Improving Stemming Algorithm for Arabic Text Search

by

Afag Salah Aldeen Babiker

Supervised by: Dr. Mohammed Mustafa Ali

Thesis is presented for the degree of Master in computer science

In the Department of Computer Science

Sudan University Science and Technology

. August 2014.

ACKNOWLEDGMENTS

My special appreciation goes to Dr. Mohammed Mustafa, my supervisor, who was very helpful and a considerable advisor.

My thanks also go to the staff members, M.Sc. students, undergraduate students at the Sudan University of Science and Technology and University of Khartoum, for their contribution in building a test collection for this thesis for their support and emotional encouragement.

ABSTRACT

Building an effective stemmer for Arabic language has always been a hot research topic in the IR field. This is because Arabic, as one of the Semitic languages, has a very rich and complex morphology.

From that perspective, several approaches have been developed for Arabic stemming and for the analysis of the best approach to index Arabic words. Formally, Arabic stemming techniques can be also classified into two major techniques: root-based techniques (known also as heavy or morphological analysis based stemming) and light stemming-based techniques (known also as affix removal stemming. Each of two approaches has major drawbacks. On one hand, root-based stemming may result in an over-stemming problem, in which words with different meanings may erroneously, grouped together. On the other hand, light-based stemming often results in an under-stemming problem, in which words with the same meaning do not stemmed together. Nevertheless, it was concluded in IR light stemming and light-10 in particular is the best developed approach for indexing Arabic documents.

Inspired by light-10, this research attempts to improve some of the drawbacks identified in light-10 stemmer. It simply adds some additional prefixes and suffixes. These extended prefixes have been added after a deep analysis and several experiments conducted by the developer to understand the nature of the Arabic words. The step has been also accompanied by developing a new algorithm, also inspired by light-10, so as to control the process of determining which prefix and/or suffix should be stripped off.

Test results showed that the proposed Extended-10 stemmer could yield significant better results when it was compared to the best known Arabic stemmer so far, that is light-10. Results also prove to be efficient for improving Arabic IR retrieval.

المستخلص

تعتبر عملية استرجاع جذر الكلمة (Stemming) في اللغة العربية من اكثر مواضيع البحث استهدافا في مجال محركات البحث , وذلك بسبب ان اللغة العربية تعتبر واحدة من اللغات السامية التي تحتوي على عدد كبير من التصاريف للكلمة الواحدة وهذا يجعلها من اللغات الاكثر تعقيدا من هذة الناحية.

ومن هذا المنظور , نجد ان هناك عدد من المقترحات والحلول تم تقديمها من الدراسات السابقة لاسترجاع جذر الكلمة العربية (Stemming) حتي يتم تخزين الكلمة العربية بصورة صحيحة, وتم تقسيم هذه الحلول الى نوعين:

- التحليل العميق لاسترجاع جذر الكلمة (Heavy based stemming)
- التحليل الخفيف لاسترجاع جذر الكلمة (Light based Stemming)

وكل من هذه التقنيات تعاني من بعض المشاكل , فمثلا, نجد مشكلة التقنية الاولى انها فشلت في تقسيم الكلمات التي تختلف في المعنى الى مجموعات مختلفة وهذه المشكلة تعرف (Over-Stemming problem) , اما مشكلة التقنية الثانية انها فشلت في توحيد الكلمات التي تتشابه في المعنى تحت جذر كلمة واحدة وهذة المشكلة تعرف (Under-Stemming problem).

من الدراسات السابقة في هذا المجال تعتبر خوارزمية Light 10 من افضل الخوارزميات لاسترجاع جذر الكلمة العربية في محركات البحث.

يركز البحث على تحسين خوارزمية Light 10 وتحسين المشاكل التي تواجه الخوارزمية , وهذا تم عن طريق اضافة prefixes/suffixes جديدة الى قائمة الاقصاء (Elimination List) , وهذة القائمة الجديدة لم تنشئ الا بعد اجراء عدد من التجارب والتحليل العميق لفهم طبيعة اللغة العربية, وهذة الخطوة اتت مصاحبة لانشاء خوارزمية لاقصاء prefixes/suffixes من الكلمة تحت قيود معينة لاسترجاع جذر الكلمة.

اوضحت النتائج ان الخوارزمية المقترحة (Extended-10) اظهرت نتائج مؤثرة على محركات البحث مقارنة مع خوارزمية Light 10 , وايضا تعتبر النتائج ذات تاثير ايجابي على عملية استرجاع الملفات العربية في محركات البحث.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

Information retrieval (IR) is the problem of satisfying users' information needs from unstructured data (like text, sound, image, etc), often known as a collection. In the context of textual IR, the major task of an IR system is to represent, store and manage information on the unstructured collections (referred to as set of documents in textual IR case) and provide a user with topical information on his information need (also referred to as a query) through an accessing mechanism to that collection. Defined in this way, the IR system should be able to: represent documents, which are often presented in a natural language, in somewhat searchable representations; represent queries, often a few words; and find documents that match the query representation with documents' representations.

In the context of the definition above, the IR task can be decomposed into three main processes. These are: a searchable document representation over which the retrieval process is performed, a representation of a user information need and a matching process between the two representations, which results in a set of retrieved documents.

The process of representing documents is called the indexing process(Manning et al., 2008), in which keywords of documents are extracted. Such extracted keywords are known as terms. The term is the basic used unit for representing both documents and queries and it can be a word, phrase, stem N-grams, etc, depending on what is needed from representing/indexing documents.

The process of producing index terms often goes through several operations, most of which are language-dependent. Examples of such processes are *tokenization* and *stemming*. Tokenization is the process of breaking a stream of characters into expressive and semantically meaningful pieces called tokens, whereas stemming renders different inflected and variant forms of a certain token to a single word stem(term). For instance, words like “participating”, “participates”, “participation” and “participant”t may all be rendered to a common single stem “participat”.

The end product of the document representation process (indexing) is a new searchable structured description of documents in a form of a set of terms (index). A user information need is also represented in the same way so as to create a query, which searches against the created index. Thus, the matching process is usually carried out between a query (information need representation) and a set of represented documents.

In a broad sense and based on this matching, a set of documents with matched scores are often retrieved as a result for the matching process. A matched document score is used to determine the relevance matching degree of a document with regards to a query. The document is considered as relevant if it covers/addresses the user information need, rather than just containing query terms. Hence, a good IR system is expected to rank relevant documents on top and its quality is measured in terms of precision and recall. Precision can be though as the proportion of the returned documents those are relevant to the query, whereas recall is the proportion of the relevant documents in a collection that were retrieved by an IR system.

Over the last decades Arabic IR becomes one of the popular areas of research especially with the explosive growth of the language on the Web and the emergent of the Cross Lingual Information Retrieval field, which shows the need to retrieve documents in other languages. This increasing interest in Arabic, however, is merely caused by its morphology and orthography. However, in spite of the significant achievements and developments in existing Arabic text retrieval systems but, yet its support is comparatively poor and much weaker than for English(Abdelali, 2006, Alansary et al., 2007). In particular, Arabic is still lacking high quality IR tools. This is mainly caused by its complex morphology, e.g., morphological variants of Arabic words may have similar semantic interpretations although they can be considered as equivalent for the purpose of IR applications, e.g., e.g., سيعلمون, علم, يتعلمون, معلمون and معلم.

For this reason, a number of Arabic stemming Algorithms, or stemmers, have been developed, which attempt to reduce a word to its stem or root form. In other words, the key terms of a query or a document are represented by stems rather than by the original words.

The morphology complexity of Arabic makes it particularly difficult to develop stemming algorithm to cover most Arabic inflection rules while on the same time it keeps the semantic meaning of word without change. Bearing in mind this unique feature for Semitic languages, including Arabic, in general, two major approaches were proposed for stemming Arabic text. These are the light stemming, which is based on the removal of some affixes from words and the heavy stemming which is primarily based on morphological analysis(Larkey et al., 2007). However, in spite of the good performance of both the employed approaches, stemming in Arabic still lacking in high quality to reach excellent accuracy.

From that prospective, this research aims to investigate extensively the problem of stemming in Arabic language. It attempts to develop a new stemming algorithm by deriving the strongest features of the current approaches while on the same time its minimizes their current drawbacks.

1.1 Problem statement

As it was shown in the literature, two major approaches are employed for stemming Arabic texts, these are heavy-stemming and light stemming. Although, light stemming is the most dominant approach for stemming Arabic text (Larkey et al., 2007), each of the two types has some pros and cons. On one hand, heavy stemming retrieves all the related text and reduces the index size significantly. Furthermore, the approach also maintains Part-Of-Speech (POS) distinctions(Levow et al., 2005), but it may erroneously cluster some different words into a single root, known as the over-stemming problem, leading to a low precision. For instance, consider the two Arabic words سيتعلمون (meaning: they will learn) and العلمية (meaning: scientific). Both of these words will be stemmed to the same root علم using the heavy stemming approaches, although the two words are different in their semantics. Additionally, heavy stemming may erroneously stem nouns if it fails to identify them. For example, using the proper noun of the US leader “Barak” (in Arabic: باراك) will be stemmed to the word برك (meaning: to sit down 'for animals).

On the other hand, light stemming achieves the goal of retrieving the most pertinent documents, but it may not succeed to cluster semantically similar words together, known as the under-stemming problem, resulting in low recall. For example, light stemming will not group the Arabic words سيتقاتلن (meaning: they will fight each other 'for plural feminine') and مقاتلات (meaning: fighters ' for plural feminine'). But, since light stemming may fail to preserve parts of speech (Levow et al., 2005), e.g., there may be a noun and a verb that are both stemmed to a single stem, it increases the possibility of matching between stemmed documents and stemmed queries. Additionally, Paice (1994) had shown that light stemming moderates the over-stemming errors but it may result increasing the under-stemming errors, while heavy stemming reduces the under-stemming errors while it may result in increasing the over-stemming errors.

Till now, it is still not clear what the correct level of conflation should be for Arabic IR – as stated by many studies (Larkey et al., 2007). Clearly, it is not good to represent Arabic words by their roots, and equate all words derived from the same root and on the same time light stemming may result in lowering recall.

From that prospective, this thesis aims to use morphological analysis to get to something better than a root and/or light-stemming methods for indexing.

1.2 Proposed Approach

This thesis introduces the computational morphology is an urgent problem for Arabic Natural Language Processing, because Arabic is a highly inflected language.

The key goal of the study is that the results of the user search in Arabic language should ultimately produce the most relevant documents. To achieve this aim the thesis proposes algorithm that could handle the two major of Arabic stemming problems in both queries and documents. First it should be capable to translate all different forms of the word that have the same meaning to a standard form.

Second it should be capable to cluster all same forms of the word that have the different meaning to different group based on their semantic.

For testing the proposed approach, however, test documents collection, with query set and relevance judgments was created for evaluation test.

It should be noted, however, that this thesis does not aim to develop a complete AIR system. Instead, the main goal is to develop and evaluate techniques which can improve the efficiency of current AIR systems.

1.3 Research Questions

The primary goal of this research is to develop a stemming algorithm for AIR to handle computational morphology in Arabic language in Arabic test collection. In particular, the following questions are the core of this thesis:

1. What are the limitations of current Arabic stemmers, when applied in both documents and queries, on retrieval performance in current AIR approaches?
2. What are the effects of light stemmer on the performance of AIR systems?
3. What is the effect of normalizing of the test dataset without apply any Arabic stemmer on the performance of AIR systems?
4. How effectively the proposed method improve the performance of current AIR systems?

1.4Contribution

The main contribution of this thesis can be summed up as follows:

New technique of the stemming that aim to improve AIR systems are developed. The techniques consider the complex morphology of Arabic language, with special focus on lightly stemming of terms in Arabic language, which the proposed light stemmer are developed, to keep the semantic of term as much as possible.

Experiment results showed that the proposed stemming techniques for AIR systems could result in significant improvement in performance and could achieve comparable results to monolingual performance.

1.5Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 gives an overview of IR. The most important components and different techniques are covered. Chapter 3 is an in-depth coverage of Arabic information retrieval. It introduces the Arabic language and its characteristics, which make this language more challenging for the IR task and the impact of these features on IR systems.

Chapter 4 introduces the proposed techniques for Arabic stemmer in AIR systems. Chapter 5 the experiments conducted to evaluate the developed technique are presented. It contains also different comparisons for effectiveness of experiments. The last chapter, chapter 6, concludes the thesis with the limitations.

CHAPTER 2

INFORMATION RETRIEVAL

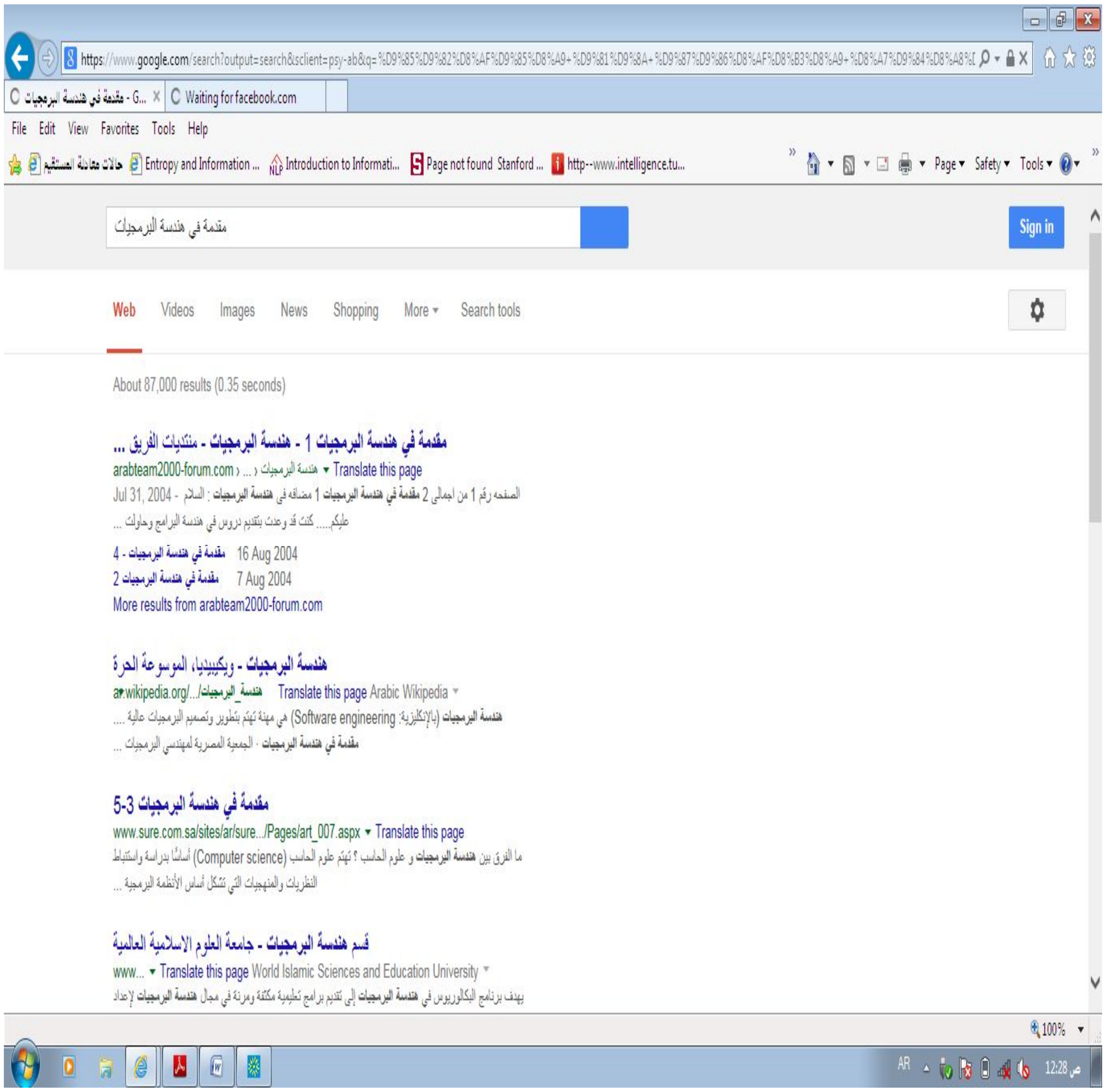
Information retrieval (IR) is the problem of satisfying users' information needs from unstructured data like (text, sound, image, etc), known as a corpus. In the context of textual IR, the major task of an IR system is to represent, store and manage information on an unstructured corpus (referred to as set of the documents) and to provide a user with relevant information on his information need referred to as query), through an accessing mechanism to that corpus. Defined in this way, the IR system should be able to: represent documents, which are often presented in a natural language~~,~~ in searchable representations; represent queries (often presented in a few words); and to find documents that match the represented query. The latter process is known as the matching problem.

Many of the techniques developed to improve IR systems retrieval effectiveness in other languages can also be applied for Arabic Information Retrieval (AIR) systems; however, techniques specially tailored for Arabic are also required. In this chapter, we introduce the main process in information retrieval, information retrieval model and Evaluation of IR Systems

2.1 Information Retrieval

Information retrieval (IR) is the process of finding unstructured data (like text, sound, image, etc), usually known as a *corpus* as a response to an information need requested by a user. Figure 2.1 shows an example of IR process through Arabic query submitted to the Google Web search engine[1].

