [2, 6, 14]. The aim of the Stem-Based approach is to eliminate the most frequent prefixes and suffixes [3, 8, 19, 20]. A lot of all those trials in this acreage were a set of rules to abbreviate the set of suffixes and prefixes, as well there is no audible account of these strippable affixes [24].

This paper is conducted to do a comparative analysis 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 that can remove, and as well as the affixes. The stemmers also compared in terms of their information retrieval performance; precision and recall. A lot of accepted and acknowledged address acclimated for bearing stems of words is the ablaze stemming techniques. 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.
- Information retrieval performance; precision and recall.
- Limitation of the stemmer

2. LITERATURE REVIEW

Literature search stated the presence of numerous Arabic stemmer algorithms. The strengths of those stemmers are varied and stemming errors are produced for every stemmer. None of the analyzed stemmers showed perfect performance, and none of them has been adopted as a standard Arabic stemmer that fulfills the user's information needs.

Arabic stemming approaches have been analyzed, weaknesses, and strengths have been pointed out. Among the studied stemmers, there have been aggressive stemmers and weak stemmers.
The Larkey [21] studied several light stemmers 10, 8, 3, 2, and 1 to prove their performance efficiency as compared based on raw and normalization. Light 10 proven its superiority over the other light stemmers in this study. In another previous study [20] Larkey tried to test the Khoja-u, Khoja, normalized, and light stemmers:1, 2, 3, and 8 without involve light 10 (because it was not proposed until that time), and he found that Light 8 gives the better results.

Other stemming techniques and algorithms such as Umass, Modified Umass and Alstem were experimented in [9], and Alstem was the best one of them in term of the efficiency. Generally, Hull [16] and James [4] discovered in two separated studies that the retrieval systems gives a high performance when the stemming techniques are used in compare with the systems that do not implement the stemming.

Light stemmer, root-based, surface-based were experimented in Aljlay study [3]. He concluded that an invalid conflation classes will be created when the root-based technique is used. His results showed that the light stemmer technique performs better than the other two techniques. Mayfield et al. [20] proposed a new retrieval system by merging 6-gram with surface-based which achieved outstandingly results for the Arabic language.

Which stemmer is the best? Which one should we use? And what is the standard Arabic stemmer to adopt? Many questions need to be answered regarding Arabic stemmers. Therefore, the stemming theory and special conflation mechanisms amplifications, advantages, performance, errors, and strength in the following sections will be discussed.

3. CONFLATION TECHNIQUES

Conflation is an accepted appellation for all processes of amalgamation calm none identical words which accredit to the aforementioned concept [11]. In this context, extension of morphological query along with the morphological conflation can be defined as conflation, where the set of characters will be resembled by the modalities of synonymous words. Conflation algorithms can be disconnected into two capital classes as Leah S. Larkey [21] states:

1. Affix removal algorithms (aka. stemming algorithms): where the variations of the morphological word are handled. The main feature of these designed algorithms is their language dependability.
2. Statistical techniques: It is constructed to handling all classes of the word variations including: string relevancy, n-grams, co-occurrence, and morphological analyses. Unlike stemming algorithm it is mainly language independent.

The domain of morphology can be classified into two subclasses, derivational and inflectional [18]. Derivational morphology could or could not impact meaning of a word. Morphological is relatively weak in English language in compare with else languages like Arabic language where the morphology is very strong, complex, and sophisticated (for example, a lot number of variants maybe given for a word word). In analyze with the changes of Inflectional assay which describes accepted changes a chat undergoes as an after effect of syntax (the plural and control anatomy for nouns, and the accomplished close and accelerating anatomy for verbs are a lot of accepted in English) [18]. There is no effect on a word's 'part-of-speech' for these changes (a noun still remains a noun after pluralisation) [3].

4. STEMMING

Arabic language needs robust stemming techniques in order to process its complex morphological. Thus, the definition of stemming and related issues is required to create the necessary basic knowledge before proceeding further with Arabic stemmers.

