A review of recent developments in term conflation approaches for Arabic text information retrieval

Chad Morris
Drexel University
cmorris@ece.drexel.edu

ABSTRACT
Conflation of related terms during indexing and query processing is an important component of effective modern information retrieval (IR) systems. This type of processing can be especially difficult, though, in IR systems that deal with morphologically complex languages such as Arabic. Consequently, many language-specific techniques and methodologies have been developed and presented over the years. This paper is concerned with exploring recent developments in this area, most notably those developments made in the field since the year 2000, the most popular of which are stemming and statistical conflation.

Categories and Subject Descriptors
H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *linguistic processing.*

General Terms
Algorithms, Standardization, Theory.

Keywords
Arabic, stemming, light stemming, lemmatization, gram characterization, n-gram, s-gram, normalization, linguistic preprocessing, machine translation, information retrieval, cross language information retrieval, IR, CLIR, string matching, term conflation, feature reduction.

1. INTRODUCTION
In a world where the proliferation of information is staggering and more and more of this information is increasingly available in a digital format, the creation of effective systems for locating and accessing this information is crucial. Consequently, the field of research known as information retrieval has been increasingly popular in recent years

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Conference'10, Month 1–2, 2010, City, State, Country.
Copyright 2010 ACM 1-58113-000-0/00/0010…$10.00.

Information retrieval is a highly interdisciplinary field, encompassing many interrelated areas of research typically found in a myriad of standard disciplines such as library science, computer science, mathematics, linguistics and psychology. At the broadest level, the field of information retrieval is concerned with building systems that effectively retrieve information - usually in the form of one or more digital objects - from a collection based on the information need of a user.

In the most general of terms, a typical information retrieval system model begins with a user inputting a query representing his information need into a system. This system then searches for objects in its collection that may contain relevant information and presents the results of its matching process to the user. A breakdown of all of the steps that are, or even can be, part of this process is beyond the scope of this short paper; instead, this paper will focus on a particular piece of this process.

Specifically, this paper is concerned with an area of research that is contingent but crucial to textual information retrieval: term conflation. This area of research is focused on linguistic manipulation of textual information for the purposes of reducing word form variation in both document collections and queries. The main aim of term conflation is to increase the overall effectiveness of a system's matching methodology by equating terms that are conceptually related but morphologically not identical. Term conflation can encompass several types of linguistic manipulation such as stemming, lemmatization, statistical string matching and root extraction and is an area of research that is also applicable to other fields such as natural language processing, speech recognition, text mining and automated document classification.

While linguistic processing, of which term conflation is a part, is an important component in all textual information retrieval systems, there are two categories of languages that have historically presented many challenges for researchers and that have been the focus of much research over recent years. The first is languages that don't have many professionally prepared linguistic resources such as word lists, dictionaries and thesauri. The second is morphologically complex languages. It is the Arabic language, one such morphologically complex language with few formal digital linguistic resources that will be the focus of this paper.

The remainder of this paper will be organized as follows. Section 2 will present a short introduction to the Arabic language, with particular attention to those features that have the biggest impact on term conflation. Section 3 will present the concepts behind term conflation. In addition, recent approaches to term conflation of Arabic text will be discussed. Finally, concluding thoughts will be presented in Section 4.

2. ARABIC LANGUAGE

Spoken as a first language by an estimated 200 to 300 million people worldwide, the Arabic language is currently in the top ten most-spoken languages on Earth. Arabic is the official language of 26 countries, the majority of which are in the Middle East and northern Africa, close on the heels of worldwide leaders English and French. In addition, Arabic is the liturgical language of Islam, a religion estimated to have the second largest number of adherents worldwide [3, 4, 10].

Arabic belongs to a group of languages known by linguists as Semitic, a group that also includes languages such as Amharic and Hebrew, and is by far the most widely spoken language of this particular language group. Overall, Arabic is a modern language with deep historical roots; and, although its popularity makes it the subject of much research in the field of information retrieval, its complexity makes it a very difficult language with which to work. Some of the linguistic characteristics, which make Arabic a challenging language for information retrieval, are presented below.

2.1 Morphology

Textual Arabic material is written in the Arabic alphabet, which consists of 28 consonants, three short vowels, three long vowels, two diphthongs and a few consonants used to represent sounds in words borrowed or transliterated from other languages. The Arabic language is a root-based language, with most of the Arabic lexicon being derived from a set of roughly 10,000 consonantal roots. The exceptions are some common nouns and particles that could consist of as few as a single character.