[1] http://www.google.com


File Edit View Favorites Tools Help
Web Videos Images News Shopping More Search tools
Sign in

مقدمة في هندسة البرمجيات

About 87,000 results (0.35 seconds)

مقدمة في هندسة البرمجيات 1 - هندسة البرمجيات - منتديات الفريق ...
arabteam2000-forum.com › ... › هندسة البرمجيات ▾ Translate this page
Jul 31, 2004 - الصفحه رقم 1 من اجمالى 2 مقدمة في هندسة البرمجيات 1 مضافه فى هندسة البرمجيات : السلام عليكم..... كنت قد وعدت بتقديم دروس في هندسة البرامج وحاولت ...
مقدمة في هندسة البرمجيات - 4 16 Aug 2004
مقدمة في هندسة البرمجيات 2 7 Aug 2004
More results from arabteam2000-forum.com

هندسة البرمجيات - ويكيبيديا، الموسوعة الحرة
ar.wikipedia.org/.../هندسة_البرمجيات Translate this page Arabic Wikipedia
هندسة البرمجيات (بالإنكليزية: Software engineering) هي مهنة تهتم بتطوير وتصميم البرمجيات عالية ....
مقدمة في هندسة البرمجيات · الجمعية المصرية لمهندسي البرمجيات ...

مقدمة في هندسة البرمجيات 3-5
www.sure.com.sa/sites/ar/sure.../Pages/art_007.aspx ▾ Translate this page
ما الفرق بين هندسة البرمجيات و علوم الحاسب ؟ تهتم علوم الحاسب (Computer science) أساسًا بدراسة واستنباط النظريات والمنهجيات التي تشكل أساس الأنظمة البرمجية ...

قسم هندسة البرمجيات - جامعة العلوم الاسلامية العالمية
www... ▾ Translate this page World Islamic Sciences and Education University
يهدف برنامج البكالوريوس في هندسة البرمجيات إلى تقديم برامج تعليمية مكثفة ومرنة في مجال هندسة البرمجيات لإعداد

FIG. 2.1: An example of the Arabic query submitted to Google.

The main task of an IR system is to represent, store and manage information on an unstructured corpus (referred to as *document collection* in the textual IR process) and to provide the user, through an accessing method, with the relevant information and according to his information need (referred to as *query* in the textual IR process). In that context, the IR system should be able to: represent documents; represent queries; and find the documents' representations which match the query representation.

2.1.1 Main Processes in Information Retrieval

According to the definition above, the IR task consists of three main processes, these are: a searchable document representation over which the retrieval process is performed, a representation of a user information need and a matching process between the two representations, which results in a set of retrieved documents. A typical IR process is shown in Figure 2.1.

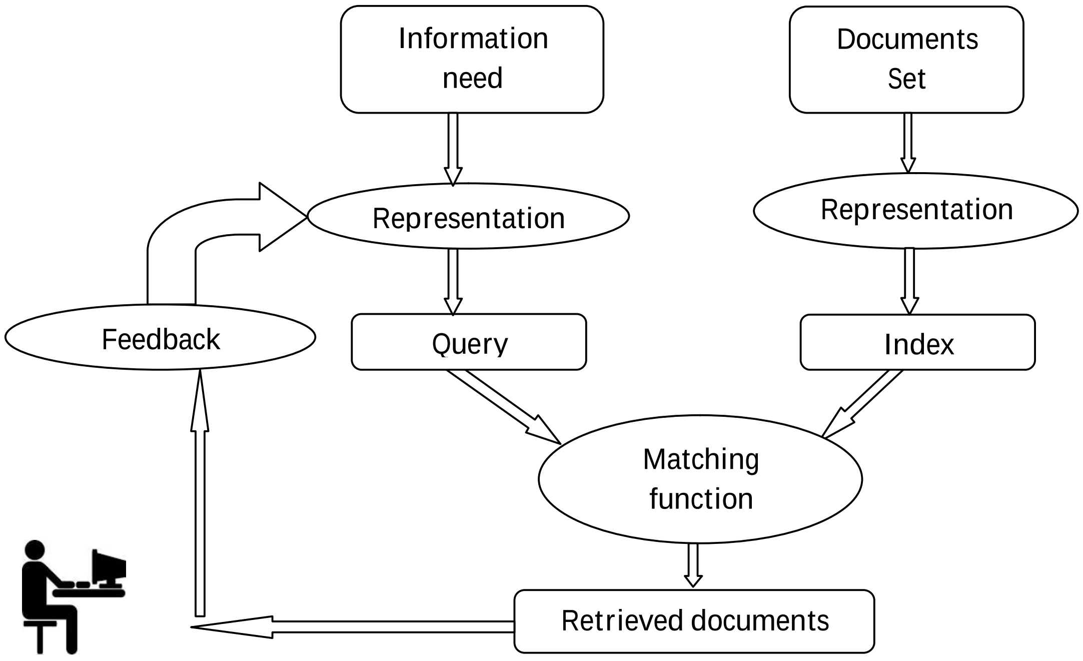


FIG. 2.2: A typical information retrieval task.

The process of representing the documents, in which keywords of documents are extracted, is called the *indexing* process (Manning et al., 2008). Such extracted keywords are known as *terms*. The term is the fundamental unit used for representing both documents and queries. A term can be a word, a phrase, a stem, etc, depending on what is needed from representing/indexing documents. The process of creating index terms often goes through several processes. Examples of such processes are tokenization and stemming. Tokenization is the process of splitting up document's text into words, phrases or other meaningful elements called *tokens*, while stemming provides a variant of a term which results from conflating the word to a particular representative stem form, e.g., root. For example, words like organization, organize, organizing, organizer, organizes May all be transformed to single stem word (organ in this example) .The final outcome of the document representation process (indexing) is a new searchable and structured documents' representation in a form of a set of terms (index). A user information need is also represented in the same way so as to create a represented query, which searches against the created index. Thus, the matching process is usually carried out between a query (information need representation) and a set of represented documents.

Based on this matching, a set of documents, with some relevance scores, are often retrieved. A matched document score is used to determine the relevance matching degree of a document with regards to a query. The document is considered as relevant if it satisfies the user information need, rather than just containing query terms. Hence, a good IR system is expected to rank relevant documents on top and its quality is measured in terms of *precision* and *recall*. The precision is a fraction of the returned documents that are relevant to the query, whereas the recall is a fraction of the relevant documents in a corpus that were retrieved by the IR system. In some certain cases, the IR system may need to re-formulate/feedback the original query so as to produce a better result list. In such a case the list of retrieved documents can be considered as an initial or intermediate result for the retrieval.

This is so-called *the relevance feedback*(Rocchio, 1971). In the relevance feedback process, the IR system may need the user to contribute in the process (by judging which

documents in the initial retrieved list are relevant) or the system could automatically perform this function based on top ranked documents. The last approach is referred to as *Pseudo Relevance Feedback* (PRF). Regardless of the feedback approach, the IR system employs relevance feedback so as to re-formulate a new information need representation and then a second retrieval process in conducted using this new query.
The steps that were highlighted above is a general overview of the retrieval process and there are still more details behind, many of which are covered in the following sections. However, the most important part of this process is what is called the information retrieval model. Retrieval models are described next.

2.1.2 Information Retrieval Models

A retrieval model (Manning et al., 2008, Nie, 2010) is a conceptual form for the information retrieval process whose task is to explain how an IR system represents documents and queries and how it predicts the relevance scores of retrieved documents with regards to a query during retrieval. From the retrieval models perspective, the final result of a matching between a query and a set of documents is an ordered or unordered result set list, depending on the used retrieval model. Retrieval models can be categorized into two main types - these are the *exact-matching* and the *best matching* retrieval models (Belkin and Croft, 1992). The categorization is derived from whether a document is exactly match the query (exact-matching model), and, or a document matches the query to some relevance degree, and thus the IR model provides the best matching documents that match queries (best-matching model). The latter models are also referred to as the ranked retrieval models.

2.1.2.1 Boolean Models

A Boolean model is an exact-matching retrieval model. In a Boolean model(Manning et al., 2008), queries are formulated by a combination of their keywords (terms) with Boolean logic operators (AND, OR and NOT) in a precise natural human language. Operators in Boolean model are handled during retrieval in a similar way to their use in conventional truth tables of the Boolean Logic. Likewise, documents are also represented as a conjunctive set of terms in a Boolean expression with a basic assumption that terms that do not occur in a certain document would not appear in its corresponding Boolean expression. Thus, a document is considered as exact-matched with regards to a certain query, if the terms represent that document satisfy the Boolean expression representing the query and, hence, a set of matched documents can be produced.

In conventional Boolean models, a result set is neither ranked nor makes use of frequency distribution statistics of terms in queries. But, some Boolean models allow non-Boolean operators like those used in proximity and wildcard operators – similar to those used by current search engines. However, in most cases document results are ordered chronologically, rather than making use of an accurate estimation for their relevance degree to queries. In fact, ranking documents had been extended to Boolean models using different approaches(Manning et al., 2008). In such a case it is called the Extended Boolean Model. In the extended Boolean models some heuristics methods, like those utilizing fuzzy set theory (Paice, 1984) and those using some types of weights (Fox, 1983) are used.

2.1.2.2 Ranked Retrieval Models

Besides the complex formulation of queries, it was previously illustrated that conventional Boolean models do not rank documents according to their relevance level. This, forces users to explore all the retrieved documents or the most relevant documents are not found (Jackson and Moulinier, 2007). There is always a possibility to find such documents at lower ranks. Furthermore, the need of the Boolean models for expert users, at least in formulating queries, to build up the query in a very precise query language with operators limit their effectiveness for non-expert users(Jackson and Moulinier, 2007). This is because formulating queries in Boolean models can be a real burden for such users, who often prefer to type just free text queries consisting of just a few words and the retrieved documents should be ranked according to the relevance degree of these documents with respect to those users' queries. In such cases a ranked retrieval model is needed (Hiemstra, 2000) so as to estimate some relevance scores for documents and to determine which of them is best matching the submitted query. In that context, the ranked models have been shown to be more effective than Boolean models(Manning et al., 2008). The next sections describe some of these ranked models.

2.1.2.2.1 Vector Space Model

As the name indicates, The vector space model measures the similarity between the query and the documents in the collection by considering the distinct query terms and the distinct terms in each document to occupy n-dimensional vectors (Salton and Lesk, 1968), where n is the number of the unique terms in the collection. The query vector and the document vector contain the weights of the distinct terms in the query and the document, respectively. Since vectors are used, the similarity between the two vectors can be simply measured using the dot product. For example, given the query vector $\hat{q}$ =< *wq1, wq2, wq3, …, wqn* > and a document vector $\hat{d}$ =< $wd_1, wd_2, wd_3, \dots, wd_n$ >, the similarity between the document and the query s(q, d) can be computed using the dot product as:

$$S(q, d) = \sum_{t=1}^{n} w_{q,t} \cdot w_{d,t} \tag{2.1}$$

Where $w_{q,t}$ the term weight in the query q, and $w_{d,t}$ is the weight of the term *t* in the document d. In order to avoid biasing the scores towards longer documents, it is often to normalize the vector length by the Euclidean normalization, which normalizes the cosine angle of vectors, of the query vector $\hat{q}$ and the document vector $\hat{d}$. In that context, the measure is called the cosine similarity measure.

$$s(q, d) = \frac{\sum_{t=1}^{n} w_{q,t} \cdot w_{d,t}}{\sqrt{\sum_{t=1}^{n} w_{q,t}^2} \cdot \sqrt{\sum_{t=1}^{n} w_{q,t}^2}} \tag{2.2}$$

where all symbols were defined above. The cosine of an angle determines the similarity between the query and the document vectors. If they are fully aligned, then the angle is zero, and thus, the similarity is one; equally, if the angle is 90 degrees, then the query and the document are totally unrelated (at least from the perspective of the query terms). Values in-between give the degree of the similarity between the two vectors and thus, these values are used to present a ranked list of results to the user.

2.1.2.2.2 Probabilistic Retrieval Model

The use of probability theory in documents retrieval created with Maron and Kuhns (1960), who discussed that documents in a collection could be ranked according to their probabilities of relevance with regards to queries. The main issue here is how to estimates a probability of each term in a query and how to assign the final probability that a document is relevant to the query. Mostly speaking, a probabilistic retrieval model uses the absence or the presence of a term in a document to predict a weight for that term(see Table 2.1). This weight match to the estimated probability of relevance of that term and the combination of all the query terms' weights is thereby used to determine whether the document is relevant or not.

	relevant (R)	not relevant(R')	Total
Term to present t	rt	st- rt	st
Term to absent t'	R-r	N- st –(R- rt)	N- st
Total	R	N-R	N

Table 2.1: Distribution of term t over the relevant and non-relevant documents in the collection.

N represents the number of documents in the collection, rt represents the number of relevant documents containing term t, St represents all documents containing t, and R is the total number of relevant documents.

Consider Table 2.9; the conditional probability that a document R is relevant if it consist a term t is given by

$$P\,(R|\,t) = \frac{r_t}{s_t} \qquad (2.3)$$

and the probability that a document R is not relevant if it contains term t is given by

$$P\,(R'|\,t) = \frac{s_t - r_t}{s_t} \qquad (2.4)$$

also, the probability that a term t is present in a relevant document is given by

$$P\,(t|\,R) = \frac{r_t}{R} \tag{2.5}$$

and the probability that a term t is present in a non-relevant document is given by

$$P\,(t|\,R') = \frac{f_t - r_t}{N - R} \tag{2.6}$$

With Bayes' theorem, the weight of term t, wt can be calculated as:

$$W_t = \frac{r_t/R - r_t}{(s_t - r_t)/(N - s_t - (R - r_t))} \tag{2.7}$$

Having calculated the term weight and assuming that terms are independent of each other, the weight for a document d is calculated by the product of its term weights.

$$W_d = \prod_{t \in d} W_t \tag{2.8}$$

The major purpose is to order documents by estimated relevance according to their weights, not the specific result of the above equation. Therefore, it is often possible to simply express this as a sum of logarithms(Witten et al., 1999):

$$W_d = \sum_{t \in d} \log W_t \tag{2.9}$$

The major problem with this model is its dependency on relevance judgments. A similar term weighting can be also used when queries are long. In such case, formula 2.23 becomes (Sparck Jones et al., 2000) that does not need pre-judged documents. Their Okapi BM25 measure considers the document frequency (f_t), the number of the documents in the collection (N), the frequency of a term in the document ($f_{d,t)}$ and it normalizes document length. The equation used to compute the similarity between a document d and a query q is:

$$\mathrm{BM25}(d, q) = \sum_{t \in q} \left[\log \frac{N - f_t + 0.5}{f_t + 0.5}\right] . \left[\frac{(k_1 + 1) f_{d,t}}{k_1((1-b) + b\ \frac{|d|}{avgdl}) + f_{d,t})}\right] . \left[\frac{(k_3 + 1) f_{q,t}}{k_3 + f_{q,t}}\right] \qquad (2.10)$$

In which |d| is the document length, avgdl is the average document length in the collection, $k_{3,}$ k_1,is another parameter to tune term frequency in query q and other symbols are as defined above.

The k_1,parameter affects the term weight. If it is 0, then the term weight is decreased, meaning that the term weight is not affected by its frequency in the document, and if it is set to a bigger value, the term weight increases as its frequency increases in the document. The tuning constant k3 affects the number of term instances that participate to the ranking. For example, if k3 is set to 0, then only one instance of each query term participate to the ranking. The constant b is used to manage the document length normalization. If it is set to 0, no normalization will take place; if it is set to 1, then normalization is in complete effect. In TREC 6, the value of k3 was 1.2, the value for k3 was in the range from 0 to 1000 and the value of the b parameter was 0.75(Walker et al., 1997) .

In our experiments in Chapter 5, we use the Okapi BM25 weighting model with default values determined for English (k1= 1.2, k3= 7, and b= 0.75).

2.2 Evaluation of IR Systems

The performance of an IR system is often considered by its ability to retrieve the relevant documents with regards to a query posted by a user (*effectiveness*). In this section, the used measures and approaches to evaluate the IR systems are described. While some researchers studied the effectiveness of an IR system by measuring user satisfaction(Al-Maskari et al., 2007, Spink, 2002), it is more common to check how well a system performs on queries. Hence, a set of queries with some well known relevant documents in some standard collection, are submitted using different ranking algorithms to the collection. Depending on the retrieved documents, the results of each system are compared to each other. But, it is often to upper-bound the effectiveness of the retrieval by using some manually judged results. The entire process is described below.

2.2.1Test Collections and Evaluation Forums

One of the essential purposes for the use of corpus in IR is the task of measuring effectiveness of an ad-hoc IR. The task is usually performed by using a corpus with three main components: a set of documents (document corpus), a set of topics (queries) and a set of relevance judgments for each query in the query set. In such a case, the corpus is known as the test collection(Croft et al., 2010). Among these types of components, relevance judgments are the most critical parts.

It is often that documents in the standard test collections consisting of several different fields (i.e. title, paragraph, etc) with every document having a unique identification number. Topics represent the user information needs. In the majority of the standard collections, e.g., TREC, search topics in the ad-hoc track are structured in many fields. The three major ones are(Croft et al., 2010): title, description and narrative. The title field is a concise query consisting only of few words. The description field is a longer sentence(s) of the query and often contains more useful details about the query. The narrative field is the longest part of the topic file. It specifies in detail the criteria of judging documents relevance. The narrative field is the major component that is often used by the assessors. Queries are often formulated from the topic files. The relevance of a document with respect to a query specifies the value of that document to fulfill the information need from the user's perspective.

