A comparison of root and stemming techniques for the retrieval of Arabic documents

Haidar Moukdad

Graduate School of Library and Information Studies
McGill University, Montreal
December 2001

A dissertation submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements of the degree of Doctor of Philosophy

To my wife Tamara, for standing by me every step of the way

This is for you my YYY

Contents

Tables and figures

Table

Figure

Abstract

Using information retrieval systems to gain access to documents in languages other than English is becoming an increasingly significant problem. Rules, theories, algorithms, and retrieval methods designed and developed for English and other morphologically similar languages may or may not apply in the linguistic environments of other languages. The problem is particularly acute in languages that differ radically from English on account of morphological rules. This thesis compares the effects of two indexing and retrieval techniques (stemming and root retrieval) on information retrieval in Arabic through an exploratory study of the handling of Arabic words by an English search engine. It also investigates how best to adapt existing English-language information retrieval systems for use with Arabic-language texts, and specifically to process words and their morphological variations. Search experiments, using 2000 Arabic documents and 40 Arabic search terms (nouns), were conducted with a Web search engine developed for English, AltaVista, to compare the performances of stemming and root retrieval and to investigate the possibility of adapting this engine for use with Arabic text. The results of the experiments show that more effective retrieval can be accomplished through stemming, and that it is possible to adapt the engine for use with Arabic without the need to develop root-retrieval features.

Résumé

L'utilisation de systèmes de recherche d'information pour repérer des documents dans des langues autres que l'anglais devient de plus en plus un problème critique. Les règles, les théories, les algorithmes, et les méthodes de recherche conçues et développées pour l'anglais et autres langues morphologiquement semblables peuvent ou ne peuvent pas s'appliquer dans d'autres environnements linguistiques. Le problème est particulièrement grave pour les langues qui diffèrent radicalement de l'anglais à cause des règles de morphologie. La présente thèse offre une étude exploratoire du traitement des mots arabes par un moteur de recherche anglais afin de comparer les résultats de deux techniques d'indexation et de recherche utilisant les tiges et les racines. L'étude explore également comment mieux adapter les systèmes anglais de recherche d'information pour le traitement de textes arabes, et spécifiquement pour le traitement des mots et de leurs variations morphologiques. Des tests portant sur 2000 documents arabes et 40 mots clés arabes (noms), ont été conduits avec un moteur de recherche de Web développé en anglais, AltaVista, pour comparer les résultats de la recherche par tiges et par racines et pour étudier la possibilité d'adapter ce moteur à l'utilisation de textes arabes. Les résultats prouvent qu'une recherche d'information plus efficace peut être accomplie en utilisant les tiges, et qu'il est possible d'adapter le moteur sans développer de dispositifs de recherche utilisant les racines.

Acknowledgements

I am greatly indebted to Dr. Andrew Large, my thesis supervisor, for his unwavering support and his dedication to this study from start to end. His intellectual guidance, constant encouragement, reading and rereading of numerous drafts, astute suggestions, and patience and understanding of my academic needs and moods made this study what it is, and allowed me to conduct the research work and bring it to fruition.

My sincere gratitude goes to Dr. Jamshid Beheshti, an inspiring and diligent member of my Doctoral Committee, for countless hours of stimulating and thought-provoking discussions. His informed commentary on research procedures and his enlightening expertise inspired many an idea, and showed me the light at the end of the tunnel.

I also wish to thank Dr. John Leide and Dr. Gerald Ratzer, the enthusiastic members of the Doctoral Committee, who provided invaluable comments at a critical junction for the thesis and added insight and new perspectives to the research process.

Lastly, special thanks go to Dr. France Bouthillier, for editing the French translation of the abstract; to Ms. Ghada Hallaq, for her expert commentary and input on the Arabic language, and for checking the accuracy and consistency of Arabic-to-English translation; and to the Faculty of Graduate Studies and Research, McGill University, for providing financial support.

1. Introduction

1.1 The problem

The last few years have witnessed the catalyzation of the Internet revolution by the World Wide Web (Web). Supported by versatile electronic publishing technologies, standards and tools, the Web has enabled the wide-spread dissemination of information to users all over the world. Catering to the demands and the needs of a linguistically diverse user population, Web information has been produced in a multitude of languages that are no longer, as they were in the pre-Internet electronic publishing era, restricted to English and other major European languages. Although English is still the dominant language, the numbers of documents in many other languages are growing at a faster rate than in English; steadily it is losing ground proportionately to languages such as Chinese, French, German, Japanese, Russian and Spanish. In 1999, English documents accounted for 86% of Web content (Inktomi 2000), but by 2001 this had fallen to 52% (Funredes 2001), with the remaining 48% in languages like German (6.97%), Spanish (5.69%), French (4.61%), Chinese and Japanese (both estimated between 5% and 8%).

This proliferation of textual information in a multitude of languages other than English, and the growing need for devising information retrieval (IR) methods to handle multilingual collections of documents, have spawned growing interest in

a research area known as cross-language information retrieval (CLIR). As opposed to monolingual IR, where the query and the documents are in the same language, in systems that support CLIR, queries can be entered in one language to retrieve documents in another language. In a CLIR system, a query is translated into the language of the documents, and depending on the actual system being used, the retrieved documents may or may not be translated into the language of the query. For example, a Chinese-speaking user can enter search terms in his/her native language and retrieve English documents. The Chinese terms in the query are translated into English and matched against the English documents; the retrieved English documents might then also be translated into Chinese. Although, for obvious reasons, most of the problems addressed by CLIR research have revolved around translation issues (Grefenstette 1998), the potential problems of using a search engine designed for one linguistic environment to search in other environments deserve attention.

The problem of creating retrieval systems for different language structures is independent of technology and media. Language-related retrieval problems manifest themselves in a traditional IR system in the same way as they do in a search engine on the Web. However, the Web is the single most important factor that has stimulated the growth of multilingual information and an interest in cross-language retrieval experiments. Since most of the improvements to IR will likely end up implemented on the Web, it is natural and practical to focus on Web applications in tackling IR problems in a multilingual environment.

In IR systems in general and in Web search systems in particular, rules, theories, algorithms, and retrieval methods designed and developed for English and other morphologically similar languages may or may not apply in different linguistic environments. Nowhere could the problem be more acute than in languages that differ radically from English on account of morphology and word-formation rules. Words (or several words formed into compound phrases) rather than complete sentences are typically entered by users of IR systems to express their information needs and query the system. And almost all search engines are designed to function with words rather than well-formed sentences. The success of the retrieval system in dealing with words greatly depends on its ability to handle their morphological structure.

On the Web, an ideal search engine of the future hopefully will have all the necessary indexing and search features needed to accommodate the different languages it indexes. While it may be logistically impossible to develop an engine that is equipped to undertake a morphological analysis of all languages, a more realistic approach would be to identify key morphological prerequisites for effective IR in individual languages and integrate them into existing English-language engines' system design. This dissertation investigates how best to adapt such systems to the morphology of one language that differs markedly from English—Arabic.

1.2 Information retrieval and language

Information retrieval is concerned with two concepts: how to represent information and how to interpret its structure (Meadow 1992). Traditionally, documents prepared for IR have gone through a process that identifies the main parts of a document and indexes them for representation within the framework of the system. Once documents are indexed and entered in the system, the system has to interpret the structure of the information included in the documents to facilitate searching and retrieving (Lancaster and Warner 1993). The IR task can simply be expressed as the endeavor to identify and retrieve from a document store all the documents, and only the documents, that match a specific, expressed information need. In a digital environment, the IR system comprises hardware (processor, input and output devices), software to match input queries against stored documents, a document store (database) and indexes to the database that normally are automatically generated by the software (unless the documents have already been assigned controlled indexing terms by a human indexer before they are entered to the database). The IR system, then, plays the role of an intermediary between the information need of the user and the information documents that might answer it. A user poses the information question in the form of a query, and the system interprets this query and searches its index files for an answer (Salton 1971). Usually, the answer comes back in the form of an output from the IR system informing the user of the existence (or non-existence) and whereabouts of documents relating to the query

(Lancaster 1968b). The question, or query, itself typically comprises one or more words linked by a Boolean operator (or several words forming a phrase) encompassing the information need, whether the language of a query is artificially imposed by the system (for example, a controlled vocabulary) or is freely expressed and formulated by the user in natural language. This normally remains the case even for those few IR systems that can accept complete sentences rather than keyword queries, although here syntactic processing as well as morphological matching probably will be applied. In the same way, the indexes and the stored documents in the database comprise individual words or phrases. Words, then, lie at the heart of IR; they are where the information query starts and what the IR system utilizes to locate information.

Searching for information relies on language to perform its functions. The content of documents and information records are represented by language elements, and the information problems of users are also expressed in terms of language (Harter 1986). All human languages have vocabularies, corpora of words whose elements constitute the building blocks from which meaningful communication constructions can be formed. Words form phrases, phrases form sentences, sentences form paragraphs, and paragraphs form documents. If we think of a document as a collection of words, then it is easy to understand the role played by the structure of a language in providing access to information within this document. Words are formed according to specific rules and guidelines that differ between languages, creating IR problems and offering

potential solutions that need to be investigated with the particular language involved in mind.

In the early days of IR systems, and for several decades subsequently, this issue was not as crucial as it is today. The majority of systems were developed to accept English-language queries for matching against English-language documents. To cite just one example, the famous early retrieval experiments conducted at Cranfield in the United Kingdom in the 1960s took place in an exclusively English-language retrieval environment. Not surprisingly, therefore, search and retrieval software, indexing methods, and user interfaces were designed specifically for this language.

For more than three decades after the term "information retrieval" was first introduced by Calvin Mooers in 1950 (Swanson 1988), IR efforts focused on ways of improving retrieval effectiveness. Based on the structure of the English language and on its linguistic properties, researchers made significant progress towards devising indexing algorithms and search techniques within the framework of theoretical models for English IR. They developed a deeper understanding of the inherent complexities in the IR process, widely adopted evaluation methods and performance measures, and tested more effective retrieval techniques (Hildreth 1989). The first major IR study was undertaken by Taube (1953), who conducted experiments on indexing systems in order to investigate their implementation in a machine-operated environment. In the

1960s, the several Cranfield studies introduced concepts like recall, precision and relevance to measure the retrieval effectiveness of IR systems (Cleverdon 1964). Recall is a measure of the extent to which the IR system retrieves all documents in the database that match the information seeker's need (at least as expressed in the search query). Precision is a measure of the extent to which the IR system only retrieves those documents that match the information seeker's need, and rejects all others. Both recall and precision in turn are governed by the concept of relevance, a subjective assessment of the usefulness of the retrieved documents to the information seeker. Although quantitative objective values commonly are assigned to recall and precision in retrieval experiments, the subjective nature of relevance, used to establish both recall and precision values, in reality means that evaluation is largely dependent on the assessor's subjective perception of the relationship between the query and the retrieved documents.

Relevance itself can be assumed to be language-independent. The relevance of English and French documents to a specific query, for example, should not be affected by the linguistic properties of these two languages. On the other hand, the construction of the query and the representation of the document in the IR system are language-dependent procedures. Linguistic problems arise in characterizing the content of documents and information requests in such a way that the characterizations can be used in an automated process to assess and retrieve relevant material (Sparck Jones and Kay 1973). It is important to emphasize this distinction, for it helps in identifying IR findings that can be

applied regardless of the specific language environment in contrast to those which may not have such a universal application. The language-dependent/independent aspects of IR have become more important in the last decade because of the increasing availability of non-English digital material and the consequent growth of interest in IR from a wide variety of languages.

1.3 Rationale and contributions

The distinction between linguistic and non-linguistic problems in IR is paramount to understanding the special problems that can arise in different linguistic environments. Since document indexing and query constructions are the most obvious linguistic elements in IR environments (Maron 1977), attention should be focused on them when investigating the need for changes in existing English-language-based IR systems to accommodate retrieval from other languages. Two general questions must be answered: 1) Are language-related IR techniques developed for one language equally applicable to other languages? 2) Or, should new techniques be developed that are more appropriate for the individual characteristics of each language? Furthermore, it cannot be assumed that these two questions will be answered identically for all languages.

This research seeks to answer these questions in just one language environment: Arabic. This is a language that may present challenges to an IR system designed

originally for an English-language environment (Khurshid 1997), because Arabic morphology and word-formation rules are radically different from those of English. These rules are based on a root -and-pattern system (discussed in detail in chapter 2) that has long been thought a major factor in hindering IR operations (Beesley 1996). However, to date there has been no attempt to explore the extent and possible ramifications of this system and its rules for an operational IR system. The cost of developing new systems or radically modifying existing ones to work with Arabic will be high. Such an investment requires clear demonstration that the pay off in improved IR efficiency will justify the cost.

While several researchers on Arabic IR (al-Kharashi 1991, Abu Salem 1992, al-Kharashi and Evens 1994, Hmeidi, Kanaan and Evens 1997, Abu Salem, al-Omari and Evens 1999) advocate the use of IR techniques such as advanced word stemming and root searching as the most effective ways to retrieve Arabic information, their research does not address the feasibility of applying these techniques in an English-language information retrieval (ELIR) environment. Nor does it investigate the effectiveness of the two techniques in a full-text environment, where the morphological properties of Arabic words are fully manifested. These researchers confined their work to experimental environments, where small-scale Arabic IR systems were developed using collections of homogeneous bibliographic records of limited vocabulary. In such environments, root retrieval might be a plausible solution, but can the same be

said in more realistic settings, for example, on the Web, where the richness of the language is fully represented, and where the development of root-searching capabilities would be a costly undertaking? Is a better solution to focus on specific aspects of Arabic word-formation rules that can be handled by stemming (Moukdad 1999)?

The purpose of this study is to investigate how best to adapt existing ELIR systems for use with Arabic-language texts, and specifically how to process words and their morphological variations. Although the linguistic problems associated with Arabic IR are independent of technology and medium, the Web was chosen for this research as the technological environment, and its search engines as the IR systems (the information delivery medium). Arabic IR researchers have set up a theoretical framework in which they argue that stemming and root retrieval are required tools for any system to effectively handle the morphology of the Arabic language. The present work takes this argument as its starting point and investigates if it holds true in existing ELIR systems. It hypothesizes that, in a realistic IR environment like the Web, stemming is a more viable approach than root retrieval to adapting these systems to Arabic texts. Can the case be made for stemming against root retrieval? If so, to what extent can the stemming of Arabic words eliminate the need for root retrieval?

The major objectives of this research are to:

1. determine the better technique to handle Arabic word variants in an IR environment: stemming or root retrieval.

2. make recommendations for adding features to existing ELIR systems to improve processing of Arabic words.

Other objectives are to:

a. compare Arabic and English word-formation rules.

   The morphologies of Arabic and English are examined to isolate noun-formation features that affect the enhancement of retrieval. Differences between the two morphologies are highlighted to determine the extent of their possible effect on IR efficiency, as well as to create a list of Arabic affixes for use in a series of searching experiments to retrieve all possible variants of a selection of nouns.

b. develop a methodology for studying the efficiency of an ELIR system in an Arabic-language database environment.

   This methodology offers an alternative to complex linguistic analysis, isolated from the context of an IR system, and it can be used as a starting point for IR research in languages other than Arabic.

c. propose an agenda for future research on Arabic IR and recommendations for research directions.

In order to better understand the stated objectives and to clarify the linguistic setting of this dissertation, it is necessary to start with the language that lies at the heart of the research. Chapter 2 provides an overview of the Arabic language and its history; then it presents a detailed look at the morphology of this language with focus on noun-formation rules.

Chapter 3 examines the similarities and differences between Arabic and English morphologies, and discusses English word-formation rules with a focus on nouns in the context of IR. Chapter 4 provides a review of the literature that has provided a backdrop for this dissertation. Chapter 5 describes the reasons for selecting the two Web-based search engines used in the research experiments, as well as their retrieval features. Chapter 6 presents the results of a preliminary study conducted to identify common Arabic prefixes that could then be used in the searches described in detail in the thesis. Chapter 7 provides a detailed explanation of the methodological procedures adopted in this research, and the results of applying these procedures are presented and analysed in Chapter 8. Finally, Chapter 9 discusses the implications of the findings, the limitations of the research and plans for future work.

2. The Arabic language

2.1 Historical overview

Arabic belongs to the Semitic family of languages, which includes Akkadian, Aramaic, Ethiopic, Hebrew, Phoenician, Syriac and Ugaritic. These languages took root and flourished in a contiguous area that covers parts of western Asia and Africa, and were divided into Northern and Southern Semitic language families (Abbott 1938). Arabic was a minor member of the Southern family, used by a small number of largely nomadic tribes in the Arabian Peninsula. The origins of today's Arabic can be traced back to the ancient dialects of these tribes, but it was not until the sixth century A.D. that these dialects developed into a language of poetry and then into the language of the Qur'an in the following century (Chejne 1969). At the height of the expansion of Islam in the eighth century, Arabic linguistics came into being as a tool for spreading the language of the Qur'an, the holy book of Islam (Hlal 1987). At that time, members of the new community felt the need to know the language of the Qur'an, which had been adopted as the official language of the young Islamic state (Gibb 1963). The Islamic conquests dispersed Arab settlers over a vast stretch of territory from Persia to Spain. As a result of this expansion, large numbers of non-Arab converts to Islam adopted Arabic, and the language entered a period of rapid and significant evolution. Fearing that the use of the language would be corrupted by the intrusion of foreign languages and dialects,

and in an effort to preserve their language, Arab scholars in the eighth century started a movement to promote the study of Arabic grammar and lexicography. A standard of correct Arabic was established and has survived, retaining its basic grammatical rules for more than 10 centuries (Beeston 1970). During this long period the language essentially has remained the same and, through periods of rise and decline, has retained the characteristics that govern its use.

Today, Arabic is a thriving language spoken by more than 200 million people, and it is one of the official languages of the United Nations. Although different spoken Arabic dialects exist throughout the Arabic world, there is only one form of the written language found in literary works, newspapers and other printed works, and it is known as *fsHh* or "Standard Arabic". The spoken dialects (*camyh)*, on the other hand, are used very marginally in writing (except in popular/folklore poetry and in conversations in some novels) and are viewed by many educated Arabs as degraded forms of the language (Mace 1998). These dialects are rarely found in print or electronic documents, making them of negligible importance for IR purposes. Standard Arabic (henceforth referred to as Arabic), on the other hand, is the universal communication medium throughout the Arab world, used in official government publications, newspapers, magazines, and other types of mass print media, and for correspondence. Although spoken Arabic dialects might have different grammatical and linguistic rules, Arabic is cited by linguists as an example of languages that have weathered the passage of time and survived intact, because

it is virtually uniform in its grammar and vocabulary throughout the Arab world (Stetkevych 1970). It has been preserved by means of a rich literary tradition and a heritage of manuscripts and print material that has recently found its way to different publication environments. In the last few years, there has been a steady increase in the use of Arabic in electronic and online environments. It has established a presence on the Web, and serious efforts have been made to facilitate the dissemination of Arabic documents in this global environment (Large and Moukdad 2000).

2.2 The script and Alphabet

The Arabic script was derived from the Aramaic via the Nabatean cursive script (Hitti 1963). As is the case with all Semitic languages, the script is written from right to left, and the user of Arabic in a computer environment will notice that the directionality of the script affects the interface design of applications (scrolling bars, navigation features, icon placement, etc.). In computer environments and in Western academic institutions, the Arabic script has traditionally been represented in converted (Romanised) form, where Arabic letters are replaced by equivalents from the Roman alphabet. In the 19th century, scholars started to use Roman letters in print and handwritten documents when they wanted to incorporate Arabic names or words into their writings in West European languages.

There are now a number of widely used Romanisation systems for Arabic script, including one developed by the Library of Congress and another adopted by the *Encyclopedia of Islam.* Script conversion can be achieved in two ways. Transliteration seeks to represent each letter in one script by one letter in a second script; transcription seeks to express the sound of letters in one script by letters in a second script (Wellisch 1975). Most of the script conversion systems from the Arabic to the Roman alphabet do not render strict (letter-by-letter) transliteration of Arabic words (Beesley 2000). They are intended also to serve as a pronunciation guide to help in reading the language, and therefore incorporate transcription techniques alongside transliteration.

Romanised versions of Arabic text can be misleading, because usually they omit unpronounced letters. Furthermore, the representation of vocalizations differs from one transcription system to another. Conversion can also create ambiguity when two Roman letters are used to represent an Arabic sound that does not exist in English. In an IR environment, words are identified by the characters (letters) that form them, not by the phonemes by which they are pronounced. Romanised Arabic letters and words in this thesis therefore are transliterated: each Arabic letter is represented by one English letter according to a transliteration scheme adapted from Buckwalter (2000). This scheme eliminates any confusion created by using more than one letter in Roman script to represent one letter in Arabic. For example the Arabic word meaning "bounty" is transliterated according to this scheme as "*xyr*", an exact replacement of the

three letters in the Arabic script by three letters in the Roman script. In other schemes, *xyr* might be represented as "*khyr*", a four-letter word. If an IR system only allows truncation after the first three characters of a word, for example, it would be confusing to discuss "*khyr*" in a truncation example - as the original Arabic word comprises three letters, truncation would not be applied after the "*y*" (seemingly, the third letter in the word), but after the "*r*" (which looks in transliteration as if it is the fourth letter).

Table 2.1 The Arabic Alphabet adapted from Buckwalter (Buckwalter 2000)

	Alone	Transliteration [a]	Initial	Medial	Terminal
1 [b]	أ	a	أ	ـأ	ـأ
2	ب	b	بـ	ـبـ	ـب
3 [c]	ت	t	تـ	ـتـ	ـت
4	ث	v	ثـ	ـثـ	ـث
5	ج	j	جـ	ـجـ	ـج
6	ح	H	حـ	ـحـ	ـح
7	خ	x	خـ	ـخـ	ـخ
8	د	d	د	ـد	ـد
9	ذ	Z	ذ	ـذ	ـذ
10	ر	r	ر	ـر	ـر
11	ز	z	ز	ـز	ـز
12	س	s	سـ	ـسـ	ـس
13	ش	S	شـ	ـشـ	ـش
14	ص	p	صـ	ـصـ	ـص
15	ض	D	ضـ	ـضـ	ـض
16	ط	T	طـ	ـطـ	ـط
17	ظ	P	ظـ	ـظـ	ـظ
18	ع	c	عـ	ـعـ	ـع
19	غ	g	غـ	ـغـ	ـغ
20	ف	f	فـ	ـفـ	ـف
21	ق	q	قـ	ـقـ	ـق
22	ك	k	كـ	ـكـ	ـك
23	ل	l	لـ	ـلـ	ـل
24	م	m	مـ	ـمـ	ـم
25	ن	n	نـ	ـنـ	ـن
26	ه	h	هـ	ـهـ	ـه
27	و	w	و	ـو	ـو
28	ي	y	يـ	ـيـ	ـي

[a] Letters 6, 7, 15, 16, 17, 18, and 19 have no sound equivalents in English. No pronunciation connection is made between these letters and their transliterated forms; the English letters are used for representation purposes only. Letter 4 sounds like the "th" in through, 9 sounds like the "th" in the, and 13 sounds like the "sh" in rush. The rest of the letters sound similar to their transliterated English forms.

[b] The first letter (*alf*) of the alphabet is a special one. It is considered a consonant when the diacritical mark (*hmzh*) occurs over or under it. The *hmzh* is a glottal stop; without it, *alf* is a long vowel. There are cases, however, where the *hmzh* is omitted from an initial *alf,* substituted with a special mark, or just ignored to simplify writing. The *hmzh* also occurs by itself in the middle or the end of a word, and is used to indicate glottal stops over letters 27 and 28, and over letter 28 without the diacritical dots. For present purposes, all these cases are represented with an *a.*

[c] There is a special form of this letter that occurs at the end of feminine nouns. It is similar to the terminal form of letter 26 with two diacritical dots over it. In electronic environments, the two dots are usually dropped and this special form is treated as letter 26.

The Arabic alphabet has 28 consonants (see Table 2.1) many of which are identical in shape but are differentiated by dots placed either above or below them. Although there are no capital letters, each Arabic letter has a different shape depending on whether it occurs at the beginning, middle or end of the word, or if it stands alone. The characters of an Arabic word are connected to preceding and following letters in a way similar to English cursive writing. However, letters 1, 8, 9, 10, 11, and 27 (Table 2.1) cannot be connected to following letters, and they have the same medial and final forms. In print and electronic material, a character (*kSydh*) similar to a long underscore is inserted between two letters as an extension of the first one.[1] The use of this character is optional (it is mainly used for aesthetic purposes). In addition to consonants, the Arabic alphabet has three short vowels that are used to clarify the pronunciation of a letter, and to indicate the grammatical case of words when they occur at the end of the word (Chejne 1969). These vowels are written as diacritical marks directly above or below the letter and they are transliterated below with their approximate English pronunciation, their English representation equivalents, and their case ending functions:

Dmh (pronunciation function equivalent to "oo" in good or book): a character similar to a "comma" above the letter; it indicates nominative case when it occurs at the end of the word.

[1] In electronic environments, the *kSydh* is treated as a separate character. In the word *Hpan* (horse), for example, a *kSydh* may be added between p and a. Retrieving this word necessitates entering the *kSydh*, unless there is a mechanism in place to ignore it.

ftHh (pronunciation function equivalent to "u" in run or but): a diagonal stroke above a letter; it indicates accusative case when it occurs at the end of the word.

ksrh (pronunciation function equivalent to "i" in hit or miss): a diagonal stroke under a letter; it indicates genitive case when it occurs at the end of the word.