Stemming is mainly affected by means of suffix lists which contain the bearable terminations of the words, and this mechanism could be successfully implemented on several languages. However, it is not widely implemented on the languages with complex morphological like Arabic, because it needs further analysis efforts on its morphological. In these cases, an absolutely morphological technique is needed to be implemented to eliminate the words' suffixes based on the internal structure of the word.

Thus, conflation is depends on process called inflexion which perform an inverse operation according to main rules of inflexion [15]. Conflating English words faces several problems due to because there are complex verbs that do not behave or follow normally with pattern group of inflexion [15], e.g. wake, woke, woken, and irregular verbs such as go and its forms. The use of a lexicon (dictionary) is a must to avoid errors.

A. Stemming Advantages and Stemming Errors

Stemming simplifies the searchers' job by making the IR system satisfy their information need. In increase in recall is gained. However, precision could be enhanced by conflation where the basic form of the word is generated by dictionary searching. By stemming the number of the index terms are minimized and thus reduce the inverted file size, as well. As a consequence for minimizing the index terms size, the processing time and storage space are minimized.
By the use of stemmers, Words in the collection must be organized into groups, multiple errors are produced and may be used to compare and evaluate stemmers.

- If the two words accord to the aforementioned semantic category, and are adapted to the aforementioned stem, again the conflation is correct. If they are adapted to altered stems, this is an understemming absurdity [15] (in added words, too abundant of a appellation is removed).
- If the two words accord to altered category, and abide audible afterwards stemming, again the stemmer has proceeded correctly. If they are adapted to the aforementioned stem, this is advised as an over-stemming [15] absurdity in added words, too little of a appellation is removed).

B. Stemmer Performance and Strength measurement

When stemming algorithms are used, the effectiveness of the retrieval system is enhanced if the size of the retrieval set is taken into account as Hull (1996) noted [16]. There are many measures to evaluate stemmer effectiveness and performance such as: Recall and Precision, Direct assessment, and counting both Stemming Errors.

When the stemmer merges a few of the most highly related words together, it is called a 'weak' or 'light' stemmer. A 'strong' or 'heavy' stemmer combines a much wider variety of forms. The set of metrics that measure stemmer strength as follows [13]:

- Number of words in each class of conflation.
- Index Compression: the ad-measurements to which a accumulating of altered words is bargain (compressed) by stemming.
- The Word Change Factor: This is an artlessly the ad-measurements of the words in a sample that accept been afflicted in any way by the stemming process.
- The average of removed characters
- Hamming Distance: The Hamming Distance takes two strings of according breadth and counts the amount of agnate positions area the characters are different. If the strings are of altered lengths, we can use the Modified Hamming Distance.

That was stemming theory and stemmers' characteristics. In the next section, the generic stemming algorithms for English will be analyzed in order to check if they may fit for Arabic or not.

5. GENERIC STEMMING ALGORITHMS

Stemming algorithms are numerous; in this section, a review of the generic stemming algorithms will be summarized as stated in [15] which they were developed mainly to English language and not to Arabic. However, they are summarized here for the purpose of proving that those stemmers are not fit for Arabic and should not be considered in the final comparative analysis.

A. *Lovins Stemmer*

The Lovins Stemmer [15] is proposed in 1968 and it is not an iterative process (i.e. a single phase), and has two main features: longest-match and context sensitive. The approach is not complex enough to stem many. Lovins' aphorism account was acquired by processing and belief a chat sample. The capital botheration with this action is that it has been begin to be awful capricious and frequently fails to anatomy words from the stems, or matches the stems of like acceptation words.

The Lovins Stemmer removes a best of one suffix from a word, consistent to its attributes as individual canyon algorithm. Lovins Stemmer uses a almost abbreviate account of about 250 altered suffixes, and eliminates the longest suffix affiliated to the word, ensuring that the axis afterwards the suffix has been removed is consistently at atomic 3 characters long. List of recording conversations elucidate the reformation process of the stem terminating.