Arabic roots can be triliteral, quadriliteral, pentaliteral, hexaliteral or heptaliteral; however, the most common root form is triliteral, with 85% of Arabic words derived from triliteral roots [Ahmed]. These roots are heavily affixed through derivation, inflection and cliticization in order to form the words found in speech and writing. A discussion of these morphological features follows. For the purposes of explanation, the Arabic root *ktb* (ب ك ت) - meaning "write" - is used as an example.[1]

2.1.1 Derivation

In linguistics, the term derivation refers to the process of forming words from other words or bases through the addition of affixes or through transformation. As stated earlier, most words in Arabic are derived from consonantal bases, or roots. The pattern of derivation for Arabic verbs and nouns consists mostly of infixing different combinations of vowels. For example, the verb *katab* meaning "to write" and the nouns *kateb* meaning "author" and *kutub* meaning "book" can all be derived from the root *ktb*. The results of derivation will be referred to hereafter as lexemes.

2.1.2 Inflection

Lexemes are then further modified by affixes and transformations to add grammatical features; this process is known as inflection in linguistics. Nouns and adjectives in Arabic can be inflected for four different grammatical features: case, number, gender and state. Verbs, on the other hand, are marked for person, gender and number. In addition, verbs are also conjugated by tense, voice and mood.

Almost all inflection is represented by a combination of prefixes and suffixes, some of which cause transformations in the initial and terminal letters of the base words to account for pronunciation. Prefixes usually denote the tense and person of verbs; whereas, suffixes are typically used to mark case, number, gender and state. Continuing with our example, the lexeme *kateb* meaning "author" can be inflected to its plural "authors" by adding a suffix resulting in *kateban*. Similarly, the lexeme *katab* meaning "to write" can be transformed into *kataba* meaning "he wrote" and *katabat* meaning "she wrote".

These are simple examples of inflection involving only affixation; however, more complicated affixation and transformations can result in other forms such as *naktubu* meaning "we write" and *yaktob* meaning "he is writing". As will be seen in the next section, the results of inflection can be further transformed; so, hereafter, the results of inflection will be referred to as bases in order to distinguish them from lexemes, the result of derivation.

2.1.3 Cliticization

While bases can appear independently as words in Arabic text, more often, nouns, verbs and adjectives are further modified by affixation of clitics in a process known as cliticization. In linguistic terms, a clitic is a grammatically independent morpheme that phonologically attaches to another dependent morpheme. Proclitics are attached to the beginning of the base and enclitics are attached to the end of the base. In Arabic, there are four varieties of clitics that attach to the base when appearing in text:

1. The prefixed definite article *al*. (DA)
2. A class of enclitic pronouns used to demonstrate possession or to distinguish the object of a verb or preposition. (PRO)
3. A class of proclitic particles. (PAR)
4. A class of proclitic conjunctions that include "and" and "then". (CON)

These clitics attach to bases in a particular order, as demonstrated by: (CON + (PAR + (DA + base + PRO))) [7].

2.2 Variants

In addition to a rich morphology, the modern Arabic language encompasses three main categories of variants: Classical Arabic, Modern Standard Arabic and dialectical Arabic. Classical Arabic, while not in phonological use outside of a religious context in the modern world, is used in historical literary and religious texts. Classical Arabic is also the language used in the Qur'an, the holy book of Islam.

[1] Note that Arabic text is typically written right-to-left. The example given here has the Arabic letters separated and presented in a left-to-right orientation. Transliterated Arabic words will be presented in italics for the remainder of this paper.

Modern Standard Arabic (MSA), also known as Literary Arabic, is an outgrowth of Classical Arabic, retaining many of its morphological and syntactic features. MSA is the predominant variant in the Arabic-speaking world, taught in schools and universities and used for formal spoken public media. Most importantly, it is the variant used in most modern Arabic publications from news content to websites to scholarly articles and government documents.

Finally, dialectical Arabic is the colloquial form of the language, localized to a given region or nation and forming the basis of the everyday spoken language of most Arabic speakers. These variants can be so widely varied as to be mutually unintelligible and are often the media of local spoken entertainment; however, they rarely appear in print.