There is also a fourth diacritical sign (*skwn*) that indicates the absence of a vowel. A letter affected by this mark is pronounced as its English equivalent would be when it is not followed by a vowel (like the "r" and "d" in "hard"). The *skwn* is represented by a small circle above the letter and never occurs at the beginning of a word or the end of a noun. It occurs sometimes, however, at the end of the singular form of a verb to indicate imperative or jussive moods. When there are two successive occurrences of a letter, the first with a *skwn* and the second with any of the three short vowels, one of the letters is omitted in writing. A doubled letter is pronounced as two but written as one; the change is expressed in writing by the diacritic *Sdh* (a 90-degree, right-rotated 3) and one of the short vowels. In the word *skr* (sugar), for example, the *k* is a double letter; when pronounced, it is more like *kk*.

Arabic short vowels have three corresponding long vowels that, as opposed to diacritical representation, are indicated by the use of three consonants from the Alphabet: *w*, *a*, and *y*. In some cases, when an unvoweled letter is followed by *w*, *a* or *y* these three letters function as long vowels and force an extended (long)

pronunciation as illustrated in the following examples and their English equivalents: *zmwr* as in the English room, *ktab* as in hat, and *rym* as in heat. A cross-over form between *y* and *a* occasionally occurs at the end of words. This is not a separate letter; it is called a "shortened *a*" that is written as *y* without the two diacritical dots but pronounced like the long vowel *a*. In addition to the main vowels, a short vowel is sometimes duplicated at the end of a noun by a doubling process called nunation.[2] This process produces (hence the term doubling) two *Dmh* (two commas above the letter), two *ftHh* (two diagonal strokes above the letter), or two *ksrh* (two diagonal strokes below the letter) to accompany the last letter of a word. Arabic does not have the equivalent of the English indefinite article (a/an), but nunation is used to indicate an indefinite noun in different cases. For example, the noun *wld* (boy) becomes (a boy) in the nominative case by doubling the vowel *Dmh* on the *d*; changing the double *Dmh* to double *ftHh* changes the case of the noun to accusative, while changing it to double *ksrh* makes it genitive.

As opposed to English and other Western languages, the short vowels have never become a permanent part of the Arabic writing system, with the single exception, perhaps, of the Qur'an, in which the vowels are always written to ensure accurate and correct reading of the sacred book (Chejne 1969). Before and during early Islamic times, the writing system did not include vowel signs, and it was difficult to distinguish between similar letters. Diacritical dots

[2] The letter *n* is pronounced like the English word noon. Orientalists devised the term nunation to indicate the sound n produced when a short vowel is doubled.

differentiating letters were not invented until the eight century A.D., and there were no spaces between words (Versteegh 1997). Vowel marks were primarily invented to ensure correct reading of the Qur'an, and they are still in use to serve this function. Text with vowels (vocalized text) can also be found in works of poetry and in elementary textbooks whose main purpose is to teach the pronunciation of words. The absence of vocalization in other texts has been a problem for a long time, and it sometimes leads to reading mistakes and misidentification of parts of speech. Arabic, being a highly inflected language as explained below, identifies the accusative, nominative and genitive cases of a noun with endings indicated by vowels. If these vowels are not written, it is necessary that the reader understand the context of the whole sentence in order to avoid mistakes. Out of context, the meaning of a word standing on its own (not in a sentence) is open to interpretation. Non-vocalized *Scr*, for example, could mean: to feel, hair, or poetry.

Despite all of the problems created by non-vocalized text, Arabic speakers are used to reading their language in this fashion. A good knowledge of grammar is necessary to read correctly in most cases, and it should be expected that Arabic text will not include vowels. In the age of the Internet, digitized Arabic text is largely non-vocalized. Software programs can make provisions for vocalization, but, as is the case with print material, the practice has been to ignore short vowels. Consequently, words in this study are treated as they would normally occur in a typical Arabic text. In an IR environment, retrieval is based on

characters that form words; therefore, attention is paid only to the 28 consonant characters listed in Table 2.1, and the vowels are ignored.

2.3 The root-and-pattern system

Although the Semitic languages differ in structure and grammar, they share one characteristic that facilitated transition from one to another. In most cases, lexical forms (words) in these languages are derived from basic building blocks with tri-consonantal roots at their bases. The word building process starts with the three letters of a root and follows a regular set of word patterns. All traditional Semitic-language dictionaries and most modern ones are arranged by root. Instead of listing alphabetic entries, these dictionaries arrange words under entries of the roots that produce them. To look up a specific word, the user has to have enough knowledge to isolate the root and then locate its entry. It is as though words like ascribe, describe, subscribe, circumscribe, proscribe, prescribe, inscribe were listed in an English dictionary under the Latin root "scribere" that describes the basic idea of writing/drawing (De Young 2000). The difference is that the words grouped under an Arabic root can be analyzed down to the letters of a root and the predefined morphological patterns that created them.

One of the standard Arabic lexicons, (*lsan alcrb* or the *Language of the Arabs*), lists 6,350 triliteral roots and 2,500 quadriliteral ones. Out of these, only about 1200 are still used in modern Arabic vocabulary (Hegazi and Elsharkawi 1985), and the great majority of words can be broken into triliteral roots consisting of three consonants or radicals (Ziadeh and Winder 1957). Although the description given here focuses on triliteral roots and the patterns that apply to them, it should be sufficient to give an idea of how the system in general works.

Words constructed from the same root constitute what is traditionally called a morpho-semantic field, where semantic attributes are assigned through patterns governed by morphological rules (see below). The meaning that is inherent in the root is shared by all words in this field. However, the patterns that produce these words make them semantically distinguished (Rafea and Shaalan 1993). A similar process can be noticed in English if we look at "cleanliness", "unclean", "cleaner" and "cleanly". While all four words share the basic meaning that is inherent in "clean" (to be physically/morally clean), they convey different semantic messages: cleanliness (the state of being clean), unclean (the opposite of being clean), cleaner (the person/substance that cleans), and cleanly (in clean condition/mode). We could say that adding "er" to the root created the noun cleaner, "ly" the adverb cleanly, and so on.

In general, each pattern is associated with a meaning which, when combined with the meaning conveyed by the root, gives a final meaning to the derived

word (Moutaouakil 1987). Using patterns to create different morphological variations from a root is a fairly regular process. It is similar to a mathematical formula, where the original letters of the root are constant variables, and the changing variables are letters added at the beginning, middle or end of the root. Patterns may also be indicated by vowel changes only; in these cases no letters are added to the root and, for present purposes (in the written form), the structure of the word is considered unchanged. Traditionally, Arab grammarians have used the letters *f*, *c* and *l* as generic letters to represent the root and the patterns. These letters were chosen because they form the root *fcl* (a basic meaning of "to do"), and because *fcl* also means verb. In derived words the order of these letters is always the same: *f* is first, *c* second, and *l* last. For example, the root *rkb* (to ride) is represented according to the following equations: $r = f$, $k = c$, and b = *l*. To create the active participle (rider), the pattern *facl* (formula: $f + a + c + l$) is used. This pattern is actually formed from the original three letters with the long vowel *a* inserted between the first and the second. To obtain the Arabic equivalent of "rider" it would simply be necessary to replace *f*, *c*, *l* with *r*, *k*, *b* and get $r + a + k + b$ (*rakb*). This is similar to adding "er" to some English verbs to indicate the performer of the action or function, as in writ-er, sing-er, think-er. However, Arabic employs a more elaborate system of patterns, and virtually every word derived from a root has to conform to a pattern.

The Arabic root is actually the simplest form of the verb, or what is called Form I. Nine other forms can be derived from a triliteral root, and they have been traditionally identified with Roman numerals as Form II through Form X.[3] These forms are manipulated versions of Form I and they represent subtle variations in its meaning (Reig 1983). It is as if we were to look at the English verbs: value, validate, and revalue as three different verb forms. These verbs have different but connected meanings: value (to appreciate) is the root, validate (to recognize or establish the value), and revalue (to repeat the action of value). However, Arabic takes this principle much further and uses verb patterns to develop a rich vocabulary of verb forms as illustrated in Table 2.2. Most of these forms are connected with certain meanings, but it is not always possible to derive all forms for all roots (Wightwick and Gaafar 1998). Some roots might produce eight or nine forms, while others are restricted to two or three. Table 2.2 shows the patterns used to derive the nine verb forms and examples of meaning variations as compared to Form I. It should be noted that: Form II (*fcl*) is distinguished from the root by adding a *Sdh* over *c*, and the difference between Form IV (*afcl*) and Form IX (*afcl*) is a *Sdh* over the *l* of the latter. Form IX is now rare in Arabic; it is only used in the context of changing color as in *axDr* (to turn green) or *azrq* (to turn blue).

[3] Other rare forms exist, but can only be found in poetry and archaic texts.

Table 2.2. Triliteral verb forms and patterns

	Pattern	**Example Verb**	**Form I**
Form II	*fcl*	*drs* (to teach)	*drs* (to study)
Form III	*facl*	*samH*(to forgive)	*smH* (to allow)
Form IV	*afcl*	*anzl* (to bring down)	*nzl* (to descend)
Form V	*tfcl*	*tclm* (to learn)	*clm* (to know)
Form VI	*tfacl*	*twapl* (to continue)	*wpl* (to arrive)
Form VII	*anfcl*	*anfpl* (to be disconnected)	*fpl* (to disconnect)
Form VIII	*aftcl*	*aHtrs* (to be cautious)	*Hrs* (to guard)
Form IX	*afcl*	*aswd* (to become black)	*sad* (to dominate)
Form X	*astfcl*	*astqbl* (to greet)	*qbl* (to accept)

Pattern rules also govern the derivation of Arabic nouns from roots. We already saw above that *facl* is a pattern for creating the active participle from the root. Another simple pattern is to add *m* at the beginning of the root (*mfcl*) to convey the meaning of a place where the action of the verb could be executed. For instance, adding *m* to *tcm* (inherent meaning of food or feeding) produces *mtcm* (restaurant); and adding *m* to *srH* (to roam or play) produces *msrH* (theater). Following the same pattern, *mktb* (office or desk) is derived from *ktb* (to write), *mdxl* (entrance) from *dxl* (to enter) and *mcml* (factory) from *cml* (to work). Table 2.3 shows a random selection of other noun-derivation patterns and illustrates examples of their usage.

Table 2.3. A sample of noun-derivation patterns

Pattern	Sample roots	Derived nouns
facwl	*jrr* (to pull), *Hsb* (to count)	*jarwr* (drawer), *Haswb* (computer)
fcal	*Hrm* (to deny), *dwm* (to last)	*Hram* (unlawful), *dwam* (work shift)
fcalh	*zrc* (to plant), *pnc* (to make)	*zrach* (agriculture), *pnach* (industry)
fcyl	*kbr* (to grow), *gsl* (to wash)	*kbyr* (big), *gsyl* (laundry)
fclan	*zcl* (to grieve), *ksl* (to neglect)	*zclan* (sad), *kslan* (lazy)
fclh	*Hrb* (to battle), *dfc* (to pay)	*Hrbh* (spear), *dfch* (installment)
fcwl	*xjl* (to hesitate), *Skr* (to thank)	*xjwl* (shy), *Skwr* (grateful)

Table 2.3 shows a small fraction of Arabic patterns. There are hundreds more that convey all kinds of meanings. It is important to keep in mind that these patterns are not arbitrary and should not be used as such. Learners of Arabic have traditionally relied on the root-and-pattern system to practise correct use of words and to enhance their vocabulary. This system is also used to derive different forms of a base noun as explained below.

2.4 Word formation

In linguistic terms, word formation is a function of morphology. Morphological analysis of human languages is largely based on the following linguistic elements: root, stem, affixes (prefixes, infixes and suffixes), and morphemes (De Guzman and O'Grady 1987). All these elements can be used in IR; therefore, a clear definition of their roles in word structures is essential. The function of the Arabic root has already been explained, but a general explanation of the term "root" as used in IR and in general linguistics will prove useful. Arabic roots are

forms of the verb, whereas in English and many other languages a root can be an adjective, a noun or a verb. A global definition of 'root' is that it is a word that can stand on its own without the need for additional morphological elements. At the same time, this word cannot be broken down into smaller words. However, a root can accept the addition of elements to create new words (Crystal 1985). Run, for example, is a root: it is a complete word with a meaningful semantic representation. This word cannot be broken down to generate new words like ru or un. However, an 's' can be added to run to obtain runs, 'ing' to obtain running, and 'er' to obtain runner. When an 's', 'ing' and 'er' are added to run, it is also called a stem. The linguistic elements 's', 'ing', and 'er' are suffixes because they are added at the end of the stem and they cannot exist in isolation from the word. That said, a morpheme is the smallest meaning-bearing unit in the composition of a word. For example, run has one morpheme (run), runs has two (run and s). Table 2.4 uses examples from English and Arabic to illustrate the relationship between root and stem, to show the differences between prefixes, infixes and suffixes, and to explain the concept of morphemes.

Table 2.4. Roots, stems, affixes, and morphemes in English and Arabic words

Word	Root	Stem(s)	Pre.	Infix	Suff.	Morphemes
attract	attract	--	--	--	--	attract
attractive	attract	attract	--	--	ive	attract, ive
attractively	attract	attract, attractive	--	--	ive, ly	attract, ive, ly
unattractive	attract	attract, attractive	un	--	ive	un, attract, ive
qbl	*qbl*	--	--	--	--	*qbl*
qbyl	*qbl*	*qbl*	--	*y*	--	*qbl, y*
mqbwl	*qbl*	*qbl, qbwl*	*m*	*w*	--	*m, qbl, w*
mqbwlwn	*qbl*	*qbl, qbwl, mqbwl*	*m*	*w*	*wn*	*m, qbl, w, wn*

In Table 2.4, an adjective is created from the verb attract by adding the suffix 'ive'; similarly, an adverb is created from the adjective attractive by adding the suffix 'ly'. The suffixes 'ive' and 'ly' are derivational suffixes, and the generation process is called derivational morphology, because new grammatical categories of the word (parts of speech) are derived: verb ➔ adjective, and adjective ➔ adverb. Conversely, the process of attaching a suffix like 's' to a noun (car ➔ cars) or to a verb (eat ➔ eats) is called inflectional morphology, because it does not create a new grammatical category from the word (word class is not affected); inflections typically encode person, number and gender features (Matthews 1974). In this case, 'car' and 'cars' are both nouns and 'eat' and 'eats' are verbs. The inflectional suffix 's' inflects the noun to indicate number contrast (singular and plural) and the verb to indicate the person of the subject (first and third).

Arab linguists identify only three parts of speech: the verb, the noun, and the particle (Mehdi 1986). This is a broad categorization by which nouns (as defined

in English), adjectives, and pronouns are all classified as nouns. As opposed to English adjectives, Arabic adjectives are not treated separately from nouns. In fact, what is considered an adjective in English can be an adjective or noun in Arabic. Take the English phrase: 'the big boy of the class'. The Arabic equivalent of this phrase can read something like *kbyr awlad alpf*, which translates roughly as 'the big of the boys of the class'. The English adjective 'big' translates as *kbyr*, but in the Arabic phrase *kbyr* is a noun. However, we could say *wld kbyr* (a big boy); *kbyr*, in this case, is an adjective. For present purposes, Arabic nouns and adjectives are simply referred to as nouns. No distinction is made between the two, and the treatment of word formation disregards any discrepancies found in English terminology.

Pronouns are divided into two categories: attached and detached. An attached pronoun is always attached to a verb or a noun, while a detached pronoun cannot be attached to a verb or a noun. For present purposes, attached pronouns are treated separately from nouns: they are identified as pronouns not nouns.

Particles, the third part of Arabic speech, include the definite article, prepositions, conjunctions, interjections, question particles and answer particles (like yes and no). For present purposes, only particles that attach to nouns will be treated because they affect IR procedures studied in this research (See 2.6). In general, these are prepositions and conjunctions like *l* (to) and *w* (and) in *llmdrsh* (to the school) and *wkrh* (and a ball). Particles that cannot be attached to

nouns are usually connected to pronouns or they occur alone like *fy* in *fyh* (in it) or *cn* in *cny* (about me).

Based on the categories of Arabic speech, the concept "word formation" is used for present purposes to describe the use of inflectional affixes to generate new forms (sub-classes) from the base form of an Arabic noun, for example, singular to plural or masculine to feminine. It does not, however, cover proper nouns (such as people, place, day and month names, etc.); these types of nouns usually do not have variants and are not affected by word-formation rules. Prefixes and suffixes in the form of particles and pronouns that do not create sub-classes from the base noun are treated separately in 2.5. The base form of the noun is the masculine singular, or the feminine singular form if a masculine one does not exist. For example, the masculine noun *ktab* (book) is a base form for the plural *ktb* (books) and for the feminine singular *ktabh* (writing). By the same token, the feminine noun *Tawlh* (table) is the base form for the plural *Tawlat* (tables) since *Tawlh* does not have a masculine form.

The Arabic base noun can be inflected to indicate gender, number and case. Gender contrasts masculine and feminine, number contrasts singular, dual and plural, and case contrasts nominative, accusative and genitive.

There is no neutral gender in Arabic; nouns are divided between masculine and feminine. This division is grammatical rather than natural, because nouns do not

necessarily have to be male or female (Cowan 1958). In general the feminine is formed from the masculine by adding the suffix *h*. For example, *Talb* is a male student and *Talbh* is a female student. In other instances, the masculine and feminine forms of a noun do not share a common root as in *rjl* (man) and *amrah* (woman). In general, the case ending of singular nouns is indicated by change of vowels: *Dmh* for nominative, *ftHh* for accusative, and *ksrh* for genitive.

Dual indicates a number of two, and is formed by adding the suffix *an* to singular masculine and feminine nouns in the nominative case. The dual form of the masculine *qlm* (pen) is *qlman* (two pens); *mdrstan*[4] (two schools) is the dual form of the feminine *mdrsh* (school). The suffix *an* is changed to *in* to indicate accusative or genitive cases: *alwld akl tfaHtyn* (the boy ate two apples); *drs fy mchdyn* (he studied in two institutes).

The plural form indicates any number higher than two; it has three types. The first type, the sound masculine plural, is formed by adding the suffix *wn* to the base masculine noun in the nominative case: *mclmwn* (teachers) is the plural of *mclm.* In the accusative and genitive cases the *wn* is changed to y*n* as in *mclmyn* (teachers in the accusative). The second type, the sound feminine plural, is constructed by dropping the suffix *h* from feminine nouns in the nominative cases and adding *at* in its place. For example, *wrqat* (papers) is the plural of the feminine noun *wrqh*. The suffix *at* does not change in the accusative and

[4] The feminine indicator *h* at the end of *mdrsh* is transliterated as a *t* when another letter follows it.

genitive cases, but case changes are indicated by changing the vowel over the *t* to *ftHh* and *ksrh* respectively. Constructing the third type, the broken plural, is more complex than the sound ones. Broken plural forms of masculine and feminine nouns are derived through the use of a pattern system similar to the one mentioned in 2.3, and case is indicated through the use of the vowels. Murtonen (1964) lists 82 of the most common patterns in addition to many rarely used ones. Table 2.5 shows a sample of ten of these patterns and their usage. The patterns are used with masculine and feminine nouns where it is not possible to construct a sound plural form. Applying these patterns might involve the addition or omission of prefixes, infixes, suffixes, or a combination of two or three of these affixes. The pattern *fcal*, for example, is applied to create the broken plural *rjal* of the masculine singular *rjl* (*man*). The pattern *fwacl* produces the plural *cwapf* of the feminine singular *capfh* (storm), and *afacl* produces the plural *agany* of *agnyh* (song).

Table 2.5. A sample of broken plural patterns

Pattern	Singular noun	Plural noun
afacyl	*Hdyv* (conversation)	*aHadyv* (conversations)
afcal	*Hzb* (political party)	*aHzab* (political parties)
fcala	*pHraa* (desert)	*pHara* (deserts)
fclan	*qtyc* (flock)	*qtcan* (flocks)
fcwl	*asd* (lion)	*aswd* (lions)
facyl	*mqcd* (seat)	*mqacd* (seats)

2.5 Particles and pronouns

Particles and pronouns affect the construction of Arabic words because, as opposed to English, they are usually attached to verbs and nouns (Haywood 1960). Possessive pronouns and particles (including the definite article) are attached to nouns in the form of non-inflectional prefixes or suffixes (See Tables 2.6 and 2.7). For instance, possessive pronouns are always attached as suffixes (the *y* in *byty* (my house)), while the definite article *al* is attached as a prefix (*albyt* (the house)). This phenomenon is so widespread in the language that the number of occurrences of nouns with these prefixes and suffixes is much higher than without them (Yahya 1989). For example, virtually every Arabic noun accepts the prefix *al* (the definite article), and the conjunction *w* (and) is always attached to the word that follows it. The prefix *k* (the equivalent of the English word 'like' in "sweet like honey") does not occur in isolation from the noun. Instead, the Arabic equivalent of "sweet like honey" is "*Hlw kalcsl*", where *alcsl* is honey.

Table 2.6. The non-inflectional suffixes (possessive pronouns)

Possessive pronouns	**Person/gender/number**	**Example**
y (my)	First/both/singular	*bldy* (my country)
k (your)	Second/both/singular	*bldk* (your country)
kma (your)	Second/both/dual	*bldkma* (your country)
km (your)	Second/masculine/plural	*bldkm* (your country)
h (his)	Third/masculine/singular	*bldh* (his country)
ha (her)	Third/feminine/singular	*bldha* (her country)
hma (their)	Third/both/dual	*bldhma* (their country)
hn (their)	Third/feminine/plural	*bldhn* (their country)
hm (their)	Third/masculine/plural	*bldhm* (their country)

Table 2.7. The most common prefix particles

Prefix particle	Meaning	Example
al	the	*alSarc* (the street)
b	in, with	*bmjalk* (in your field)
f	and, therefore	*frays* (and president)
k	like, as	*kjamch* (like university)
l	for, to	*lmdynh* (to city)
w	and	*wjrs* (and bell)

Particles are far more common than possessive pronouns and they can occur alone or in combination at the beginning of a noun. Up to three can be attached to a noun. For example, the definite article can be preceded by any one of the other five prefixes. Table 2.8 shows some of the most common combinations and gives examples of their use.

Table 2.8. Prefix particle combinations

Combination	Meaning	Example
bal	in the	*balSarc* (in the street)
fal	and the, therefore the	*flmdynh* (therefore the city)
kal	like the	*kalrays* (like the president)
lal	for the, to the	*lalmjal* (to the field)
wal	and the	*waljamch* (and the university)
fbal	therefore in the	*fbalHaq* (therefore in the right)
wbal	and in the	*wbalwsT* (and in the center)
wkal	and like the	*wkalSms* (and like the sun)
wlal	and for the	*wlalysar* (and for the left)
fb	and in, therefore in	*fbnwm* (therefore in sleep)
wb	and in	*wbHrkh* (and in movement)
fl	and for, therefore to	*flmcrkh* (and for battle)
wl	and for, and to	*wlzman* (and to time)

2.6 Arabic nouns in IR

The most salient problem in an IR system is to improve recall rates while retaining a high level of precision (van Rijsbergen 1979). Retrieving morphological variants of a word is a technique that is meant to enhance recall. Because of the dominance of the root system, and the large number of derivation possibilities, morphological variants of a word are not always semantically related. Under the root *qpd*, for example, we can find *qpd* (intention) and *qpydh* (poem). It is safe to assume that a user searching for *qpydh* would not be interested in *qpd*. Instead, this user would be interested in *qpydtan* (two poems), *qpaad* (poems), and in all occurrences of these words with possessive pronouns and prefixes as mentioned in 2.5. For present purposes, morphological variants of an Arabic noun are divided into three groups: root based (nouns grouped under one root), inflected (feminine, dual, plural, etc.), and affixed (attached to particles and possessive pronouns).

In theory, looking up a word in an IR system with root searching capabilities is like using a traditional lexicon as explained in 2.3. However, instead of figuring out the root of the word and then looking it up in the lexicon, the IR system analyzes the word down to its root and retrieves documents that contain any morphological variation derived from that root (Al-Kharashi and Evens 1994). The IR system should also retrieve inflected and affixed variants of a noun, which are not usually listed in a lexicon. With this ultimate variant retrieval,

potential problems might arise. As explained above, save for the root, the search noun might not have much in common semantically with many of the retrieved nouns. For instance, entering the word *clm* (flag) as a search term will retrieve any document that contains words such as *clamh* (scholar), *tclym* (teaching), and *clym* (expert). It will also retrieve all affixed and inflected variants of these words in addition to all possible forms of the verb *clm* (to know).

The problem of retrieving inflected and affixed variants of Arabic nouns has to be approached from two directions: one dealing with suffixes (feminine, dual, sound plural, personal pronouns, etc.), and another dealing with prefixes and infixes (particles and broken plurals). In a traditional IR system, suffixes can be handled through stemming (at the indexing stage) or right-hand truncation[5] (using a wild card character, like * or ?, to replace a string of characters at the end of the word at the search stage); this will reduce the search term to a stem and allow the retrieval of documents containing its variants. Searching for the English truncated term run*, for example, will retrieve runner, runners, and running. Arabic Suffixes can be handled in the same way. To retrieve variants of a noun, it is sufficient to truncate the search term: *mktwb** (letter) will retrieve *mktwby* (my letter), *mktwban* (two letters), *mktwbha* (her letter), etc.

[5] Although Arabic is written from right to left, since most Arabic words are represented in this thesis in transliterated forms, right-hand and left-hand truncation are used for present purposes to respectively indicate end-of-the-word and beginning-of-the-word truncation.