B. *Dawson Stemmer*

In 1974 Dawson proposed a novel Stemmer; it is mainly depends on the Lovins Stemmer. However, it makes the list of the suffix rules approximately to 1200 suffixes. It acquire the longest bout and individual canyon attributes of Lovins, and exchanges the recording rules, which were begin to be wildcat, application instead a constancy of the fractional analogous action as well authentic aural the Lovins Stemmer.

The agnate affair amid the Lovins and Dawson stemmers is that every catastrophe independent aural the account is associated with an amount that is acclimated as a basis to seek an account of exceptions that accomplish assertive altitude aloft the abatement of the associated ending.

The above aberration amid the Dawson and Lovins stemmers is the address acclimated to break the botheration of spelling exclusions. The Lovins stemmer employs the technique known as recoding. This action is advised as allotment of the capital algorithm and performs n amount of transformations based on the belletrist aural the stem. In comparing with the Dawson stemmer employs fractional analogous which attempts to bout stems that are according aural assertive limits.

C. *Paice/Husk Stemmer*

The Paice/Husk Stemmer was developed in the backward 1980s; the Stemmer has been implemented in Pascal, C, PERL and Java. When operating with its accepted rule-set, it is a rather 'strong' or 'heavy' stemmer. It is a simple accepted Stemmer; it removes the endings (suffixes) from a chat in a broad amount of steps.

D. *Porter Stemmer*

The Porter stemmer was first presented in 1980. The stemmer is an ambience acute suffix abatement algorithm. It is based on the abstraction that the suffixes in the English accent (approximately 1200) are mostly fabricated up of an aggregate of abate and simpler suffixes. It is a lot of broadly acclimated of all the stemmers and implementations in abounding languages are available. The stemmer is divided into a number of linear steps, five or six; a linear step Stemmer. Porter himself implemented the algorithm in Java, C and PERL. Porter developed an Improved Porter stemmer as well.

E. *Krovetz Stemmer*

In 1993, the Krovetz Stemmer [18] was developed as a 'light' stemmer. The Krovetz Stemmer finer and accurately removes inflectional suffixes in three steps:

1. The about-face of a plural to its individual anatomy (e.g. `-ies', `-es', `-s'), the about-face of accomplished to present tense (e.g. '-ed'), and the removal of '-ing'.
2. The about-face action firstly removes the suffix, and again admitting a action of analytical in a concordance for any recoding (also getting acquainted of exceptions to the accustomed recoding rules), allotment the axis to a word. The concordance lookup as well performs any transformations that are appropriate due to spelling barring and as well converts any axis produced into an absolute chat that acceptation can be grasping.
3. Due to the high accuracy of the stemmer, but weak strength, it is implemented as a type of pre-processing achieved before the master stemming algorithm (such as the Paice/Husk or Porter Stemmer). This would provide partly stemmed ascribe for the stemmer that deals with accepted situations accurately and effectively, and accordingly could abate stemming errors.

F. *Truncate (n) Stemmer*

This algorithm mainly keeps the word commencement, where these retained letters should a suitable *n* integer (for instance, from 4 to 6 letters). If the word has beneath than *n* letters, then it is unchanged. After truncation, words are compared to each other. If the retained parts are similar, they are conflated to the same group, otherwise they are not. This approach suffers from several problems such as: conflation groups depend on topic of original text and the organized word collection is time-consuming because it is constructed manually.

G. *N-grams (String Similarity)*

String-similarity approaches to conflation absorb the arrangement artful admeasurements of affinity amid an ascribe concern appellation and anniversary of the audible agreement in the database. Those database expressions which are like the query terms are displayed to the user based of the users' needs. The N-gram is considered as one of the mostly used matching technique in compare of the others [14]. A set of *n* successive letters are isolated from a word in N-gram. This technique depends on the concept that every several comparable words take an elevated amount of n-grams in common. The optimal values for n are 3 or 2 which correspond to the implement of diagrams and trigrams. For instance, the word (computer) results in the generation of n-grams as shown in table (1).