2.3 Orthography

The standard orthography of Arabic also poses its own problems for term conflation and other text processing tasks. Firstly, the letter shapes exhibit transformation based on placement in the written word. In addition, traditional letter doubling is often applied inconsistently in modern written Arabic. More problematic is that written Arabic – like many Semitic languages - is mostly consonantal, meaning that vowels are omitted. As can be seen from the earlier discussion of Arabic morphology, vowels are often what distinguish one word from another. While human Arabic readers can decipher the meaning of a term by context, the standard practice of omitting vowels in Arabic text causes ambiguity that is difficult for computers to handle. The one major exception to the practice of vowel omission is the text of the Qur'an, in which all diacritics must be included to avoid this inherent ambiguity.

2.4 Influence of Foreign Languages

The long and varied history of the Arabic language as well as its place as a popular modern language means that Arabic has adopted the use of many foreign words. In information retrieval these words are known as out of vocabulary (OOV). Transliteration of these words into the Arabic alphabet is not standardized. Additionally, OOV words do not follow standard Arabic transformation rules. Thus, OOV words complicate the process of term conflation and are a common source of error in IR systems.

3. TERM CONFLATION

As stated earlier, term conflation is focused on linguistic manipulation of textual information for the purposes of reducing word form variation in both document collections and queries, with the goal of equating terms that are conceptually related but morphologically not identical. Term conflation aims to increase the overall effectiveness of an IR system's matching methodology. In general, term conflation approaches can be categorized into two broad groups: feature reduction approaches and statistical conflation approaches. Typical feature reduction methods include approaches such as stemming and lemmatization; whereas, typical statistical conflation approaches include string similarity scoring approaches such as gram characterization, edit distance and Bayesian models. Most modern approaches use some combination of feature reduction and statistical conflation approaches.

As can be seen from the discussion in the previous section, term conflation in systems that deal with Arabic text can be complicated and difficult. Because many of the traditional approaches developed for Indo-European and Romance languages have been largely unsuccessful in adapting directly to use in Arabic settings, there has been a proliferation of new approaches proposed for Arabic term conflation over the last few decades. Over the past ten years, research in this area has focused on two main approaches: stemming and statistical conflation. The remainder of this section will be devoted to discussing these two main categories and the associated research that has been proposed over this same time period.

3.1 Stemming

The term stemming is used to refer to conflation approaches that attempt to reduce words as they appear in the text to some common stem. With this approach, it is hoped that conceptually related words will reduce to the same stem and therefore be considered equivalent during indexing and retrieval. Stems need not and typically are not valid morphological roots.

Stemming approaches can be divided into three general groups: lookup-based, rule-based and probabilistic [1]. Lookup-based approaches compare words in text against a precompiled list of possible word forms and their associated stems. If a match is found, the associated predetermined stem is returned. While this approach yields highly accurate results, it has several obvious drawbacks including the need for linguistic expertise, a labor-intensive list formation process, the inability of the system to handle words that do not appear in the precompiled list and a lot of processing power.

Rule-based approaches also require a sizeable amount of linguistic knowledge and a fair amount of labor prior to implementation. Rule-based stemmers try to apply linguistic rules as a linguist would to produce a stem. Traditionally, these rules deal with the correct removal of suffixes and, in some cases, prefixes and infixes. The general process sees words compared to one or more sets of these rules and, if they meet the criteria set forth in the rule, the associated transformation rule is applied and the resultant stem is returned. While processing time is reduced when compared to lookup-based approaches, rule-based approaches can be prone to erroneous stripping of affixes that don't actually exist, especially when dealing with morphologically complex languages such as Arabic.

Finally, probabilistic approaches, as the name suggests, utilize probability-based techniques to determine stems. In these stemmers, rules are applied to inflected forms in order to produce stems, in a similar fashion to rule-based stemmers. The difference is that these rules are part of a probabilistic model, not a strictly linguistic one. Some of these models are trained using a set of root and inflected form relations; others are developed based on language independent models such as gram characterization and similarity scoring. In addition, while most other stemming approaches produce stems without using native contextual information, some probabilistic stemmers utilize contextual analysis to help increase the accuracy of stem identification. In general, probabilistic approaches are appealing as they required the least amount of linguistic knowledge when compared to the other two types of approaches; however, some are highly dependent on the availability of accurate training materials for model development and others are limited in their scope.