A traditional IR system will handle infixed variants, but the user has to be well versed in Arabic to use middle truncation. In the simplest forms of broken plurals, this will involve correct insertion of the wild card character in the middle of the word. The term *dr*s* will retrieve the singular form of *drs* (lesson) and its broken plural *drws* (lessons). Other more complex broken plural forms (Table 2.5) have more than one infix, or a prefix and an infix added to the base form, making truncation a challenging task. The plural of *msjwn* (prisoner) is *msajyn*; middle truncation involves inserting the wild card betweens s and j, and between j and n in the singular form. In the case of the singular *mrD* (disease), the plural *amraD* is formed by adding *a* as a prefix and an infix. This poses a new problem, because middle truncation is not enough: the beginning of the word has to be truncated too (left-hand truncation). This type of truncation is also needed to strip nouns of any particles (Tables 2.7 and 2.8) that might be attached to them. In this case, an IR system should have left-hand truncation capabilities or be able to identify and isolate these particles at the indexing stage. A search using the truncated term **bryd* (mail) would retrieve documents that contain *bryd* or any of its variants like *albryd* (the mail), *wbryd* (and mail), and *kalbryd* (like the mail). By the same token, if the indexing mechanism can isolate the particles, *albryd*, *wbryd*, and *kalbryd* would be stripped of *al*, *w*, and *kal* and indexed under *bryd*.

Most of the noun-formation rules that may hinder retrieval in Arabic either do not exist in English (infixes) or have minimal effect on retrieval (prefixes). The

magnitude of the problem created by these rules cannot be fully understood and appreciated without an examination of English noun-formation rules and their role in IR. Any attempt to adapt ELIR systems to Arabic will have to take into account the similarities and differences between the noun-formation rules of these two languages. Although some morphological rules are shared among languages, attention in an IR environment should be focused on the differences between languages and on ways to accommodated them. Arabic rules differ radically from those of English, and this degree of difference is likely to adversely affect the processing of Arabic nouns in an ELIR system. Where do Arabic and English morphologies meet, and how do their differences manifest themselves in search and retrieval environments? In an attempt to answer these questions, the next chapter takes a look at the morphology of the English language, discussing its word-formation rules with a focus on nouns in the context of IR.

3. An overview of English

Grammars of human languages are roughly equal in complexity. While parts of grammar differ in complexity from one language to another, these parts, as a unified whole, ultimately create language rules and conventions that are complex in nature and require thorough understanding on the part of language learners and analysts. As the dominant language in the world, English has developed over the years into a common-sense language that has accommodated structural changes and simplification of speech and grammar rules as needed. While most non-native speakers of English struggle with its huge vocabulary corpus and its seemingly arbitrary pronunciation rules, linguists and language learners alike make note of the versatility of this language and the relative simplicity of its morphological structure and rules. By virtue of its universal appeal, English has become the language of technology and of computer terminology and applications. This universal appeal, however, is not a consequence of the linguistic structure of the English language, but, rather, of British, and especially of later American political, economic, cultural and military power. As opposed to Arabic and other morphologically complex languages, the morphological rules of the English language lend themselves easily to treatment in computational environments and have developed into a linguistic system against which other systems can be studied and evaluated. In this thesis, the morphological parts of this system that affect IR are examined,

namely the rules of word formation and the presence of affixes in English words.

3.1 Word formation

Chapter 2 explained how the formation of Arabic words revolves around roots, and how complex morphological rules govern the creation of new word meanings. Conversely, English words tend be formed on the basis of a limited and relatively straightforward number of rules and processes. These processes have been at work in the language for some time, and many words in English daily use today were, at one time, considered mis-uses of the language. Regardless, there exist in today's English nine common processes by which words are formed: eight of these involve formation methods ranging from combining words to abbreviating them, and the ninth involves the use of affixes to expand English vocabulary (called derivation). Derivation is so common that it will be treated separately under the topic of affixes. The remaining eight processes are coinage, borrowing, compounding, blending, clipping, backformation, conversion and acronyms (Bauer 1983).

3.1.1 *Coinage*

Coinage is one of the least common processes of word formation in English, and it involves the invention of totally new words that have no relatives in the

dictionary--someone will coin the new term (word). This practice is accepted by linguists, and typical sources for these types of words are invented trade or product names associated with specific companies that later become generic in their use. Examples of these words include aspirin, nylon, zipper, kleenex and teflon. After the initial coinage of such words, they gradually become accepted as everyday words in the language.

3.1.2 *Borrowing*

The concept of borrowing is common across all human languages. As a result of historical, trade, cultural and military contacts, words are borrowed by one language from another to supplement the vocabulary or to define a term that is foreign to that language. These words are identified by linguists as loan words, and they can create linguistic analysis problems because of their foreign origins. As a general rule, many of the morphological rules that apply to a native word do not apply to a word that has been borrowed from another language.

Borrowing is a common source of new words in English and can be traced back to the early encounters of England with other civilizations and especially to its colonial possessions throughout the world. A vast number of loan words has been adopted both from European and Eastern languages, including words such as alcohol (Arabic), boss (Dutch), croissant (French), lilac (Persian), piano (Italian), pretzel (German), tycoon (Japanese), and yogurt (Turkish). More

recently, foreign words have trickled into English usage as a direct consequence of the dominance of American culture and of the waves of non-English speakers who have immigrated to the United States. Although American cultural icons and practices are recognized all over the world and have forced the use of many English words in other languages, foreign loan words have been adopted by modern American English to accommodate a growing need to extend the vocabulary and accept words of cultural or religious significance in other languages like bagel (Yiddish) and mafia (Italian).

3.1.3 *Compounding*

A concept of English word formation that is strange to Arabic is compounding. While this is a common practice in English and Germanic languages, it is virtually non-existent in Arabic. Technically, this process involves bringing together two separate words and producing a single word form (a compound word). While it is a very productive source of new English words, compounding is so common in English that compound words are often mistaken for a single word, and they include such obvious examples as doorknob, fingerprint, mailman, sunburn and wastebasket.

3.1.4 *Blending*

This is similar to compounding, except that the two words lose some of their letters. Typically, the beginning of one word is blended with the ending of another to form a new word. For example, a restaurant meal that is served on Sunday as a breakfast or lunch is called brunch: br- from breakfast and -unch from lunch. By the same token, the unpleasant product of modern city life pollution is called smog: sm- from smoke and -og from fog. Other interesting examples are the blending of binary (b-) and digit (-it) to produce the computer term bit, and of channel (ch-) and tunnel (-unnel) to create an appropriate title word for the tunnel under the English Channel that links England and France. As with compounding, blending does not exist in Arabic.

3.1.5 *Clipping*

Again, this is another way of getting rid of a part of the word or reducing it to create a new word. English has developed creative ways to shorten words and make them more accessible for daily use, especially in casual speech. Although the word gasoline is the official term describing the oil by-product used to run cars, it usually referred to as gas (after clipping the -oline). Fax is a clipped form of facsimile as is ad in relation to advertisement, cab to cabriolet, and condo to condominium. The clipping phenomenon is also common in educational

environments, where just about every word gets reduced--hence the existence of words such as chem, exam, lab, math, prof and typo.

3.1.6 *Backformation*

Words of a specific grammatical type (usually a noun) are reduced to produce another grammatical type (usually a verb). This process is called backformation, because it contradicts the norm of creating nouns from existing verbs. In a typical situation, the word exists in the form of a noun and then the need arises to create a verb to convey the meaning of this noun. For example, when the word television came into existence, the verb televise was not an English word, but it was formed later through the process of backformation. Similar circumstances necessitated the formation of babysit from babysitter, liaise from liaison and opt from option.

3.1.7 *Conversion*

Although conversion does not alter the appearance or structure of a word, it is nevertheless considered a way of forming new words. Technically, through the process of conversion, a word stays the same but its grammatical category changes. The noun paper becomes the verb paper so we could say: "he is papering the wall". Also, vacation is converted to the verb vacation as in "he vacations in Florida every winter". The conversion process can also be reversed,

and verbs become nouns. For example, the verbs guess, kill, spy are the sources for a guess, a kill and a spy.

3.1.8 *Acronyms*

A word-formation process that is common in English but rare in Arabic is the use of acronyms. In English, acronym words are usually formed from the initial letters of a set of other words, and they can be divided into two groups: Alphabetisms and regular words. Alphabetisms are acronyms that are pronounced as a string of Alphabet letters like VCR (video cassette recorder) or CD (compact disk). Regular words are acronyms that are pronounced as singular words and they are accepted in usage as a word that is governed by pronunciation rules. These include words such as NATO (North Atlantic Treaty Organization) and NASA (National Aviation and Space Agency). While these examples have kept their capital letters, others have lost these with time and are written like any other regular English words as we can see in radar (radio detecting and ranging) and scuba (self contained underwater breathing apparatus).

3.2 Affixes

Attaching affixes to English words is by far the most common process of word formation. New words are derived from existing ones by means of using one or

more affixes to alter the meaning, grammatical category, verb tense, count or gender of these words. These affixes are usually prefixes (attached at the beginning of the word) or suffixes (attached at the end of the word). On the other hand, infixes (inserted in the middle of the word) which occur in Arabic do not exist in English. Affixes can be derivational or inflectional. Derivational affixes change the grammatical function/category of a word or its meaning (modern (adjective) modern-ize (noun)), while inflectional affixes inflect nouns and verbs to indicate tense, gender or count changes as in kill/ kill-ed, lion/lion-ess and school/school-s.

Covering all of the rules of English word formation is well beyond the scope of this thesis. As with Arabic, the focus here is on the variations in occurrences of the basic form of the noun in English text. Inflectional variations of nouns are the most common terms that affect IR. These variations include forms contrasted by gender, number and case. While Arabic nouns are either masculine or feminine, English nouns are in most cases neuter. Gender is contrasted only when it is naturally necessary to differentiate between nouns, i.e., when nouns are referring to a male or a female (man/woman, actor/actress, king/queen and witch/warlock)

In Arabic a *-h* is usually attached to a masculine noun to create a feminine form. The use of an equivalent suffix is rare in English, but there are instances where a feminine noun is created by adding the suffix -ess to the singular form of the

masculine. For example, prince becomes princess and steward becomes stewardess.

Another inflectional form of English nouns is the number: While Arabic has singular, dual and plural forms of the noun, English distinguishes only between the singular and the plural forms. Usually the plural form of a singular noun is formed through adding a suffix, although some nouns are converted to plural by changing some part of their internal structure. The most common plural-indicating suffixes are: -s, -es. These two suffixes are usually added to the singular noun without any changes to its structure, while other less common suffixes might replace the ending of the noun. Table 3.1 lists all the possible suffixes and examples of their usages (Young 1984). In addition to suffixes, some English singular words do not follow regular rules and go through internal changes to produce irregular forms of the plural. Some examples of these words are listed in Table 3.2.

Table 3.1. Plural-indicating suffixes and their usage

Suffix	Singular/Plural	The process
-s	teacher/teachers	-s is attached to the noun without any changes to the stem
-es	princess/princesses	-es is attached to the noun without any changes to the stem
-es	analysis/analyses	-es replaces -is in the original stem
-en	ox/oxen	-en is attached to the noun without any changes to the stem
-ren	child/children	-ren is attached to the noun without any changes to the stem
-a	curriculum/curricula	-a replaces -um in the original stem
-a	criterion/criteria	-a replaces -on in the original stem
-i	Alumnus/alumni	-i replaces -us in the original stem

Table 3.2. Irregular plural forms

Singular forms	Plural forms	The process
foot, tooth, man	feet, teeth, men	The vowel in the stem is replaced by another vowel
loaf, calf, leaf	loaves, calves, leaves	-es is added and the last consonant of the stem is replaced

Lastly, an English noun can be inflected to indicate possessiveness (the genitive case). This form is usually obtained by the use of an apostrophe or a combination of an apostrophe and -s as suffixes at the end of nouns. If a book belongs to the teacher, then we can say: "this is the teacher's book". In a different situation, the book might belong to the students; therefore, it is "the students' book".

3.3 English nouns and IR

Retrieving different variations of an English noun should, in theory, enhance recall in an IR system. For present purposes, these variations (morphological variants) are those mentioned in 3.2 and could be divided into two groups: suffixed variants (feminine, regular plural and genitive forms) and non-suffixed variants (irregular plurals).

The problem of retrieving English noun variations is not as difficult as that of Arabic nouns. While Arabic nouns can be present with all kinds of affixes, the process of isolating the basic form of a suffixed English noun is relatively straightforward. A simple stemming procedure, for example, is all that is needed to reduce many plural forms to their singular forms (Table 3.1). A similar procedure will also isolate the basic form of a feminine noun or a genitive form (through the elimination of -ess, the apostrophe, and -'s).

Theoretically, at least, most of the IR problems in English environments are not related to morphological variations of search terms (they are related to the particularly rich vocabulary of English, drawn from several linguistic sources, that has produced large numbers of synonyms and homonyms). As we are going to see in the next chapter, stemming has been traditionally implemented to handle word variants, although effectiveness has been debated. The negligible effect of prefixes on the retrieval of English nouns, coupled with the absence of

infixes in English morphology, have made stemming and truncation stable features of IR systems designed for this language, and they are virtually the only features needed to handle its morphology. The morphology of the language is simple enough to eliminate the need to undertake complex morphological analyses that might be necessary for other languages, a fact that has been illustrated in research on English IR and on IR in other languages.

4. A review of prior work

At the outset must be emphasized the paucity of previous work on Arabic IR in general, and the virtually non-existent theoretical work on modifying ELIR systems to work with Arabic texts. Research on Arabic IR has focused on small-scale experimental systems that were developed for research purposes only. Neither the extent to which these systems will work effectively in an operational environment, nor their relevance to the adaptation of ELIR systems for use with Arabic texts have been investigated. That said, the experimental systems and the evaluation studies that have been conducted on them offer invaluable information on the different approaches to Arabic language processing in IR environments, and provide a good starting point for this dissertation.

The methodological approach adopted in this study was inspired by and is based on the long tradition of IR research in general, and on the findings of research on Arabic IR in particular. In addition, IR research on CLIR and on languages other than Arabic and English have provided a backdrop for this dissertation and situated it within the larger picture of research on global information exchange.

This literature review is divided into five parts. The first part deals with general works on IR, identifying the landmark works in the field and summarizing their contribution. It also covers the evaluation of IR systems, a research and development area that has occupied researchers for a long time. The second part

focuses on CLIR, a relatively new IR research area closely related to the topic of this dissertation. Works relating to IR from languages other than Arabic and English are treated in the third part, while the fourth part deals with stemming as a specific technique used to handle morphological variations of words in IR. The last part exclusively treats works relating to the Arabic language in electronic environments and to the efforts that have been made to develop and evaluate Arabic IR systems.

4.1 IR and system evaluation

4.1.1 *Introduction to IR systems and the literature*

The term, "information retrieval" encompasses the whole process of locating and retrieving information. At the heart of this process is the IR system. It comprises an established set of procedures and rules, as operated by humans and/or machines, to perform some or all of the following operations: Indexing, search formulation, searching, feedback, and index language construction (Robertson 1981). While the label 'IR system' can be pasted on a wide variety of information containers, ranging from a simple card catalog to the huge repository of the Internet, it is limited here to electronic systems that need hardware and software to function. Electronic IR systems are capable of storage, retrieval, and maintenance of information components, including text, audio, images, and other multimedia objects. However the text component of an IR

system is the foundation on which the process of retrieval primarily has been based.

Historically, Bush (1949) and Shaw (1949) were among the first to introduce the concept of automated document storage and retrieval, at a time when computers were prohibitively expensive and a few privileged individuals had the luxury of working on them. About twenty years later, online search services, such as DIALOG and ORBIT, were introduced to a wider audience (Saffady 1989), heralding a new generation of IR systems and paving the way for an exponential growth of electronic information. This, in turn, has necessitated an ever-growing need for the development of new systems that have the capacity and the operational structure to handle a large amount of information and provide adequate access to it. The information explosion also stimulated a vibrant research field that has generated numerous works and experiments dealing with the structure, design, functions, performance and effectiveness of information retrieval systems. A vast literature now can be found on the topic, with subjects ranging from the development and testing of basic retrieval techniques to advanced cognitive analyses of the search process and the ramifications of this process for the development of IR systems.

Whether dealing with an operational or experimental IR system, the major preoccupation of IR researchers has been to improve retrieval effectiveness and build theoretical foundations on which improved IR systems could be

constructed. Major outcomes of this scientific work include: 1) new theoretical models of the IR environment, 2) the development of a deeper understanding of the inherent complexities in the IR process, 3) widely applicable evaluation methods and performance measures, and 4) tested, more effective retrieval techniques and friendlier user-system interfaces (Hildreth 1989).

4.1.2 *Evaluation of IR systems*

In recent years, IR technologies have been introduced into the everyday life of millions of users through the fast-expanding information domain of the Internet. IR is no longer restricted to users of online library catalogs or to individuals who have access to online information services (Kowalski 1997). One consequence has been a preoccupation on the part of IR researchers with the evaluation of systems and with the techniques for searching and retrieving information (Saracevic 1995).

IR systems are evaluated to ascertain their value and appraise their performance. The results of evaluation are then ideally used to change and improve existing systems (Kiewitt 1979). Thus, evaluation must have a purpose and must not be an end in itself (Bryant 1968). One obvious purpose is to investigate the capability of the system to satisfy the user's information needs. After all, the user is one side of the equation that has to be kept in mind when looking at the effectiveness of a system. Since the search process involves entering queries and

retrieving documents to answer those queries, the success of an IR operation is tightly related to how the user evaluates the information retrieved by the system. An IR system is evaluated to determine its usefulness and establish procedures to test and observe how well it functions, and to investigate the extent to which it can be improved (Lancaster and Fayen 1973).

Logistically, examining the current performance of the system is called macro evaluation, because it only studies when the system performs well and when it does not, without going any further. On the other hand, an in-depth investigation that goes beyond superficial description to diagnose causes of failure and suggest ways of improvement is called micro evaluation (King 1971). These causes of failure are related to two groups of factors that control the effectiveness of an IR system: database factors and factors associated with the exploitation and manipulation of the database (Lancaster and Warner 1993). The first group includes the documents in the database, how completely and accurately their subject matter is recognized and represented in the indexing process, and how adequately the system vocabulary represents the subject matter. The second group includes how well the information needs of the user are understood, how well these needs can be transformed into search strategies, and how adequately the system vocabulary represents the subject interests of the user.

The evaluation process involves deciding on the scope of the evaluation, on the design and execution of the evaluation, and on the analysis and interpretation of the results to suggest modifications to the system (Lancaster 1981). Once the need for evaluation has been settled, the next step is to decide on the factors or components to be measured. Borko (1962) suggests that a system is best evaluated by measuring user satisfaction and comparing the system's response to inquiries to other systems' responses to the same inquiries. Cleverdon (1964) was the first to detail the specifics of the process, when he listed the following six criteria that may be used to evaluate an IR system:

1. Coverage
2. Recall
3. Precision
4. Response time
5. User effort
6. Form of output

Although Cleverdon identified these criteria more than three decades ago, they are still applicable today to most evaluation studies.

Traditionally, and until 1993, evaluation studies were done primarily by academicians, who restricted their research to small selections of test documents within a controlled testing environment (Kowalski 1997). Search algorithms

were the primary focus of research, with their effectiveness measured and compared to the performance of other algorithms. This process continued to be the standard of IR system evaluation for more than three decades (in the 1960s, 1970s, and 1980s). An important development occurred in 1992, when a series of annual experiments coordinated by the National Institute of Standards and Technology was launched in the United States. Since then these Text REtrieval Conferences (TREC), have been held regularly, ushering a new era of evaluation standards by providing a huge standard database complete with search statements and expected results that can be provided to researchers and commercial companies for testing their systems. Initially, participants in TREC tested their IR systems on a large collection of heterogeneous but monolingual documents, using preset queries, and employing relevance judgements to measure recall and precision (Harman 1996). In the last few years, however, TREC has expanded its focus to include documents in a variety of languages and to conduct CLIR experiments (see below).

To better understand the problem of evaluation, it is helpful to take a quick look at the history of evaluation studies and trace their development through the last forty years. This will also give an idea of the methodologies applied, and their advantages and disadvantages. The first evaluations of IR systems began in the early 1950s when Taube (1953) conducted experiments on indexing systems in order to investigate their implementation in a machine-operated environment. Taube concluded that the evaluation of information systems can be based on two

major criteria: the characteristics of the systems, and user satisfaction. A few years later, in the 1960s, a series of studies that became the examplar for experimental evaluation of IR systems was conducted in Cranfield, England. These studies addressed issues that are still present in the IR field today (Cleverdon et al. 1966). In addition to comparing major indexing systems, the Cranfield investigation developed a method of evaluation and introduced the concepts of recall and precision (as mentioned above) and the measurement of their ratios as they are still applied today.

Building on the Cranfield investigation, Lancaster (1968a) evaluated MEDLARS, a database of medical literature, using different user groups and real (rather than experimentally established) searches. He aimed at studying users' requirements in order to determine how effective and efficient the system was in meeting their needs. Another research undertaking was headed by Salton in the US, who designed and implemented an automatic document retrieval system (SMART) during a long-running project from 1960 to 1970. He used the basic evaluation measures of precision and recall to study the effectiveness of his system (Salton 1962-1970). The same method was employed in the UK by Jackson and Sparck Jones (1970), who used a collection of 200 documents from Salton's collection and searches from the Cranfield project to perform comparisons. They summarized the work carried out on the automatic construction of keyword classifications and their use in IR, and they concluded that classified indexing terms are more suitable for retrieval purposes.

These landmark studies took place at a time when access to electronic databases was not widely available. Starting in the 1980s, the availability of online catalogs, and online and CD-ROM databases, increased, and with this came a growing interest in using real users and letting them judge the relevance of retrieved information for their needs (Tague-Sutcliffe 1996). Instead of having the researcher judge the relevance of documents or delegate the task to subject experts (as in the Cranfield studies and TREC experiments), the retrieved documents have to be judged by the users who retrieve them.

The user-oriented evaluation trend took hold in the 1990s and more evaluation studies appeared, with the focus shifting towards qualitative studies. Park (1994) discusses the need to develop the concept of user-based relevance for the benefit of users and for the meaningful development of future research in information retrieval. Park examines the characteristics of users' criteria of relevance, and suggests the use of a qualitative research approach as an alternative methodology for studying user-based relevance. Wood, Ford and Walsh (1994) also tackle the issue of relevance and search effectiveness. They consider the effect of postings information on the effectiveness of searches. The authors compared searches, made by postgraduate information studies students, of the LISA (Library and Information Science Abstracts) database on CD-ROM with and without postings information. They found that performance (the number of relevant references, precision and recall) was not significantly different but

searches with postings information took more time, and more sets were viewed than in searches without postings.

From a different perspective on users' satisfaction with their searches, Janes (1994) questions the validity of considering the user as the best choice to measure the relevance of retrieved documents. He asks how well do other people, especially those involved in information work who make such judgments as part of their training and work, perform as judges of documents for information needs they did not originate? Janes tests the question using three groups of subjects: incoming students to a school of information/library science, continuing students in that school, and academic librarians (holders of the MLIS degree). The subjects in the three groups were asked to judge the relevance of two document sets to the original users' stated information need. The outcome of these judgments was compared to those made by the users; the most important conclusion was that subjects' judgments compared reasonably well to those of the users who submitted the information needs.

Along the same line, Su (1994) conducted a study to investigate the appropriateness of 20 measures for evaluating interactive IR performance, representing four major evaluation criteria. Among the 20 measures studied were recall and precision. The user's judgment of IR success was used as the measure with which all other 20 measures were to be correlated. The study group included 40 end-users from an academic environment with individual

information problems. The users interacted with six professional intermediaries searching on their behalf in large operational systems. The author concluded that high precision does not always mean high quality (relevancy, completeness, etc.) to users because of differing users' expectations. Therefore, the user's perception is the most important factor affecting the judgment of a search.

No matter what the measured factors are, and no matter who judges the relevance of retrieved documents in the case of search evaluation, the evaluation of IR systems has been undertaken in two environments: investigation and experiment. Sparck Jones (1981) defines the investigation environment as research conducted in a laboratory with controlled elements, and the experiment environment as a measurement of a descriptive nature. Laboratory testing involves an environment where the variables are controlled as tightly as possible in order to eliminate extraneous variations that might distract the researcher from the scope of the research and ultimately confuse the results. In general, laboratory evaluations have been conducted to test new IR theories and examine their applications within IR systems (Bawden 1990). One example of such experiments is the study conducted by Robertson and Belkin (1978), who sought to prove that an IR system's performance can be improved by means of ranking retrieved items according to their probable relevance to the user's query.

While the importance and wide use of laboratory evaluation are well documented in Sparck Jones (1981), it is generally accepted now that the

relevance of results for operational systems can and should be questioned (Bawden 1990). Isolating the IR system from its users and from its actual operating environment makes the measurement of effectiveness questionable and imposes restrictions on the implementation of research findings. Testing in a real environment, or operational evaluation, has the advantage of providing results that can be applied to systems with direct benefit to users and operators alike (Barraclough 1981). The performance of an operational IR system can be evaluated by testing one or all of its three main functional parts: search formulation, searching, and output. Some of the common methods of collecting data for this type of evaluation are: logging search sessions, surveying users, and observing the search behavior of users. The obvious observation here is that the human factor (the user) is at the center of the evaluation process: there are real users with real needs. Therefore, the major difference between evaluating an operating IR system and evaluating a system in a laboratory environment is that in the latter case some form of an 'ideal' performance must be set as a standard for performance (Lancaster 1981).

To sum up, evaluation is essential to isolate the sources of weakness and areas for improvement in existing systems, and recall and precision have been the major measures of retrieval effectiveness for a long time. Are recall and precision measurements as employed by IR evaluation studies appropriate for evaluating the ability of an ELIR system to handle other languages?