Table 1. Different Digrams and Trigrams for ‘computer’

N-grams	Grams	Size
Digrams	*C, CO, OM, MP, PU, UT, TE, ER, R*	N+1
Trigrams	**C, *CO, COM, OMP, MPU, PUT, UTE, TER, ER*, R**	N+2

Where ‘*’ denotes a padding space.

We approved to administer anniversary of the antecedent stemmers to Arabic but abominably none of them seems to be acceptable candidate. Hence, it is required to elaborate further on the Arabic language characteristics in order to understand and analyze the Arabic stemmers.

6. CHARACTERISTICS OF ARABIC LANGUAGE

Arabic accent is announced by over 300 actor people; as compared to English language, Arabic characteristics are assorted in abounding aspects. In [22] Nizar summarized most of the Arabic language characteristics.

A. *Arabic alphabets or script*

Arabic is accounting from right to left, consists of 28 letters and can be continued to 90 by added shapes, marks, and vowels [3]. Arabic alphabets and script diverge significantly if they are compared with other languages in the following areas:

(1) Numerals, Style (font), Tatweel (Kashida), diacritics, marks, and shapes
(2) Distinctive letters (ش س ث ت ب), and none distinctive letters ((ؤ الإأآىء)

B. *Arabic Phonology and spelling*

(1) 28 Consonants, 3 long vowels (ي و ا), 3 short vowels(ِ ُ َ), and 2 diphthongs (إدغام حرفا علة متصلان معا)
(2) Encoding could be in Unicode and CP1256 at the same time.

C. *Morphology*

(1) Consists from bare root verb form that is triliteral, quadriliteral, or pentaliteral.
(2) Pattern and Root is equivalent to Lexeme which is called Derivational Morphology
(3) While Features and Lexeme is equal to word and it is called Inflectional morphology
(4) Noun specific: (conjunction, preposition, article, possession, plural, noun)

- □ Number: collective, plural, dual, singular.
- □ Gender: feminine, masculine, Neutral.
- □ Case: nominative, genitive, accusative.
- □ Definiteness: indefinite, definite.
- □ Possessive clitic.

(5) Verb specific: (conjunction, tense, verb, subject, object)

- □ Aspect: imperfective, perfective, imperative.
- □ Tense: future, present, past.
- □ Voice: passive, active.

 Subject: gender, number, person.
 Mood: subjunctive, indicative, jussive.
 Object clitic.
(6) Others: single letter prepositions and conjunctions

D. *Morphological ambiguity*

Derivational (قاعدة) base or rule?, Inflectional (تكتب) it could take two meanings she writes or you write?, because the ambiguity of the spelling due to misspelling or missing diacritics (ي ة، ا،), and Combined ambiguity.

For the purpose of advice retrieval, this affluence of lexical vocabularies or variability, forms and orthographic variable spelling would increase the mismatch possibility between forms in documents and the word form in a query that are similar for the query. Stemming is a tool that is used to combat this vocabulary mismatch problem.

7. ARABIC STEMMERS

The variation between morphological world's languages properties is high, and stemmers are language dependent as stated earlier. Hence, it is expected to see distinct stemmers for the Arabic language that are different form the English ones. Section (4) describes the criteria that make Arabic language is so complex to stem. Abu-Salem [1] stated that in Arabic language, the index terms of the roots or stems are helpful. He concluded that Arabic could be considered as a root based language and the benefit of its index terms is better than English language.

The affair of whether roots or stems are the adapted akin of assay for IR has been one aggravation that has accustomed acceleration to added approaches to stemming for Arabic accent besides Affix abatement and Statistical Stemming approaches as declared in section(3). Added approaches include manual dictionary construction, morphological analysis, and new statistical methods involving alongside corpora [3].

However, this paper concentrates on the analysis of Arabic stemming algorithms only. In this section, most of the major stemming techniques in Arabic are analyzed and compared.