Relevance judgment is complex process because it is essentially subjective, as it can vary according to the person who makes the assessment or for the same person at different times (Voorhees and Harman, 2001).

Whenever relevance judgments are produced, it is essential to know total number of relevant documents for each topic. But, this is infeasible, especially for large test collections, due to the large effort needed. As a result, a sample of documents for each topic is only assessed. This is known as *pooling* (Robertson and Jones, 1976).

The majority of relevance judgments in TREC collections are of binary scale(Croft et al., 2010). In such a case, the retrieval task focus on higher recall, where it is important to retrieve any relevant document. However, for some tasks, like to what degree a document is relevant to the query, multiple levels of relevance (graded non-binary relevance) can be used. For example, relevance assessment for a particular document with respect to a specific query can be done on a four-point scale (0-3), with 0 = irrelevant, 1=marginally relevant, 2= good and 3=excellent. In such non-binary relevance, the retrieval task emphasizes highly relevant documents or document di should be ranked higher than document dk because it is more relevant.

2.2.2 Measuring Effectiveness

Several measures have been proposed to evaluate the effectiveness of an IR system. However, the used measure depends solely on the required retrieval task and used relevance judgment. If a binary relevance judgment is being used then the employed measures will be the *precision* and the *recall*. The recall measures the ability of a system to retrieve all the relevant documents of a particular query (Jones and Van Rijsbergen, 1975). Formally,

$$\text{Recall} = \frac{\text{number of retrieved relevant documents}}{\text{number of relevant documents in the collection}} \qquad (2.13)$$

In the same context, the precision is a fraction of the retrieved documents that are relevant to the submitted query(Jones and Van Rijsbergen, 1975). Formally,

$$\text{Precision} = \frac{\text{number of retrieved relevant documents}}{\text{number of retrived documents}} \qquad (2.15)$$

For a specific query, the precision evaluates the capability of the IR algorithm to remove non-relevant documents, while the recall evaluates its ability to retrieve all the relevant documents. The underlying idea behind the two measures assumes that users need to retrieve relevant documents as much as possible, while on the same time it is beneficial to reduce the irrelevant documents whenever it is possible.

The major drawback of using precision and recall is that both measures are not able to distinguish between documents rankings (Croft et al., 2010). For instance, if there are two relevant documents that are being retrieved at ranks 1 and 4 using some particular algorithm and the same documents are ranked at positions 7 and 10 using another algorithm, then the two algorithms would have the same precision and recall values at rank 10. Therefore, the precision at a predefined rank position, e.g., at rank p, has been developed to measure IR systems effectiveness. As the name indicates, precision at rank p changes the search task to focus on retrieving the most relevant documents at a given rank, rather than just finding all the relevant documents (Manning, et al., 2008; Croft et al., 2010). But, the measure also may not differentiate differences in the ranking at positions 1 to p. Therefore, precision is often used at 11 standard recall levels: 0, 0.1, 0.2,…, 1.0. Some interpolation mechanism is used also, in order to obtain the precision values at all the standard recall levels. If the values of the precisions from the rank positions are averaged for each query, where a relevant document is retrieved, then the measure is called the *average precision*, which is a single summarization value. The overall conclusion about the performance of a specific algorithm is then determined by averaging the values of the average precision over a sufficient number of queries (Manning, et al., 2008; Nie, 2010). The used measure is known as the *Mean Average Precision* (MAP). The MAP is the most widely used measure in evaluation of IR and CLIR systems (Croft et al., 2010; Nie, 2010).

2.2.3 Significance Test of Retrieval Performance

In IR, it is essential to know if there is an improvement of a particular used retrieval algorithm over another and whether this improvement is caused by a real difference between the two algorithms or the difference has just appeared by chance. Such difference between algorithms is usually measured using statistical significance tests. In particular, in IR the concern is in the paired tests, as the algorithms under evaluation use the same set of queries, which in turn should be of a reasonable amount.

When a statistical test is used for comparing performance of two ranking algorithms (say algorithm A and algorithm B), a typical confidence level of a 95% is utilized. This value means that in 95% of choices of A and B the performance of algorithm A will be higher than that of algorithm B. In other words, if the chance of the practical difference between the algorithm A and the algorithm B, known as significance value, is small enough (i.e. < 0.05), then this difference is considered as significant as there is 5% probability of being false positive. Since the significance value represents the probability of error in accepting that the result is correct, the value 0.05 is considered as an acceptable error level. The most commonly used significance tests in IR are the Student’s t-test and the Wilcoxon signed-rank test(Croft et al., 2010).

2.3 Summary

In this chapter, the major approaches used to parse, index and retrieve documents using IR systems are described. It is explained in the chapter how the IR system extracts the proper tokens from both documents and queries. Many techniques are employed for this purpose, e.g., tokenization, normalization and stemming. However, these techniques are often language-dependent.

The retrieval models that are used to match queries and document have been reviewed and it is concluded that the majority of the existing IR systems employed the ranked models for retrieval.

The current used approaches for measuring IR systems are briefly covered in this chapter. It was shown that the use of precision at rank p is better than using only precision and recall as the former distinguishes between documents rankings.

CHAPTER 3

ARABIC INFORMATION RETRIEVAL

Over the last decades Arabic IR has become one of the most interest areas of research in IR, particularly with the explosive growth of the language on the Web, which require retrieving documents in other languages. This growing interest in Arabic, but, is caused by its morphology, which is totally different from the European and the East Asian languages (Larkey et al., 2007, Moukdad, 2006, Xu et al., 2002, Yahya and Salhi, 2011). However, in spite of the great achievements in existing Arabic retrieval systems, but current AIR is comparatively poor and much weaker than for English(Abdelali, 2006, Alansary et al., 2007, Yahya and Salhi, 2011) .In particular, Arabic is still lacking in high quality IR and NLP tools, e.g., the need for efficient Arabic stemming. . In this chapter, we review the basic characteristics of the Arabic language, Orthographic Variations, morphology and its Diacritisation. The remainder of this chapter we describe Arabic stemmer under three broad categories: morphological analyzers, light stemmers, and statistical approaches. Morphological analyzers attempt to identify the affixes, stem, and root of a given word, and are primarily used for natural language processing (NLP) tasks such as part-of-speech tagging. In contrast, light stemmers focus on removing affixes to improve retrieval effectiveness, and do not attempt to identify grammatically correct stems. Finally, statistical approaches extract n-grams for indexing and retrieval, and operate independently of any language-specific rules.

3.1 The Arabic Language

Arabic is one of the most widely spoken languages on the world. It is a member of the Semitic languages group. Semitic languages are among the first languages that has an old written script, for example Aramaic, Hebrew and Phoenician(Arabic-History, 2014).The first documented inscription in Arabic was found around 328 C.E. There is evidence that the Arabic script was derived from the ancient Nabatean (Aramaic) alphabet, but the language flourished independently with the rise of Islam in 622(Arabic-History, 2014). Arabic also is a liturgical language of 1.6 billion Muslim speakers and is one of six official languages of the United Nations.

Arabic can be classified into three forms (Saad and Ashour, 2010): Classical Arabic (CA), Modern Standard Arabic (MSA) and Dialectal Arabic. Classical Arabic was the language of the Quran. Modern Standard Arabic (MSA) is modernized version that has been derived from the classical Arabic. MSA is the formal language in the Arabic world for reading and writing(DeYoung, 1999). MSA is used to write all books, newspapers, magazines and media text. Dialectal Arabic, as the name indicates, is used in informal communication in Arabic countries. Each dialect has its own new terms such as those borrowed from other languages(Bishop, 1998). Accordingly, the term ‘Arabic’ refers to both MSA and Dialectical Arabic(Abdelali, 2006).

3.1.1 Character sets

The Arabic alphabet has 28 letters(Tayli and Al-Salamah, 1990).Table 2.1 illustrates the complete set of the Arabic alphabet . Arabic script is written from right-to-left while Arabic numbers are written and read from left-to-right. Script in Arabic consists of two types of symbols (Habash and Rambow, 2007): these are the letters and the diacritics (known also as short vowels), which are certain orthographic symbols that are usually added to disambiguate Arabic words. For instances, MEEM (م) is a letter equivalent to ‘M’ in English, whereas مٌ is a diacritized letter with the sound ‘MW’, like in the word Modern.

ص	ش	س	ز	ر	ذ	د	خ	ح	ج	ث	ت	ب	أ
ي	و	هـ	ن	م	ل	ك	ق	ف	غ	ع	ظ	ط	ض

TABLE 3.1: The complete set of the Arabic letters.

Short vowels are always omitted in written MSA texts as Arabic speakers could differentiate easily between words with similar forms from the context in which they occur. Arabic letter shapes can be changed according to their positions within words. Informally, Arabic letter can have four different shapes: isolated, initial, medial, and final. For example, the shapes "مـ ,”"ـمـ" , "ـم” ," م" are the four different representations of the single letter MEEM respectively. But, it is important to notice these different representations of a character shapes are all mapped to single base code.

3.1.2 Major Part-of-Speech in Arabic

Arabic words can be categorized into three major categories(part-of-speech): nouns, verbs and particles. A noun is a word that indicates a meaning and usually it is not associated with time. Nouns are often preceded by definite article, personal pronouns, demonstrative pronouns and relative pronouns. Table 3.2 illustrates the different types of the noun predecessors.

Predecessor	Example	Meaning in English
Definite	ال	The
personal pronouns	انا – انت	Me –You
demonstrative pronouns	هذا– هذه	This - this feminine
Relative pronouns	الذي - التي	Which masculine- Which feminine

TABLE 3.2: Different types of Nouns proceeded

A verb is a word that indicates a meaning but, unlike nouns it is usually associated with time. The verb can be classified in two types, these are the perfect and the imperfect (imperfect represent the present and future tenses) forms. Perfect verbs indicate whether the event for which the verb expresses, is completed (meaning the past) for example the boy was *leaved* to the home (the action was completed happened), whereas imperfect indicate uncompleted events,) for example the boy will *leave* to the home (the action will happen in future). A particle is a word that has no meaning unless it has been attached to other words, for example, a preposition.

Words in Arabic are either masculine or feminine, each of which is formed in a different way. For example, the feminine form of the word مُعلم (meaning: single masculine teacher) is the word مُعلمة which is formed by adding the letter 'ه' to the end of the word. The same characteristic appears also in both nouns and verbs in Arabic in order to specify number (singular, dual 'two' and plural) as in مُعلم, مُعلمان and مُعلمون (meaning: singular teacher, two teachers and more than two teachers, respectively).

3.1.3 Morphology

As in the majority of the Semitic languages, Arabic has a rich morphology with a highly derivational system usually conflated from roots. Roots in Arabic can be either trilateral, quadrilateral and in very rare forms consist of five consonants. There are 6,350 triliteral roots and 2,500 quadrilateral ones listed in Lisan-Alarab, one of the major thesaurus for referencing to the Arabic language (Moukdad, 2006). Among the 10,000 roots, only about 1200 are still in use in MSA (Hegazi and El-Sharkawi, 1985).

Words and morphological variants in Arabic are often derived from roots using *patterns*. Patterns are used as the standard frames for Arabic words. Grammatically, the major standard frame is the trilateral pattern فعَل (transliterated as f-à- l), in which each letter corresponds a single consonant in the trilateral root. The pattern preserves “f”, “à”, and “l” in the same order. For example, in the trilateral root كَتبَ (meaning: wrote), “f” corresponds to the first letter "ك" in the root, “à” corresponds to the middle letter "ت" and “ب” corresponds to the last letter “ل”. More patterns can be derived from the main pattern فعل (f-à-l) by attaching some letters to the root at the beginning (prefix), the middle (infix), or at the end (suffix). Examples of some patterns are أفعِلة,أفعَالّ , transliterated as a-f-à-l, a-f-i-l-à, respectively. Using these rich rules, it is possible to derive more than 50 derivatives from only one single root. Table 3.6 illustrates several words derived from the root كتب, which corresponds to the major pattern فعل (f-à-l), according to some different patterns, in which some letters are added to the main pattern.

Word	Pattern	Pattern Transliterated	Meaning
كتب	فعل	f-à- l	write (a three consonants root)
يكتب	يفعل	y- f-à- l	He writes
كتبنا	فعلنا	f-à- l-n-a	We write
كتبن	فعلن	f-à- l-n	They write (plural feminine)
يكتبون	يفعلون	y- f-à- l-o-n	They write (plural masculine)

TABLE 3.3: Different derivatives from the root كتب

More affixes can be added to the derived patterned word so as to create a more complex structure. Definite articles, e.g., ال (meaning: the), conjunctions, particles and other prefixes can be added to the beginning of a word (known as prefixes), whereas suffixes can be added to the end.

Antefixes are usually prepositions added to the beginning of words before prefixes. Prefixes are attached to illustrate the present tense and imperative forms of verbs and usually consist of one letter. Suffixes are added to indicate genders and numbers as it was illustrated before in section 3.1.3. Postfixes are used to represent pronouns or the absence of a person while talking. For example, the word لتكسرنهم (meaning: you will surely break them) can be decomposed as follows: (antefix: ل, prefix: ت , root: كسر , suffix: ن and postfix: هن). For the purpose of understanding stemming, all Arabic affixes are listed in Table 3.7, quoted in(Kadri and Nie, 2006).

Antefixes	Prefixes	Suffixes	Postfixes
وبال، وال، بال، فال، كال، ولل، ال، وب، ول، لل، فس، فب، فل، وس، ك، ف، ب، ل	ا، ن، ي، ت	تا، وا، ين، ون، ان، ات، تان، تين، يون، تما، تم، و، ي، ا، ن، ت، نا، تن	ي، ه، ك، كم، هم، نا، ها، تي، هن، كن، هما، كما
Prepositions meaning respectively: and with the, and the, with the, then the, as the, and to (for) the, the, and with, and to (for), then will, then with, then to (for), and will, as, then, and, with, to (for)	Letters meaning the conjugation person of verbs in the present tense	Terminations of conjugation for verbs and dual/plural/female/male marks for nouns	Pronouns meaning respectively: my, his, your, your , their, our, her, my, their, your, their, your

TABLE 3.4: Arabic affixes in MSA (Arabic is read from right to left).

3.1.4 Orthographic Variations

Orthographic Variations in Arabic represent as a one of the main challenges to Arabic information retrieval. This is because in most cases the matching process between the query and the document collection may fail.

Orthographic variations in MSA, has six forms(Zawaydeh and Saadi, 2006), these are Holy Quran variations, Typographical variations, Phonological variations, Cross-Linguistic variations, Morphological variations and Spelling variations. Figure 3.1, illustrates the different types of orthographic variations in Arabic (Zawaydeh and Saadi, 2006).

The Holy Quran variations are stemmed from the fact that the Holy Quran has special rules for deleting and substituting of letters. For example, the letter ALIF in the Quran is deleted after the vocative Yaah, e.g. in the pharse يأ أ يها الناس

(Meaning: oh people) the letter ALIF in the middle is usually removed, resulting in the phrase يايها الناس .

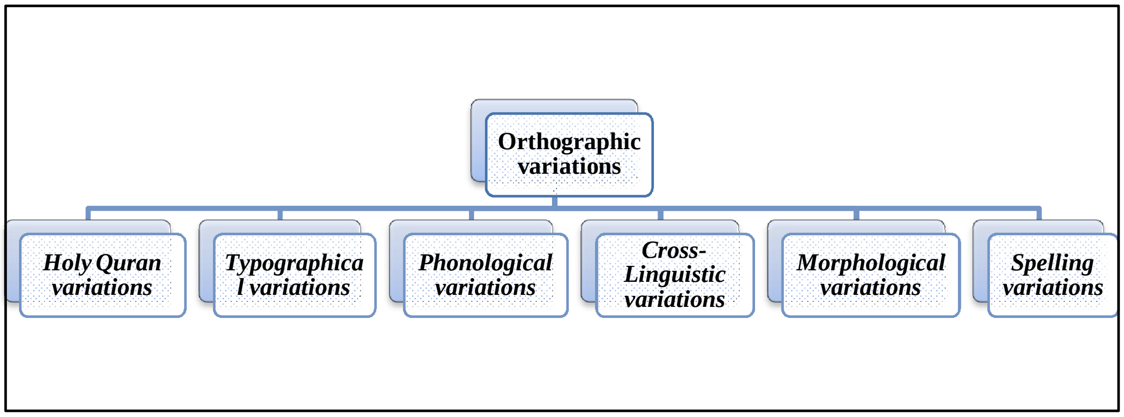


FIG. 3.1: Types of orthographic variations in MSA.

Typographical variations are simply caused by the fact that Arabic words may erroneously written. One major reason for such type of errors is that there are some certain Arabic letters, which have different glyphs. For examples, the Arabic letter ALIF has four different glyphs (أ, إ , آ and ا) and the letter YAA may be written with its dotted or un-dotted forms (ي and ى). The letter HAA also has two forms; these are the forms ه and ة, respectively. Figure 3.2 shows some different typographical variations for the word الأحسان(meaning: charity).