4.2 CLIR

The term CLIR has been adopted recently to denote an area of research and development that has also appeared in the literature as multilingual IR (Hull and Grefenstette 1996) and as translingual IR (Carbonell et al. 1997). The main pre-occupation of CLIR research is to solve the problem of matching queries and documents across different languages. Why is it a problem, and why is it necessary to pursue this type of research? With the advent of the Web and the growth in the diverse linguistic communities that make use of its services, it became apparent that IR systems in the future might have users who do not understand the language of the stored documents and, therefore, are unable to formulate queries to match against those documents. Although users might have some foreign language knowledge, their proficiency might not be good enough to appropriately express their information needs in the language of the documents. The designers of traditional IR systems did not have this problem to deal with. In most cases, everything was designed in English without any thoughts given to the difficulties that may face non-English speakers. Researchers and developers of IR systems now, however, are considering mechanisms to cross language barriers between users and systems. For example, many search engines now provide translation services for retrieved pages, and a few services even offer translation of the queries.[1]

[1] (e.g.): http://crl.nmsu.edu/users/madavis/mundial.html

The origins of CLIR can be found in the work dealing with the development of CLIR systems in 1964. In that year, the International Road Research Documentation system was developed using a controlled-vocabulary thesaurus with index terms in three languages: English, French and German (Pigur 1979). Another system was developed by Pevzner (1969), who used English and Russian to experiment with retrieval. Salton (1970) augmented his SMART system with hand-constructed English-German index terms utilizing a thesaurus containing entries in these two languages. He set up a small collection of English documents and their German translations, and created a set of parallel queries in the two languages. Salton matched German queries against English documents, and English queries against their equivalent German documents, concluding that the manual translation of queries from English to German did not negatively affect the retrieval performance.

This early research, as well as the operational systems developed, used controlled indexing rather than natural-language indexing. This approach simplifies CLIR, as it is only necessary to accurately translate the thesaurus of controlled terms into the second language to produce good CLIR (as Salton (1970) did and demonstrated). But current research on CLIR has become much more challenging, because it has moved from a controlled to a natural language environment. In such cases, it is no longer sufficient merely to develop a bilingual thesaurus that will convert the controlled index terms used in the query into their equivalents used to represent the stored documents. The volume of

research is growing, and TREC has included a CLIR track beginning at TREC-6 in 1997. Since then, CLIR experiments at TREC have focused on retrieval evaluation studies initially using collections of documents in English and French, and then expanding to cover Italian and Chinese.

Most CLIR research has tackled the issue of finding the best approach to query (and in some cases retrieve document) translation, and the problems that it creates depending on the languages being used. Adriani (2000), for example, focuses on resolving term ambiguity in translation between English and Indonesian. She sees ambiguous terms as one of the main factors affecting translation and, consequently, the effectiveness of retrieval, but she also found that differences in word-formation patterns between English and Indonesian render query translations from one language to the other difficult.

Jones et al. (1999) report the results of an investigation into English-Japanese CLIR. They employed different query translation methods and found out that full machine translation outperforms dictionary-based translation when this method is applied to queries with little linguistic structure. On the other hand, Nie et al. (1999) deal with CLIR based on parallel texts and automatic mining of parallel text from the Web. They used a probabilistic translation model for CLIR in English and Chinese, and found that the performance of this approach equalled that of machine translation.

Working on English-Spanish CLIR, Ruiz and Srinivasan (1998) advocate the use of a meta-thesaurus instead of dictionary-based translation. Hedlund, Turid, Pirkola and Jarvelin (2001), analyze Swedish from the viewpoint of CLIR and show that this language has unique word-formation features that necessitate correct word normalization and compound splitting in a dictionary-based CLIR.

A query translation model was developed by Sperer and Oard (2000), based on a structured bilingual dictionary (English-Chinese) in which the translations of each term are clustered into groups with distinct meanings. They adopted a novel approach to query translation, where a query passes through a two-stage process: The system first determines the intended meaning of its terms and then selects translations appropriate to that meaning.

In a CLIR system, documents may be in English or any other language, and queries can be entered in English or any other language to retrieve them. In an ELIR system that contains documents in other languages, these documents can only be retrieved by queries in these languages: an English query cannot be used to retrieve non-English documents. On the Web for example, a page containing Russian words can only be retrieved by entering a query that contains one or more of these words in Russian. CLIR research has so far focused on translation methods, and even on the few occasions when linguistic matters relating to the search algorithms were investigated this was done to facilitate translation. Little attention has been paid to the morphological structure of the search terms, or to

indexing and search techniques. For example, the issue of stemming has not been discussed in CLIR, and IR problems related to individual languages have not been treated. In reality, CLIR and monolingual IR cannot be separated. Translation is only a tool for CLIR, and its use does not mean that all the problems associated with monolingual IR will not exist in CLIR. Building successful CLIR systems has to take into account all these problems and deal with them at the individual language level.

4.3 IR in languages other than English and Arabic

The early works in IR concerning problems that could be related to the (natural) language of the retrieval system are best labelled as being descriptive. For example, Zheng and He (1986) describe the problems associated with character entry and encoding in Chinese, which they consider to be the primary problems in Chinese IR. They discuss Zheng's Code, which is a completely new system designed in 1985 at the Chinese Academy of Agricultural Sciences' Institute of Information of Agricultural Science and Technology. At that time, three systems designed in the People's Republic of China were generally considered among the best. However, they were not widely accepted and used since each scheme required an expanded keyboard in order to accommodate all possible configurations needed for Chinese vocabulary. Due to the ideographic nature of the Chinese written language, an efficient and logically designed Chinese character coding system for computer storage and retrieval was needed, and

Zheng's Code was developed to overcome this problem by using a standard QWERTY keyboard for data input.

Shi and Larson (1989) investigate ways of providing effective techniques to enter and access stored Chinese information. They start with a description of three fundamental differences between English and Chinese IR: differences in character encoding, character storage, and character entry. They also discuss how to facilitate character entry and retrieval by regular expression searching. A regular expression in text editing and in information retrieval is a search pattern composed of a mixture of symbols and metasymbols. The symbols match exactly the same symbols in the source file; metasymbols on the other hand, have special meanings that are specified by the system designer. Regular expression searching in Chinese character entry and retrieval has the advantages of saving users' search time and mental effort, and reducing mental model mismatch errors. But regular expression searching will also retrieve noisy information because of the metasymbols contained in the regular expression. The authors developed mathematical models to control noisy information and to conduct cost benefit analysis in regular expression searching for Chinese characters or character strings. Four sets of mathematical models were worked out based on the assumption of random naive user searching. These models can be used to fulfil three tasks: (1) analyze the benefit and cost of employing metasymbols in the regular expression and find regular expressions which provide the largest net benefit; (2) for a given amount of acceptable noisy

information, find regular expressions which employ the maximum number of metasymbols; and (3) for a given number of metasymbols, determine regular expressions which generate the minimum amount of noisy information.

The problem of automatic Chinese text segmentation is tackled by Wu and Tseng (1995). Because Chinese texts do not contain word boundaries, they cannot be readily segmented. The authors developed an automatic segmentation system as a prototype for Chinese full text retrieval. The idea of this system is to apply partial syntactic analysis (analyzing morphemes, words, and phrases). It was built on the hypothesis that Chinese words and phrases exceeding two characters can be characterized by a grammar that describes the concatenation behavior of the morphological and syntactic categories of their formatives. The hypothesis was examined through segmentation, category disambiguation, and parsing. The experiment was carried out on a small sample of 30 texts, and it showed that the majority of significant words and phrases can be retrieved with a high degree of accuracy.

Hebrew IR research has been done mostly in Hebrew, and the few available English works describe existing operational systems or systems that support the Hebrew script, in addition to exploratory studies dealing with different aspects of cataloging and transliteration (Vernon 1991). One widely used IR system that can handle Hebrew is ALEPH, developed in Israel in the early 1980s. Lazinger and Levi (1996) provide a detailed description of the ALEPH system, tracing the

history of its development and presenting technical details about its ability to handle multilingual information storage and retrieval. Aliprand (1990) discusses in detail the features of RLIN (Research Libraries Information Network), which support the Hebrew script. The bibliographic utility provides original-script cataloguing of and searching for Hebrew materials.

Much of the research done on Russian IR has been reported in Russian-language publications. The few English works that exist deal mostly with the transliteration of the Cyrillic script that is used in Russian. For example, Pasterczyk (1985) outlines a strategy to help online searchers of Russian in large scientific databases overcome the problem created by the numerous schemes in use for the transliteration of the Cyrillic alphabet. More comprehensive research has been carried out by Aissing (1995), who discusses the problems created by the wide diversity in the current practice of transliterating Cyrillic scripts for use in bibliographic records in OPACs. Many different transliteration tables have been used, and without knowing which table was used it is difficult to retrieve desired records successfully or efficiently. The author explores the problems besetting three groups of Russian-language students, at Florida University at Gainesville, faced with Romanised Cyrillic bibliographic records, and he investigates the students' ability to search the Russian records according to the Library of Congress transliteration table. He concludes that transliteration is one of the factors limiting access by Russian-language students.

While descriptive works have their value, the most important contributions to IR research in different languages started to appear in the early 1990s, with a focus on the morphological properties of these languages and on stemming. Popoviç and Willet (1992) examine the use of stemming on Slovene-language documents and queries. This language is similar to English in that variant word forms are created by adding suffixes to a basic term. However, there are big differences between the morphological structures of the two languages, because the morphology of Slovene is more complex. This in turn means that a stemming algorithm for Slovene will need to be more complex than a stemming algorithm for English. For example, there are no less than 94 different forms of the stem raziskova* (for research).

The stemming algorithm used by Popoviç and Willet was developed by them (1990). It has a list of common suffixes, and rules that govern when each suffix can be removed. The list of suffixes is very long (more than 5270), and each suffix is accompanied by a minimum stem length to control the allowable stem length after suffix removal. To test the algorithm a collection was constructed containing the abstracts of 217 articles and the full text of 287 articles. These constituted the great bulk of all of the articles that had ever been published on library and information science in the Slovene language. A set of 48 queries was searched against the database in three ways: stemmed form, non-conflated form, and by an experienced intermediary, who used truncation as was felt appropriate. The stemmed and the truncated searches gave similar results that

were far superior to those obtained when conflation was not carried out. Statistically, there was a very substantial performance difference between the conflated and non-conflated text presentations. This difference is far greater than that observed in tests of stemming algorithms with English-language document test collections. The authors conclude that suffixing can be very effective in information retrieval, if the language used has a sufficient degree of morphological complexity.

The morphology of Modern Greek and its role in IR was investigated by Kalamboukis (1995). The researcher listed 41 different inflectional suffixes in the Greek language and argued that any attempt to store every possible form of every word in a database would require a vast amount of computer storage space. He presented a simple procedure for stripping the suffixes of Greek words covering both inflectional and derivational morphology. The performance of this suffix-stripping method was evaluated within an IR system with acceptable results.

Working on the Malay language, Ahmad, Yusoff and Sembok (1996) developed an algorithm for stemming Malay words. The researchers argue that the identification of the correct root for each word in Malay text is necessary for indexing purposes. A data set was used to test the algorithm and check for errors, but this algorithm was not tested in an IR context.

Ekmekcioglu and Willett (2000) investigated the Turkish language. They conducted an experiment using a file containing the titles and abstracts of 6,289 economic and political news stories extracted from newspapers in the period 1991-1993. Using a morphological analyzer, the researchers applied stemming to the indexes and experimented with stemmed and non-stemmed searches. The searches were conducted using queries provided by 30 native Turkish students, who also provided the relevance judgements on the stemmed and non-stemmed search outputs. It was concluded that stemming algorithms are necessary in Turkish IR.

4.4 Stemming

IR systems can be characterized by the human language within whose boundaries they are operated. Understanding differences among languages might be the key to developing IR systems with real multilingual capacities, enabling storage and retrieval at equal levels of efficiency and flexibility. Blair (1990) considers the way in which a document is represented in an IR system to be the fundamental issue of IR. This representation is a linguistic process, and the problem of describing documents for retrieval is, first and foremost, a problem of how language is used. Understanding how documents should be presented for effective retrieval is primarily a problem of language and meaning. In theory, the same should apply to indexing procedures and to formulating search queries, because the success of a particular search and the quality of

retrieval performance depends chiefly on the match between representation (indexing) and the requirements of the individual query, and on the adaptation of the query formulation to the characteristics of the retrieval system. Therefore, the formulation of the search query can be described as a linguistic process too, lending great importance to the argument that the linguistic properties of a given language affect the entire process of information retrieval.

Stemming is a common approach adopted by IR retrieval systems designed for use with English, and it has been studied by, among others, Lovins (1968), Porter (1980), and Salton (1971). Salton (1971) concluded that suffix removal improved the effectiveness of retrieval, while Harman (1991), on the other hand, suggests that when dealing with an online system, stemming should be applied differentially (to some queries but not others). Stemming is essentially a linguistic procedure performed at the query stage by using truncation. The purpose of using stemming with words is to achieve a quick approximation to the word root: "A word for which we want to find an exact or near match may be written as a stem or root word, and the retrieval system, asked to find words in storage that match the root" (Meadow 1992). At the search stage, users of IR systems are usually provided with the means to search on parts of words through the use of truncation, which can be applied in four ways: right truncation, left truncation, simultaneous right and left truncation, and middle truncation. Respectively, right and left truncation ignores the ending and the beginning of a

word, while infix truncation specifies the beginning and the end of a word and leaves the middle unspecified (Lancaster and Warner 1993).

A key article researching stemming was written by Harman (1991) who investigates the interaction of suffixing algorithms and retrieval techniques in retrieval performance in an online environment. Three general purpose suffixing algorithms were used: An "S" stemming algorithm, the Lovins (1968) algorithm, and the Porter (1980) algorithm. These algorithms were tested on three test collections: Cranfield 1400, Medlars, and CACM.

The "S" stemming algorithm is a basic one, conflating singular and plural word forms, and it is used for minimal stemming of words that have three or more characters. Its principal usage is to remove, when grammatically appropriate, the ending of plural words and restore them to their singular forms. The Lovins stemmer operates in much the same manner as the S stemmer, although at a higher level of complexity. First, it finds the longest possible suffix to allow the length of the remaining stem to be two characters or greater. This stem is then checked against an exception list for the given suffix. If the stem passes, it is processed into the final stem in a cleanup step. The Porter algorithm looks for about 60 suffixes in order to produce a word variant conflation that is intermediate between the S algorithm and the Lovins algorithm. Instead of removing the longest possible suffix, this technique uses the successive removal of short suffixes.

To provide information about the variation of stemming performance, sets of queries were matched against records from the above three databases. None of the three suffixing techniques achieved any significant improvement over term weighting when used in the Cranfield collection. The same results were found in the second database (the Medlars Collection). On the other hand, the CACM collection showed the most improvement in performance for suffixing, but the improvements were not statistically significant. The author attributes the lack of meaningful improvement for stemming to the fact that improvement and degradation in performance are produced simultaneously in stemming. Stemming adds non-useful terms, which cause non-relevant documents to have higher ranks (in terms of occurrence), and therefore often lowers the ranks of the relevant documents. Keeping the above in mind, Harman suggests some recommendations for the use of suffixing in online environments. First, stemming reduces storage requirements, especially for small databases on machines with little storage. Second, since the use of a stemmer is intuitive to many users, some type of selective stemming should be used, truncation for example. Finally, the automatic use of stemming algorithms like the Porter and the Lovins algorithms should be implemented in an online environment, but the ability to keep a term from being stemmed (the inverse of truncation) should be provided as well. This way, the user will have full advantage of stemming, while having the ability to improve the results of those queries adversely affected by stemming.

4.5 IR in the Arabic language

Interest in Arabic IR did not materialize until the 1990s. Before that, specialists in Arabic computing focused their efforts on presenting the language in a computer environment and finding solutions for display and coding problems. In the early 1990s, this changed, and research started to appear on the automation of Arabic online library catalogs and on IR issues. The literature on Arabic in electronic environments includes works ranging from descriptive articles to experimental research on IR systems. In between, there are evaluation studies of Arabic Online Public Access Systems (OPACs), overviews of automation, and works on indexing and thesaurus construction. To present the diversity of this literature, the review of works on Arabic has been divided into three parts: General works on processing and indexing, Library automation and OPACs, and IR experiments.

4.5.1 *General works*

Processing the Arabic language using a computer system has presented hardware and software developers with challenges related to the script itself and to the morphological structure of the language (Chapters 1 and 2). Nowhere is the latter more felt than in indexing, and subsequently, in retrieval. Experiments

with computer systems in this area have dealt mostly with finding ways to overcome the display, indexing and encoding difficulties.

An early work on Arabic in computer systems by Aman (1984) addressed the problem of Arabic computerised information exchange. Since the Arabic alphabet is completely different from the Latin alphabet, standard computer equipment is unsuitable for the Arab market. Librarians who attempt to use computers intended for the Roman alphabet quickly become frustrated by the need to transliterate Arabic bibliographic information. Script conversion through transliteration is both unfamiliar and cumbersome to Arab users who constitute in most library situations the major group for Arabic documents. Aman sought to identify problems associated with the use of Arabic in input and output devices, efforts then being made to introduce a unified code for the Arabic language, and Arabic in information systems. Advocating the need for more research, he concluded that the efforts to find a standardized approach to Arabic computerized information exchange had yielded some results.

Another early work by Ghani (1987) was concerned mainly with the Uniterm system for Coordinate Indexing, developed by Mortimer Taub in 1951 at the Technical Information Services of the Atomic Energy Commission in Washington D.C., and considered to be the most practical and most popular among the many pre-computerized indexing systems. Although its utility for storage and retrieval of literature in languages based on Latin alphabets had been

established, it had never been tried for Arabic. Ghani investigated the possibilities of using the Uniterm indexing system for storing information in Arabic and reached the following conclusions:

1. The frequent use of prefixes in Arabic words (especially prepositions and the definite article) scatters throughout an alphabetic sequence terms that are related to one word—a problem much less common in English.

2. There are many homographs in Arabic.

3. The usage of technical English words is common in Arabic, but there is no standard system of spelling.

Hegazi, Ali and Abed (1987) tackled the measurement of redundancy caused by the morphological nature of the Arabic language (compared to English, redundancy in Arabic was assumed to be higher, because Arabic words are derived from roots according to certain patterns, depending on fixed rules, in addition to suffixes, prefixes and infixes). Their study measured the information content per letter and per letter complexes. This kind of measurement can be helpful in many areas, such as information retrieval or text compression. In order to reveal the true characteristics of the Arabic language, full-text documents were used, i.e., full words as they appear in any text with their morphological extensions and not merely their roots. The n-gram technique was applied. (the n-gram is defined as a string of n letters occurring frequently in a

text, justifying their consideration as symbols by themselves in addition to the symbols that comprise the text). Examples of the full-text documents that were used in the study are books, newspapers, and social magazines. Systematically, studies of the dependencies of characters on each other were done, as well as a study on the average distribution of word lengths. This identified the most and the least frequent characters in any Arabic text. By comparing the results with those from research on English, Arabic was found to have a greater redundancy, and the average word length for Arabic is greater than for English, making Arabic potentially more compressible than English.

Bachir and Baxton (1991) tried to provide a partial answer to the question of whether Arabic periodical article titles can be relied on as a basis for keyword indexing techniques. Another aim of their research was to compare the characteristics of Arabic titles with those of English titles, which according to previous studies have been found sufficiently informative to be used for indexing. They examined the information content of Arabic titles in 16 scientific and non- scientific fields by counting their number of substantive words and comparing the results with those for English periodical articles in the same subject areas. Although significant differences were found between the two samples in some subjects, such as agriculture, philosophy, linguistics, law, and library and information science, Arabic titles generally appear to be as informative as English titles. Where there is a difference, the main problem is that Arabic titles tend to be longer, and contain words that are not indicative of

the subject matter. Some practical problems are found in using Arabic titles for indexing, for example, the need to strip prefixes from keywords, and the presence of some words in Roman rather than Arabic script.

Sakai, Terashita and Takenmoto (1986) presented the results of an experiment at Knanazawa Institute of Technology (KIT), Japan, to develop a prototype system that can manage catalogue records of Arabic materials in computerised form, by adopting a 16-bit character-encoding scheme. Another purpose of their study was to demonstrate that the 16-bit encoding scheme can be used as the technical basis for developing international bibliographic information systems capable of integrating textual materials of various languages in an effective way. More specifically, the intention was to show that the above technology could further be extended to include Arabic information, whose characteristics are somewhat different from other common languages. In order to examine the feasibility of a 16-bit encoding scheme for Arabic bibliographic information, an experimental Arabic database was constructed for retrieval experimentation. The 16-bit character-encoding scheme offered greater ease and flexibility than the conventional 8-bit scheme. Since Arabic and non-Roman characters are defined on a different portion of the single, large bit-code domain, Arabic catalog records, even with accompanying Roman texts, can be handled in a straightforward way. That makes it possible to store Arabic and non-Arabic records in a single database, controlled by a single computer system. Furthermore, although the experiment used a small database that could be stored

on a personal computer (PC), the authors suggested that handling much larger databases on a mainframe should not be a problem. The mainframe itself need not have multilingual capabilities; to access the database, it is sufficient to have multilingual PCs hooked to the mainframe that can manipulate the Arabic data and display them in the right form.

Musa (1986) focused on the technical problems encountered in processing bilingual Arabic/English text in an electronic environment. These problems are caused by the dissimilarity of the two languages. Three general problems were identified: 1) Written English has rules which sharply contrast with those of written Arabic (Arabic is written from right to left); 2) the shape of each Arabic character depends on its position in the word and on the nature of other neighbouring characters; and 3) almost all computer systems for processing text are built upon English's morphosyntactical structure, which is completely different from that of Arabic. Musa addressed the issue of printing and communicating in a bilingual Arabic/English environment, with the objective of building a system that would perform these tasks in a high-quality manner. The research presented a complete mathematical model for an Arab/English processor. To solve the problem of the different shapes of Arabic characters, an algorithm was developed to create the proper shapes without at that point taking into consideration the different directions for writing Arabic and English. Then, another algorithm was developed to take this output from the first one and put

the Arabic and English into two different buffers, subsequently merging the buffers to produce a high-quality display.

The problem of handling Arabic text compression was tackled by al-Fedaghi and al-Sadoun (1990). Their research is concerned with finding a method to reduce the storage space necessary to contain Arabic text in a computer system, in order to decrease the cost of data storage. The morphological compression of Arabic text was thought to be the most effective compression method, replacing some words in the original text by their roots and morphological patterns. In order to examine its effectiveness and measure its reduction ratio, a new combinational method was developed and tested utilizing different texts. The morphological compression was performed in two steps. First, a triliteral root for a compressible word and a morphological pattern were extracted; and second, the compressible words were stored in a three-byte format while the uncompressible words were stored at one character per byte. Large sample data were used to test experimentally this morphological compression scheme. The reduction effect of the morphological property of the language was between 25% and 31.2%, but if the method is used in conjunction with other compression techniques (space elimination from the original text), it is not difficult to achieve reduction ratios of above 40%.

Salem (1991) discusses the construction of two thesauri developed in the Arab region: the Arabic Thesaurus in Social, Economic, and Political Activities

(ATSEPA) and the Arabic Petroleum Thesaurus (APT). ATSEPA was the first Arabic thesaurus to be developed in the Arab region, starting in 1980. APT, the second such project, (developed for the Arab Petroleum Training Institute) was started in 1985 and finished in 1987. The construction of these thesauri revealed the major problems that face thesaurus construction in Arabic. Salem summarises the problems and ways to alleviate them as follows:

- In order to avoid redundancy and the confusion caused by the use of plurals and singulars in descriptors, words chosen as descriptors were used in general in the singular form, except in those cases where the plural form was the only choice, because the singular would give a completely different meaning (e.g., the Arabic equivalent of public relations will give a different meaning if used in its singular form).

- The definite article (*al*) created sorting problems, and it was decided to discard it except in compound descriptors, and in geographical or proper names.

- Many slang terms were present, due to the wide range of dialects used in different Arab countries. The decision was made to use the standard Arabic terms and discard the slang terms.

- The use of transliterated terms, such as "television" and "radio" was minimized, and an effort was made to substitute them with their Arabic equivalents.

- A decision had to be made on the use of singular, plural, masculine, and feminine forms of words. The plural is especially a problem, because it has multiple forms and the majority of them are irregular. The feminine form of a word may have the same letter as other terms that are related to the same word. It was decided to use the singular (as mentioned above) when possible, and to always use the masculine form of a word.

4.5.2 *Library automation and OPAC evaluation*

Library automation in an Arabic-language environment requires Arabised systems that can fully utilise Arabic script for input and presentation of data as well as system operation and management. This practice is not yet widespread, but several projects have been undertaken in Saudi Arabia and Kuwait. In addition, a few library and cataloging systems can handle the Arabic script, including ALEPH and a system implemented by the Research Libraries Group (RLG) through its online bibliographic utility, the Research Libraries Information Network (RLIN) in California.

RLIN, a multi-million-record database, is the creation of the RLG, whose members include academic and research institutions in the United States, Canada, and Europe. Records on RLIN are in some 350 languages that utilize non-Roman scripts such as Arabic, Chinese, Cyrillic, Hebrew, Japanese (Kanji and Katakana) and Korean (Hannon 1992). RLIN is the world's largest bibliographic database for material in Middle Eastern languages: Arabic, Persian, Hebrew and Urdu are the major such languages represented in the database. Most of the Middle Eastern language records are in ALA/LC Romanisation, and the majority are completely Romanised. However, since RLIN's Arabic script capability was released in November 1991, Arabic, Persian and Urdu records have been entered in their original Arabic script.

In her article on the implementation of the Arabic script on RLIN, Aliprand (1992) provides a comprehensive description of how Arabic materials are indexed. The extensive and powerful indexing of the bibliographic system allows for searching on names, phrases, or words, as well as on call numbers, ISBNs and LCCNs. All Arabic fields are as fully indexed and in the same way as their Romanised equivalents, and search expressions can be written in multiple character sets (for example, an English name and an Arabic title phrase can be combined in one search statement). In Arabic records, the title, subtitle, and series statements are the core fields that are indexed (in the title word and the title phrase indexes). While these indexes can be used for minimal searching, the inclusion of other access points is left to the discretion of member libraries.