A. Affix Removal

(1) *Normalization* which functions as follows [20]:

- Converting to windows Arabic encoding (CP1256)
- Remove punctuations
- Remove diacritics
- Remove none letters
- Replace أ،إ،آ with ا
- Replace ى with ي
- Replace ة with ه

The affix removal process is mainly achieved before the stemming process as one of the pre-processing steps. However, there are many conflicts in the literature if this process is necessary or not. The author of this paper substantially confirms on necessity to implement of some of those steps. For example, inflectional ambiguity could be produced by diacritics removal. Generally, recall and precision will be decreased.

(2) *Surface-based stemmers* that comprise from at least two morphemes as stated in [3]:

- Conveying semantics by the three consonantal root
- Syntactic information could be carried by a word pattern الوزن

Conflation is based on the surface words that exist in the user query and the corpora documents.

(3) *Root-based stemmers*: the main goal of this type of stemmers is to separate the root of a specific surface word. The prefixes and suffixes are removed and they are followed by the extraction of root. The residual stem is then compared with the similar patterns and length to extirpate the root as depicted in table (2) by projecting the matching related letters [3]. Weaknesses of root-based stemmers are:

- Increases word ambiguity
- All possible patterns are not involved
- Conflation of Irregular triliteral verbs
- Conflation Double triliteral verbs

(4) *Algorithmic Light Stemmers* which eliminate a few number of suffixes and prefixes without dealing with recognize patterns or infixes, and find roots that listed in [3] and [20]. Their drawbacks are as follows:

- No absolute abundant lists of strippable prefixes and/ or suffixes or algorithm had been published.
- Adjective mainly does not provide conflated especially with its singular form
- It fails to conflate *broken plurals* for nouns
- Conflate is not given by with the present forms of past tense

Many versions exist for the light stemmer approach follow the same following steps:

- Remove و, remove the definite article as mentioned in Appendix A, and remove suffixes that found in Appendix B.
- According to the versions of the light stemmers shown in Appendixes A and B, **Alstem** is the best light stemmer, while the weakest is **Light 1**.

(5) *Simple Stemmers* are considered as types of Light stemmers using them the infixed vowels ا،و،ي،ء are removed from variant patterns as concluded in [20]. Table (3) depicted several versions of exist simple stemmers.

The algorithms of affix removal can be categorized from strong to weak stemmers. Appendix C shows the analysis of most of the existing stemmers based on: mean average recall/precision, and several other attributes. According to the appendices A, B, and C, it can be concluded that it is not an easy task to make a fair comparison between those stemmers except they are experimented under the same circumstances such as the corpus.

B. Manually Constructed Dictionaries

They are manually congenital dictionaries of roots and stems were made for each word to be indexed.

C. Morphological Analyzers (Stemmers)

For each word, the morphological analyzers find the root or any potential root automatically by a software program.

D. Statistical Methods involving Parallel Corpora

Statistical stemmers, which accumulation chat variants application absorption techniques and Co-occurrence assay methods that are activated to automated morphological assay software systems. However, those techniques cannot be predictable to achieve correctly on the Arabic language due its strong morphology.

Table 2. Extracting root from stem by comparing patterns

	كتابة				
Canonical Pattern	ة	ل	ا	ع	ف
Surface word	ة	ب	ا	ت	ك
Root	كتب				

Table 3. Versions of Simple Stemmers

versions	**How it works**
Simple	Vowels are removed from normalized words
Simple 2	Vowels are removed after light 2 is applied
Simple 8	Vowels are removed after light 8 is applied

8. CONCLUSIONS AND RECOMMENDATIONS

In this work, the definitions concerning stemming approaches, analysis of the generic stemmers, the Arabic morphological structure, and most of the published Arabic stemmers are discussed and presented. The previous works done by many authors proved greatly that their works were independent, and no one of them biased towards to the stemmer of standard Arabic. Most of the stemming algorithms were proposed without enough or clear details lists for prefixes and suffixes.