Stemming has been in fairly widespread use in IR and related fields since it was first introduced in the late 1960s and, while earlier stemmers were based on one of the approaches just outlined, most modern stemming algorithms exhibit a combination of these approaches. Although research in stemming was dominated by English language models during the early days of research in the area, stemmers have more recently been developed for many modern languages, including Arabic. Most recently, research for stemming of Arabic text has been fairly popular. This popularity has resulted in the development of a large number of stemming approaches over the past ten years, most of which can be divided into two main groups – light stemming and root extraction – distinguishable by the lexical character of their resultant stems.

3.1.1 Root Extraction

The term root extraction is used in the field of Arabic language processing to refer to stemming algorithms that aim to produce lexically accurate roots from inflected words. Root extraction stemming algorithms are typically complex and multifaceted in their approach, combining several techniques. Root extraction has by far been the most researched and widely applied term conflation paradigm in Arabic language systems over the years and, as such, new algorithms are still produced today.

The now predominant stemmer in this paradigm is known as the Khoja stemmer, which was developed in the 1990s. The general outline of the algorithm is as follows:

1. Normalize the text by removing all diacritics that denote vowelization, stopwords, punctuation and numbers.
2. Remove known proclitics such as the definite article and conjunctions.
3. Remove suffixes using a rules-based approach.
4. Remove prefixes using a rules-based approach.
5. Remove infixes by comparing the result of step 5 to a predefined list of patterns. If a match is found, extract the characters from the pattern designated as root characters.
6. Match the result of step 6 against a list of lexically valid roots.
7. Perform some character transformations, if necessary.
8. Check two letter roots to see if a double character should be added [2, 13].

The first step in the Khoja algorithm is referred to as normalization. The process of normalization is common to all stemming algorithms, although each algorithm dictates a particular set of techniques and rules. During this process, the system tries to isolate the terms needing stemming by doing such things as removing punctuation, standardizing text by removing diacritics and performing some character transformations.

Along a similar line, Taghva, Elkhoury and Coombs [13] have tried to create a more effective version of the Khoja stemmer by making two significant alterations to the Khoja algorithm. The first change was to remove the check of generated roots against a dictionary of lexically valid roots. This elimination helped reduced processing time and reduced the workload of maintaining an additional linguistic resource. The second alteration was made to the way in which affixes are evaluated and removed. In particular, the removal of suffixes was relegated to the same step that handled pattern matching for infix removal. This process calls for the iterative evaluation and removal of suffixes and infixes simultaneously based on word length. Overall, performance of the algorithm was not significantly different from the Khoja stemmer, showing that similar results could be achieved with fewer linguistic resources.

The Khoja stemmer and stemmers like it are, by some accounts, more traditional in nature in that they rely heavily on formal linguistic resources that are labor-intensive to produce. More recent developments in root extraction seek to instead rely more heavily on probabilistic models that require fewer linguistic resources and less linguistic expertise.

Some, like Boudlal et al. [5], feel the most effective method to extract roots may result from combining more traditional stemming methods with probabilistic models based on contextual analysis in a multipart process. The approach they propose passes words to be stemmed through two modules. The first module identifies all possible segmentations of an inflected word based on prefix and suffix lists and further identifies all possible infixes for each stem produced by the segmentation process based on a root-pattern dictionary. The second module attempts to identify the most likely root from the list of possibilities determined in the first module by the inflected word's position in the sentence in which it appears in the original text. It does this by utilizing a hidden Markov model that is trained from a hand-annotated corpus in order to find hidden patterns in sentence-level root sequencing based on the observable state of sentence-level inflected word sequencing. The results of experiments with this algorithm showed exceptional performance.

Others, like Hmeidi et al. [8] propose combining simplified traditional methods with gram characterization matching techniques. Similar to the Khoja stemmer, the algorithm presented by Hmeidi et al. begins with normalization – including a set of novel transformation rules developed by the authors – and prefix and suffix removal; however, the affix removal performed is done so using a much reduced and simplified list of prefixes and suffixes compared to Khoja and Boudlal et al. These stems are then matched with roots in a standard root dictionary using statistical string-similarity measures based on the comparison of the set of two-character *n*-gram (bigram) decompositions of the stems to the same decomposition of the roots in the dictionary.[2] The experiments that Hmeidi et al. conducted used two separate statistical measures to match stems to roots: Dice similarity scores and the Manhattan Distance measure. In general, Dice similarity scoring seemed to work better that the Manhattan Distance measure and the algorithm proved effective at extracting not only trilateral roots – the main focus of other algorithms – but roots of all lengths.