The search engine on RLIN offers right-hand (suffix) and internal truncation, but left-hand (prefix) truncation (crucial for Arabic) is not supported. A partial solution to this problem is the implementation of particle removal, which removes the definite article *al* (the) as well as some other particles (Aliprand does not state exactly which ones). Since the main searching field available for Arabic-script documents is the title, it can be used for topical searching when standard English subject headings are inadequate. The option of searching for nouns without particles in this field was at the implementation stage when Aliprand was writing. Base word indexes were planned to allow searching for particle-less words, that is, a search for an Arabic word would retrieve any records containing that word, either with or without a particle.

Vernon (1991) deals with non-Roman script languages in an automated library environment, identifying the issues raised in the case of two such scripts: Hebrew (Hebrew and Yiddish) and Arabic (Arabic, Persian and Ottoman Turkish). When a library decides to automate its catalogue, policy decisions have to be made regarding Arabic and Hebrew script materials: Romanisation versus the original script. Either way, problems are bound to appear. The problems posed by Romanisation can be classed as "theoretical", i.e., how to represent the Arabic and Hebrew scripts in a meaningful and efficient way; using the original scripts poses problems that can be described as "technical", i.e., how to display and sort information in a different, that is, non-Roman script. Vernon noted that while the drawbacks of the Romanisation system should be

acknowledged, they will be with librarians for quite some time. Although advances in computer technology have provided new possibilities for using Hebrew and Arabic scripts within the online record, they have not eliminated the need to Romanise, because this method is cheaper and more accessible, due in big part to the fact that many libraries share their information through bibliographic utilities--it is impossible to use Arabic and Hebrew scripts in cataloguing and share the bibliographic records with libraries that do not have the capability to utilise them.

While RLIN's development of an Arabic script capability appears to be mainly directed towards helping universities in North America handle their Arabic-script holdings, the development of Arabised OPACs in the Middle East aims at providing fully operational Arabic online catalogues that satisfy the needs of native speakers and facilitate access to Arabic material in the region's libraries and research centres. Two particularly successful Arabisation projects are the Arabic versions of DOBIS/LIBIS, and MINISIS. DOBIS/LIBIS is an integrated library automated system developed in Europe specifically to handle library applications. IBM provides the system as a software package for mainframe computers. In 1990, the Arabic version of DOBIS/LIBIS was running in eight libraries in the Middle East. MINISIS was developed in Canada by the International Development Research Center (IDRC) for use on a Hewlett Packard 3000 minicomputer. It is a general-purpose information management program that is being used by many libraries for major library functions; its

Arabic version was used in 1990 by 18 libraries in the Arab world (Chaudhry and Ashoor 1990).

A background work on Arabic library automation has been prepared by al-Anzi and Collier (1994) who discuss Arabisation and its possibilities in the future. They argued that the most important development in Arabisation was the conversion of DOBIS/LIBIS and MINISIS for use in Arab libraries. At the technical level, they conclude that modern library systems vendors who claim multilingual and multi-script capability do not seem to understand the problems of information retrieval in Arabic. The basis of their argument is that converting text from English to Arabic is one thing, and developing effective OPAC retrieval is something else. There is a need to develop OPACs that take into consideration the characteristics of the Arabic language, and that use appropriate indexing and searching software. Keeping that in mind, and based on the results of the survey of the development of Arabisation, it is suggested, without any further elaboration, that in order to solve the problem of an effective Arabic OPAC, lexical analysis of the Arabic language will be necessary.

The first work to deal with Arabic OPACs was the evaluation of DOBIS/LIBIS and MINISIS conducted by Chaudhry and Ashoor (1990). Their research was motivated by the need to respond to questions about the suitability of these particular systems. They examined data on the functions, performance and user satisfaction of the two systems. In order to collect this data, the major functions

and features of the two systems were grouped into ten categories, and each category was divided into five further sub-functions or components. The designated ten categories were: acquisition, circulation, periodicals control, cataloguing, online public access catalogue, management information, processing Arabic information, support services, documentation, and special features. A systematic comparison of the two systems was conducted using the data collected on the above ten categories and applying a scoring scheme to assess their functions. The study showed that both systems are very good for handling library automation, but DOBIS/LIBIS excelled in circulation and periodical control work, while MINISIS was superior in cataloguing, OPAC handling, and dealing with Arabic data.

A second evaluation of the Arabic version of DOBIS/LIBIS was conducted by Khurshid (1992) through a case study of automation at the King Fahd University of Petroleum and Minerals Library (KFUPM) in Saudi Arabia. When the initial planning for automation started in early 1975, the system features that KFUPM included were: integration, MARC and AACR/AACR2 compatibility, distributed access throughout the campus, multiple language capabilities, networking capabilities, and IBM compatibility. After the initial period of investigation, DOBIS/LIBIS was chosen over four other systems because its multilingual capability was considered to be appropriate for adaptation to Arabic. However, the Arabic version of DOBIS/LIBIS was not developed until 1986 and installed in 1987. Khurshid discusses the implementation of the Arabic

version of DOBIS/LIBIS and its effect on the library community, and identifies some of the main problems and limitations of the hardware and the software. The hardware limitations are mostly related to the keyboard, because it does not support the various diacritical characters found in the Arabic script. Khurshid cites as an example of this limitation the escape character, which is used in combination with alpha and/or numeric characters to form substitutes for diacritical characters. Regarding software problems, he identifies the input, sort, and display forms of the Arabic definite article "*al*" as the major problem. If this article is not ignored in sorting, it would result in a large number of entries being clustered together in the file and would impede searching. To alleviate this problem, "*al*" is generally ignored in filing. This results, however, in sorting problems with words that start with "*al*" as an integral part (rather than as a definite article) where it should not be ignored in filing. Khurshid concludes that for any system to become successful in the Middle East, it must support processing of Arabic script materials. The users of the library were satisfied with the Arabic OPAC, because it has almost all the features of an English-language OPAC, and because it is more complete than the previous card catalogue.

4.5.3 *IR experiments*

The differences in morphological structure between English and Arabic have inspired most of the research conducted so far on Arabic IR. It mainly deals with using word roots and stems as index terms in collections of bibliographic

records, based on the assumption that the affix-rich morphology of Arabic will make any other indexing method ineffective.

The first experiment that heralded interest in Arabic IR was conducted by al-Kharashi (1991), who explored the problems of storing and displaying Arabic bibliographic data, selection of index terms, ranking of Arabic records, and stemming algorithms for Arabic index terms. This work was supplemented by that of al-Kharashi and Evens (1994). The basic goal of the two works was to find the best way to solve the problem of stemming for documents in Arabic. To test the proposed indexing methods, the Micro-AIRS System, a microcomputer system for Arabic information retrieval developed by al-Kharashi, was used. A series of experiments was performed using three indexing methods: the word itself, the stem, and the root. The root is defined as a bare verb form that can be triliteral, quadriliteral, or pentaliteral. The stem is a combination of a root and derivational morphemes to which one or more affixes can be added. The bibliographic records were extracted from the databank at King Abdulaziz City for Science and Technology in Saudi Arabia. A small word-stem-root dictionary was created and used during the indexing and retrieval process to identify the stem or the root of a given word and also to identify stop words. In order to assess the effectiveness of the three indexing methods, 29 queries were performed against a database of 355 Arabic bibliographic records, covering computer and information science. The results demonstrated the superiority of root/stem-retrieval methods over word-retrieval methods, and underlined the

contrast with IR methods in English. Moreover, the root performs as well as or better than the stem at low recall levels and definitely better at high recall levels. This experiment was limited in scope, however, because the collection had short records without abstracts, and the title field alone could be used for information retrieval.

Abu-Salem (1992) constructed an experimental Arabic IR system with 120 records (fewer than al-Kharashi) but this time including abstracts. He used the same indexing methods as al-Kharashi (1991) and repeated the latter's experiments. He confirmed the results of al-Kharashi, rating roots as the best indexing terms in Arabic, followed by stems and words. He also concluded that the presence of abstracts improves retrieval regardless of the indexing method, and that the interactive use of a relational thesaurus, linking morphologically related words, gives the same good results as using roots as index terms.

Building on the experiments of Abu-Salem (1992) and al-Kharashi (1991), Hmeidi, Kanaan and Evens (1997) built a database comprising 242 records, all with abstracts, with the intention of determining the usefulness of automatically indexing Arabic words and investigating the use of roots, stems and full words as index terms. The authors defined automatic indexing as a task performed by a program that would take Arabic text and index every word according to specific rules and guidelines. Traditional measures of recall and precision were applied to searches using manual and automatic indexes, and the superiority of the latter

was proved. One reason given for the feasibility of automatic Arabic indexing is that Arabic words typically appear less often than English ones. This has to do with the pattern and root rules mentioned above and with the morphological structure of Arabic. Because one root can produce a large number of words, and many words are created by adding affixes and connecting the definite article "*al*", a large proportion of Arabic roots will appear only once, making the frequency of index terms (roots) low. As for index terms, this research found that Arabic documents were best indexed by word roots, because root indexing increased recall and bypassed complex problems created by Arabic morphology: a root index term would retrieve all variations of this root and eliminate the need to enter complex search queries. As for the effectiveness of searching, the authors argued that roots made better index terms than words or stems, at least when phrases were not involved.

4.5.4 *Conclusion*

Most of the research on Arabic language processing and IR have focused on the script, on the linguistic properties of the language in general, or on its morphological structure in particular. Display, encoding and indexing problems have been studied; some have been resolved and some are still being debated. Crucial to IR is the treatment of Arabic morphology for indexing and retrieval purposes. Stemming and root indexing have been adopted by researchers as necessary tools for effective IR. Complex linguistic analyses have been

conducted to prove this point, but the feasibility of implementing Arabic IR tools in an ELIR system has not been discussed. In the experimental Arabic IR systems that have been developed so far, stemming and root indexing have been employed to find word variants, and their effectiveness in IR environments has been measured using recall and precision.

5. Search engine selection

As a prelude to the research methodology, this chapter discusses how the two search engines used in this research were selected, and describes their retrieval capabilities.

5.1 Search engines

The Web offers a publishing medium for all written languages whose user communities have access to the technology, and it presents these communities with an opportunity to have their cultures and ideas introduced and disseminated throughout the world. The increasingly multilingual environment offered and fostered by the Web has stimulated interest in the search for and development of better and more efficient tools capable of handling multiple languages. Services and technologies such as translation tools, multilingual HTML coding, and character-encoding schemes have become popular, and the steady increase in the volume of multilingual information will probably contribute to the appearance and development of new tools to handle the demands of linguistically diverse user populations.

The Web holds a huge number of documents and links that cannot be easily accessed without sophisticated tools and search services. Two major types of

search services have facilitated access to Web information: directories and search engines. Directories rely on hierarchical subject classification schemes that employ browsing as the primary access method; search engines index the content of Web documents, and then permit users to query these indexes and retrieve information. Search engines constitute the primary search approach for 85% of Web users (Lawrence and Giles 1999), and they have evolved in the last few years into sophisticated tools.

There are many English-language search engines on the Web, and most provide a full array of sophisticated search commands and advanced search capabilities (Savoy and Picard 2001). Common indexing and search capabilities include Boolean search, inclusion or exclusion of terms, truncation (right-hand and middle), exact phrase matching, word proximity searching and case-sensitive searching (Schwartz 1998). The search engines are constantly changing, although many of the changes are superficial and cosmetic in nature, affecting only the appearance, layout or interface of the system without any substantial changes or additions to search features. Hock (2000) surveys search engine features and commands, focusing on the main searching features offered by eight major Web search engines and directories: AltaVista, Excite, Fast Search, Go Network, Google, HotBot, Lycos, and Northern Light. In addition to a simple retrieval approach, each of these engines provides an advanced version that often differs considerably from the simple version. For example, AltaVista has

AltaVista Advanced; Excite has Excite Power Search; and Lycos has Lycos Advanced.

5.2 English-language search engine selection

The selected search engine had to meet four criteria: word-by-word indexing, availability of truncation, capability to index Arabic words, and availability of a version that could be locally installed and controlled.

5.2.1 *Word indexing*

Most search engines employ a list of stop words that are not indexed and are ignored in searching. These include common English words which are usually articles, prepositions, or conjunctions like the, a, an, and, for. A few others (AltaVista and Northern Light) do not use stop-word lists and index every word in a Web document. Search engines developed for English do not have stop-word lists for other languages and the effect, if any, of their English lists on other languages is not clear and might be difficult to assess. For this research, it was essential that all words in a Web document (regardless of what part of speech they belong to) are indexed. This will eliminate any malfunction in indexing operations and ensure that every Arabic word is indexed.

5.2.2 *Truncation*

As opposed to the universal use of Boolean operators by search engines, truncation is not available on all of them. Some search engines provide truncation search capabilities; others provide simple automatic stemming (retrieving the singular and plural forms of the search term), while others do not offer truncation or stemming capabilities. For example, AltaVista and HotBot include truncation symbols to retrieve variants of words; an (?) may be used in place of one character, while an (*) can replace up to five characters. Northern Light provides automatic retrieval of singular and plural forms of words and gives users the choice of truncation search terms: A (%) replaces one character, and an (*) replaces multiple characters. Excite provides simple stemming, while Lycos and Google only search for exact words. The ELIR system to be used in this research had to include truncation so that it could handle the different and numerous endings of Arabic nouns.

5.2.3 *Non-Roman character handling*

A third criterion in the selection of an English-language search engine for this research is that it should have the ability to handle non-Roman characters, and especially Arabic characters. Just a few years ago, this type of capability was not available on most engines. In order for the engine to index documents in

non-Roman characters, its indexing mechanism must be able to handle the different encoding schemes that allow the presentation of non-Roman characters on the Web. For example, in order for a search engine to index Arabic Web documents, it must understand the encoding language used to represent these documents; if it does not, it simply skips the documents altogether or indexes whatever it understands in the document (there might be some Roman characters, for example). AltaVista, Excite, Google and HotBot index Arabic documents, while other engines, such as Go, Lycos and Northern Light, do not.

5.2.4 *Locally installable version*

Finally, for logistical reasons the search engine had to have a version that can be installed and controlled locally. This would provide the researcher with the ability to control indexing and searching operations, as explained in the next chapter.

The only English-language search engine available in 1998 (when the research was initiated), which met these four criteria was AltaVista. It indexes every word in Arabic texts, provides truncation capabilities, and was offered (at that time) as a personal version that could be installed and controlled locally.

5.3 Arabic-language search engine selection

The Arabic search engine had to meet one major requirement: it must offer root-searching capabilities. These capabilities are specifically designed for the Arabic language to retrieve all words that belong to one root, and they are probably the most difficult and costly features to implement in a retrieval system. Searching by the root of Arabic words retrieves the highest number of documents (al-Kharashi 2000) and, therefore, provides a test bench against which the performance of an ELIR can be measured.

The selected Arabic search engine is al-Idrisi, a Web-based engine that was available for use without charge. In 1998, al-Idrisi was the only Arabic search engine that provided root-retrieval capabilities. Another search engine (ArabVista) appeared in 2000. It is an Arabized version of AltaVista, and offered capabilities for root searching for a limited time (this search feature has disappeared recently from the engine's page without any explanation).

5.4 AltaVista

5.4.1 *AltaVista (Web version)*

AltaVista was developed in spring 1995 by scientists at Digital Equipment Corporation's Research Lab in Palo Alto, California, which owned and operated the engine until it was bought by Compaq Computer Corporation in January 1999. In August 1999 CMGI, Inc. acquired AltaVista, and it has been operating the engine since then. Today, AltaVista is one of the major search engines on the Web, receiving millions of queries every day.

In 1995 the scientists who developed AltaVista devised an indexing mechanism that could index every word in documents on the Web. This mechanism was the first to produce a full-text searchable database of Web documents. It was also the first major Web search engine to introduce multilingual search capabilities. Initially AltaVista allowed users to enter search terms in more than 15 languages (because the engine indexes Web documents published in these languages). Currently, AltaVista allows users to limit their Web searches to any one of 25 languages; this feature is provided through a pull-down menu from which the user can select the desired language and restrict searches to it. AltaVista was equipped with a machine translation service (Babel Fish) that can translate words, phrases or entire Web documents to and from French, German, Italian,

Portuguese, Russian and Spanish. The translation service has since been imitated by other search engines, and many now offer translation as a standard feature.

The AltaVista search environment consists of two major components that are important for understanding the operation of the search engine: the engine itself (http://www.altavista.com) and the interface, which allows queries to be submitted to the engine. As with other search engines, AltaVista is constantly changing and improving its interface.

The Web version of AltaVista is based on Boolean search algorithms provided through two major search modes: simple querying and advanced querying. The searching is performed against the full-text (every word) of Web documents that are indexed by the engine's indexing software. In simple querying, the user may enter one term or a collection of terms, where the default Boolean operator is OR, if none is used and more than one term are entered. The "+" and "-" signs may be used to indicate the presence or absence of a term respectively. When "-" is placed in front of a search term, AltaVista ignores Web documents containing that term (a NOT Boolean operator); a "+" in front of a term instructs the engine to only retrieve documents containing the term, acting as an AND Boolean operator. Double quotes ("...") may be used to enclose a phrase, instructing the engine that the terms within these quotes must be adjacent for a document to match the query.

In the advanced querying mode, more explicit use of Boolean operators is required. AND, OR, and AND NOT are interpreted as Boolean terms rather than as search terms, as in the simple mode. Also in this mode, the Boolean operator NEAR may be used between two terms to indicate that they must be close to each other (within ten words) in the document, without necessarily being adjacent. In addition to querying variations, the advanced searching page provides the user with tools to control aspects of the search in ways other than modifying the query. For example, the user can restrict the results to items added (or modified) on a specific date, or within a range of dates. The user can also specify how results are displayed and how many links to documents matching the query are displayed on each screen. By default, AltaVista returns a screen displaying links to a maximum of ten Web documents matching the user query. If a search produces more than ten documents, the documents are organized in numbered blocks, each containing ten documents. The default display of ten documents per screen can be changed in the advanced mode, where the user can choose 20, 30, 40, or 50 documents. More information on the main engine and its development may be obtained from its site (http://www.altavista.com).

5.4.2 *AltaVista (PC version)*

The indexing mechanism of AltaVista on the Web, its simple and advanced modes, and its search interfaces were incorporated in 1997 into a personal

version as a Windows application and can be installed on a personal computer. Its official name is "AltaVista Personal 97". For the sake of simplicity, it is referred to hereafter simply as AltaVista. It is a fully functioning version of the main AltaVista engine as it was in 1997, with the same indexing and searching features, albeit presented in a slightly different interface. It is this PC version that was used in this research.

In addition to indexing and providing a search interface to HTML files (Web documents), AltaVista also indexes a variety of other files that may be present on a user's computer. These include many file formats created by Microsoft software products and E-mail message files of popular clients like Outlook and Eudora. For the sake of simplicity and to ensure uniform terminology, files indexed by AltaVista will henceforth be referred to as documents. Starting in 1997, AltaVista was made available as free software, until a modified version called AltaVista Discovery was introduced in 1999; this version was available free for a short time, but was soon discontinued. Currently, there is no equivalent tool available from the company that owns the Web-based engine. Instead, a commercial version of the search engine is offered for sale to corporations and businesses that wish to index their files and make them searchable across Intranet networks. The following description of AltaVista is based on the documentation provided with the software.

Once installed on a local computer, AltaVista may be configured to index local documents, search them and provide a Web page as an interface to searching facilities. The main components of AltaVista are the Indexer, the Query Dispatcher, and the Browser. The Indexer is the tool that provides access to the indexing configuration and performs indexing operations. This tool can be configured to index specific types of files or specific folders on the hard drives. For example, the software can be instructed to index HTML documents in the "web" folder and ignore all other documents. Once the configuration is complete, a command is issued to index the documents, and an index file containing all words in the specified HTML documents is created. A precise count of the indexed documents and the number of indexed words are given at the end of the indexing operation. To search the index, the Query Dispatcher must be running and the Browser must be opened to enter queries. The Browser is the main search page that can be opened in any Web Browser, while the Query Dispatcher, as the name implies, is required to dispatch queries from the Browser to the index file.

The AltaVista PC Indexer works in the same manner as the Web AltaVista; it indexes every string of characters in a document. In other words, it indexes every word regardless of the number of characters it contains. A word is defined in AltaVista, as any string of letters and digits that is separated by either white space, or special characters and punctuation, such as "%", "$", "/", "#", "-", and

" _". For example, the Indexer interprets and indexes EVE5000, 89068, www, a, the, for, and EasierSaidThanDone all as single words, because they are continuous strings of characters, surrounded by characters that are neither letters nor digits. It will also consider "hot/cold" as two words and separately index them as hot and cold, ignoring the character "/". Following these rules, the Indexer collects all the words that it finds in a document, regardless of whether the word exists in a dictionary or is spelled correctly, and enters them as index terms in its index files.

Once the index is built, simple and advanced searches can be performed. Searching using AltaVista can be performed by querying the indexes created by the Indexer; it requires opening the Browser page in a Web browser. This page provides the interface to AltaVista and, by default, displays the Simple Search page; switching to Advanced Search is achieved through clicking a button on this page. Before going into the differences between the simple and advanced searches, let us look at some of the search rules and feature that they share.

Both the simple and advanced search functions use the same syntax rules regarding phrasing, case sensitivity and truncation. To indicate a phrase in a search query, the words must be enclosed within double quotes. This ensures that the engine finds the words together, instead of looking for separate instances of each word individually. If double quotes are not used in the query

(international relations), the engine would find instances of "international" alone and "relations" alone, as well as any instances where the two words happen to appear together as a phrase. Indicating a phrase can also be achieved by the use of punctuation in some instances. AltaVista ignores punctuation except to interpret it as a separator for words. In a few instances, placing punctuation or special characters between words, with no spaces between the characters and the words, is also a way to indicate a phrase. For example, the term CD-ROM is treated as a phrase: the hyphen is automatically interpreted as an indicator of a phrase (CD and ROM are not rejoined as a single word).

Case sensitivity is another rule shared by the simple and advanced search modes. To ensure a case-insensitive search, the query must be entered all in lowercase letters. For example, entering turkey in the query field, will instruct AltaVista to find all occurrences of the word turkey, including those spelled TUrkey, TURKEY, turkey, and so forth. Conversely, if a query contains any uppercase letters, the search is case-sensitive: AltaVista finds all occurrences of Turkey with initial capitalization only. It will ignore any documents containing the words TURKEY or turkey only.

The rules of truncation are the same in both simple and advanced searches. An asterisk (*) may be used at the end of a string of characters as a wildcard (right-hand truncation) to indicate that AltaVista should find all words containing a

match for the specified pattern of letters. This is convenient for finding derivatives and spelling variants of the same word. For example, to look for the word walk and any derivatives, such as walker, walkers, and walking, it is sufficient to enter walk* in the query field. The asterisk may also be used as a middle truncation (in the middle of a search term), to instruct AltaVista to find words that start and end with the same string of characters but have different ones in between (For example: wom*n to retrieve woman and women, and col*rful to retrieve colourful and colorful). In order to limit extraneous searching, AltaVista requires that at least three letters must be specified in front of the (*) notation: car* and den*m are allowed, while ca* and de*m are not. Also, the asterisk interchanges with a maximum of five letters: internat* will retrieve international but not internationally.

The main differences between Simple Search and Advanced Search lie in the way retrieved documents are ranked and the use of natural language and Boolean operators. In the simple mode, retrieved documents are ranked based on a series of factors that ensures that the most relevant ones appear at the top. These factors include:

1. The position of words or phrases in the document (higher ranking if they are in the first few lines of the document).

2. The frequency of occurrence of a word or phrase in the document.

3. Whether all of the specified words or phrases appear in a document. A document containing all three words specified in a three-word query would rank higher than a document containing only two or one of the words.

4. The proximity of query words to each other. The closer they are to each other, the higher the ranking of the document.

In the advanced mode, retrieved documents are displayed in no particular order. If the user wishes to rank them, he/she can do so by entering ranking rules in the advanced search interface.

In the simple mode, it is possible to use natural-language queries. In this case, AltaVista looks for documents that contain all or any of the words of the query and ranks the ones that contain all of them at the top. Alternatively, the use of the + and - symbols as simple operators that require the presence or absence words, gives the user control over which terms should be retrieved and which ones should not. The advanced search, by contrast, allows the user to enter more precise and logical syntax structures. The Boolean operators AND, OR, and AND NOT can be used to construct queries and they can be nested as in the following example: (history OR geography) AND Ireland AND Canada.

In simple and advanced searches, AltaVista displays the search results in the form of hyperlinks to the documents that match the query. The number of the retrieved documents is given at the top of the page, while a precise count of the number of occurrences of the search word is given at the bottom of the page. For example, a query containing the two words apples and oranges, could return 12 documents and the term count at the bottom of the page might look like this: apples 36, oranges 42. This means that the index contains 36 instances of apples and 42 instances of oranges.

5.5 Al-Idrisi

Al-Idrisi is a product from the Sakhr Software Company that specializes in Arabic computing solutions. The company was founded in Kuwait in 1982, and was relocated to Egypt after the Iraqi invasion of Kuwait in1990. Sakhr is a leading producer of Arabic products, including machine translation, word processing, Web browsing, and search and retrieval software, as well as speech recognition and educational programs. Information on these products and on the company may be accessed at its web site (http://www.sakhrsoft.com). Al-Idrisi was developed by Sakhr in 1996 to provide a search and retrieval mechanism for the growing number of Arabic documents available on the Web in general and on the company's site in particular. Until 1999, the original search engine was available free of charge on the company's site. This service now has been

discontinued, and al-Idrisi software is currently offered as a commercial product. As an alternative to al-Idrisi's site, Sakhr developed in late 1999 al-Dalil (http://www.aldalil.com), an Arabic directory and search engine modeled after Yahoo! Al-Dalil utilizes the indexing and searching features of al-Idrisi, but it did not retain all the features that were available on al-Idrisi's site. The following describes al-Idrisi as it was offered on the Web in 1999.