Every columnist called an accumulation of stemmers for appraisal purposes, and the after-effects of the appraisal are referred to that accumulation only. Diverse evaluation samples were used and many stemmers were developed. This can be easily seen from the previous work mentioned in section (2).

As a result of this study, the author suggests the followings:

A. The researches should be constructed to study the important proposed Arabic stemmers to determine the significant features of every stemmer on a standard Arabic.
B. Diacritics in standard Arabic stemmer must be taken into account, because they have significant affecting on the semantics. Consequently, the stemming errors and ambiguities will be reduced.
C. The future Arabic stemmer has to be very intelligent in order to deal with all kinds of word variants. This will probably use what the authors of this work call, a Hybrid Intelligent Arabic Stemmer (HIAS).
D. All proposed stemmers must be experimented against standard selected collection for each language especially the Arabic language. The author proposes to consider the Holy Quran as a standard collection.

REFERENCES

[1] Abu-Salem H., Al-Omari M. & Evens M., "Stemming methodologies over individual query words for an Arabic IR system". Journal of the American Society for Information Science, 50: 524 – 529, 1999.

[2] Al-Fedaghi S. and Al-Anzi F., “A new algorithm to generate Arabic root-pattern forms”. In proceedings of the 11th national Computer Conference and Exhibition. PP 391-400. March 1989.

[3] Aljlayl M. and Frieder, O., "On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach". Proceedings of the eleventh international conference on Information and knowledge management, 340-347, 2002.

[4] Allan J. and Kumaran G., "Details on stemming in the language modelling framework". In UMass Amherst CIIR Tech. Report, IR-289, 2003.

[5] Al-Shalabi R., Kanaan G., & Al-Serhan H., "New approach for extracting Arabic roots". In proceedings of The International Arab Conference on Information Technology, 2003.

[6] Al-Shalabi R. and M. Evens. “A computational morphology system for Arabic”. In Workshop on Computational Approaches to Semitic Languages, COLING-ACL98. 1998.

[7] Buckwalter T., "Buckwalter Arabic Morphological Analyzer Version 2.0". Linguistic Data Consortium (LDC) catalogue number LDC2004L02, ISBN 1-58563-324-0, 2004.

[8] Chen A. and F. Gey. “Building an Arabic Stemmer for Information Retrieval”. In Proceedings of the 11th Text Retrieval Conference (TREC 2002), National Institute of Standards and Technology, 2002.

[9] Darwish K. , Douglas W. Oard, "CLIR Experiments at Maryland for TREC 2002: Evidence Combination for Arabic-English Retrieval". In TREC, 2002.

[10] Darwish K. and Oard D., “Term Selection for Searching Printed Arabic”, in Proceedings of the 25th ACM SIGIR Conference, pp. 261–268, 2002.

[11] Farag A., Andreas N., "N-Grams Conflation Approach for Arabic Text", SIGIR'07 iNEWS07 workshop, 2007.

[12] Fouzi H., Aboubekeur H., and Abdulmalik S., "Comparative study of topic segmentation Algorithms based on lexical cohesion: Experimental results on Arabic language", The Arabian Journal for Science and Engineering, Volume 35, Number 2C, 2010.

[13] Frakes W., Fox C. “Strength and similarity of affix removal stemming algorithms”. ACM SIGIR Forum, Volume 37, No. 1., 26-30, 2003.

[14] Freund, G. and Willett P., "Online Identification of word variants and arbitrary truncation searching using a string similarity measure". Information Technology: Research and Development 1: 177-187, 1982.

[15] Hoopr R. & Paice C., "The Lancaster Stemming Algorithm", 2005. http://www.comp.lancs.ac.uk/computing/research/stemming/index.htm.

[16] Hull D., "Stemming algorithms: a case study for detailed evaluation", Journal of the American Society for Information Science, 47(1), 70-84, 1996.