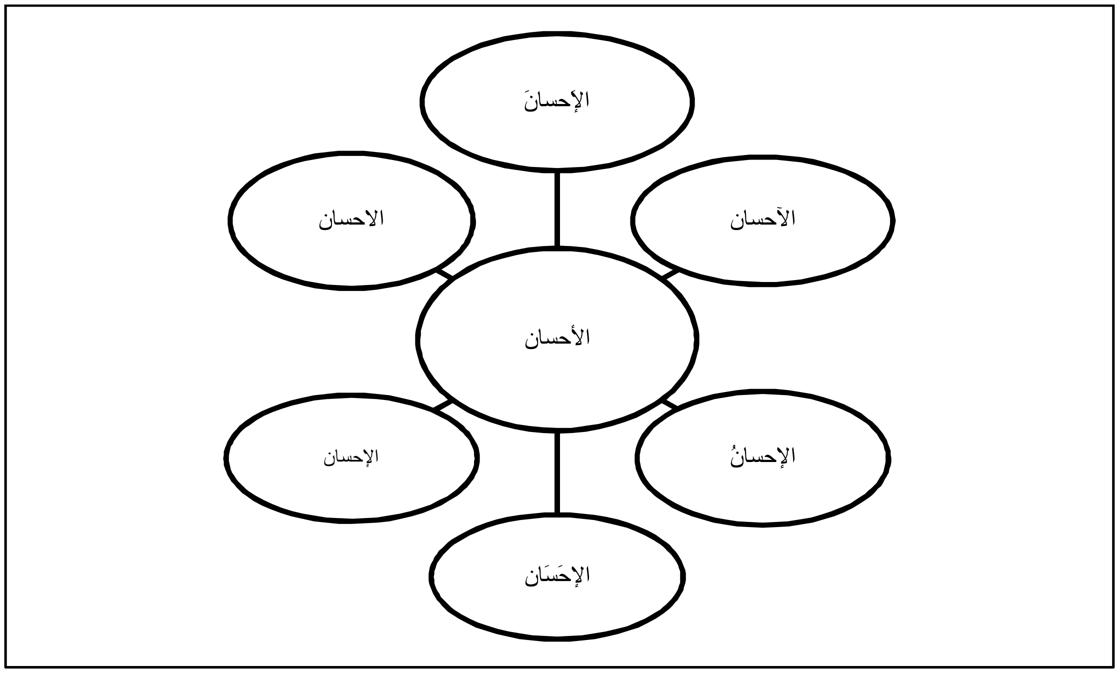


FIG. 3.2: shows different typographical variations in word الأحسان

Phonological variations happen due to regional variations. The regional variation problem in Arabic is the one of the major causes for adding a new level of complexity and ambiguity to Arabic information retrieval. This is because there are 22 countries with Arabic as the mother tongue language. Thus, in the Arabic world, dialects differ from one country to another resulting in different levels in variations (pronunciation, phonology, vocabulary, morphology and syntax(Habash and Rambow, 2007)).

Cross-Linguistic variations occur mainly when foreign words are transliterated into Arabic and, thus, resulting in different transliterated forms. For example, the transliterated Arabic words for the English proper noun ‘Angelica’ may look like أنجليكا , انجلكا , أنجليكه And أنجليكة .

Beside the contribution of the complex morphology of Arabic, *Morphological variations* usually appear as for unclear morpheme boundaries of words, e.g., a word like أشكي لك (meaning: I complain to you) may be written as أشكيلك. Run-on words problem contributes also to morphological variation difficulty (Buckwalter, 2004, Population, 2014). Run-on words in Arabic usually occur when the word ends with a non-connecting letter such as YAAH, ALIF, WAW or RAA. For example, sometimes the word نور الدين (meaning: the proper noun Nour Eldin) may be written as نورالدين without space between the word نور and the word الدين.

Spelling variations, occur usually as a result of misspelling of MSA words. For example, the word سماء (meaning: the sky) may erroneously be written as سما (meaning: to get a higher level)

Orthographic variations in Arabic play a big role in confusing IR systems and usually lead to a larger possibility of mismatching between queries and documents. According to Beesley (1998), cited in Aljlayl and Frieder (2001), ambiguous written Arabic words have an average of five valid morphological analyses per word and, thus, many non-relevant documents will be retrieved when a query is submitted in Arabic, whereas many relevant ones may not appear in the final retrieved list.

3.1.5 Diacritisation

According to Arabic grammar rules, diacritisation refers to short vowels that may appear above or below Arabic letters and it is usually used to remove any ambiguity that may occur in both meaning and pronunciation of words .An example of ambiguity that may occur due to the miss of diacritical marks is the word ولد. In this example, the word ولد is very ambiguous unless it is disambiguated using diacritical marks as it can be diacritised in many different forms, e.g., وُلد (meaning: he was born) and وَلد (meaning: boy). Diacritised texts are found in early educational schools, dictionaries and the Holy Quran. In formal letters and published articles and journals, Arabic text is usually written without diacritical marks as Arabic speakers can easily distinguish the meaning of the words from the context in which they appear but, this is usually a problem for non-Arabic speakers who represent more than 85% of the Muslims(Population, 2014).

3.1.6 Synonyms

Arabic has a numerous terms of synonyms, in which a particular word could have multiple synonyms. For example, in Arabic the lion (meaning: أسد) has more than 150 names (Fustat Adab, 2014) according to its age as well as its name’s synonyms: (الكُرْدوسُ, العَرَنْدَسُ, مَرْزُبانُ). This represents as a real difficulty to Arabic IR because in order to catch all relevant documents the word needs to be expanded by its synonyms.

3.2 Text Pre-processing in Arabic Information retrieval

Preprocessing in Arabic includes the tokenization, the normalization and stemming of words and the removal of the stopwords. The Tokenization is the process of breaking a stream of characters into meaningful pieces called tokens and possibly omitting some particular characters - such as punctuation. The output of the tokenization phase is that words in documents are segmented and identified from one another and be ready for indexing, as in the IR process. After a text has been broken up into tokens, those tokens are usually normalized. The *Normalization* is the process of producing the canonical form of a token in order to maximize matching between a query token and document collection tokens. One common approach in normalization, for example, is to remove a certain character/symbol from a token, e.g., any *non-character* symbol and the *stopwords* . The Removal of non-characters(Abdelali, 2006) includes the removal of punctuation marks, diacritics and *Kasheeda*, known also as *Tatweel .for* example, the word عثمان (a proper noun) can be written with *kasheeda* as عثمـــــــان. the *stopwords* are words with no useful meaning. These are typically prepositions i.e. الي (meaning in English by), , pronouns i.e. هي (meaning in English she), etc. . The normalization is also a language-dependent process. For example, in the Arabic language, the Normalization used to represent different forms of a letter with a single Unicode representation. This is important to moderate the orthographic variations. Such normalization includes: replacing ALIF in HAMZA forms (*ALIF* combined with *HAMZA* that is written above or below the *ALIF* like in 'أ and إ') and *ALIF MADDA* (آ) with bare *ALIF* (ا); replacing final un-dotted *YAA* (ى) with dotted *YAA* (ي); replacing final *TAA MARBOOTA* (ة) with *HAA* (ه); replacing the sequence ىء with ئ; replacing the sequence يء with ئ and replacing ؤ with bare *ALIF* (ا). In its complex forms, normalization is used to handle morphological variation and inflation of words (Levow et al., 2005). This is called stemming. Since documents and/or queries may have several forms of a particular word, stemming is the process of mapping and transforming all the inflected forms of a word into a common shared form. In monolingual IR, stemming appears to have a positive impact on recall more than precision(Kraaij and Pohlmann, 1996, Pirkola et al., 2001).

3.3 Classification of Arabic Stemmer in AIR

It was previously explained in section 3.2 that stemming has a positive impact on both precision and recall. However, stemming approaches are language-dependent as the process is mainly based on the languages characteristics. For Arabic IR, several approaches have been developed. These approaches can be classified into three major categories: morphological or heavy analyzers (root-based), light stemmers, and statistical approaches. Morphological analyzers attempt to discover the affixes, stems and roots of words and they are mainly used in the tasks of Natural Language Processing (NLP) paradigms such as part-of-speech tagging. Light stemmers attempt to remove the affixes of words so as to improve retrieval effectiveness and they do not perform heavy linguistic processes to provide the correct stems. Statistical approaches attempt to isolate the stemming process from its dependency on language feature and thus they use some statistical methods like n-grams approaches for indexing and retrieval of documents.

3.3.1 Morphological Analyzers

Early researchers were influenced by the conventional ways of indexing contents and thus, early attempts developed systems that depend on morphological analysis and root extraction.

El-Sadany and Hashish (1989) developed a morphological analyzer that deals with text words. Sadany and Hashish analyzer accepts a word and returns its different morphological characteristics, such as the root, and the pattern .The analyzer has the ability also to accept a sentence written by a user provides the roots of the words in that sentence. The analyzer also allows the user to clarify ambiguity occurs in sentences. No evaluation has been provided for this system.

Al-Fedaghi and Al-Anzi (1989) developed a system to discover the trilateral root of Arabic words. The system consists of two check lists: a list of Arabic patterns and a list of valid trilateral Arabic roots. The pattern list does not contain only the essential Arabic patterns, but also those patterns with the valid appended affixes. Instead of removing affixes, the system compares the input words with the patterns of the same length and it returns the matching root if it is existed in the valid root-word list. The authors reported that their algorithm successfully extracts up to 80% of words roots in a small text collection. However, Khoja and Garside stated that no accurate figures were reported(Khoja and Garside, 1999).

Al-Shalabi and Evens (1998) extended Al-Fedaghi and Al-Anzi algorithm so as to discover the quadrilateral roots for Arabic words. They improved the efficiency of the algorithm by removing the longest possible prefixes and searching for the root. The algorithm creates roots by comparing patterns with the remaining first five letters. From that prospective, Al-Shalabi and Evens algorithm produces both trilateral and quadrilateral roots. The algorithm was tested for accuracy and efficiency. One major drawback in this work was stated by (Khoja and Garside, 1999)., who reported that it is not known how the algorithm deals with weak letters.

Khoja and Garside (1999) developed a new algorithm that extracts roots from Arabic words. The algorithm is different from the earlier morphological analyzers in that it considers weak letters when producing roots. The algorithm uses predefined lists of valid Arabic roots and patterns. After every prefix or suffix removal, the algorithm compares the remaining stem with the defined patterns. When a pattern matches a stem, the root is extracted and checked against the list of valid roots. If no root is found, the original word is returned. , Khoja's stemmer has some drawbacks. For example, Khoja may result in an over-stemming problem. For instance, Khoja produces the root طفل (meaning: child) for the word طفيليات (meaning: parasites), and produces the root لَعِب (meaning: play) for the word لعوب(meaning: irresponsible), which are totally different in meaning.

Al-Kharashi (1991) and Al-Kharashi and Evens (1994) checked the effectiveness of indexing Arabic text with their Micro-AIRS system using roots, words and stems. Using 355 bibliographic records, they manually created a lexicon of 1,126 words, 725 stems and 52 roots and the lexicon was used to discover roots, words and stems. Using a set of 29 queries and equivalent relevance judgments, they reported that the root-word index produce both the stem and the word index, with the word index being the least effective.

Similar experiments were developed by Abu-Salem (1992) who conducted a series of experiments using words, stems, and roots as index terms. His experiments on a small corpus consisting of 120 documents and 32 queries definite the conclusions of Al-Kharashi (1991) that root-based indexing outperforms both stem-based and word-based indexing.

Abu-Salem et al. (1999) experienced the effects of three weighting schemes on the performance of the three different retrieval methods. They used the cosine similarity coefficient with a binary weighting scheme, the tf.idf weighting scheme, and a mixed stemming method between the query and the document. In the mixed stemming method they used a lexicon of stems, words, and roots along with their particular average weights across all documents to discover the best weight for terms in the query. They choose how to index every term based on the best weight of its root, word, or stem. Their result concludes that the root indexing (mixed stemming method is the best of the methods they used.

Hmeidi et al. (1997) compared whether automatic indexing using words, roots, and stems is better than manual indexing. They used a test collection of 242 abstracts and 60 queries with relevance judgments, Their results showed that manual indexing using roots as index terms gives better results than using words and stems. They also concluded that automatic indexing using roots as index terms gives better results than using words.

```
                                    تعمل    Processing token :
                                    tEml     Transliteration :
                                                  SOLUTION #1
                                          Eamil       Lemma  :
                                        taEomalVocalized as  :
                                                  Morphology :
                                   prefix : IVPref-hy-ta
                                               stem : IV
                                       suffix : Suff-0
                                       Grammatical category :
                                    IV3FS    prefix : ta
```

FIG. 3.3: Two solutions for the word تعمل (meaing: she works/you work) using the Buckwalter analyzer.

Buckwalter (2002) developed an Arabic morphological analyzer that returns the possible pattern of an Arabic word (see figure 3.3). The analyzer uses three dictionaries consisting of the common Arabic prefixes, stems and suffixes along with another three compatibility tables to validate the prefix-stem, stem-suffix, and prefix-suffix combinations. Buckwalter analyzer accepts an Arabic word and provides its possible patterns by a translated Arabic letter in English. However, the fundamental dictionaries and rules of Buckwalter analyzer were updated later(Buckwalter, 2004). The morphological analyzer cannot be used directly in IR experiments as it provides more than one possible solution for the same word.

Darwish and Oard (2002) developed *SEBAWAI*, an Arabic analyzer that is based on automatically derived rules and statistics. SEBAWAI has two main modules. The first module constructs a list of "word-root" pairs i.e. (وسحابهم, سحب). Then it extracts a list of prefixes, suffixes and stem templates and follows this process by estimating the probability that a prefix, suffix or stem template would occur. For example, given the pair (وسحابهم, سحب) in the example above, the system produces و (meaning: and) as the prefix, هم (meaning: theirs) as the suffix and "CCAC" as the stem template (C's represent the letters in the root).The second module of SEBAWAI takes a word and produces the possible combinations among prefix, suffix and template. These combinations are obtained by eliminating prefixes and suffixes from words and then comparing all the produced stems to templates. As a result, a list of ranked roots is produced. These roots will be matched automatically against the list of the 10,000 roots extracted from an electronic copy of Lisan Al-Arab to confirm their existence. SEBAWAI has some limitations stated by its developer. First, it cannot stem transliterated words such as entity names because it binds the choice of roots to a fixed set. Second, SEBAWAI is not able to deal with words that have one letter length. Such words are very limited in Arabic, e.g., ع (meaning: grasp). Third, SEBAWAI cannot deal with some individual words that constitute complete sentences, like لَنَهْدِيَنَّهُم (meaning: we will surely guide them).

Lee et al. (2003) developed an Arabic morphology system that divided words within sentences to prefix-stem-suffix form. The system adopts a trigrams language model and a list of valid prefixes and suffixes. The system achieved 97% accuracy on a test corpus consisting of 28,440 words. The system does not handle Arabic infixes.

Daoud and Hasan(2011) begins the stemming process by the stem(root based stemmer), rather than the removal of the affixes. This is done by segmenting the Arabic word (or string) according to a lookup dictionary that contains only valid stem. In that context, the longest match is returned whenever number of words is minimized. Daoud and Hasan(2011) claimed that their algorithm is the ideal stemmer when it was compared to both Khoja and light 10 stemmers.

3.3.2 Light Stemmers

Morphological analyzers for stemming need intensive analysis and they have shown to be unhelpful for AIR(Larkey et al., 2007). Additionally, it requires comprehensive lists of prefixes, suffixes, stems, roots and rules to be set in advance. Such lists are often imperfect due to ambiguity and the derivational system in the Arabic language. For such reasons light stemming, in which words are lightly stemmed, were proposed.

Aljlayl(2002) developed a novel light stemmer that help to eliminate the most frequent prefixes and suffixes, rather than to discover the correct root of an Arabic word.

The stemmer begins by eliminating diacritics from Arabic words. Then it normalizes words according to what have been shown in section 3.2. Aljlayl put some condition that a word should be consisting of three or more letters in order to remove its prefixes and/or suffixes. Thus, the algorithm initially remove affixes by eliminating the conjunction و (meaning: and) and then it removes the definite articles with any preceding prepositions and conjunctions, e.g., بال (meaning: with the). The stemmer removes also the most common suffixes starting with the longest ones and it then removes prefixes such as the prepositions ي, ب, and ي if the second letter is "ت". One good feature for the algorithm is that it uses a list of foreign words to avoid stemming them. It is not clear when the checking is done. Nevertheless, the work of Aljlayl neither provides complete lists of the eliminated affixes nor a clear mechanism for the removal of stoppwords. The stemmer participated in the TREC 2001 evaluation and was the second-best stemmer out of seven stemmers used in the evaluation. Aljlayl conducted his experiments using TREC 2001 and it was shown that his stemmer performance outperforms Khoja, which is a root-based. The results reported in Aljlayl’s work showed also that the stemmer improved the performance, over Khoja, by 24.3% and 19.6% with and without relevance feedback, respectively.