Al-Idrisi's indexing and search features were developed to handle the complex morphological structure of Arabic. It has features to accommodate prefixes, suffixes, and derivatives of Arabic words. It also indexes words based on their roots (see below). In 1999, al-Idrisi provided access to a small collection of Web documents (approximately 12,000 documents). These documents covered topics ranging from news items to technical reports, including literature, history, and geography. Most of the documents were produced by Sakhr and held on its servers, but others were collected from different Arabic sites on the Web.

Searches using al-Idrisi may be conducted on two different pages: the simple search page and the advanced search page. The simple page does not provide much control over queries, save for a menu that allows users to specify phrase searches and to search for any word or all words in a query. The advanced page, on the other hand, provides access to a wide array of search features that control the matching level of query words as follows:

1. Exact match: The engine matches the word with documents containing the exact match of this word (with and without diacritical marks). For example, *rjl* (man) is only matched to *rjl*.

2. Stem matching: Matches a word with documents containing the exact form of this word or any form of the word with prefixes, suffixes or a combination of both attached to it. Entering *wTn* (nation), for examples, searches for documents containing *wTn*, *alwTn* (the nation), *wTnh* (his nation) and *alwTny* (the national).

3. Derivative matching: This retrieves documents containing the exact word or any of its derivatives (words derived directly from the search word by a process that may involve attaching prefixes, suffixes, or infixes). For example, *slm* (peace) retrieves *alsm* (the peace), *slym* (safe) and *slamat* (greetings).

4. Root matching: This matches a word with documents containing the word or any word that shares the same root with it, including the words retrieved by stem and derivative matching. As a result, of the four types of matching, root matching is the one that retrieves the highest number of documents. It combines the results of the first three types with documents containing words derived from the root. Let us look at the

word *clm* (science), for example. Root matching will retrieve *clm* (exact matching), *alclm* (the science) (stem matching), *clwm* (sciences) (derivative matching), and a word like *clamh* (sign), which shares the root *clm* with the word *clm*.

The documents retrieved by al-Idrisi's searches are ranked according to relevancy criteria that include the number of occurrences of a search word in the retrieved document, the proximity between search words in the document, and the position of the word in the document. If a search query containing the word *mlk* (king) is entered in al-Idrisi, a document containing three instances of *mlk* will be ranked ahead of documents containing one or two instances. If two documents contain two instances of *mlk*, the document where these two instances are separated by ten words is ranked ahead of the document where they are separated by 12 words. If the two documents only contain one instance of *mlk,* one document is ranked ahead of the other when it has *mlk* in its first paragraph and the other document has it in its second paragraph.

The search results are displayed in a similar way to the display approach of other search engines: each retrieved document is represented by its hyperlinked title followed by the first few lines of the contents and another hyperlink (see below). However, al-Idrisi provides two ways to display the full document: regular display and highlighted display. The regular display is activated by clicking the

title of the retrieved document, which shows the document in its original format. Clicking a link below the first few lines of the document activates the highlighted display, where the document is displayed with the words that caused the document to be retrieved highlighted in red. For example, a document retrieved by using *mlk* with stem-searching option will have words such as *mlk* and *almlk* highlighted in red to indicate that they match the query criteria.

5.6 Al-Idrisi versus AltaVista

Al-Idrisi differs from AltaVista in the way it handles Arabic. As an engine designed specifically for the Arabic language, it offers indexing and searching features that are not offered on AltaVista. The morphological variations of Arabic words are handled through the four types of matching described above. AltaVista, on the other hand, handles the morphological variations of English words through truncation, a searching tool that could be implemented in Arabic texts.

6. Prefix identification

6.1 Introduction

As a preliminary to making comparisons between an ELIR system and an Arabic-language IR system, it was necessary to identify the prefixes and prefix combinations in Arabic that potentially these systems would have to cope with if they were to achieve high recall rates. The Arabic language is very rich in prefixes (see Chapter 2). Arabic nouns can accommodate up to 15 possible prefixes and prefix combinations (Cohen 1970). They are numbered here A1 through A15 for identification purposes and are listed in Table 6.1.

Table 6.1 includes six prefixes (A1 to A6) along with nine possible combinations of their occurrences (A7 to A15). It should be emphasised that in Arabic text up to three different prefixes can be attached to a word at the same time. The objective of this preliminary research step was to ascertain how many of these prefixes and prefix combinations have high occurrence rates that might affect the retrieval of Arabic nouns and documents.

Table 6.1. Arabic prefixes and prefix combinations (see Appendix A for the Arabic-script list)

Arabic	Possible English meaning(s)
A1: *al-* (P4)	the
A2: *b-* (P1)	in, inside, by
A3: *f-* (P3)	so, then, and
A4: *k-* (P2)	as, like
A5: *l-* (P6)	to, for
A6: *w-* (P5)	and
A7: *bal-* (P1+P4)	in the, inside the, by the
A8: *fal-* (P3+P4)	so the, then the, and the
A9: *kal-* (P2+P4)	as the, like the
A10: *ll-*(P6+P4)	to the, for the
A11: *wal-* (P5+P4)	and the
A12: *wb-* (P5+P1)	and in, and inside, and by
A13: *wbal-* (P5+P1+P4)	and in the, and inside the, and by the
A14: *wl-* (P5+P6)	and to, and for
A15: *wll-* (P5+P6+P4)	and to the, and for the

6.2 The test database

The first step was to create a test database. After preliminary investigation of Arabic sites on the Web, a site dedicated to the publications of an Egyptian religious scholar, Yusuf al-Qaradawi (http://www.qaradawi.net) was selected. It includes electronic versions of a number of his published books in their original Arabic. While the subject matter of all the books deals with religious issues, the subject and organization of a particular one, *the Lawful and Unlawful in Islam*, made it an appropriate choice, for the following reasons:

1. A test collection created from this book would be complete in the sense that it is a comprehensive coverage of a specific area of Islamic teachings.

2. It deals with al-Qaradawi's interpretation of Islam's governance of the daily and social life of its adherents.

3. It is divided into four chapters containing individual rulings and opinions on a wide range of topics.

4. Each one of these rulings and opinions has its own unique title, and therefore can be treated as a separate information record.

Reasons 1 and 2 are mentioned, because the document collection in this case can be realistically viewed as a simulation of a real database covering a specific topic or field of knowledge. Reasons 3 and 4 were taken into consideration because of logistical factors: the organization of the book makes it easy to construct individual documents and bypass the complications of creating a test database from scratch, a task beyond the scope of this research.

The entire text of the book was downloaded and saved. Next, section by section, the book was broken down into individual documents, each representing one ruling. Each ruling was copied and pasted into a new HTML file that was saved under the title of that ruling. In total, 271 rulings were thus identified and, consequently, 271 HTML files created and saved in one directory (955,492 bytes of storage space). The HTML files (documents) range in length between one paragraph and three pages. The indexing and searching software (see 5.4.2) was configured to index every word in each file, resulting in an index of 69,209 words.

6.3 Search queries

A list of 35 nouns was compiled from a set of 43 questions posed to al-Qaradawi by a group of his followers on topics covered in the test database, and published in a book called *ftawa mcaprh* (Current Rulings) that is available on the Web in Arabic (http://www.qaradawi.net/arabic/books/fatawa-moasera/index-all.htm).

A total of 376 unique nouns were extracted from the text of the 43 questions and numbered. Using the random-number-generator function of a calculator, 35 nouns were randomly selected. These nouns (with their English equivalents) are listed in Table 6.2. Using the prefixed and non-prefixed forms of the 35 nouns, systematic searches were undertaken with AltaVista. This involved taking each noun and searching it 16 times: once without prefixes as its English equivalent would be used (naked form), and 15 times with the six prefixes and their nine possible combinations. In total 560 searches were undertaken. Each search statement contained one term only in order to get an accurate figure for the number of retrieved documents for each word.

Table 6.2. Test nouns and their English equivalents (see Appendix B for the Arabic-script list)

	Arabic	English equivalent
N1	*rswl*	prophet
N2	*awlad*	children
N3	*qran*	quran
N4	*Tlaq*	divorce
N5	*zwj*	husband
N6	*kZb*	lying
N7	*dyn*	religion
N8	*tjarh*	trade
N9	*xmr*	alcohol
N10	*Scr*	hair
N11	*lbas*	clothes
N12	*tHrym*	prohibition
N13	*zwaj*	marriage
N14	*asrh*	family
N15	*Hq*	right
N16	*Hb*	love
N17	*arwaH*	souls
N18	*Hkm*	ruling
N19	*mslm*	muslim
N20	*Src*	law
N21	*Hlal*	lawful
N22	*Hram*	unlawful
N23	*lHm*	meat
N24	*dm*	blood
N25	*faadh*	interest
N26	*Tach*	obedience
N27	*Tcam*	food
N28	*ayman*	faith
N29	*mwt*	death
N30	*Srb*	drinking
N31	*bnt*	daughter
N32	*nfqh*	support
N33	*arD*	land
N34	*amanh*	honesty
N35	*rSwh*	bribe

6.4 Results

6.4.1 *Documents in naked and prefixed noun searches*

The numbers of discrete documents retrieved by the 560 searches against the 271 Arabic documents using non-prefixed and prefixed nouns are shown in Table 6.3. The first column contains the 35 search nouns, randomly numbered N1 to N35. The second column shows the number of discrete documents retrieved using nouns, in the naked form, with no prefixes attached to them. The rest of the table list the numbers of discrete documents retrieved after attaching the 15 prefixes or combinations of prefixes to nouns.

Table 6.3. Prefixed and non-prefixed (naked) noun searching: Number of retrieved documents

	Naked	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**	**A10**	**A11**	**A12**	**A13**	**A14**	**A15**
N1	97	58	1	0	0	1	1	0	0	0	1	2	0	0	0	0
N2	2	5	0	0	0	0	0	0	0	0	0	1	0	0	0	0
N3	0	69	0	0	0	0	0	0	0	0	1	4	0	0	0	0
N4	6	17	0	0	0	0	0	2	0	0	0	1	0	0	0	0
N5	8	19	1	0	0	1	0	0	0	0	7	1	0	0	0	0
N6	0	5	0	1	0	0	0	0	0	0	0	1	0	0	0	0
N7	24	31	0	0	0	0	2	5	0	0	0	1	0	0	1	0
N8	8	9	2	0	0	1	0	0	1	0	2	3	0	1	0	0
N9	4	22	0	0	0	0	0	3	0	2	3	2	0	0	0	0
N10	3	3	1	0	0	0	0	1	0	0	0	2	0	0	0	0
N11	5	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
N12	38	27	0	2	0	1	7	3	0	0	0	4	0	0	0	0
N13	11	24	1	0	1	0	0	2	0	0	2	3	0	0	0	0
N14	3	9	0	0	0	0	0	0	0	0	0	2	0	0	0	0
N15	34	24	4	1	0	1	1	5	0	0	3	2	1	0	0	0
N16	6	3	0	0	0	1	1	1	0	0	0	1	0	0	0	0
N17	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N18	17	16	3	0	1	0	1	0	0	0	0	2	0	0	0	0
N19	30	65	0	0	0	14	1	3	0	0	39	4	0	0	0	1
N20	7	4	1	1	0	1	2	1	0	0	0	1	0	0	0	0
N21	18	25	0	1	0	0	0	2	0	0	0	1	0	0	0	0
N22	46	31	3	2	0	0	0	1	0	0	0	7	0	0	0	0
N23	11	1	0	0	0	0	4	1	0	0	0	0	0	0	0	0
N24	7	8	0	0	0	0	0	0	0	0	0	7	0	0	0	0
N25	4	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0
N26	3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
N27	5	12	1	0	0	0	5	0	0	0	0	0	0	0	0	0
N28	0	13	0	0	0	0	0	2	0	0	0	2	0	0	0	0
N29	1	7	1	0	0	0	0	0	0	0	0	0	0	0	0	0
N30	8	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0
N31	9	2	0	0	0	0	1	0	0	0	0	1	1	0	0	0
N32	0	5	0	0	0	1	0	1	0	0	0	3	0	0	0	0
N33	5	33	0	0	0	0	0	0	1	0	0	7	0	0	0	0
N34	2	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0
N35	1	3	0	0	0	0	0	1	0	0	1	0	0	0	0	0

6.4.2 *Occurrence frequency of prefixes/prefix combinations*

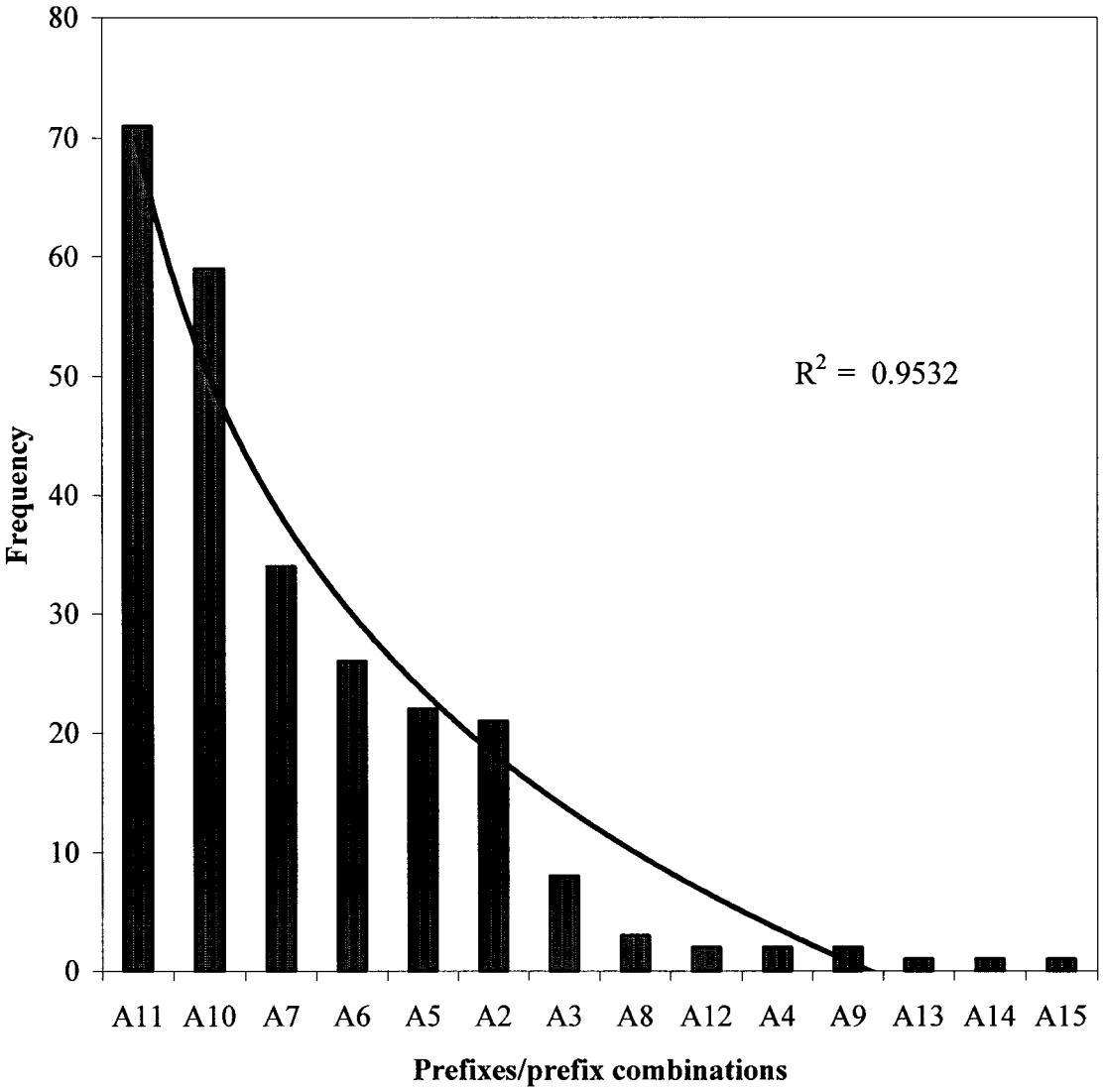
Generated from the numbers in Tables 6.3, Table 6.4 ranks the 15 prefixes and prefix combinations by the number of documents they retrieved. Each of the four prefixes (A1, A2, A5, A6) and three prefix combinations (A10, A11, A7) retrieved more than 20 documents, with A1 retrieving a number of documents higher than the total number retrieved by all the other 14 prefixes/prefix combinations. The remaining two prefixes (A3, A4), as well as six of the prefix combinations (A8, A9, A12, A13, A14, A15) retrieved less than nine documents each.

Table 6.4. Ranking of prefixes and prefix combinations

	A1	**A11**	**A10**	**A7**	**A6**	**A5**	**A2**	**A3**	**A8**	**A12**	**A4**	**A9**	**A13**	**A14**	**A15**
Documents	556	71	59	34	26	22	21	8	3	2	2	2	1	1	1

An analysis of the frequency distribution of the prefixes/prefix combinations is shown in Figure 6.1. Excluding the most frequently occurring prefix (A1), this frequency distribution shows a logarithmic trend (R squared = 0.9532). Based on this trend, it was decided to choose the six most frequently occurring prefixes/prefix combinations (in addition to A1, the most frequent one) for later application in the comparative search engine experiments (see Chapter 7). These prefixes/prefix combinations are: A1 (*al*), A2 (*b*), A5 (*l*), A6 (*w*), A7 (*bal*), A10 (*ll*) and A11 (*wal*).

Figure 6.1. Frequency distribution of prefixes/prefix combinations (excluding A1)



7. Methodology

7.1 Introduction

Adapting an ELIR system to use with other languages can be a challenging task, not to be lightly undertaken. The morphological properties of the language are the single most important issue that must be tackled in indexing, searching and retrieval. Chapter 4 has indicated how research on IR in different languages has focused on indexing methods, and how experimental systems were developed to handle the morphological variations of words both at the indexing and search stages. The developers of these systems took into consideration the morphological features of individual languages, devised indexing methods that suit these features, and then tested their retrieval effectiveness. One common observation to be made about these systems is that they provide invaluable information on the effectiveness of their indexing methods and, more importantly, offer a starting point for investigating the adaptation of ELIR systems to use with their respective languages. For example, if researchers have found that stemming is a necessary method for IR in the Turkish language, then stemming should be investigated when studying the possibility of adapting an ELIR system to use with Turkish text. In the case of Arabic, the developers of IR systems have identified stemming and root indexing as two methods that must be implemented in any system to effectively handle the language (Abu Salem,

al-Omari and Evens 1999). Stemming is a universal IR technique that is used with different degrees of success to enhance retrieval in any language (Pirkola 2001), while root indexing is a language-specific technique that has been developed for Arabic.

This dissertation is based on the premise that the process of adapting an ELIR system to use with Arabic texts must start at the word level, and specifically with the morphological variants of nouns. This involves providing the system with indexing and searching features that enable users to retrieve the variants of a noun by simply entering that noun or any of its variants as a search term. In English, this is accomplished through stemming. Stemming also has been used along with root indexing in experimental Arabic IR systems to ensure effective IR (Abu-Salem, al-Omari and Evens 1999) and by Arabic search engines on the Web (ArabVista and al-Idrisi). Root indexing requires morphological analysis to identify the Arabic root of words and then group all words that are derived from one root under one index term: the root itself (al-Fedaghi and al-Sadoun 1990). Logistically, implementing a mechanism to handle these analyses and performing them within an ELIR system will likely be more time- and resource-consuming than stemming, which is a common feature of most ELIR systems. On the other hand, implementing a stemming mechanism may not be enough to ensure retrieval of word variants in an IR environment. How does root indexing compare with stemming, and which technique is a better choice for Arabic nouns?

7.1.1 *Search engines*

Two IR systems were selected for this research: an Arabic-language IR system (al-Idrisi) that employs stemming and root-indexing, and an ELIR system (AltaVista) that employs stemming only (see chapter 5). These systems were used to answer the following questions:

1. How do AltaVista's stemming search features compare with root searching in al-Idrisi?

2. How might the performance of AltaVista be improved?

3. Does root searching actually outperform stemming?

7.1.2 *Test database and queries*

In experimental IR studies, the most common methodological approach involves creating an experimental text collection with known relevant documents, and computing evaluation measures to validate the effectiveness of the strategy (Hull 1996). The experimental collection of documents (document database) used in this research comprises Web pages (documents) retrieved by initial searches using al-Idrisi to locate nouns extracted from real Web searches.

In traditional IR experiments, queries expressing information needs are selected and matched against documents to measure recall and precision. This research, however, deals with the issue of matching nouns to documents that contain not only those nouns but also other nouns belonging to the same noun blocks (see 7.1.4). Therefore, the queries were selected to include only one noun and/or its variants: the command issued to the IR system can be conceptualized as follows: find documents containing this noun or any of its variants.

7.1.3 *Searches*

Searches were designed to compare the performance of AltaVista with al-Idrisi, and to evaluate stemming as an alternative to root-retrieval. The document database was created using the results of root-retrieval by al-Idrisi. The only way to determine if a retrieved document is relevant to a query is to display the document and then check the highlighted terms to see if any belong to that query's noun block (see 7.1.4). It is important to stress that because the initial objective was to compare the performance of AltaVista, with its stemming capabilities, against that of al-Idrisi and its root capabilities, the AltaVista searches were performed against the document set retrieved by al-Idrisi; at this stage, an assumption was made that all the documents retrieved by al-Idrisi were relevant.

One purpose of the experiment is to explore how well AltaVista performs in terms of document retrieval using its current search features. A second purpose is to establish to what extent manual manipulation of AltaVista's search features can increase the number of retrieved documents, thereby suggesting ways in which it might be improved for Arabic-language searching. The final purpose is to identify those documents that still have not been retrieved by AltaVista in order to determine whether in fact they should have been retrieved (that is, they contain nouns belonging to the block of the noun in the query) or not (they do not contain nouns belonging to the block of the noun representing the query). Such an examination of the documents not retrieved by AltaVista after all stemming manipulations have been implemented will reveal two critical pieces of information: how many of the documents initially retrieved by al-Idrisi using its root indexing technique have wrongly been missed by AltaVista (that is, the shortcomings of stem in comparison with root searching); but also, how many of those documents were wrongly retrieved by al-Idrisi in the first place (that is, the shortcomings of root in comparison with stem searching).

7.1.4 *Recall, precision and relevance*

Experiments were conducted to investigate how closely AltaVista can approach the performance level attained by al-Idrisi, and to suggest ways of improving the former to make it get even closer. The experiments called for noun queries

searched using AltaVista to be matched against documents retrieved earlier by the al-Idrisi. In theory, investigating how close AltaVista can get to al-Idrisi involves counting how many documents it retrieves out of the ones retrieved by al-Idrisi using the root of a specific noun. It was not at all clear at the outset of this research whether the search-by-root feature of al-Idrisi always retrieves relevant documents, but earlier research by others (al-Kharashi 1991, Abu Salem 1992, al-Kharashi and Evens 1994, Hmeidi, Kanaan and Evens 1997, Abu Salem, al-Omari and Evens 1999) had strongly suggested that this indeed was the case. The initial assumption, therefore, was that all documents retrieved by root searching would be relevant to the query containing the noun that retrieves them. The question of whether in fact this is the case was left to the final stages of the methodology.

Relevance is one of the most critical concepts in information retrieval (and indeed in the whole of information science). Earlier research on Arabic IR (al-Kharashi 1991, Abu Salem 1992, al-Kharashi and Evens 1994, Hmeidi, Kanaan and Evens 1997, Abu Salem, al-Omari and Evens 1999) concluded that root retrieval extracts the highest number of Arabic noun variations and, therefore, produces the maximum possible number of retrieved documents. An ELIR system can only match this recall performance if it provides indexing and search capabilities that facilitate the retrieval of an equal number of documents. For example, if a search-by-root query in al-Idrisi retrieves 35 documents, AltaVista

also should be able to retrieve those same 35 documents. This assumes, however, that all 35 documents are relevant to the subject encapsulated in the search statement. If this is not the case, then it does not follow that any shortfall by AltaVista represents in fact a criticism of or a failing in the search engine. In practice, then, the question of relevance cannot be ignored; criteria must be in place to judge relevance and to compare the performance of the two systems using this as a measure.

In traditional evaluation studies of IR systems, recall and precision measures are based on the relevance of retrieved documents to the information needs expressed in queries. Determining relevance is not a morphological/linguistic process and should be considered a language-independent exercise: it does not involve looking at the success of a system in retrieving variants of words included in a query; rather, it assesses the extent to which a retrieved document matches the information needs represented by those words. This standard definition of relevance does not apply in this research work. It is looking at the performance of a system only at the word level, where a retrieved document is examined in order to check if it contains a variant of a query word or not. More specifically, since the definition of a word is limited to include only (as described in Chapter 2), a document is relevant to a query if it contains the noun included in the query or any variant of that noun. The variants of a noun form a block of nouns; the occurrence of any noun in this block within a document

makes it relevant to a query that contains a noun belonging to the same block. In English, a noun block contains the masculine noun, the feminine noun (if any) and their plural and genitive forms. For example, the block for the noun waiter contains waiter, waitress, waiters, waitresses, waiter's, waitress', waiters' and waitresses'. The presence of any of these seven nouns in a retrieved document makes it relevant, using this restricted definition, to a query containing any one of them. In Arabic (see Chapter 2), the number of nouns in a noun block is much higher and can run into the hundreds. A block can include the masculine (m.) noun, the feminine (f.) noun, their dual (d.) and plural (p.) forms, and any forms of these nouns attached to the definite article, to particles, or to possessive pronouns. Given the large number of variants, in addition to the fact that this number can vary from noun to noun, only a very partial listing is shown in Table 7.1, using the noun *mclm* (teacher).