[17] Khoja S. and Garside R., "Stemming Arabic text". Technical report, Computing Department, Lancaster University, Lancaster, 1999.

[18] Krovetz R., "Viewing Morphology as an Inference Process", ACM Press, p.p. 191202, 1993.

[19] Larkey L., and M. E. Connell. “Arabic information retrieval at UMass in TREC-10”. Proceedings of TREC 2001, Gaithersburg: NIST. 2001.

[20] Larkey S., Ballesteros L., Margaret E. Connell, “Improving Stemming for Arabic Information Retrieval: Light Stemming and Occurrence Analysis”, in Proc. of the 25th ACM International Conference on Research and Development in Information Retrieval (SIGIR’02), Tampere, Finland, pp.275–282, 2002.

[21] Larkey S., Ballesteros L., Margaret E. Connell, "Light Stemming for Arabic Information Retrieval, Arabic Computational Morphology Text", Speech and Language Technology Volume 38, 2007, pp 221-243.

[22] Nizar H., "Introduction to Arabic Natural Language Processing", ACL’05 Tutorial University of Michigan - 2005.

[23] Sawalha M. and Atwell E., “Comparative Evaluation of Arabic Language Morphological Analyzers and Stemmers”, in Proceedings of COLING-ACL, 2008.

[24] Syiam M., Fayed Z. & Habib M., "An Intelligent System for Arabic Text Categorization", IJICIS, Vol.6, No. 1, 2006.

Stemmer	List of Prefixes (27)																											count
	ال	وال	كال	بال	فال	و	بت	يت	لت	مت	وت	ست	نت	بم	لم	وم	كم	فم	ولل	لل	وي	لي	في	وا	فا	لا	با	
Light 1	X	X	X	X	X																							5
Light 2	X	X	X	X	X	X																						6
Light 3	X	X	X	X	X	X																						6
Light 8	X	X	X	X	X	X																						6
Light 10	X	X	X	X	X	X														X								7
Alstem	X	X		X	X		X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	24
Umass	X	X	X	X	X	X																						6
Khoja	Information is not available																											
Khoja-u																												
Mofified Umass	X	X	X	X	X	X													X	X								8

- Appendix A –

Versions of light stemmers with prefixes

Stemmer	List of suffixes (24)																								count
	ه	ة	ها	ـان	ـات	ون	ين	يه	ية	ي	وا	وه	تي	ته	تم	كم	هم	هن	تك	نا	ـه	ـة	ا	ـي	
Light 1	None																								0
Light 2	None																								0
Light 3	X	X																							2
Light 8	X	X	X	X	X	X	X	X	X	X															10
Light 10	X	X	X	X	X	X	X	X	X	X															10
Alstem			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		21
Umass			X	X	X	X	X	X	X												X	X		X	10
Khoja	information is not available																								
Khoja-u																									
Mofified Umass			X	X	X	X	X	X	X												X	X		X	10

- Appendix B -

Versions of light stemmers with suffixes

Stemming Algorithm	Strength	Mean Average Precesion	number of Prefixes	number of Suffixes	Total number of Affixes
Light 1	Weakest	0.273	5	0	5
Light 2	Weak	0.291	6	0	6
Light 3	Weak	0.317	6	2	8
Light 8	Average	0.39	6	10	16
Light 10	Average	0.413	7	10	17
Alstem	Strongest	0.316	24	21	45
Umass	Average	0.321	6	10	16
Khoja (Root-based)		0.313			
Khoja-u (Root-based)		0.341			
Mofified Umass	Average	0.331	8	10	18

- Appendix C -

Comparison of Arabic Light Stemmers

Author

Mohammed A. Otair is an Associate Professor in Computer Information Systems, at Amman Arab University-Jordan. He received his B.Sc. in Computer Science from IU-Jordan and his M.Sc. and Ph.D in 2000, 2004, respectively, from the Department of Computer Information Systems-Arab Academy. His major interests are Mobile Computing, Databases, ANN. He has more than 30 publications.