Using the same test collection, TREC2001, Larkey and Connell (2005) extended Aljlayl’s stemmer and proposed a set of light stemming package, named as light1, light2, light3, and light8 stemmers. The stemmers differ only from each others in the prefixes and suffixes they remove, as it will be shown later in this subsection. A final extension to light8, which is called light10, had been also proposed by the same developers. In Larkey and Connell works, all stemmers share the same first step when normalizing Arbic words. The first version, which is light1", removes only the definite articles with their possible preceding particles which consist only from one letter. However, the removal was subjected to a condition that the remaining stem should have more than two letters.

stemmer	Removing from front	Removing from end
Aljalyal	و, ال, وال, كال, ست, سي, ل, ب,ت, ي,لل	ين, ون, ات, ة, ان, ي, هم, هن
light10	ال، وال،لل، بال، كال، فال، و	ها، ان، ات، ون، ين، يه، ية، هـ، ة، ي
Al-Stem	وال, فال, بال, بت, يت, لت, مت, وت, ست, لم, بم, نت, وم, كم, فم, ال, لل, وي, لي,في,وا,فا, لا, با	ات, وا, ون, وه, ان, تي, ته, تم, كم, هم, هن, ها, ية,تك,نا,ين,ية,ة,ه, ي,ا
Chen	وال, بال,فال, كال, ولل, مال, اال, سال, لال, فا, كا, ول, وي, وس, سي, لا, وب, وت, وم, لل, با, و, ب, ل	ها، يه، هم، هم، ن، ما، و، يا، ني، يا, ه,كم, كن, تم, تن,ين,ان, ات, ون, ه, ة, ي, ى

TABLE 3.5: Prefixes and suffixes removed by the Arabic light stemmers

an exception has been made here to the preposition"ل"(equivalent to the English letter L). Extending light1 to light2, Larkey and Connell added the prefix "و" to the set of the prefixes that should be removed from words. The third version of the stemmers, which was light3, extends light2 to remove the suffixes "ه" and "ة". More suffixes were eliminated in the fourth version light8. Eventually, in light10 stemmer the prefix "لل" has been added light stemmer

. Table 3.5 shows all the prefixes and suffixes that have been proposed to be removed by the different light stemmers of Larkey and Connell.

Darwish and Oard (2003b) porposed another light stemmer(Al-Stem) which removes a set consisting of 24 prefixes along with another 21 suffixes. Table 3.5 shows these eliminated suffiexes and prefixes. Their experiment was performed using pre-translation query expansion. In each document, Darwish and Oard first extracted the 20 most expressive terms. Later, Darwish and Oard combined the 10 top-ranked documents into a single *large-document* and then the 20 most descriptive terms in that large-document were selected. The resulting set of 20 terms was then added to the representation of document that was being expanded. Results showed that further work on document expansion is needed (Darwish and Oard, 2003b).

Chen and Gey (2002) proposed a new light stemmer known as Chen's stemmer. Chen and Gey used an English stemmer to stem English words in an English-Arabic parallel corpus. Then, Arabic words are clustered together into a stem category depending on their mapping to English stems in the corpus after being aligned and processed with *GIZA++*, which was described in the previous chapter. The Results showed that the increase in performance was substantial when it was compared with Al-stem.

In their work to evaluate linguistic-based stemming approaches and light stemming, Kadri and Nie (2006) conducted some experiments to compare the two approaches. They used the corpus statistics of TREC 2001 to resolve ambiguity about whether a letter sequence is a prefix or a suffix. Thus, using the corpus statistics they created all the possible stems with their frequencies of occurrences. To stem a word, the developers divide it to its possible stems and the most possible one is chosen based on its statistics in the corpus. Kadri and Nie called this approach the linguistic-based stemmer. For the light stemming approach, Kadri and Nie built a stemmer that truncates the most frequent prefixes and suffixes in the same corpus. Using TREC 2001 and TREC 2002 test collections, results concluded that using linguistic-based stemming produces better results than the light stemming, and that the light stemming "is not the best approach for Arabic IR".

3.3.3 Statistical Approaches to Arabic Stemming

Statistical methods have been also used to find Arabic words stems. The major employed approach involves the use of n-grams methods, in which a word is divided into a number of n characters depending on the used window size of the n-gram. Next, a similarity measure is used to cluster words that have similar n-grams to some degree.

Xu et al. (2002) implemented a stemming approach based on the Buckwalter analyser. In particular, the developers used two techniques: *sure-stem* and *all-stems*. With the *sure-stem* technique, a word is stemmed if it has just one stem. If a particular word does not have a stem, then it will be left as it is. With the *all-stems* technique, a word is converted to all its possible stems probabilistically, with the assumption that all stems are equally probable because there is no training data but later a probabilistic IR model will handle such kind of ambiguity. Results showed an improvement on the recall of over 10% when it is compared with full-word stem. Results also showed that *sure-stem* is somewhat better than *all-stems*, but the improvement is not statistically significant.

Since statistical methods based on clustering words that may not be related to each other, e.g., a bank of a river and a bank for business, Larkey et al. (2002) implemented a statistical approach that was primarily depending on the analysis of the co-occurrence of terms in Arabic text. However, Larkey, Ballesteros and Connell used a bilingual lexicon derived from an online dictionary, so it contains fewer variants. This means that query terms were not matched against the dictionary entries unless they were stemmed. From this note, Larkey,and her team concluded that stemming may be unnecessary with sufficient parallel data.

Mustafa and Al-Radaideh (2004) proposed an approach for searching Arabic text using n-grams. They used bigrams and trigrams the Dice similarity measure was employed for the purpose of discovering inflectional forms of words. In the work of Mustafa and Al-Radaideh 0.6 of a similarity value was considered for determining whether the word is similar or not. Results concluded that the use of infixes in Arabic words because their variants to show low similarity.

Mohamed, et al., (2011) proposed a new technique for Arabic documents retrieval using Wikipedia. The main idea was based on collecting concepts with their synonyms in a dictionary from the downloadable dumped database of the Arabic Wikipedia, in which redirect pages usually represent other different names (abbreviations, synonyms, etc) for concepts (articles). Accordingly, documents, after being tokenized, are processed in term of n-gram and if a particular n-gram matches any of the synonyms of a certain concept, then the term would be substituted by its right concepts, which are demonstrated by their synonyms, for example, in the concept dictionary. Using Arabic TREC-2001 with Arabic queries, results showed that the effect of using such an approach was not better than stemming techniques, but it is hoped that the continuous growth of Wikipedia may result on changing this effectiveness tendency towards concept-based IR.

3.4 Summary

In this chapter, some of the published works on Arabic Information Retrieval have been reviewed. It was shown that many studies have focused on analyzing the morphology.

The main objective was to extract the root of an Arabic word. Experiments using small corpora have shown that root indexing is more effective than both stem-based indexing and word-based indexing. Nevertheless, light stemmers have shown to be more effective than root-based stemming on large and standard test collections. Light stemming approaches differ in the types of prefixes and suffixes they remove and it was shown that light 10 is the best one among them. Statistical approaches to AIR have been also tested so as to make the stemming process independent of target language and trigrams have been reported to be the best gram size for indexing Arabic text.

CHAPTER 4
ARABIC STEMMER

Stemming is the process of merging different forms of the same word that are semantically equal and share the same stem. For IR systems, Stemming is used to improve retrieval effectiveness and to reduce the size of indexing files.

Arabic words have various forms. For example, a noun can have up to 519 different forms, whereas a verb can have up to 2 552(Attia, 2007). To alter words to their root or stem, extra character that append to the word either at the beginning (prefix), middle (infix), or at the end (suffix) have to be removed by *stemming*. For example the word "يشرَبَ" (meaning: he is drinking),"شَراب" (meaning: the drinking) and "شارب" (meaning: the moustache) convert to "شرب" after heavy stemming process completed..

As described in Section 3.2, two major approaches are the most dominant for Arabic stemming. These are light stemming (known also as affix removal stemming) and heavy stemming (morphological analysis stemming).Stemmers usually removes affixes by comparing the specific pattern of the word with a pre-prepared list of affixes. These lists are usually constructed based on the language morphology and statistical analysis of Arabic text (Aljlayl and Frieder, 2002). Using such a fixed list to match the beginning or the end of the word is more effective, but also affects core letters. This can happen in any language, but is a main problem for Arabic, where pronouns conjunctions, prepositions, and particles are attached directly to words. The same letter sequence may also be core characters, and removing such core characters leads to incorrect results.

In this chapter, we examine approaches for proper Arabic stemmer. We compare approaches to normal affix removal, and demonstrate that our technique increases text retrieval effectiveness.

4.1 The Problem of Arabic Stemming Matching

The major goal behind building any Arabic stemmer is to improve search effectiveness, so an IR system could match user's queries with relevant documents. However, users often outline their query terms in many different forms but, they are in fact searching for the same object. Consider two different users who are searching for relevant documents on Arabic language challenges. One might submit a query like ‘تحديات اللغة العربية’ whereas the other might write ‘التحدي في اللغة العربية’, in which the word ‘التحدي’ has been placed instead of the word تحديات in the first query. From that perspective, an IR system must be able to handle two major difficulties that are very related to stemming. First, it should be able to unify all the different forms of any word that have the same meaning or at least derived from the same root. Second, the IR system should have the ability to cluster correctly all the derived forms of a single word and thus, forms of words with different meanings should be clustered into different groups.

For example, consider a collection that contains 7 Arabic documents as illustrated in Figure 4.1. A user might submit a query like 'مقتل' (meaning: the person who was killed) to this collection. With these arguments, if a good stemming algorithm is applied, then only documents D2, D3, D4 and D5 should be retrieved, while documents D6 and D7 should not, as they are irrelevant to the submitted query (They didn’t include any inflectional form of the submitted query). Document D1 should not also be retrieved as it is semantically not relevant to the submitted query, e.g., the document covers a topic related to a fighter battle. This is not the same topic that has been covered in the other documents, which are mainly focused on 'murder and killing'.

If this example is related to the two major problems that were highlighted, then the stemmer could be perfect as it was able to cluster documents that are relevant to the submitted query while on the same time it excludes document D1 since it is irrelevant due to topic semantic.

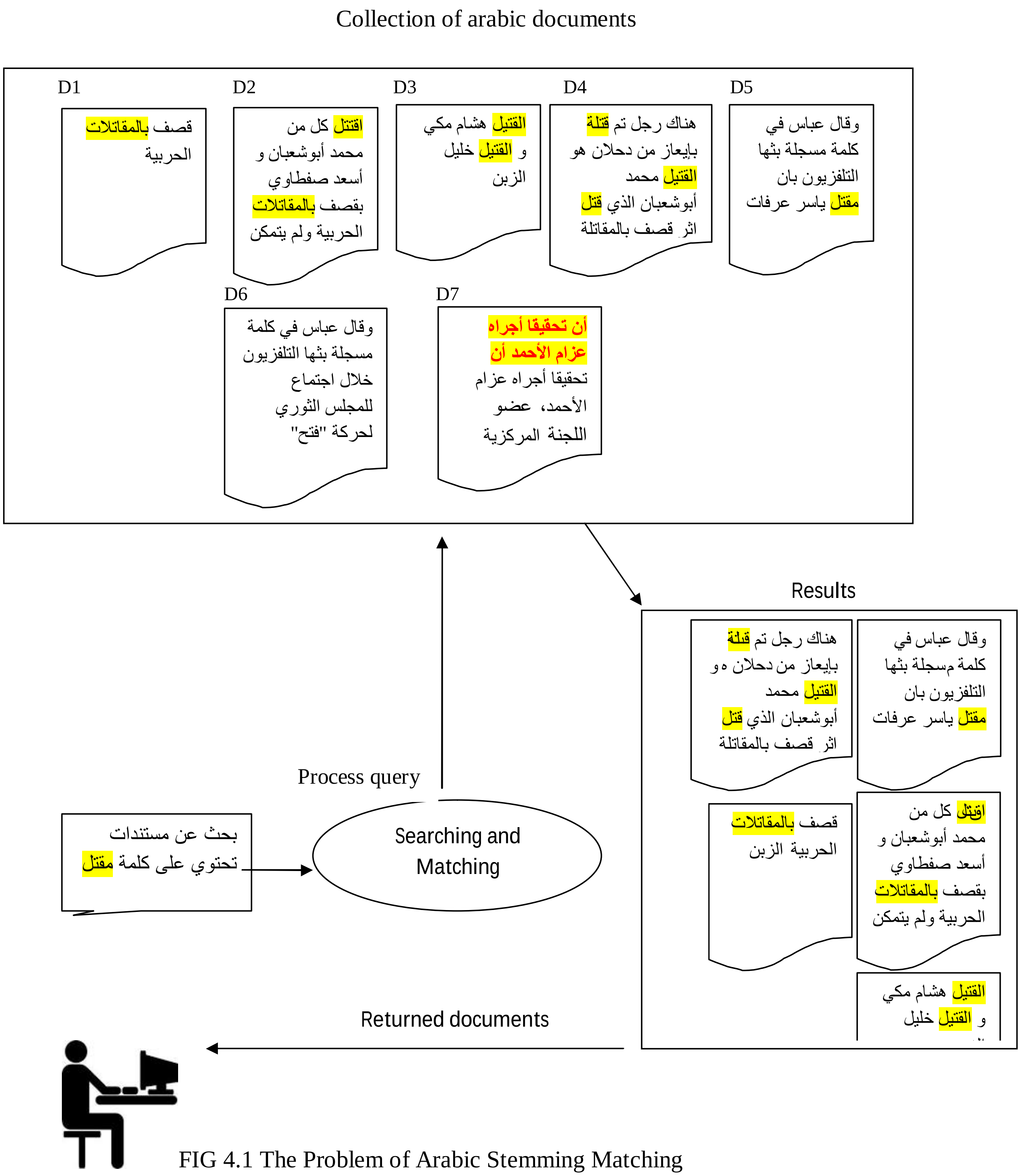

Collection of arabic documents
D1
قصف بالمقاتلات الحربية
D2
اقتتل كل من محمد أبوشعبان و أسعد صفطاوي بقصف بالمقاتلات الحربية ولم يتمكن
D3
القتيل هشام مكي و القتيل خليل الزبن
D4
هناك رجل تم قتلة بإيعاز من دحلان هو القتيل محمد أبوشعبان الذي قتل اثر قصف بالمقاتلة
D5
وقال عباس في كلمة مسجلة بثها التلفزيون بان مقتل ياسر عرفات
D6
وقال عباس في كلمة مسجلة بثها التلفزيون خلال اجتماع للمجلس الثوري لحركة "فتح"
D7
أن تحقيقا أجراه عزام الأحمد أن تحقيقا أجراه عزام الأحمد، عضو اللجنة المركزية
Results
وقال عباس في كلمة مسجلة بثها التلفزيون بان مقتل ياسر عرفات
هناك رجل تم قتلة بإيعاز من دحلان ه و القتيل محمد أبوشعبان الذي قتل اثر قصف بالمقاتلة
اقتتل كل من محمد أبوشعبان و أسعد صفطاوي بقصف بالمقاتلات الحربية ولم يتمكن
قصف بالمقاتلات الحربية الزبن
القتيل هشام مكي و القتيل خليل
Process query
بحث عن مستندات تحتوي على كلمة مقتل
Searching and Matching
Returned documents

FIG 4.1 The Problem of Arabic Stemming Matching

With an on these two problems, especially the first one, a new and effective stemmer is developed in this thesis. However, in order to develop such stemmer, we first evaluate the current major approaches for Arabic stemming. In particular, the best Arabic stemmers in IR were evaluated to identify their major drawbacks and then these drawbacks are mitigated. The next section evaluates the current approaches.

4.2 Evaluation of existing Stemming Approaches

It was previously explained in section 3.1, that two major approaches were proposed to Arabic stemming: the root based stemmers like Khoja and the stemmers which lightly stem words, e.g., Light10 by Larkey. Light10 stemmer has become a widely used in stemming for Arabic documents and has been added to the *Lemur* toolkit[2] and *Lucene* IR system[3].

Each of the two approaches has its drawbacks as it will be described in the nest sub-section.

4.2.1. Root based Stemmer

A root based stemmer aims to pull the essential form for any given word by performing morphological analysis. In other words, for any submitted query a root-based stemmer attempts to extract the root of the word, even if the word is a proper noun, e.g., the proper noun محمد (translation: Mohammed). From that perspective, the morphological analysis always tries to determine some longest matching mechanism in order to put the original word into a certain Arabic pattern. Following this, the affixes are to be

[2] http://www.lemurproject.org/

[3] http://lucene.apache.org/core/index.html

determined so as extract the root that can be placed with the pattern فعل (transliterated as *f-à- l*). After the root is extracted it will be compared with a list of predefined roots in order to determine its correctness. For example, root-based stemmer produce the root, ذهب for the word لنذهبن (meaning: we will surely leave) because antefix: ل, prefix: ن and suffix: ن are removed. Thus, it can be said that the focus of the stemmer is on verbs rather than nouns as it always results in root.

This implicit underlying assumption may hold for the inflectional forms of verbs. For example, it is correct to unify the words يتقاتلون, يتقاتلن, يقتتلن and يقاتلا to a single root قتل (meaning: the person who was killed). However, in contrast, this is a major drawback if the characteristics of the Arabic language are considered because while there are 10,000 roots, only about 1200 are still in use in MSA (Hegazi and El-Sharkawi, 1985) .Thus, attempting to put every Arabic word into a root form may result in a very different meaning for the original word. Let’s consider the following words: السودان, طفيليات, باراك أوباما and الستائر (meanings respectively: the republic of the Sudan, parasites, the US leader Barak and curtain). These words may be stemmed to a very different meanings or very chaotic words. For instance, the stems of these words in using khoja stemmer results in (سود, طفل, مهرج, برا وبا and ستر) whereas it results in (سود, طفل, مهرج, برك وبم and ستر) using Buckwalter analyzer. to The main theme of all these words is that they are nouns not verbs.