Table 7.1. A partial list of variants in an Arabic noun block

mclm (m.)	*mclmh* (f.)	*mclman* (m. d.)	*mclmwn* (m. p.)
mclmtan (f. d.)	*mclmat* (f. p.)	*mclmy* (my teacher m.)	*almclm* (the m. teacher)
almclmh (the f. teacher)	*mclmtha* (her f. teacher)	*wmclm* (and a teacher)	*almclmwn* (the m. teachers)
mclman (the m. d. teachers)	*wmclmat* (and f. teachers)	*mclmhm* (their m. teacher)	*wllmclm* (and for the m. teacher)

Recall used in this restricted sense is a measure of the extent to which the IR system retrieves all documents in the database containing a noun or nouns that belong to the noun block of a noun present in a query. Precision is a measure of

the extent to which the system only retrieves those documents that contain block nouns, and rejects all others.

7.2 Methodological steps

7.2.1 *Arabic noun selection*

The first step was to construct a set of Arabic nouns that can be used in the experiments. This was done by selecting and translating into Arabic 40 nouns from a collection of 907 English nouns entered by real users in real searches conducted on the Web. A logical choice would have been to use queries entered in AltaVista, but access to such queries was not available at the time and, theoretically, the use of queries entered in other engines should not make a difference: a search term is a search term no matter where it is used. The nouns were originally obtained in English from a search identifier service provided by WebCrawler (http://www.webcrawler.com), a popular Web search engine. This service is called (Search Voyeur) and allows the monitoring of real-time queries as they are entered to WebCrawler by searchers all over the World. During March 1999 for a period of three days, three 5-minute sessions of Search Voyeur were captured during different times of the day: 9:05- 9:10 A.M. on March 5, 3:20-3:25 P.M. on March 12, and 10:45-10:50 P.M. on March 23. A total of 1769 search queries (predominantly in English) were captured during the

sessions, and 4236 individual English search terms (strings of characters separated by spaces) were extracted from these queries.

Of the 4236 terms obtained in Step 1, 2109 terms were nouns. If a term was a homograph, it was considered a noun ('show', for example, was considered the noun 'show' not the verb 'to show'). These nouns were entered into a Microsoft Access database file. Using the sorting facilities available in Access, 808 duplicate nouns were eliminated, leaving 1301 unique nouns in the set. Further examination of the noun set revealed the presence of 394 proper nouns (countries, cities, people, etc.). Proper nouns were excluded because they do not usually have dual, plural, or feminine forms in Arabic, and typically do not generate morphological variations (the object of the investigation). Loan words from other languages were also excluded because they are untypical in Arabic, usually not having an identifiable Arabic root, a fact that makes them fall outside the scope of this research.

After these exclusions, 907 nouns remained in the set. Each noun was numbered for identification purposes. Applying a random selection process, 40 numbers were generated, and the corresponding nouns were selected from the list of 907 nouns. These 40 nouns were translated into Arabic by the researcher (and rechecked by a second Arabic speaker), and formed the noun data set (see Table 7.2). It comprises the basic forms of Arabic nouns: singular masculine or singular feminine nouns that are not attached to any prefixes or suffixes.

Table 7.2. Noun data set (see Appendix C for the Arabic-script list)

Arabic	English eq.	Root
wkalh	agency	*wkl*
Hywan	animal	*Hyw*
fnan	artist	*fnn*
wladh	birth	*wld*
wld	boy	*wld*
Srkh	company	*Srk*
wpl	connection	*wpl*
mtsabq	contestant	*sbq*
tHkm	control	*Hkm*
xlq	creation	*xlq*
wkyl	dealer	*wkl*
dfac	defense	*dfc*
qsm	department	*qsm*
tnzyl	download	*nzl*
byah	environment	*bwa*
nar	fire	*nwr*
pdyq	friend	*pdq*
lcbh	game	*lcb*
dlyl	guide	*dll*
taryx	history	*arx*
byt	house	*byt*
pnach	industry	*pnc*
mclwmh	information	*clm*
sakn	inhabitant	*skn*
mchd	institute	*chd*
bryd	mail	*brd*
wjbh	meal	*wjb*
mktb	office	*ktb*
xyar	option	*xyr*
qpydh	poem	*qpd*
Hml	pregnancy	*Hml*
vmn	price	*vmn*
qraah	reading	*qra*
wpfh	recipe	*wpf*
ntyjh	result	*ntj*
xdmh	service	*xdm*
tswq	shopping	*swq*
crD	show	*crD*
jhh	side	*wjh*
jamch	university	*jmc*

7.2.2 *Document data set creation*

The next step involved the creation of a document data set that could form a test database for searches using AltaVista. Each of the 40 nouns was entered one at a time to al-Idrisi as a single search term using the search-by-root option on the Advanced Search page. The searches on these terms produced hits ranging from 85 to 1046 documents (Table 7.3). From the results of each of the searches, 50 documents were selected randomly and displayed using the "highlight feature" implemented by al-Idrisi to distinguish words that cause a document to be retrieved (see 5.5). The randomness was achieved using a process similar to the one used to select the 40 nouns. For every search, each retrieved document was numbered for identification purposes. Applying a random selection process, 50 numbers were generated through the random-number-generator function of a calculator, and the corresponding documents were selected from the list of documents retrieved by the search. Because root searching was used, every occurrence of a word that is derived from the root of the noun used as a search term was highlighted. The 50 selected documents from each search were saved in a separate folder on a local computer. For example, the search for *xlq* (creation) produced 113 documents; 50 documents were randomly selected out of 113 and saved in a folder named "creation" on the local hard drive. As a result of this process, 40 folders (one for each search) were created, each containing the 50 randomly selected documents resulting from the corresponding search. This procedure resulted in 2000 HTML documents that formed the test database.

Table 7.3. Al-Idrisi's search results (number of hits)

Arabic	English eq.	Hits
wkalh	agency	119
Hywan	animal	176
fnan	artist	169
wladh	birth	400
wld	boy	400
Srkh	company	643
wpl	connection	605
mtsabq	contestant	271
tHkm	control	407
xlq	creation	113
wkyl	dealer	119
dfac	defense	167
qsm	department	230
tnzyl	download	163
byah	environment	120
nar	fire	85
pdyq	friend	132
lcbh	game	112
dlyl	guide	315
taryx	history	250
byt	house	173
pnach	industry	418
mclwmh	information	1046
sakn	inhabitant	113
mchd	institute	190
bryd	mail	347
wjbh	meal	128
mktb	office	807
xyar	option	344
qpydh	poem	169
Hml	pregnancy	252
vmn	price	146
qraah	reading	382
wpfh	recipe	137
ntyjh	result	409
xdmh	service	715
tswq	shopping	332
crD	show	489
jhh	side	473
jamch	university	829

7.2.3 *Document indexing*

The PC version of AltaVista was installed on the same personal computer as the test database.

AltaVista was then used to index the 2000 HTML documents contained in the 40 folders. A separate index was built for each folder to allow searching against individual folders (as explained below). Statistics related to the indexing process are presented in Table 7.4. The "Words" column indicates the number of indexed words in the folder that contains the 50 HTML documents retrieved by a specific noun. The "Average" column indicates the average number of words per HTML document page in the indexed documents. For example, the 50 documents retrieved by the noun *wkyl* (dealer) contain 37909 words, for an average of 758 words per document.

Table 7.4. AltaVista's indexing statistics

Arabic	English eq.	Words	Average
wkalh	agency	39061	781
Hywan	animal	47093	941
fnan	artist	34101	682
wladh	birth	44352	887
wld	boy	52017	1040
Srkh	company	51877	1037
wpl	connection	59831	1196
mtsabq	contestant	50973	1019
tHkm	control	65298	1306
xlq	creation	35730	714
wkyl	dealer	37909	758
dfac	defense	34808	696
qsm	department	40439	808
tnzyl	download	22334	446
byah	environment	48752	975
nar	fire	19702	394
pdyq	friend	32385	647
lcbh	game	24045	480
dlyl	guide	39951	799
taryx	history	22512	450
byt	house	28471	569
pnach	industry	46725	934
mclwmh	information	62518	1250
sakn	inhabitant	39727	794
mchd	institute	39280	785
bryd	mail	46902	938
wjbh	meal	33871	677
mktb	office	55577	1112
xyar	option	34762	695
qpydh	poem	49005	980
Hml	pregnancy	44896	897
vmn	price	30735	614
qraah	reading	34265	685
wpfh	recipe	40995	819
ntyjh	result	58729	1175
xdmh	service	33901	678
tswq	shopping	61629	1233
crD	show	63536	1271
jhh	side	82820	1656
jamch	university	47468	949

7.2.4 *AltaVista searches*

In Stage 1 of the searches, each of the 40 Arabic nouns was matched against its corresponding database using AltaVista. First, the nouns were entered in their complete form, exactly as they had been entered earlier using al-Idrisi. In this way AltaVista searched for an exact match of the noun (the column labelled SS (simple searches) in Table 7.5).

The next stage (Stage 2) was to use the truncation feature available on AltaVista in order to ignore the endings of Arabic nouns that are not part of the root (a manual stemming of the nouns). Each noun was truncated after the occurrence of the third and last letter of the root. For example the noun *xyar* (from the root *xyr*) was truncated after the letter "*r*", the last letter of the root; and the noun *wjbh* (from the root *wjb*) was truncated after the letter "*b*". When a noun had only three letters, it was truncated after these letters, because this is the minimum number of pre-truncation characters allowed by AltaVista, as explained in 5.4.2. The column labelled AS (advanced searches) in Table 7.5 shows the truncated forms of the 40 nouns as they were entered in AltaVista.

At the next stage (Stage 3) it was necessary to use AltaVista to retrieve documents using the 40 nouns and after specific prefixes and prefix combinations had been added to these nouns. Because AltaVista does not offer

left-hand truncation (truncation of the beginning of the Arabic noun), this stage involved manual modification of the search nouns. The seven prefixes/prefix combinations (hereafter referred to as prefixes) that had earlier been identified (see Chapter 6) as the most common prefixes in Arabic text were one by one added to the noun. To ensure the retrieval of the noun in its basic form as well as attached to any of these prefixes, each noun was entered in eight forms: its exact form (as described above) and in the other seven forms with the seven prefixes attached to it. The column labelled MMS (manually modified searches) in Table 7.6 shows samples of how these searches were entered. For example, the query of the noun *byah* (environment) contains eight nouns: *byah, albyah, walbyah, llbyah, balbyah, wbyah, lbyah,* and *bbyah.* (Note that when no Boolean operators are used between search terms, AltaVista defaults to OR, and a document is retrieved when it contains any one of the terms). The first noun is the basic form, with no attached prefixes, the remaining seven nouns are forms of the basic noun attached respectively to the prefixes: *al* (the), *wal* (and the), *ll* (for the), *bal* (in the), *w* (and), *l* (for), and *b* (in).

Table 7.5. Simple and advanced searches in AltaVista (see Appendix D for the searches in Arabic script)

English eq.	Arabic	Root	SS	AS
agency	*wkalh*	*wkl*	*wkalh*	*wkal**
animal	*Hywan*	*Hyw*	*Hywan*	*Hyw**
artist	*fnan*	*fnn*	*fnan*	*fnan**
birth	*wladh*	*wld*	*wladh*	*wlad**
boy	*wld*	*wld*	*wld*	*wld**
company	*Srkh*	*Srk*	*Srkh*	*Srk**
connection	*wpl*	*wpl*	*wpl*	*wpl**
contestant	*mtsabq*	*sbq*	*mtsabq*	*mtsabq**
control	*tHkm*	*Hkm*	*tHkm*	*tHkm**
creation	*xlq*	*xlq*	*xlq*	*xlq**
dealer	*wkyl*	*wkl*	*wkyl*	*wkyl**
defense	*dfac*	*dfc*	*dfac*	*dfac**
department	*qsm*	*qsm*	*qsm*	*qsm**
download	*tnzyl*	*nzl*	*tnzyl*	*tnzyl**
environment	*byah*	*bwa*	*byah*	*bya**
fire	*nar*	*nwr*	*nar*	*nar**
friend	*pdyq*	*pdq*	*pdyq*	*pdyq**
game	*lcbh*	*lcb*	*lcbh*	*lcb**
guide	*dlyl*	*dll*	*dlyl*	*dlyl**
history	*taryx*	*arx*	*taryx*	*taryx**
house	*byt*	*byt*	*byt*	*byt**
industry	*pnach*	*pnc*	*pnach*	*pnac**
information	*mclwmh*	*clm*	*mclwmh*	*mclwm**
inhabitant	*sakn*	*skn*	*sakn*	*sakn**
institute	*mchd*	*chd*	*mchd*	*mchd**
mail	*bryd*	*brd*	*bryd*	*bryd**
meal	*wjbh*	*wjb*	*wjbh*	*wjb**
office	*mktb*	*ktb*	*mktb*	*mktb**
option	*xyar*	*xyr*	*xyar*	*xyar**
poem	*qpydh*	*qpd*	*qpydh*	*qpyd**
pregnancy	*Hml*	*Hml*	*Hml*	*Hml**
price	*vmn*	*vmn*	*vmn*	*vmn**
reading	*qraah*	*qra*	*qraah*	*qraa**
recipe	*wpfh*	*wpf*	*wpfh*	*wpf**
result	*ntyjh*	*ntj*	*ntyjh*	*ntyj**
service	*xdmh*	*xdm*	*xdmh*	*xdm**
shopping	*tswq*	*swq*	*tswq*	*tswq**
show	*crD*	*crD*	*crD*	*crD**
side	*jhh*	*wjh*	*jhh*	*jhh**
university	*jamch*	*jmc*	*jamch*	*jamc**

The fourth and last stage of the searches utilized queries that produced the maximum possible number of documents (the highest recall level) for each of the 40 nouns. These queries were designed to retrieve all documents retrieved by the first three stages, in addition to documents that were retrieved by modifications made to the noun forms used in queries in Stage 3, the stage of manually modified searches (MMS). The noun forms used in the searches in Stage 3 were truncated after the last letter of the root. That meant a query would retrieve documents containing the basic truncated noun or any of its prefixed forms. The column labelled AMMS (advanced manually-modified searches) in Table 7.6 shows samples of how these searches were entered.

Table 7.6. Samples of manually modified and advanced manually-modified searches (see Appendix E for the searches in Arabic script)

Arabic	**English eq.**	**MMS**	**AMMS**
wpl	connection	*wpl alwpl walwpl llwpl balwpl wwpl lwpl bwpl*	*wpl* alwpl* walwpl* llwpl* balwpl* wwpl* lwpl* bwpl**
xlq	creation	*xlq alxlq walxlq llxlq balxlq wxlq lxlq bxlq*	*xlq* alxlq* walxlq* llxlq* balxlq* wxlq* lxlq* bxlq**
wkyl	dealer	*wkyl alwkyl walwkyl llwkyl balwkyl wwkyl lwkyl bwkyl*	*wkyl* alwkyl* walwkyl* llwkyl* balwkyl* wwkyl* lwkyl* bwkyl**
tnzyl	download	*tnzyl altnzyl waltnzyl lltnzyl baltnzyl wtnzyl ltnzyl btnzyl*	*tnzyl* altnzyl* waltnzyl* lltnzyl* baltnzyl* wtnzyl* ltnzyl* btnzyl**
byah	environment	*byah albyah walbyah llbyah balbyah wbyah lbyah bbyah*	*bya* albya* walbya* llbya* balbya* wbya* lbya* bbya**

Upon completion of the four stages of searches, each document that had not been retrieved by AltaVista (a missed document (MD)) was displayed to identify the words that had caused its initial retrieval by al-Idrisi. This was easily accomplished because, as explained above, the documents were saved in "highlighted" formats, where the words that caused their retrieval were highlighted in red. After the highlighted terms were extracted, a database file was created in Access to organize the terms and link them to their respective documents, and consequently to the noun. Let us suppose that after performing all four stages of the searches using the noun *nar* (fire), 15 MDs were identified. Each one of these MDs is displayed and the highlighted words in it are extracted and entered in a record containing pointers to the document that contains them and to the noun *nar*. Later, this type of information can be consulted to analyze the causes of retrieval failure and to determine if a document should have been retrieved by AltaVista or, alternatively, if it should not have been retrieved by al-Idrisi in the first place. For example, if an MD has one highlighted term *nwr* (light), which does not belong to the noun block (see 7.1.4) of *nar*, it is judged irrelevant: it should not have been retrieved by al-Idrisi. By contrast, if an MD contains the highlighted term *nyran* (fires), which belong to the noun block of *nar*, it is judged relevant: it ideally should have been retrieved by AltaVista.

Following similar procedures, each document that was retrieved by AltaVista at the last stage of searches was displayed to verify that it was relevant to the noun.

Each retrieved document was checked to confirm that it contained at least one highlighted term that belonged to the noun block of the noun that retrieved it, and all retrieved documents were judged relevant to their respective queries.

8. Results and analysis

8.1 Introduction

Searches were conducted in four stages on AltaVista using Arabic nouns to retrieve documents that earlier had been retrieved by al-Idrisi. The first two stages employed search features already provided by AltaVista, without undertaking any manipulation of the nouns to simulate new search features. The last two stages were conducted using manually manipulated search techniques to simulate features that could potentially be added to AltaVista in order to assess their effectiveness.

The first stage involved simple searches (SS), where the noun was entered without stemming/truncation and without the addition of prefixes. The second stage involved advanced searching (AS), right-hand (suffix) truncating the noun (a stemming-related procedure) to retrieve more documents (increase recall). The third stage involved manually manipulated searches (MMS), where prefixes were added to the noun to make up for the absence of left-hand (prefix) truncation in AltaVista. The last stage involved an advanced version of the third stage (advanced manually-manipulated searches (AMMS)), where the truncation feature of AltaVista was used together with MMS to further increase recall levels.

These search experiments in AltaVista were conducted with the following objectives in mind:

1. To compare the recall of AltaVista with that of al-Idrisi--in document retrieval; that is, to determine how many of the documents originally retrieved by al-Idrisi could also be retrieved by AltaVista.

2. To explore ways of improving the recall achieved by AltaVista in order to identify ways of adapting AltaVista for use with Arabic text.

3. To isolate documents retrieved by al-Idrisi that were not retrieved by AltaVista in order to analyse these documents to see if they are relevant to the nouns used in the searches; that is to determine whether al-Idrisi is retrieving documents that are unrelated to the search noun.

Objective (1) was achieved through the first two stages of the searches (SS and AS) using AltaVista's existing search algorithms. Objective (2) was achieved through the last two stages of the searches (MMS and AMMS), using a manually enhanced AltaVista that involved adding prefixes to the search nouns. This procedure simulated an AltaVista that in effect has left-hand truncation capabilities or automated prefix attachment to Arabic nouns. AMMS by itself, provided the highest maximum recall level (closest to that achieved by al-Idrisi through its root searching capability). Therefore, the documents that were not retrieved after this stage were assumed to be missed documents (MDs) pending

the analysis of AltaVista's failure to retrieve them. Once the missed documents had been identified (after the four stages of the searches), each was examined, and the word/words (keywords) that caused its retrieval in al-Idrisi were analysed to determine if they belong to a block of the noun used in the search, that is, if they are relevant.

How did Objectives (1), (2) and (3), help in achieving the two main objectives of the thesis: the adaptation of an ELIR system for use with Arabic text, and a comparison of stemming with root retrieval. By comparing AltaVista's performance to that of al-Idrisi, and using manually manipulated searches, achieving objectives (1) and (2) allowed us to determine if adding prefixes to Arabic nouns improved the performance of the ELIR system (AltaVista). Isolating MDs and analysing the reasons that caused AltaVista not to retrieve them allowed us to determine if there are other required features that AltaVista should have besides the capability to search on the prefixes of Arabic nouns. Finally, isolating those MDs that should not have been retrieved by al-Idrisi sheds further light upon the effectiveness of root retrieval and whether or not it is a helpful feature in any search engine designed to retrieve Arabic-language documents.

In the following sections any improvements in recall rates attained in the four stages of searches in AltaVista are examined, followed by a systematic analysis of the MDs for each of the 40 nouns used in the search experiments.

8.2 The searches

An overview of the AltaVista searches is provided in Table 8.1. For each of the 40 Arabic nouns it shows the results for the four stages, plus the number of missed documents. The lower is the number in the MD column, the closer AltaVista's performance is to al-Idrisi's. To simplify the process of listing the nouns and facilitate understanding of the tables for the non-Arab reader, the nouns will be from now on used in their English translation, and they are listed in the first column. The remaining columns list the number of documents retrieved in the four stages of searching and the number of documents that were not retrieved after the fourth stage (the stage that produced the highest recall). The SS (Simple Search) column lists the number of documents retrieved when the exact Arabic noun was entered as the only search term, without prefixes or truncation. The AS (Advance Search) column lists the number of document retrieved when the truncation feature of AltaVista was used to retrieve any occurrence of the exact Arabic noun truncated after the first three characters. The MMS (Manually Modified Search) column contains the search results when eight search terms were entered: the exact noun and the nouns attached to the seven prefixes/prefix combinations. The AMMS (Advanced Manually-Modified Search) column indicates the number of documents retrieved when using the truncated noun but attached to the seven prefixes/prefix combinations. It shows the highest possible number of document that could be retrieved in any of the four stages of searching and, for present purposes, it is assumed to

represent the optimal performance of AltaVista. Subtracting the number in this column from the original number of 50 documents in the document set that was retrieved by al-Idrisi produces the number in the last column (MD/missed document). For example, if the number in the MD column is 11, this means that the fourth stage of searching retrieved 39 documents out of 50 and failed to retrieve 11 (as in the case of the noun industry). The documents referenced in the MD column are analyzed later to determine why they were not retrieved and if the keywords that retrieved them in al-Idrisi belong to the noun block and are, therefore, relevant.

Table 8.1 shows that there were many MDs. In total 1120 documents were not retrieved by AltaVista out of the 2000 documents retrieved by al-Idrisi. For some nouns, the number of MDs was almost 100% (meal, 49 MDs, and contestant, 46 MDs). The lowest MDs are for the noun environment (2 MDs). The rest of the nouns have MDs ranging from 3 to 45, with the largest concentration of numbers in the 20s and 30s. But, for now we are assuming that al-Idrisi has retrieved only relevant documents.

Table 8.1. Number of documents retrieved in the four search stages in AltaVista

Noun	SS	AS	MMS	AMMS	MD
meal	0	1	0	1	49
contestant	0	2	0	4	46
download	2	3	5	5	45
price	3	3	5	5	45
boy	5	7	6	7	43
artist	1	2	3	7	43
poem	1	2	3	8	42
control	3	5	6	8	42
defense	0	1	8	11	39
dealer	3	3	3	12	38
institute	5	5	5	13	37
connection	8	13	10	15	35
fire	1	2	14	15	35
option	7	12	11	16	34
friend	4	11	7	16	34
pregnancy	5	13	6	17	33
game	6	10	8	17	33
university	11	14	12	19	31
animal	0	9	2	19	31
side	7	7	19	19	31
information	8	9	20	21	29
creation	16	18	22	23	27
recipe	0	15	0	24	26
service	7	10	16	25	25
result	20	20	26	26	24
department	18	18	24	26	24
birth	1	19	21	27	23
office	3	16	16	28	22
shopping	3	11	4	28	22
agency	9	25	11	29	21
reading	10	13	18	29	21
show	19	23	27	30	20
guide	15	15	31	31	19
house	23	33	29	37	13
industry	10	18	19	39	11
mail	21	24	38	41	9
company	32	35	37	42	8
inhabitant	17	25	25	45	5
history	28	32	41	47	3
environment	22	30	40	48	2

8.2.1 *Exact Arabic nouns in IR*

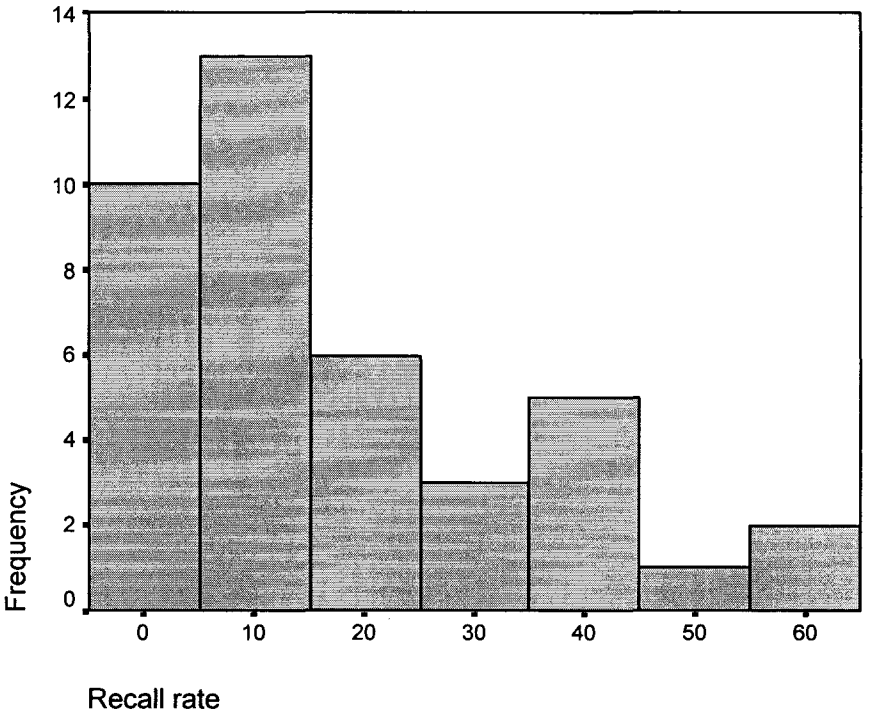
The simple searches (SS) conducted in the first stage used the exact form of the Arabic noun (the basic noun without prefixes or suffixes). How did AltaVista fare? Entering exact Arabic nouns in search queries does not seem to be a viable option for effective IR from Arabic databases. Using non-affixed nouns in searching substantially reduces the number of retrieved nouns and therefore adversely affects the number of retrieved documents. Table 8.2 tabulates the numbers of documents retrieved using AltaVista and the exact Arabic noun. The SS column indicates the number of documents retrieved by a simple search, and the Recall rate column indicates the percentage found of the original 50 documents retrieved by al-Idrisi. Only two exact nouns (company and history) retrieved more than 50% of the documents. Four nouns (house, environment, mail, and result) retrieved documents accounting for more than 40% and less than 50% of the documents. Eight nouns retrieved numbers of documents ranging between 20% and 38%, while the remaining 28 nouns retrieved less than 20% of the documents, including five nouns that retrieved no documents at all. The distribution of recall rates is shown in Figure 8.1.