A new line of research has emerged recently as well, in which researchers attempt to dispense with traditional linguistic resources during processing altogether, feeling that these algorithms still rely too heavily on linguistic expertise and require too much processing time. Al-Serhan and Ayesh [3] developed a

[2] See discussion of n-gram matching in Section 3.2 for a more detailed explanation of the approach.

method for extracting trilateral roots using a backpropagation neural network model. In this model, five-character inflected words were encoded into binary representations based on a set of predetermined rank values for each character. These rank values were assigned for each character on the basis of a frequency analysis of common Arabic affixes; characters that appear more frequently in Arabic affixes are given a higher rank. These inputs are fed into a neural network that has been trained using a set of 500 random lexically valid input-output pairs. Based on its training, the network returns a binary output string showing which three characters of the original five compose the root. Although limited in scope to five-character inflected forms and trilateral roots, a demonstrated high degree of accuracy shows research into similar models may be worth exploring.

Within the past ten years, however, the question has been raised by many researchers, especially in the IR field, as to whether root extraction is really the most effective stemming method for Arabic [9]. These researchers point out that a major drawback to root extraction is that it creates ambiguity in IR settings by creating too few conceptual categories. Conflation to a lexical root in a highly morphological language results in a loss of distinguishing semantic information present in the inflected forms that could be important for more accurate information retrieval.

For example, if a searcher using an Arabic-based web search system is interested in finding information about available office space in Cairo, he might include the word *maktab* meaning "office" in his search. Root extraction algorithms would ideally reduce this inflected form to its root *ktb*, the root used as an example in Section 2. As a result, it is likely that the search would return pages about authors, books and writing as well as those about offices, as the words for these concepts are all derived from the same root in Arabic. Because of this shortfall, there is a current interest among researchers in less aggressive stemming algorithms, collectively known as light stemming.

3.1.2 Light Stemming

In Arabic language processing, the term light stemming is used to refer to stemming algorithms that aim to produce stems, as opposed to the root extraction approaches that aim to produce lexically accurate roots. These stems may or may not approximate lexically accurate bases or lexemes. When compared to root extraction algorithms, light stemming approaches are less aggressive, removing a much smaller set of morphological features from the text that it processes. The main attraction of light stemming is the fact that it retains disambiguating conceptual information that is normally lost in the more traditional root extraction conflation approaches and thus increases precision in IR settings.

Larkey, Ballesteros and Connell [9] were one of the first to demonstrate this shortcoming experimentally. They developed a series of light stemmers, each removing a different combination of morphological features. They then performed a set of both monolingual and cross-language retrieval experiments to compare their light stemmers to runs using the Khoja root extraction stemmer as well as baseline runs without conflation. The results of these experiments showed that one of the light stemmers tested performed on par with the Khoja stemmer for monolingual retrieval tasks, and significantly outperformed the Khoja stemmer for cross-language tasks. This particular stemmer is known as Light8. This light stemming approach removes stopwords, the definite article, the conjunction و meaning "and" and a small set of suffixes.

Aljlayl and Frieder [4] came to a similar conclusion when testing their light stemming approach against the Khoja stemmer and baseline runs. Aljlayl and Frieder feel that the most important morphological step for word sense disambiguation is derivation. As can be seen from the earlier discussion of Arabic morphology, derivation is the morphological stage at which different conceptual categories are created from a root. For example, derivation allows for the distinction between the concept of writing and the concept of a book, both from the same root. Consequently, in order to retain this word sense disambiguation during conflation, the light stemming algorithm they propose focuses on the removal of clitics and inflectional features in an attempt to closely approximate lexemes.

3.1.3 Hybrid Methods

Although helpful, neither root extraction approaches nor light stemming approaches have proven to be perfect solutions to the problem of term conflation. Given this fact, some researchers have begun to look into hybrid stemming methods that combine these two basic approaches into a single stemming solution. What makes these new stemming approaches unique is that they stem nouns and verbs separately, allowing for modifications to more traditional stemming methods based on the unique morphological features of each of these classes.