It is true that some morphological analyzers are able to identify the type of the part-of-speech of the words, e.g., Buckwalter analyzer, but yet the use of POS approaches is inefficient during ad-hoc retrieval as the process may result in a very slow retrieval and an extremely long process for indexing any document. This is why always approaches attempt to provide a quick technique for performing the stemming process.

In fact, some studies Al-Shammari and Lin (2008a, 2008b) proposed a novel algorithm for stemming Arabic words, known as Educated Text Stemmer (ETS), by the use of syntactic knowledge and morphology. The hypothesis of the study is that in Arabic there are some useful stopwords, which can be used to identify verbs and nouns. Accordingly, the Khoja stemmer was apllied to stem verbs while light stemming is applied to nouns. but, it was concluded that they will not be effective for ad-hoc retrieval due to the reasons illustrated above.

Bearing in mind the discussed arguments above, the major drawback of the root-based stemmers is the over-stemming problem in which a stemmer may erroneously cluster words with different meanings. This is very evident when words like طفيليات (meaning: parasites) and أطفال are considered. Both words will be stemmed to the root طفل. Such a problem results in over-stemming difficulty, in which words with very different meanings are grouped into the same stem.

Another major problem as discussed above is the fact that it is not always correct to produce the root of proper nouns, or nouns in general. For instance, a word like باراك أوباما should not be stemmed to برك وبم as Khoja stemmer usually does.

Let’s also consider the proper noun هشام which is usually stemmed to the word هشم (meaning: to broke something). It is very evident that the meaning is totally different. Sometimes even the root-based stemmer may result in a very chaotic and ambiguous word. For example, consider the stem كيى which is results from stemming the proper noun مكي. The stem makes the word to lose its meaning as it is simply not an Arabic word. In fact it becomes a Parisian word.

Another example for nouns (that are not proper nouns) is the word مقاتلات (meaning: fighters). The word should not be stemmed to the stem قتل as both Khoja and Buckwalter did since the stem will produce many irrelevant documents. Table 4.1 shows more examples to incorrect stems that may result from using root-based approaches. Words in the table were stemmed using Khoja.

The word	Stemmed word(khoja)
القتيل	قتل
هشام	هشم
مكي	كيى
خليل	خلل
الزين	زون

TABLE 4.1: the stemmed word using khoja algorithm in the document two

Such problems may bias result list dramatically as the stem may result in increasing and/or decreasing the weight of terms dramatically. If the word is erroneously grouped with another word, then the term frequencies of this word in documents will be increased (term frequency component in the weight formula). At the same time, the importance of the word will be decreased as the document frequency component will increase too due to wrong grouping of the words. If we add to this problem the fact the relevant documents, which its stemmed word was grouped with another word, will not be retrieved, then we can conclude that root-based stemmer is not a good option for indexing Arabic word.

To illustrate the impact of the stemming problem on weighting, let’s to briefly discuss it by referring to our example in section 4.1, in particular Figure 4.1. Table 4.2 shows briefly the term frequencies of the word that present in the query ‘مقتل’ in the document collection. The documents and the query in this example were stemmed using Khoja stemmer.

Document no	The word	Khoja Stemmer results	Term frequency
D1	بالمقاتلات	قتل	1
D2	اقتتل	قتل	3
	بالمقاتلات	قتل	
	القاتل	قتل	
D3	القتيل	قتل	2
	القتيل	قتل	
D4	قتلة	قتل	3
	القتيل	قتل	
	قتل	قتل	
D5	مقتل	قتل	1
Query	مقتل	قتل	1

Table 4.2: Stemming the documents in the example collection using Khoja stemmer

It is obvious that all words have been stemmed to a single root قتل although the word المقاتلات in documents D1 and D2 should not be stemmed to this word. However, the impact of this over-stemming on the weight is that the term frequency in document D2 increases by 1 and thus, it results in a TF component of weight similar to its peer in document D4 although document D4 is more relevant. On the other side, the word المقاتلات in document D1 results in increasing the document frequency component of the word in the query and thus, its effect on the final score of documents will be suppressed if the query contains more than one word.

4.2.2. Light Stemmers

The underlying idea behind light stemming is to avoid grouping words with different meanings together as much as possible. To achieve this goal light stemming approaches attempt to lightly stem words and thus, the underlying assumption is to avoid performing deep morphological analysis. From this perspective, several approaches for stemming Arabic texts lightly were proposed. The most, and yet the famous and elegant, ones are the Larkey's stemmers.

Larkey proposed several light stemmers (*light1, light2, light3, light8* and *light10*) as it was discussed earlier. Currently the most widely used one of them is Light 10 which has been identified as the best effective stemmer for indexing Arabic documents when it was compared to other stemmers (Larkey et al., 2007) and it was included in many widely used IR systems such as Lucene and Lemur, as it was shown earlier.

Light 10 as illustrated in section 3.3.2 is based on some heuristics that are able to eliminate some strippable prefixes and suffixes from words. The robust feature of light stemming approaches in general and light 10 in particular, is that the stemmer minimizes the impact of the over-stemming problem. Since only few prefixes and suffixes are removed then words will not lose their meanings and thus, they are not going to be clustered together. Consider the word المقاتلاتin the explanatory example above. If a root-based stemmer is employed for stemming the word, then the resulted stem is قتل (meaning: to kill), which has a different meaning to the word fighters, resulting in the over-weighting problem. In contrast, if the word is lightly stemmed, then only some prefixes and suffixes will be removed. For example, if light 10 stemmer is used to stem the word, then the resulted stem is مقاتل (as only the prefix ال and the suffix ات will be eliminated). It can be noted that both the word and the stem have the same meanings. This is a very strong feature for the light-stemming approaches. Another example is the name of the American President باراك أوباما (Barak Obama). Using light 10 the same name will be preserved as neither strippable prefixes nor strippable suffixes are present in that name. Table 4.3 shows more examples for the success of the light 10 stemmer in stemming Arabic words correctly.

The word	Stemmed word(ligth10)
القتيل	قتيل
هشام	هشام
مكي	مكي
خليل	خليل
الزين	زين

TABLE 4.3 the stemmed word using light 10 algorithm in the document two

In spite of the light stemming characteristic of the light 10 stemmer but, some major drawbacks can be identified.

The major drawback in light stemming techniques in general and light 10 in particular is the under-stemming problem, in which words with the same meanings may be clustered into different groups. Thus, the stemmer may result in low recall as many relevant documents will not be retrieved since the stemmer did not succeed to cluster semantically similar words together.

Table 4.4 shows such a problem and its impact on term frequency component by referring to the document collection in our explanatory example. When light 10 stemmer was used, the stemmer fails to group the words اقتتل (meaning: who is Fought) and القاتل

(meaning: the killer) in D2 although both words are semantically similar. This is because the algorithm of the stemmer uses two different heuristics for stemming the two words. The same problem also appears in D4, in which the stemmer fails also to cluster the words القتيل (meaning: the dead person), قتلة (meaning: killers) and قتل (meaning: to kill), resulting in low recall.

As in root-based approaches, the problem of under-stemming could have a major impact on weighting of terms. On one hand, it reduces the term frequency of terms and, thus, resulting in low weight for the term (and consequently low score for document in which that term appears). Such a drawback appears in document D4 when the words the words القتيل, قتلة and قتل have been stemmed to different stems, instead of resulting in a term frequency equals to 3 as the words are semantically similar.

Document no	The word	light10 stemmer	Term frequency
D1	بالمقاتلات	مقاتل	1
D2	اقتتل	اقتتل	1
	بالمقاتلات	مقاتل	1
	القاتل	قاتل	1
D3	القتيل	قتيل	2
	القتيل	قتيل	
D4	قتلة	قتل	1
	القتيل	قتيل	1
	قتل	قتل	1
D5	مقتل	مقتل	1

Table 4.4 the stemmed word using the Larkey stemmer in the corpus

On the other hand, the under-stemming problem may increase the weight of terms due to reduction in the document frequency component. Since documents that contain similar terms but in different inflectional forms may be grouped differently then the document frequency of each form will be reduced and thus the importance of that inflected term will be increased.

4.3 The overview of Proposed Stemmer

With an eye on the major drawbacks of both root-based and light-stemming approaches, a new stemming algorithm has been proposed in this thesis. The major aim is to mitigate and minimize the effects of the main problems on techniques.

First, we started with raising a question, that is: on which approach does the new stemmer is to be built? On one hand it was shown that root-based approaches increase the recall but, they usually result in many irrelevant documents. On the other hand, light stemming is elegant and it often provides more relevant documents. On the same time many of the irrelevant documents will not be retrieved, however, it usually results in lower recall.

It was also shown also that three major, and yet severe, problems are related to root-based stemmers. These are: the over-stemming problem, in which words with different meanings may erroneously be grouped together as in المقاتلاتand يقتتلون, which are both be stemmed to a single root; the production of the root for proper nouns such as باراك أوباما and طفيليات; and biased weight of terms, and thus for scores of documents too, since the importance of terms will be decreased while its term frequencies will be increased.

From that perspective, it would be a wise decision to avoid using root-based approaches for indexing documents in any rich and inflectional language as in Arabic. A similar conclusion was also stated by many researchers in IR (Larkey et al., 2007)

It was also shown that light stemming has a very robust characteristic, which is its ability to avoid stemming of proper nouns as much as possible while on the same time it attempts to preserve the meaning of words regardless of its POS type. In our analysis to the stemming problem, it was found that the major reason for the success of light 10 stemmer is the fact that a significant portion of Arabic words are grammatical or/and proper nouns. Bearing this good feature of the Arabic language in mind, beside the drawbacks of the root-based stemmers, our new proposed stemmer, which has been called as the *EXTENDED-10*, has been built on the top of the light 10 stemmer. The next section describes the approach and the reason behind its developing.

4.3.1. The EXTENDED-10

As in all light stemming methods, the proposed approach depends totally on removing some prefixes and suffixes but under certain conditions that will be controlled by some heuristic rules. In that context, the problem of stemming will be changed to which prefixes and suffixes should be stripped from Arabic words?

To answer this question, we firstly did a very deep analysis to identify the types of problems that may occur when using the light 10 stemmer. Several snippet codes were written for this purpose and was concluded that beside the under-stemming problem, light 10 still have some drawbacks.

First, in our analysis, it was noticed that most strippable prefixes and suffixes in light 10 are mainly focused around nouns. For instance, the prefixes of light 10, which are ال، وال، بال، كال، فال, و, لل, وبال are all determined for the purpose of removing these prefixes from grammatical nouns and proper nouns. Thus, in a word like السودان only the define article will be removed while in the name of the American President باراك أوباما the name will be preserved as it was submitted. Although the argument here indicates that this is a good feature can be accounted to light 10 stemmer but, in contrast it is the major reason for the under-stemming problem. This is because Arabic verbs are partially ignored when removing such prefixes, for example. Consider the verbs تتجادل and جادل (meanings respectively: arguing and argued). Using light 10, verbs will not be stemmed to the same group as there is no prefix to be removed from their beginnings since removal of such a prefixes in light 10 are focused on grammatical and proper nouns. Another example for the ignorance of verbs in light 10 is the prefix فل, which is usually used to emphasize doing the action (verb in this case). For example, in a word like فليكتب (meaning: you should write), which consists of the verb كتب and the prefix فل, the light 10 stemmer will maintain the word as it appears during indexing and thus, other inflectional verbs, e.g., يكتب, كتبا, will not be grouped with فليكتب. Thus, one of the major aims of the proposed Extended-10 is to consider some of the neglected prefixes and suffixes in light-10, especially for verbs. Note that the same arguments also apply for suffixes that always occur with verbs. For instance, the absent pronouns in Arabic like كم and هم (meanings, respectively: your, their) are not included in light 10 and thus a word like قاتلوكم (meaning: they are kill you) will be indexed and stemmed separately from the word قاتل, for example, although both words have the same semantics.

Second, when we tackle the light 10 behaviour, it was found that there are many prefixes and suffixes related to nouns that were not included in the stemmer although it was shown that Arabic nouns represented a considerable part of the words in the language. This is not a trivial notice as the ignorance of some of the major prefixes and suffixes that are related to nouns may easily cause the IR system to miss the documents.

For example, the attached preposition ل(equivalent to the English letter L) to nouns is simply neglected in light 10 and thus, a word like لدرجة (meaning: to the degree) will be preserved although the attached preposition should be eliminated so as to be grouped with the original word درجة. Another example for such a preposition that is not included in light 10 is the preposition ب (equivalent to the English letter B), e.g., باسم (meaning: in the name), which will not be stemmed using light 10.

For these two reasons and performing a deep analysis, our proposed Extended-10 stemmer adds more prefixes and suffixes, beside those described by light 10, so as to account for verbs as well as nouns. The new added prefixes are وبال, ولل, فل, ول,وب, فب, تت, ب, ل and the new extended suffixes are وا, تي, هما, نا, هم, ت . Table 4.5 illustrates the final sets of the strippable prefixes and suffixes in the proposed Extended-10 stemmer.

Stemmer type	Removing from front(prefix)	Removing from end(suffix)
EXTEND10	ال، وال، بال، كال، فال, و, لل, وبال, ولل, فل, ول,وب, فب, تت, ب, ل	ها، ان، ات، ون، ين، يه، ية، هـ، ة، ي, وا, تي, هما, نا, هم, ت

TABLE 4.5: Strippable strings removed in EXTEND10 stemmer.

However, one might ask if we remove the prefixes like ب and ل from the beginning of a word, for example and as the Extended10 stemmer proposes, how we could handle words like وليد (meaning: a proper noun) or لقمان, which is another proper noun. Note that the light 10 stemmer stem these words as ليد and لقم, respectively. In other words, light 10 stemmer avoids removal of letters like ل since many proper nouns begin with this letter, so how our proposed stemmer handle such a situation? The answer for this question is the use of the proposed heuristic rules which remove only certain prefixes and suffixes under certain conditions. The next section describes our proposed algorithm for handling these heuristic rules.

4.3.2. The EXTENDED-10 Algorithm

The underlying assumption behind our proposed algorithm is that since the majority of Arabic words are derived from tri-literal or quad-literal roots then any resulted stem for a word should not be less than 3 letters and should not exceed 4 letters, too but under certain criteria. This is totally different from the assumption behind light 10, which stated that stemmed words may be consisting of 2 or 3 letters under certain conditions. For instance, a word like وجه (meaning: face) is stemmed to وج, which is meaningless word consisting of 2 letters, when using light 10 stemmer, whereas a word like لقمان will be stemmed to لقم, which is a chaotic word consisting of three letters, but it may erroneously grouped with the verb لقم (meaning: stuff).

Examples include, but are not limited to, also words like صحون (meaning: the plates), ساعة (meaning: clock or hour) and السودان (meaning: the Sudan). Using light 10 stemmer, these are three words will be to صح, ساع and سود, respectively.

To avoid such problems the proposed stemmer changes the rule while considering words characteristics. Our stemmer attempts to maintain words consisting of three letters and four letters as they appear, e.g., وجه and السودانin the examples above will be stemmed as وجه andسودان, in which only the definite article ال was removed.

On the algorithm the idea is also extended to include verbs. In the light 10 stemmer a verb like تتنافسون (meaning: they are to compete for something) will be stemmed to تتنافس,

which would result in an under-stemming problem as the new stem will not be clustered with the original root نافس. Thus, our proposed stemmer attempts to mitigate such types of problems. The next section describes the algorithm in terms of steps.

Algorithm

The algorithm consists of the following steps:

Step 1, the algorithm begins with the removal of the diacritical marks that are written below and above the Arabic letters. Thus, the set of the diacritics: ً, ُ, ٌ, ٍ, ِ, َ and ْ is removed. Note that the diacritic "*Shaddah*" (ّ) is not strippable in the algorithm since it is often placed above the letter to indicate that the letter is duplicated. Hence, in the algorithm the *shaddah* is replaced by duplicating the letter above which it occurs. Also it removes the diacritical marks called the Kasheeda, known also as Tatweel.

Step 2: in the second step the algorithm perform some normalization so as to unify the different orthographic forms for certain letters. In particular, the algorithm changes the letters (أ) "HAMZA-above-ALIF", (إ) "HAMZA-under-ALIF" and "ALIF-Maddah" (آ) to a plain "ALIF" (ا). Thus, the verb أشرب, which means drink and the plural noun أجراس (meaning: bells), for examples, can be written as اشرب and اجراس with bare ALIF. Similarly, the "TAA-MARBOOEH" (ة) is altered to the letter "HAA" (ه) as in the word شريفة, which will be changed to شريفه. Additionally, the sequence of "ALIF-MAKSORA"(ى) and "HAMZA" (ء) is replaced with (ئ) and the sequence of "YAA" (ي) and (ء) is altered to (ئ). Similarly, the sequence of "YAA" (ي) and (ء) is modified to (ئ).

Step 3, in step no. 3, the connector و (pronounced as WAW and it means and), the letter ب (equivalent to the English letter B) and ل (equivalent to the English letter L) are removed if and only if the remainder of the word is greater than 3. For examples, the words وجد (meaning: who is found something), بسم (meaning: in the name) and لساعة (meaning: to one hour) are changed to وجد, in which the letter و, ب, in which no letter has been removed as it contains only three letters. لساعة, in which the ل is only eliminated.

```
Let T denote the set of characters of the Arabic word
Let $K_i$ denote the position of letter i in term T
Let Stem denote the term after stemming in each step
Let S denote the set of suffixes
Let P denote the set of prefixes
Let n to be the total number of letters in the Arabic word
**Step 1**: Remove any diacritic in T,
**Step 2**: Normalize أ, إ, آ in $K_1$ of T to ا (plain ALIF)
            Normalize ى in $K_n$ of T to ي
            Replace the sequence of ى in $K_{n-1}$ and ء in $L_n$ to ئ
           Replace the sequence of ي in $K_{n-1}$ and ء in $L_n$ to ئ
           Normalize ه in $L_n$ of T to ة
**Step3**: If the length of T is greater than or equal to 3 characters then, Remove the prefix WAW "و",
           BEH "ب" and LEEM "ل" in position $K_1$.
**Step 4**: For all variations of P do,
Find the specific prefix $P_i$ in T
If $P_i$ matches in T
$P_i$ = $P_i$ + Characters in T ahead of $P_i$
Stem = T - $P_i$
**Step 5**: If the length of Stem is greater than or equal to 3 letters then,
For all variations of S, obtain the most frequent suffix,
Match the region of $S_i$ to longest suffix in Stem
If the length of (Stem -$S_i$) greater than or equal to 4 characters then,
```

FIG 4.2: The proposed algorithm of the Extended 10 stemmer

Step 4**:** in the fourth step, the algorithm strips out the prefixes. This is can be achieved by firstly matching the word with the longest prefixes listed in our predefined set and. This means that if there are two prefixes with different length, the longest one is chosen. If any matched prefix is encountered, the algorithm remove that prefix from the input word if and only if the retained stem contains 3 letters or more, otherwise, the algorithm didn’t eliminate the prefix.

In ***step 5***, the algorithm focuses on suffixes. As in step 4, it matched the longest suffixes first. If the algorithm fails to locate the longest suffix, then it considers shorter ones. When there is a part of the term under stemming matches a suffix, the algorithm removes that suffix. Before removing the suffix the algorithm checks the length of the target stem. If there are fewer than 4 letters, then it leaves the entire term; otherwise, the algorithm returns the stemmed term. Figure 4.2 describes the complete set of the steps of the proposed algorithm.

The word	The meaning in the English	EXTEND10	Light10 (Larky)	Root-Extraction Stemmer (Khoja)
الساعة	The hour	ساعة	ساع	سوع
أعلنت	They was announced	اعلن	اعلنت	علن
شركة	The company	شركة	شرك	شرك
للضمان	For the grantee	ضمان	ضم	ضمن
بالتالي	The next	تالي	تال	تلا
لدرجة	To the degree	درجة	درج	درج
أعمالهم	Their works	اعمال	اعمالهم	عمل
البطون	The bellies	بطون	بط	بطن
ليوم	For a day	يوم	ليوم	لوم

TABLE4.6. A comparison between the three stemmers

To this end, let’s now see how the proposed approach can affect the stemming process. Assume that we would like to stem the words that are listed in Table 4.4. The words were stemmed using three different stemmers: the proposed Extended 10 stemmer, The light 10 stemmer and Khoja analyzer.

As it can be seen in the table, the proposed stemmer achieved more reasonable results. For example, it is very evident that the proposed stemmer correctly stems all the words in the Table, e.g., لدرجة has been stemmed to the stem درجة whereas the word للضمان has been stemmed to ضمان. On the other hand, both light 10 and Khoja stemmers fail to provide good stems for the majority of words. For instances, light 10 stemmed the word ضمان to ضم (meaning: combine), which has a different sense from the original meaning and thus, resulting in clustering two words that are not semantically similar. Another example is the word أعلنت, which has been stemmed to علن using Khoja stemmer. The resulting word is too ambiguous.

4.4 Summary

The main purpose of using a stemming process is to discover the representative indexing forms of words. Since the process is language-dependent and Arabic is a highly inflectional language, the use of a good stemming algorithm is expected to have a very positive impact on retrieval effectiveness of IR system.

In this chapter, it was shown that yet there is no standard approach to Arabic stemming but, the approaches can be classified into two main classes, these are the root-based and light-based techniques. Each of which has its pros and cons. Root-based stemmers usually results in higher recall but, it may erroneously cluster two semantically different words into the same group, resulting in over-stemming. Light stemmers preserve the meaning of words but, it may fail to cluster words that have the same meanings. In the chapter, both approaches were evaluated with example to determine their weaknesses and strengths and it was concluded that light stemmer is better than using root-based approaches as it results in more accurate relevant documents. The same result was also concluded in many other studies.

Accordingly, in the chapter a new stemming technique has been proposed and it was built on the top of the light 10 stemmer. The new stemmer, which was called as the Extended-10, attempts to mitigate those drawbacks that result from the use of light-10. This was achieved by extending the set of the removable prefixes and suffixes in light 10. However, the decision of whether to remove an antefix depends on a well analyzed algorithm stated in the chapter. Initial examples show that the proposed approach is more accurate. The next chapter evaluates the proposed approach.

CHAPTER 5

EVALUATION

This thesis challenges to improve the search effectiveness of Arabic information retrieval through building good Arabic stemmer. An AIR system can match user's queries with relevant documents. Users applied their query keyword in a lot of different formats but they are searching for the same thing. The good stemming algorithms that are used in search engine can capable solve two major problems.

First it should be capable to translate all different forms of the word that have the same meaning to a standard form.

Second it should be capable to cluster all same forms of the word that have the different meaning to different group based on their semantic. . This chapter shows how such a proposed stemming algorithm in AIR system was evaluated. In particular, the chapter will evaluate the newly developed approach, which was shown in the design chapter. This evaluation aims to show the significant impact of using these proposed approaches on Arabic retrieval effectiveness and determine if they provide a significant improvement over some well-established baselines. To achieve this, the experiment was carried out using the test collection and the previously produced query. The chapter presents initially the general work that was set. Such work includes before the indexing the normalization in texts, stemming .These details of the test environment are presented in section 5.1. Section 5.2 is devoted to experiments and results.

5.1 Experimental Setup and Test Environment

The test environment of the experiments conducted had been created. First, a new test collection was created. The document collection was collected from the internet. In particular, a web crawler[4] (*WebReaper*) was run against some website to collect data in computer science. The only constraint is the availability of computer science data. One gigabyte of raw data was collected in this way. Following this step, an HTML parser *Jericho* [5] was used to clean the data. Through this process, only text is preserved and thus, figures, odd symbols, and pages headers are removed. At the end, a test collection containing 1734 documents was extracted.

Query #	Query	Counterpart in English
Q01	تحليل وتصميم النظم	System analysis and design
Q02	عرف امن المعلومات	Define the information security
Q03	التشفير بالمفتاح العام	encryption using public key
Q04	المسجلات في الاسمبلي	registers in assembly
Q05	ماهي الشبكات اللاسلكية	What is the wireless network

TABLE5.1: Examples of some sample queries (Q01-Q05) in the created query set.

Next, a new set of 15 queries on common computer was created. The query set itself was created by asking 10 students at the College of Computer Science and Information Technology, Sudan University of Science and Technology, to submit a sample of 5 queries. This randomization is important to build representative test queries. There was

[4] http://www.webreaper.net/

[5] *http://jericho.htmlparser.net/docs/index.html*

no constraint on the query lengths. The language in queries was Arabic. All queries were put together and duplicates were removed and a set of 30 queries was chosen to represents as test queries for the experiments conducted in this thesis.

Queries were numbered for referencing purposes. Table 5.1 shows some examples for the test queries.

To create a relevance judgments set it is important firstly to specify the retrieval task in terms of the application's goals. In particular, the kind of the required judgments is often influenced by the type of the required retrieval task. It was previously shown that binary relevance is the most dominant retrieval task in the different editions of experiments conducted, e.g., TREC. Accordingly, it is determined to conduct the assessment on such retrieval task. Thus, two-point scales (1 for relevant documents and 0 for irrelevant documents) were determined. In a descending order, these two points are as follow:

[1] Relevant document: The document discusses some or all the themes of the topic of a query. Thus, if the document has information that can be beneficial on writing a report on the query topic, then the document can be considered as relevant. The same criterion is being used also by the TREC and CLEF forums.

[0] Irrelevant document: The document can be considered as irrelevant to the submit query if it does not contain any information about the topic of that query.

Thus, using these conditions, the top 10 retrieved documents, for each query, by all algorithms in the experiments, were collected. Duplicates were eliminated and the final list is presented to a group of assessors, who are 3 M.Sc. students in computer science, so as to determine its relevance. It was described in the review chapter such a process is known as pooling and it is widely used in common IR conferences.

Following these steps, an IR system was used to index the document collection. After creating the index of some statistics, e.g., number of Arabic documents and average number of words/document, about the corpus were extracted. Table 5.2 shows the Statistics of the test collection

Description	Total
Number of documents	1734
Number of words	4,683,724
Number of distinct words	162,032
Average number of words per document	3.7

TABLE 5.2: Statistics of the test collection, Figures are computed without stemming.

5.1.1 Used IR System

In all experiments, the Lucene IR system6 was used. Lucene is an experimental information retrieval system that has being widely used in previous editions of the CLEF, NTCIR and TREC joint evaluation experiments. Apache Software Foundation7 describes Lucene as a high-performance search engine with many full-featured libraries to process and manipulate texts. Lucence also has the ability to index and retrieve files in different Unicode encodings. It is written entirely in Java with many powerful query types.

6 http://lucene.apache.org/core/index.html

7 http://www.apache.com

The size of index in Lucene is approximately 20-30% compared to the size of text to be indexed. Lucene also supports the BM25 retrieval model, which is used in our experiments. It is being concluded that BM25 is one of the best models for retrieval (Manning et al., 2008). The Lucene has a diagnostic tool known as Luke8 that is able to access indices that are being created by Lucene. Through Luke, it is possible to: browse documents; display frequent terms; analyze search results and optimize the index. Thus, using both Lucene and Luke, all documents in the corpus were indexed and analyzed, respectively. Thus, using the Lucene, an index for the document collection was created. During the indexing process, Arabic documents were normalized and stemmed. This is what next sections will describe.

5.1.2 Normalization

Normalization begins with processing and eliminating the kasheeda (see section 3.2.1 in the Arabic IR chapter). Next, diacritical marks were also removed. Following this, a letter normalization process for some letters was performed so as to unify their orthographical forms. In particular, the letter normalization that had been performed for Arabic words in documents includes:

- Replacing the letters ALIF HAMZA with its forms: (أ،إ) and MADDA (آ) with bare ALIF (ا);
- Altering the final un-dotted YAA (ى) with dotted YAA (ي);
- Replacing the final TAA MARBOOTA (ة) with HAA (ه); and
- Modifying the sequence ىء with ئ.
- Modifying the sequence يء with ئي.

8 http://www.getopt.org/luke

5.1.3 Stopwords Removals

As in other languages, Arabic also contains functional words (or stop words), which do not carry a particular and useful meaning for IR. A stopword list in Arabic includes some words translated from English stopword lists, such as prepositions, pronouns particles, normal and demonstrative prepositions, e.g."لكن"(meaning: but). Thus, during the indexing process, stopwords were removed from the documents. Note that the Arabic stopwords list that was used in our implementation is a modified version from the one included in the Lucene IR system.

5.1.4 The Employed retrieval model

The retrieval model which was used in the implementation so as to predict the relevance scores of the retrieved documents with regards to a query is an extended version of the probabilistic model BM 25, which was described in section 2.2.2.2.2. This extended model, which was proposed by Robertson, et.al, 2004, is based on refraining from doing linear combination of scores obtained from scoring every field in documents. Instead, the proposed alternative is to calculate a single score for the linear combination of term frequencies (and also document frequencies) of terms in the different fields, but weighted by the corresponding weighted fields. The scoring function in this way is applied only once.

Thus, in the implementation the relevance scores of documents are computed for all fields together and the default values for K1 and b were 1.2 and 0.75, respectively and the used formula was:

$$BM25\,(d, q) = \sum_{t \in q} \left[log \frac{N - f_t + 0.5}{f_t + 0.5} \right] . \left[\frac{(k_1 + 1) f_{d,t}}{k_1 ((1-b) + b \frac{|d|}{avgdl}) + f_{d,t})} \right] . \left[\frac{(k_3 + 1) f_{q,t}}{k_3 + f_{q,t}} \right] \qquad (5.1)$$

The details of the used formula are provided in section 2.2.2.2.2

5.2 Experiment and results

This section reports the results of the experiments that were conducted to test the effectiveness of the techniques shown in the design chapter. As previously described in that chapter, a new developed stemmer, called as Extended-10, has been developed. For the purpose of the experiments two studies/experiments, called as study I and study II, have been conducted. The first experiment compares the no-stemming approach, in which the base/surface words in documents are used, with the light10 stemmer. Study I will represents as a baseline for the second experiment, which evaluates the proposed Extended-10 stemmer and compares it to both the no-stemming approach and light10.

5.2.1 Study I:

Aims

Study I seeks to evaluate the use of light-10 to the use of no-stemming approach. Thus, it investigates whether light stemming approach will have a positive impact on the retrieval of Arabic documents. In particular the study investigates the impact on the top ranked documents as the employed measure for evaluation is the MAP measure.

Methodology

In the study, documents in the test collection were firstly normalized. This includes the steps which were described in section 5.1.2, e.g., removal of diacritical marks. Following this, all documents were located into a single pool index using the Lucene information retrieval system. Throughout the indexing process texts in documents were tokenized on both white spaces and punctuation marks. Hyphenations were not considered, meaning that if the word lies at the end of a line while its remainder placed in the next line, the word will be considered as two separate words, e.g., nour aldein

In the study, two runs were conducted, named as *no-stemming* and *Light10* stemming. In the no stemming run, all extracted words were not stemmed at all and only their surface forms were considered. For instance, a word like مقاتلون (meaning: gladiators OR Fighters) will be kept as it appears. The standard analyzer of the Lucene was used for this process. The standard analyzer didn't do any type of stemming and it just tokenizes the text on white spaces and punctuations only. In the methodology of the second run, Light10, all extracted words were stemmed using light 10 stemmer. Light 10 is provided as a library in Lucene. For instances, words like مقاتلة ،مقاتلات and مقاتلون were stemmed to a single stem, which was مقاتل. Thus, two different indices were created.

In the study, the query set was submitted to the two runs separately. The queries were submitted one by one. In the first run, the first index was used and thus, each query, after being analyzed using the standard stemmer, was submitted to this index. In the second run, the queries were submitted to the second index and they were stemmed using light 10. At each run, the top 10 documents for each query were extracted and their relevance judgments were evaluated using the pooling mechanism, which was described in the review chapter. Using such a method the precision was computed to evaluate the performance of each query. Next, the precision values obtained by all queries were averaged to compute the mean average precision (MAP).

Results and Discussion

Figure 5.1 shows the results of the light stemming in terms of mean average precision curve, compared to the no-stemming run. Table 5.3 lists the corresponding results in tabular form.

Query collection	Measures	NO-Stemming	Light stemming
@ 5 topics	MAP	0.55	0.86
@ 10topics	MAP	0.56	0.78
@ 15 topics	MAP	0.58	0.75

TABLE 5.3: Shows the results of no-stemming approach, compared to those obtained by ligh-10

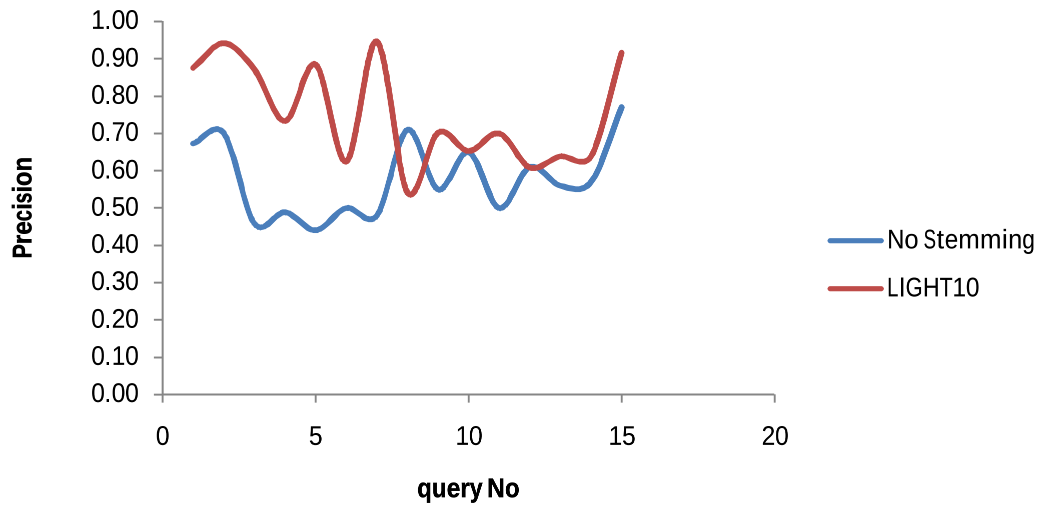


FIG5.1: Retrieval effectiveness, in terms of mean average precision, of no-stemming compared to light10

In the figure, it is obvious that in the majority of the submitted queries, light10 performance outperforms the performance when there was no stemming technique was used. This means that the use of light 10 have a positive impact on retrieval effectiveness even for languages with rich morphology as Arabic. However, the same results were also concluded by previous studies. It is clear that the major reason for the better performance of light 10 is that many words with the same stem will be clustered together unlike in the use of the surface words in which words will not be grouped into a single stem although they are similar in their meanings and in their root as least, e.g., مقاتلون and مقاتل.