In particular, Al-Shammari and Lin [2] have proposed a stemmer in which verbs undergo root extraction and nouns undergo light stemming. If one thinks about the morphological character of Arabic, this approach makes sense. In Arabic, roots typically express the concept of action, thus verbs derived from a particular root express the same concept of action, making root extraction an appealing approach to conflate verbs. Nouns, on the other hand, are also derived from these same roots; however, by definition, they express concepts of objects and ideas. This difference means that nouns do not take on their distinctive word sense until they are derived, and thus differentiated, from their root's concept of action.

The algorithm proposed by Al-Shammari and Lin uses sets of stop words that typically precede nouns and verbs, rules about Arabic sentence structure and a multiple pass structure informed by saved global noun and verb arrays to perform part-of-speech disambiguation for the collection undergoing stemming. During this step, each term to be processed is tagged as either a noun or verb. After undergoing a final normalization step that removes any remaining stop words, verbs are subjected to the Khoja stemmer, as outlined earlier. Nouns and any terms for which a lexical category cannot be determined undergo light stemming that involves removal of common prefixes and suffixes. This stemmer shows promising results when compared to the Khoja stemmer and will likely prove to be the catalyst for research into similar approaches to Arabic text conflation in the future.

3.2 Statistical Conflation

Stemming is not the only conflation approach for Arabic that has been the subject of research over the past ten years. Statistical conflation has also received some attention. Unlike stemming, statistical conflation attempts to group related terms based on statistical similarity measures. This type of approach to term conflation is appealing in that it typically requires little to no

linguistic expertise and is language independent, meaning that a single approach can often be applied to more than one language. This is a very appealing prospect for those researchers trying to create IR systems that deal with morphologically complex languages and languages for which very few linguistic resources exist.

Statistical conflation is a fairly recent development in the field of term conflation research. Almost all of the approaches that have been developed in this category focus on string similarity matching. The concept behind these approaches is that the character structure of words can be used to identify semantic relationships. Of the few approaches that currently exist, the pure *n*-gram approach to string similarity matching appears to be the most effective for Arabic [6, 12]. Additionally, *n*-gram based approaches have been shown to be better at handling misspelled and transliterated words than stemming algorithms.

In the pure *n*-gram approach, the two terms to be compared are decomposed into sets of sub-strings, *n* characters in length. For example, the bigram (*n*=2) decomposition of the word *kutub* would be: *ku*, *ut*, *tu*, *ub*. These sets of character substrings are then compared to one another and a similarity score is computed, typically using Dice's similarity measure:

$$sim(a,b) = \frac{|a \bigcap b|}{|a \bigcup b|}$$

Terms whose similarity scores are above a certain threshold are grouped together.

While there has been much research [1, 6, 12] regarding the optimal gram size for use with Arabic in recent years, Ahmed and Nürnberger [1] have proposed a modified version of this pure *n*-gram approach that seems to work well with Arabic. The problem, Ahmed and Nürnberger posit, is that the pure *n*-gram model does not take into account the order of the character substrings when determining matches. Because of the heavy use of prefixes and suffixes in Arabic, this type of approach is prone to conflating semantically unrelated words based on the matching of these affixes and not the more distinctive character substrings present in the lexemes. Consequently, they proposed an approach that takes into account character ordering when determining matching grams between terms. Through a series of IR experiments, they were able to show that their revised *n*-gram approach outperformed pure *n*-gram approaches in both precision and recall.

3.3 Combined Methods

Alas, neither pure stemming nor pure statistical conflation methods have been shown to be the perfect solution for Arabic IR applications, demonstrating 100% precision and recall. Accordingly, some researchers have sought to combine the two most popular term conflation methods in an effort to create a conflation approach that is more effective than either one alone. Mustafa [11] is pioneering research into these combined methods by utilizing light stemming and n-gram conflation approaches in the same algorithm.

Mustafa proposed that words in the text collection and the query be first stemmed using light stemming. Then, subsequently, the collection should be searched using bigram decompositions of these stemmed text and query terms. In his IR experiments, Mustafa compared two different levels of light stemming against each other and a run of *n*-gram matching without stemming. One set of light stemming rules attempted to approximate bases, by removing clitics such as conjunctions, pronouns and the definite article; the second set approximated lexemes by trying to remove clitics and inflectionally related affixes. In both cases, only prefixes and suffixes were removed according to affix lookup trees; infixes were not removed. Combining the second set of light stemming rules with *n*-gram matching yielded the best results of the three runs. These findings confirm that the morphological structure of the Arabic language has a significant impact on the success of string similarity matching approaches for IR, therefore validating the need for term conflation approaches that minimize its effects.