The difference in the performance in the first 8 queries, approximately, is large while is it not (meaning the difference between the two runs was relatively small) is some other queries especially those from query 12 to query 17, for examples. We tackled the reason behind the variation in the resulting retrieval lists by each run and it was found that the large difference between the two runs was mainly caused by the fact that light10 stems many documents lightly without losing its meanings, while the surface word approach only attempts to retrieve the exact words in queries and thus, many documents were not retrieved because they are easily missed. However, this is not the case in the second set of queries, in which the difference in performance between the two runs was small. In particular, the queries after 12 were containing words that have no much prefixes and suffixes to be removed. For example, in the query no 11, which says 'الفرق بين الكائن والفئة' (meaning: difference between class and object), no much strippable prefixes and suffixes to be removed via light 10. The same criterion also holds for the remainder of the queries when they were tackled. Thus, only small difference between the two runs (no-stemming and ligh10) was occurred.

In the figure, it is also noted that there is an exceptional query in which the performance of the no-stemming approach is better than the performance of light 10. This phenomenon was tackled and it was found that the reason behind this exception was the query words. In particular, the query was requesting documents in database. It says ‘ مقدمة في قاعدة البيانات’. While no-stemming approach retrieved very relevant documents, light 10 stemmers stems the word البيانات to بيان by removing the letters ات and, thus, the new stemmed word loses it meaning completely and consequently many irrelevant documents were retrieved.

Nevertheless, it can be concluded that the use of stemming in general and light 10 in particular, it beneficial to retrieval performance as it increases both the precision and the recall .

5.2.2 Study II:

Study II was main concerned with testing the impact of the proposed stemmer. In that context, we would like to check whether the Extneded-10 has a positive impact on Arabic retrieval.

Aims

The main aim of study II is to evaluate if the use of the proposed Extended-10 method can have a significant effect on Arabic retrieval performance. It also compares whether the proposed stemming is better than the best known light stemmer, which is light 10. In that context, light-10 is considered in this study as the upper bound run.

Methodology

In the study, documents in the test collection were firstly normalized. Following this, all documents were located into a single pool index using the Lucene. Throughout the indexing process texts in documents were tokenized on both white spaces and punctuation marks.

In the study, two runs were conducted, these were Light10 and ExtendedLight10. Accordingly, two corresponding indices were created. The only difference between them is the used stemmer to index documents. While in the first run, which was light10, the light 10 stemmer were used to index and analyze both queries and documents, the new proposed stemmer, which is the Extended-10, was being used for indexing the test collection.

As in the methodology of study I, queries in the two runs were submitted one by one and the top 10 documents in each were collected to check their relevance, which was computed using the precision measure. Following this, the values of all precision fr each query were averaged to conclude a final single value, which is the mean average precision (MAP).

Results and Discussion

Figure 5.2 shows the results of the Extend10 stemming in terms of mean average precision, compared to light10 while Table 5.4 lists the corresponding results in tabular form and query-by-query for all those in our query set.

Query collection	Measures	Light Stemming	Extended Light stemming
@ 5 topics	MAP	0.86	0.88
@ 10topics	MAP	0.78	0.81
@ 15 topics	MAP	0.75	0.80

TABLE 5.4: Shows the results of proposed stemming, compared to the baseline (light 10) algorithm

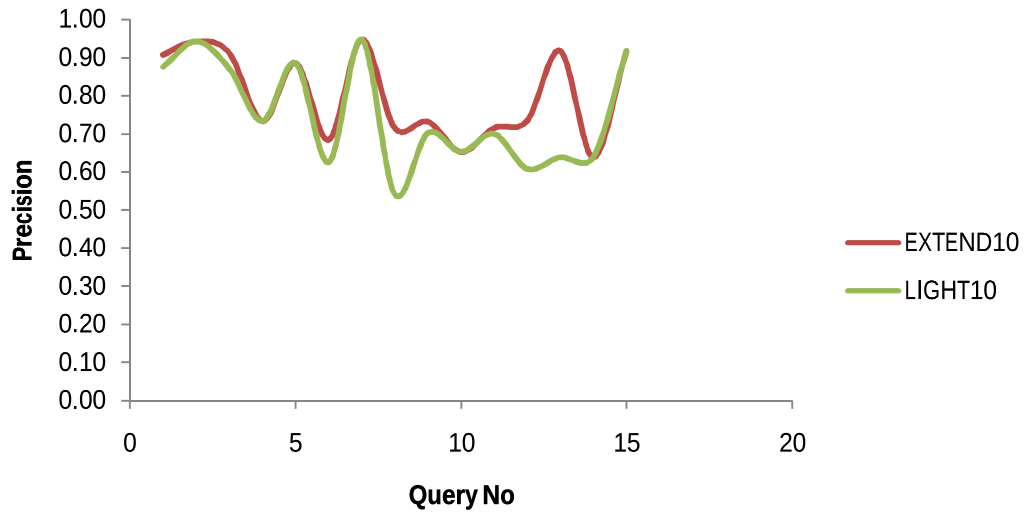


FIG5.2: Retrieval effectiveness of light10 and the proposed Extend10 stemmer. Results were shown in terms of mean average precision.

As it can be seen in the figure, at all the queries the performance of the proposed Extended-10 stemmer was better than the performance of light 10. As it be noticed in the table, the difference at top 5 queries was small (the MAP at the first 5 queries was 0.86 when using light 10, while it was 0.88 for the same queries when the proposed technique for stemming was used). In the same table, Table 5.4, it is noticed that the difference in performance of the two runs begins with small values and then it increases as more queries were used and more documents were retrieved until it reached its upper at query 15, in which the difference between performance of the two runs reaches 0.05 for the proposed stemmer Extended-10.

This improvement in the performance of Extended-10 was caused by the fact that there were many words in documents, which are similar and Extended-10 could easily group them together, while light 10 did not. Note that the proposed Extended-10 stemmer removes more prefixes and suffixes as it adds more prefixes and suffixes. For example, in the query no. 9, which says 'مقدمة في قواعد البيانات', the Extended-10 stemmer didn't remove the last letters ات due to its conditions in the algorithm, which adds more restriction to the removal of words. This is not the case in light 10 stemmer, which results in removing the letters ات and, thus, resulting in the stem بيان . The same arguments also hold for the remaining queries. In other words, as the removal of light-10 may result in removing some important letters, the proposed Exended-10 stemmer removes only letters when the words satisfy certain conditions. Another example, in the query no. 8, which says 'تعريف الشبكات اللاسلكية', light 10 removes the letters ات in the word الشبكات, resulting in شبك, while the proposed stemmer did not and it preserved the same word as it appears.

It is also observed that the difference in performance between the two runs, meaning light 10 and Extended-10, was large in some few queries. This is mainly caused by the same reasons that were stated in the previous paragraph. Additionally, there are some suffixes and prefixes that were not included in light 10 while they were added in the proposed Extended-10 and, thus, resulting in better performance for the latter stemmer.

5.3 The summary

In this chapter, it was shown that the stemming has a large positive effect on Arabic information retrieval. It was seen in the chapter that two different sets of experiments were conducted. The first set of experiments compares the use of the surface words (no-stemming) approach to the use of light10. Results showed that the light stemming techniques are beneficial to the retrieval of Arabic documents.

In the second study, which compares light 10 to the proposed Extended-10 stemmer, it was shown that the latter results in better performance for retrieval. This is because the Extended-10 controls the removal of Arabic prefixes and suffixes and it adds more of them to the set of the strippable antefixes.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

Conclusions

Stemming in Arabic is one of the challenging problems in IR. This is because the language is very rich in its morphology and it has a very systematic approach for producing new words from only the root. From that perspective, it was concluded that stemming is important in inflectional languages such as Arabic.

This thesis investigates the problem of stemming Arabic IR. It shows the unique features of the Arabic language that makes it challenging to information retrieval. It also shows that root-based stemming approaches do not always improve performance as they may erroneously clusters some words with different meanings and, thus, resulting in an over-stemming problem. This is the main reason for developing light-stemmers approaches, which lightly stem words and without performing deep linguistic analysis. Light stemming techniques reported better results and improve the retrieval performance of Arabic documents.

Bearing this important point, this thesis developed a new stemming algorithm that is based on light stemming approaches. In particular, the new stemmer was inspired by the work of Larkey and her colleagues in what is so-called light 10. Light 10 has been classified as the best stemming algorithm for Arabic IR and it has been included in several famous IR systems like Lucence and Indri Lemur.

The new stemmer, which was called the Extended-10 stemmer, considers the unique characteristics of the Arabic words. In tact context, the proposed stemmer adds additional rules to control these unique features and, thus, elimination of prefixes and suffixes is not similar to its peer in light 0. As light 10 considers many strippable prefixes and suffixes in nouns, the Extended-10 stemmer considers both nouns and verbs. Accordingly, it adds some additional prefixes and suffixes to the list of the strippable antefixes.

In the results the thesis shows that the use of stemming is beneficial to Arabic IR. It is always good to stem Arabic texts. The results also concluded that the new proposed stemmer is better than ligh-10 and the new added set of prefixes and suffixes beside the new algorithm which adds some constraints to the removal process have a positive impact to retrieval. It is concluded that the proposed stemmer is better than the best known one in IR. Thus, the new proposed approach could have a positive effect on the entire inflectional languages, which have the same derivational systems, e.g., Hebrew and Amharic languages.

In spite of the improvement in retrieval, the proposed stemmer still have some drawbacks as it may fail to correctly stem some words. In many cases removing some strippable prefixes and suffixes may hurt other words but, this is the nature of the Arabic language, in which a single root may result in more than 90 derived words. From the results it is also concluded that the problem of stemming does not have a perfect solution but, it can be mitigated through performing deep analysis to the feature of Arabic words.

Future Work

It is noted that there are major limitations is the work reported in this thesis. These are the sizes of the test collection and the query set. It is true that many standard collections are available but for only those could pay for them. Thus, we were enforced to build our own test collection, which was small in its size. However, it is known that building such collections needs real investment, funding and requires much efforts.

From that perspective, in the future work

We have not ruled out the possibility that a better morphological analyzer, and better use of morphological analysis to conflate words, could work better than the Extended-10 stemmer. We have only tried a few obvious alternatives. Ultimately, one would like to be able to conflate all the inflected forms of a noun together, including broken plurals, adjectives, and all the conjugations of a verb, which we cannot handle in the proposed algorithm.

REFERENCES

ABDELALI, A. 2006. *Improving Arabic Information Retrieval Using Local Variations in Modern Standard Arabic.* New Mexico Institute of Mining and Technology.

AL-MASKARI, A., SANDERSON, M. & CLOUGH, P. Year. The relationship between IR effectiveness measures and user satisfaction. *In:* Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, 2007. ACM, 773-774.

ALANSARY, S., NAGI, M. & ADLY, N. Year. Building an International Corpus of Arabic (ICA :(progress of compilation stage. *In:* 7th International Conference on Language Engineering, Cairo, Egypt, 5–6 December 2007, 2007.

ALJLAYL, M. & FRIEDER, O. Year. On Arabic search: improving the retrieval effectiveness via a light stemming approach. *In* :Proceedings of the eleventh international conference on Information and knowledge management, 2002. ACM, 340-347.

ARABIC-HISTORY. 2014. *World's Muslim Population*[Online]. Available: http://www.indiana.edu/~arabic/arabic_history.htm [Accessed.[2014

ATTIA, M. A. Year. Arabic tokenization system. *In:* Proceedings of the 2007 workshop on computational approaches to semitic languages: Common issues and resources, 2007. Association for Computational Linguistics, 65-72.

BELKIN, N. J. & CROFT, W. B .1992 .Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM,* 35, 29-38.

BISHOP, C. M. 1998. *A history of the Arabic language* [Online]. Available: ttp://linguistics.byu.edu/classes/ling450ch/reports/arabic.html. [Accessed 24 April 1998.[

BUCKWALTER, T. Year. Issues in Arabic orthography and morphology analysis. *In:* Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, 2004. Association for Computational Linguistics, 31-3.4

CROFT, W. B., METZLER, D. & STROHMAN, T. 2010. *Search engines: Information retrieval in practice*, Addison-Wesley Reading.

DARWISH, K. & OARD, D. W. 2003b. CLIR Experiments at Maryland for TREC-2002: Evidence combination for Arabic-English retrieval. TREC 2003 proceedings.

DEYOUNG, A. J. 1999. *Arabic language history* [Online]. Available: Available: http://www.indiana.edu/ ~arabic/arabic_history.htm [Accessed.[

FOX, E. A. 1983. Extending the boolean and vector space models of information retrieval with p-norm queries and multiple concept types.

HABASH, N. & RAMBOW, O. Year. Arabic diacritization through full morphological tagging. *In:* Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers, 2007. Association for Computational Linguistics, 53-56.

HEGAZI, N. & EL-SHARKAWI, A. Year. An approach to a computerized lexical analyzer for natural Arabic text. *In:* Proceedings of the Arabic Language Conference ,Kuwait, 1985.

HIEMSTRA, D. 2000. *Using language models for information retrieval*, CTIT Ph.D. thesis.

JACKSON, P. & MOULINIER, I. 2007. *Natural language processing for online applications: Text retrieval, extraction and categorization*, John Benjamins Publishing.

JONES, K. S. & VAN RIJSBERGEN, C. 1975. Report on the need for and provision of an "ideal" information retrieval test collection. British Library Research and Development Report 5266. *Computer Laboratory, University of Cambridge,* 46.

KADRI, Y. & NIE, J.-Y. Year. Effective stemming for Arabic information retrieval. *In:* proceedings of the Challenge of Arabic for NLP/MT Conference, Londres, Royaume-Uni, 2006.

KHOJA, S. & GARSIDE, R. 1999. Stemming arabic text. *Lancaster, UK, Computing Department, Lancaster University*.

KRAAIJ, W. & POHLMANN, R. Year. Viewing stemming as recall enhancement. *In:* Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, 1996. ACM, 40-48.

LARKEY, L. S., BALLESTEROS, L. & CONNELL, M. E. 2007. Light stemming for Arabic information retrieval. *Arabic computational morphology.* Springer.

LEVOW, G.-A., OARD, D. W. & RESNIK, P. 2005. Dictionary-based techniques for cross-language information retrieval. *Information processing & management,* 41, 523-547.

MANNING, C. D., RAGHAVAN, P. & SCHÜTZE, H. 2008. *Introduction to information retrieval*, Cambridge university press Cambridge.

MOUKDAD, H. 2006. Stemming and root-based approaches to the retrieval of Arabic documents on the Web. *Webology,* 3.

NIE, J.-Y. 2010. Cross-language information retrieval. *Synthesis Lectures on Human Language Technologies,* 3, 1-125.

PAICE, C. D. 1984. Soft evaluation of Boolean search queries in information retrieval systems. *Information Technology :Research and Development,* 3, 33-41.

PIRKOLA, A., HEDLUND, T., KESKUSTALO, H. & JÄRVELIN, K. 2001. Dictionary-based cross-language information retrieval: Problems, methods, and research findings. *Information retrieval,* 4, 209-230.

POPULATION, W. S. M. 20 .14*World's Muslim Population [Online].* [Online]. [Accessed http://islam.about. com/od/muslimcountries/a/population.htm 2014.[

ROBERTSON, S. E. & JONES, K. S. 1976. Relevance weighting of search terms. J*ournal of the American Society for Information science,* 27, 129-146.

ROCCHIO, J. J. 1971. Relevance feedback in information retrieval. 313–323.

SAAD, M. K. & ASHOUR, W. Year. OSAC: Open Source Arabic Corpora. *In:* 6th ArchEng Int. Symposiums, EEECS, 2010.

SPARCK JONES, K., WALKER, S. & ROBERTSON, S. E. 20 .00A probabilistic model of information retrieval: development and comparative experiments: Part 1. *Information processing & management,* 36, 779-808.

SPINK, A. 2002. A user-centered approach to evaluating human interaction with web search engines: an exploratory study. *Information processing & management,* 38, 401-426.

TAYLI, M. & AL-SALAMAH, A. I. 1990. Building bilingual microcomputer systems. *Communications of the ACM,* 33, 495-504.

VOORHEES, E. M. & HARMAN, D. Year. Overview of TREC 2001. *In:* Trec, 20.01

WALKER, S., ROBERTSON, S. E., BOUGHANEM, M., JONES, G. J. & JONES, K. S. Year. Okapi at TREC-6 Automatic ad hoc, VLC, routing, filtering and QSDR. *In:* TREC, 1997. Citeseer, 125-136.

WITTEN, I. H., MOFFAT, A. & BELL, T. C. 1999. *Managing gigabytes: compressing and indexing documents and images*, Morgan Kaufmann.

XU, J., FRASER, A. & WEISCHEDEL, R. Year. Empirical studies in strategies for Arabic retrieval. *In:* Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, 2002. ACM, 269-274.

YAHYA, A. H. & SALHI, A. Y. Year. Tools for Arabic People Names Processing and Retrieval. *In:* the proceedings of the 3rd Arabic language Technology International Conference(ALTIC), 2011 International Conference on, 2011 Alexandria, Egypt. 71-76.

ZAWAYDEH, B. & SAADI, Z. Year. Orthographic Variations in Arabic Corpora. *In:* Goverments Users Conference, 2006.