4. CONCLUSION

Ultimately, the field of term conflation for Arabic text has received a lot of attention over recent years and, given the globalized nature of today's world and the growing importance of the Arabic language, it is likely to get a lot of attention in the future. As can be seen from this short introduction to the recent developments in this area, no single approach has been proven to be the perfect solution, warranting standardization across all systems. From this investigation of the current state the field, it seems that the most promising direction of research lies in conflation methods that combine hybrid stemming techniques with statistical string matching techniques, especially *n*-gram based approaches.

5. REFERENCES

[1] Ahmed, F. and Nürnberger, A. 2009. Evaluation of n-gram conflation approaches for Arabic text retrieval. *J Am Soc Inf Sci Tec*. 60, 7 (Jul. 2009), 1448-1465. DOI=10.1002/asi.v60:7

[2] Al-Shammari, E. and Lin, J. 2008. Towards and error-free Arabic stemming. In *Proceedings of the 2nd ACM Workshop on Improving non-English Web Searching*. (iNEWS'08, Oct., 2008). Association for Computing Machinery, 9-16. DOI= 10.1145/1460027.1460030

[3] Al-Sheran, H. and Ayesh, A. 2006. A triliteral word roots extraction using neural network for Arabic. 2006. In *The 2006 International Conference on Computer Engineering and Systems*. (Nov., 2006). IEEE, 436-440. DOI=10.1109/ICCES.2006.320487

[4] Aljlayl, M. and Frieder, O. 2002. On Arabic stemming: improving retrieval effectiveness via a light stemming approach. In *Proceedings of the eleventh International Conference on Information and Knowledge Management*. (CIKM'02, Nov., 2002). Association for Computing Machinery, 340-347. DOI=10.1145/584792.584848

[5] Boudlal, A., Belahbib, R., Lakhouaja, A., Mazroui, A., Meziane, A. and Bebah, M. 2010. A Markovian approach for Arabic root extraction. *The International Arab Journal of Information Technology*. 7, 2 (Apr. 2010), 13-20.

[6] Darwish, K. and Oard, D. 2002. Term selection for searching printed Arabic. In *Proceedings of the 25th annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. (SIGIR'02, Aug., 2002). Association for Computing Machinery, 261-268.

[7] Habash, N. and Sadat, F. 2006. Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*. (Jun., 2006). Association for Computational Linguistics, 49-52.

[8] Hmeidi, I., Al-Shalabi, R., Al-Taani, A., Najadat, H. and Al-Hazaimeh, S. 2010. A novel approach to the extraction of roots from Arabic words using bigrams. *J Am Soc Inf Sci Tec*. 61, 3 (Mar. 2010), 583-591. DOI=10.1002/asi.v61:3

[9] Larkey, L., Ballesteros, L. and Connell, M. 2002. Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. In *Proceedings of the 25th annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. (SIGIR'02, Aug., 2002). Association for Computing Machinery, 275-282. DOI=10.1145/584376.564425

[10] Momani, M. and Faraj, J. 2007. A novel algorithm to extract tri-literal Arabic roots. In *IEEE/ACS International Conference on Computer Systems and Applications, 2007*. (AICCSA'07, May, 2007). IEEE, 309-315. DOI=10.1109/AICCSA.2007.370899

[11] Mustafa, S. 2005. Combining n-grams and stemming for Arabic word-based inexact matching and term conflation. *Journal of Information & Knowledge Management*. 4, 1 (Mar. 2005), 29-35.

[12] Mustafa, S. and Al-Radaideh. 2004. Using n-grams for Arabic text searching. *J Am Soc Inf Sci Tec*. 55, 11 (Sep. 2004), 1002-1007. DOI=10.1002/asi.20051

[13] Taghva, K., Elkhoury, R. and Coombs, J. 2005. Arabic stemming without a root dictionary. In *International Conference on Information Technology: Coding and Computing*. (ITCC, 2005). IEEE, 152-157, Vol. 1. DOI= 0-7695-2315-3/05