Using the simple form of the Arabic noun in an ELIR system risks, then, missing many of the noun variants (the nouns that belong to the simple noun block), and reduces, therefore, recall levels. As a result, many of the documents that should be retrieved will be missed.

Table 8.2. Recall rates for simple searches (SS)

Noun	SS	Recall rate
company	32	64%
history	28	56%
house	23	46%
environment	22	44%
mail	21	42%
result	20	40%
show	19	38%
department	18	36%
inhabitant	17	34%
creation	16	32%
guide	15	30%
university	11	22%
reading	10	20%
industry	10	20%
agency	9	18%
information	8	16%
connection	8	16%
service	7	14%
side	7	14%
option	7	14%
game	6	12%
pregnancy	5	10%
institute	5	10%
boy	5	10%
friend	4	8%
dealer	3	6%
control	3	6%
office	3	6%
shopping	3	6%
price	3	6%
download	2	4%
poem	1	2%
birth	1	2%
fire	1	2%
artist	1	2%
contestant	0	0%
recipe	0	0%
meal	0	0%
defense	0	0%
animal	0	0%

Figure 8.1. Distribution of recall rates for simple searches (SS)



8.2.2 *Traditional methods of truncation*

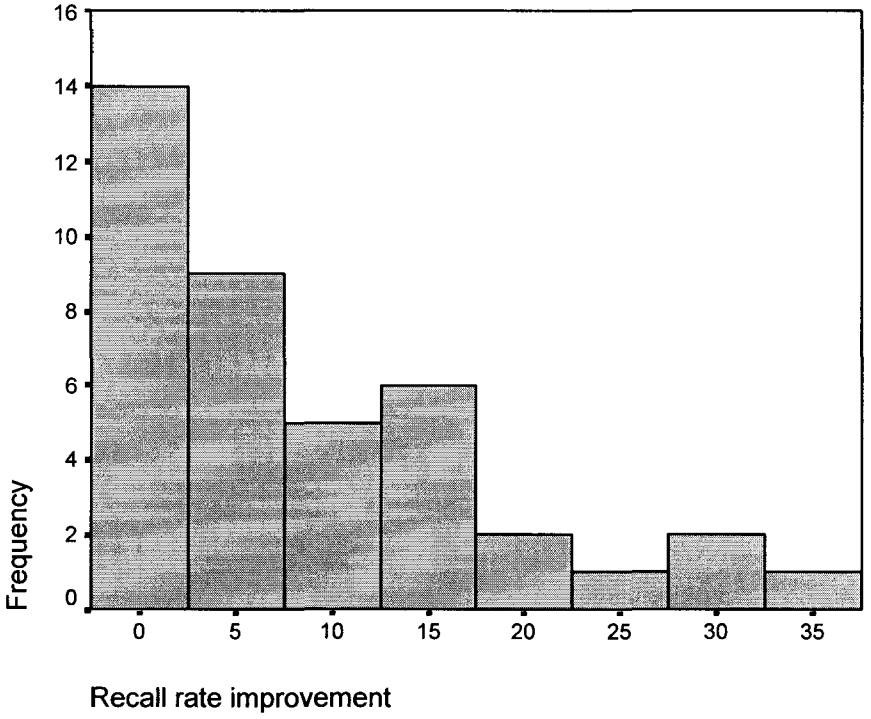
One of the traditional methods of enhancing the retrieval of documents in an IR system is truncation. AltaVista provides right-hand (suffix, or end of the word) truncation as a search feature to help users in retrieving different variations of words (and especially the plural form of a singular noun). How did this feature improve the results of searches that earlier had been conducted using the non-truncated forms of nouns? Right-hand truncating an Arabic noun after the third letter of the root (manual stemming) certainly improves recall levels, but what is the extent of this improvement and is it significant? Table 8.3 shows the numbers of documents retrieved by each truncated noun, listing the percentage of documents retrieved (recall rate) from the original 50 documents retrieved by al-Idrisi, and compares it with the rate of simple searches (SS). The AS column indicates the number of documents retrieved by the truncated-noun search; the third column indicates the recall rate; the fourth column indicates the SS recall rate; and the last column indicates the improvement in the recall rate between SS and AS searches. For example, the noun "university" in its truncated form retrieved 14 out of the 50 documents for a 28% recall rate with 16% improvement over the SS search. Five truncated nouns (the truncated forms of inhabitant, company, house, history, agency, and environment) retrieved 50% or more of the documents. Eleven truncated nouns retrieved between 30% and 48% of their respective documents. The remaining 24 nouns retrieved less than 30% of their documents, with nine of them retrieving less than 10%. The

improvement rates over SS ranged from 0% to 36%. The rate was improved by 20% or more in six searches, between 10% and 18% in 11 searches, and less than 8% in 23 searches (0% in seven of these 23 searches). The distribution of improvement of recall rates in AS over SS is shown in Figure 8.2.

Table 8.3. Recall rates for advanced searches (AS)

Noun	AS	Recall rate	SS Recall rate	Improvement rate
company	35	70%	64%	6%
house	33	66%	46%	20%
history	32	64%	56%	8%
environment	30	60%	44%	24%
agency	25	50%	18%	32%
inhabitant	25	50%	34%	16%
mail	24	48%	42%	6%
show	23	46%	38%	12%
result	20	40%	40%	0%
birth	19	38%	2%	36%
industry	18	36%	20%	16%
creation	18	36%	32%	4%
department	18	36%	36%	0%
office	16	32%	6%	26%
recipe	15	30%	0%	30%
guide	15	30%	30%	0%
university	14	28%	22%	16%
pregnancy	13	26%	10%	16%
connection	13	26%	16%	10%
reading	13	26%	20%	6%
option	12	24%	14%	10%
shopping	11	22%	6%	16%
friend	11	22%	8%	14%
service	10	20%	14%	14%
game	10	20%	12%	8%
animal	9	18%	0%	18%
information	9	18%	16%	2%
boy	7	14%	10%	4%
side	7	14%	14%	0%
control	5	10%	6%	4%
institute	5	10%	10%	0%
download	3	6%	4%	2%
dealer	3	6%	6%	0%
price	3	6%	6%	0%
contestant	2	4%	0%	4%
poem	2	4%	2%	2%
artist	2	4%	2%	2%
fire	2	4%	2%	2%
meal	1	2%	0%	2%
defense	1	2%	0%	2%

Figure 8.2. Distribution of improvement of recall rates in AS over SS



The level of improvement in the second stage of searches (truncated nouns) is not high. This is related to the morphological nature of Arabic words discussed in chapter 2. While truncation helps in solving the problem of suffixes, the prefix-rich forms of Arabic nouns cannot be retrieved with right-hand truncation. This works well for English words because of the importance of suffixes compared with prefixes, but for Arabic it is not enough because it does not solve the important problem created by prefixes (which requires left-hand truncation). That said, was the improvement in recall rates in AS over SS significant? To answer this question, a paired-samples t test was conducted to evaluate whether AS recall rates were significantly higher than those of SS. The results indicated that the mean recall rate for AS ($M = 26.70$, $SD = 19.07$) was significantly greater than the mean recall rate for SS ($M = 17.70$, $SD = 17.07$), $t(39) = 6.00$, $p = 0.001$. The mean difference was 9.00 between the two recall levels.

8.2.3 *Language-dependent term selection*

The selection of an English search term in traditional IR systems involves choosing a word to define a concept and usually does not involve morphological considerations except for singular/plural and occasional spelling variations. Arabic nouns, in contrast, occur often with prefixes. The third stage of searching in AltaVista involved manually (i.e., the user entering one by one the term plus

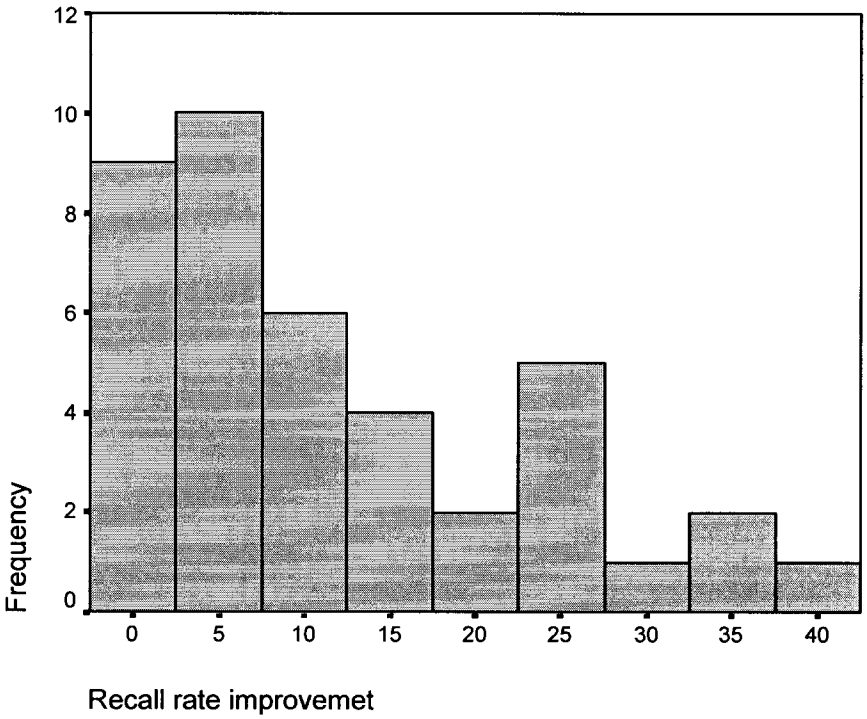
any potential prefixes) modifying the nouns to include prefixes as part of the search term. This produced eight search terms for each original noun query.

Table 8.4 shows the recall levels achieved for each of the 40 nouns by manually attaching prefixes to these nouns but not truncating them on the right-hand side, and it lists the percentage of documents retrieved (recall rate) from the original 50 documents retrieved by al-Idrisi and compares it with the rate of simple searches (SS). The MMS column indicates the number of documents retrieved after attaching the various prefixes to the noun; the third column indicates the recall rate; the fourth column indicates the SS recall rate; and the last column indicates the improvement in the recall rate between SS and MMS searches. Attaching the seven prefixes/prefix combinations to two of the nouns (history and environment) improved the recall rate from 56% and 44% (in the SS searches) to 82% and 80% respectively. The recall rate of seven queries ranges from 50% to 76%, while a range of 20% to 48% represents the recall rates of 14 queries. The remaining 17 queries retrieved less than 20% of their documents, including three that did not retrieve any documents. The improvement rates over SS searches ranged from 0% to 40%. The rate was improved by 24% or more in nine searches, between 10% and 18% in 11 searches, and less than 8% in 20 searches (0% in five of these 20 searches). The distribution of improvement of recall rates in MMS over SS is shown in Figure 8.3.

Table 8.4. Recall rates for manually modified searches (MMS)

Noun	MMS	Recall rate	SS Recall rate	Improvement rate
history	41	82%	56%	26%
environment	40	80%	44%	36%
mail	38	76%	42%	34%
company	37	74%	64%	10%
guide	31	62%	30%	32%
house	29	58%	46%	12%
show	27	54%	38%	16%
result	26	52%	40%	12%
inhabitant	25	50%	34%	16%
department	24	48%	36%	12%
creation	22	44%	32%	12%
birth	21	42%	2%	40%
information	20	40%	16%	24%
industry	19	38%	20%	18%
side	19	38%	14%	24%
reading	18	36%	20%	16%
office	16	32%	6%	26%
service	16	32%	14%	18%
fire	14	28%	2%	26%
university	12	24%	22%	2%
agency	11	22%	18%	4%
option	11	22%	14%	8%
connection	10	20%	16%	4%
game	8	16%	12%	4%
defense	8	16%	0%	16%
friend	7	14%	8%	6%
pregnancy	6	12%	10%	2%
boy	6	12%	10%	2%
control	6	12%	6%	6%
institute	5	10%	10%	0%
download	5	10%	4%	6%
price	5	10%	6%	4%
shopping	4	8%	6%	2%
dealer	3	6%	6%	0%
poem	3	6%	2%	4%
artist	3	6%	2%	4%
animal	2	4%	0%	4%
meal	0	0%	0%	0%
recipe	0	0%	0%	0%
contestant	0	0%	0%	0%

Figure 8.3. Distribution of improvement of recall rates in MMS over SS



Utilizing a language-dependent term selection in the third stage of searches (adding prefixes to the basic noun) increased the recall rates with clear improvement over those attained in the SS stage, as illustrated in Table 8.4 and Figure 8.3. Was this improvement significant? A paired-samples t test was conducted to evaluate whether MMS recall rates were significantly higher than those of SS. The results indicated that the mean recall rate for MMS ($M = 29.90$, $SD = 19.07$) was significantly greater than the mean recall rate for SS ($M = 17.70$, $SD = 17.07$), $t(39) = 6.85$, $p = 0.001$. The mean difference was 12.20 between the two recall levels.

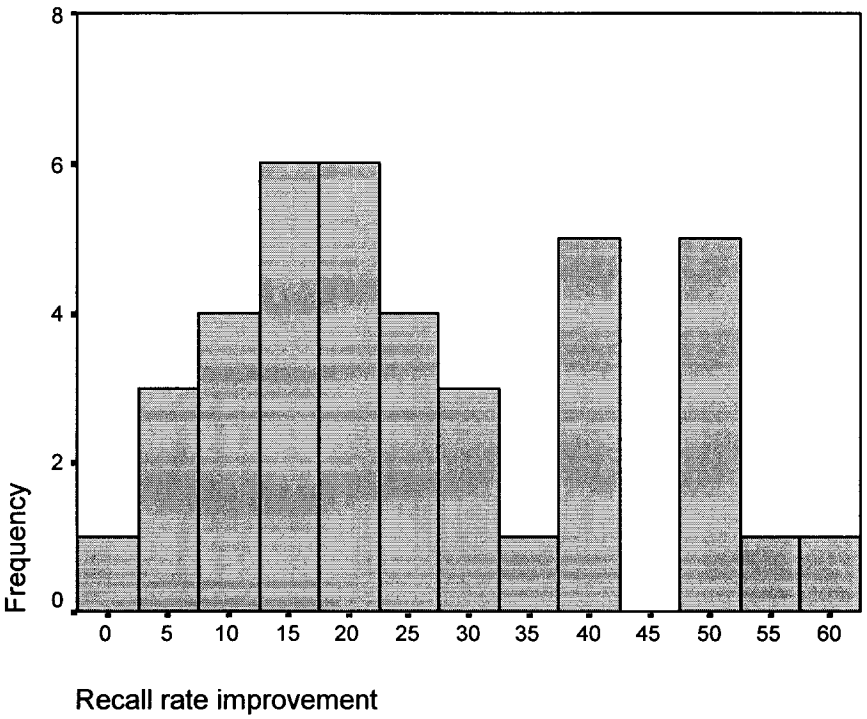
To further illustrate the effect of utilizing language-dependent term selection, the terms used in the third stage of AltaVista searching (MMS (manually modified searches)) were also right-hand truncated after the occurrence of the last letter of the root. Table 8.5 shows the results of the AMMS (advanced manually-modified searches) stage, and compares the recall rates with those of the SS (simple searches) stage. The AMMS technique produced the highest number of retrieved documents: 880 documents (44% of the 2000 documents found by al-Idrisi). Three truncated and prefixed nouns (environment, history, and inhabitant) respectively retrieved 96%, 94% and 90% of the documents. Fourteen other nouns retrieved numbers of documents ranging from 50% to 84% of the total. A range of 22% to 46% of documents was retrieved by 15 queries, and the remaining eight queries retrieved less than 20% of their documents. The improvement over SS recall rate ranged from 2% to 58%. Fourteen queries had

an improved recall rate of more than 38%; 10 queries improved from 20% to 28%; and improvement rates between 2% and 18% were registered for 16 queries. Figure 8.4 shows the distribution of improvement in AMMS recall rates over SS recall rates.

Table 8.5. Recall rates for advanced manually-modified searches (AMMS)

Noun	AMMS	Recall rate	SS Recall rate	Improvement rate
environment	48	96%	44%	52%
history	47	94%	56%	38%
inhabitant	45	90%	34%	56%
company	42	84%	64%	20%
mail	41	82%	42%	40%
industry	39	78%	20%	58%
house	37	74%	46%	28%
guide	31	62%	30%	32%
show	30	60%	38%	22%
agency	29	58%	18%	40%
reading	29	58%	20%	38%
shopping	28	56%	6%	50%
office	28	56%	6%	50%
birth	27	54%	2%	52%
department	26	52%	36%	16%
result	26	52%	40%	12%
service	25	50%	14%	36%
recipe	24	48%	0%	48%
creation	23	46%	32%	14%
information	21	42%	16%	26%
university	19	38%	22%	16%
animal	19	38%	0%	38%
side	19	38%	14%	24%
pregnancy	17	34%	10%	24%
game	17	34%	12%	22%
friend	16	32%	8%	24%
option	16	32%	14%	18%
fire	15	30%	2%	28%
connection	15	30%	16%	14%
institute	13	26%	10%	16%
dealer	12	24%	6%	18%
defense	11	22%	0%	22%
poem	8	16%	2%	14%
control	8	16%	6%	10%
boy	7	14%	10%	4%
artist	7	14%	2%	12%
download	5	10%	4%	6%
price	5	10%	6%	4%
contestant	4	8%	0%	8%
meal	1	2%	0%	2%

Figure 8.4. Distribution of improvement of recall rates in AMMS over SS



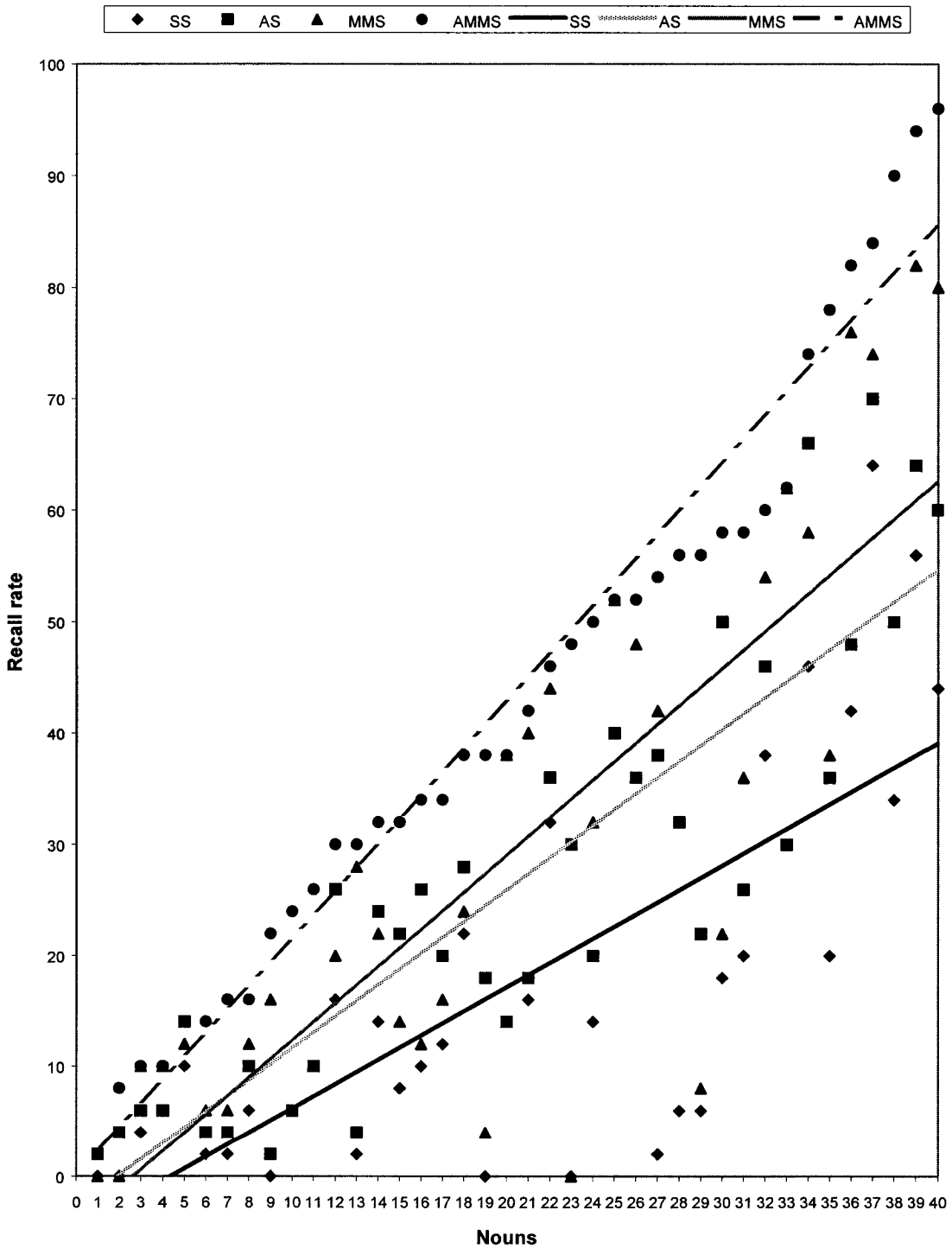
Adding prefixes to the truncated basic noun in the AMMS stage produced the highest rates of recall among the four stages of the searches, because this technique produced the combined results of the previous three stages. The significance of improvement in AMMS recall rates over the recall rates of SS was evaluated by conducting a paired-samples t test. The results of this test indicated that the mean recall rate for AMMS ($M = 44.00$, $SD = 25.27$) was significantly greater than the mean recall rate for SS ($M = 17.70$, $SD = 17.07$), $t(39) = 10.52$, $p = 0.001$. The mean difference was 26.30 between the two recall levels.

8.2.4 *Recall trends*

The last three stages of searches (AS, MMS and AMMS) produced different levels of improvement in recall rates, but they all significantly improved the recall rates over those of the SS stage. Improvement in recall rates also varied from noun to noun, suggesting that the different search techniques employed in the search stages did not have exactly the same effect on all nouns. As a whole, however, the searches using the 40 nouns showed a trend of recall rate improvement as illustrated in Figure 8.5. This figure plots a regression analysis of the recall rates produced in the four stages of searches. The different four patterns (explained in the legend on top of the plot area) represent SS, AS, MMS and AMMS, showing the recall rates for each one of the 40 nouns in each one of the four stages. The four different lines in the plot area represent the trend lines

of recall rates in the four stages. The lines show a clear and progressive improvement in the recall rates from SS to AS, from AS to MMS, and from MMS to AMMS, although the trend lines for AS and MMS slightly overlap.

Figure 8.5. Recall rate trends in the four stages of searches



8.2.5 *Failure rates*

Even with the manual addition of prefixes coupled with AltaVista's own right-hand truncation, the AMMS recall levels did not reach those initially achieved using al-Idrisi. The failure rates varied considerably from noun to noun (see Table 8.6). The MD column indicates the number of documents that were missed after the fourth stage of searches in AltaVista, and the third column indicates the rate of failure. Of the 40 nouns, 24 experienced a failure rate of 50% or more, with four (meal, contestant, download, and price) reaching rates of 90% or higher. Eight of the nouns experienced a failure rate between 40% and 48%, five between 16% and 38%, and only three nouns experienced rates of 10% or less. Do the high failure rates reflect the reality of the effectiveness of AltaVista? Or are there morphological explanations for these rates? Should all the documents retrieved by al-Idrisi using a root search really have been retrieved? These important questions are addressed in the next section in order to understand the root factor in retrieving relevant documents.

Table 8.6. AltaVista's search failure rates

Noun	MD	Failure rate
meal	49	98%
contestant	46	92%
download	45	90%
price	45	90%
artist	43	86%
boy	43	86%
poem	42	84%
control	42	84%
defense	39	78%
dealer	38	76%
institute	37	74%
fire	35	70%
connection	35	70%
option	34	68%
friend	34	68%
pregnancy	33	66%
game	33	66%
animal	31	62%
side	31	62%
university	31	62%
information	29	58%
creation	27	54%
recipe	26	52%
service	25	50%
result	24	48%
department	24	48%
birth	23	46%
office	22	44%
shopping	22	44%
reading	21	42%
agency	21	42%
show	20	40%
guide	19	38%
house	13	26%
industry	11	22%
mail	9	18%
company	8	16%
inhabitant	5	10%
history	3	6%
environment	2	4%

8.3 The root factor

Queries entered in the fourth stage of AltaVista's searches produced the highest number of documents. Until now, any document that was not retrieved by any of the queries on the 40 nouns has been considered a missed document (MD) that should have been retrieved in an Arabic IR environment, using a search-by-root feature. Therefore, AltaVista's failure rate is measured by comparing the number of MDs from the AltaVista searches with the number of documents retrieved by al-Idrisi (Table 8.6). Since al-Idrisi retrieved all the documents that AltaVista failed to retrieve, each one of these documents must be related in one way or another to the noun by the Arabic root: the document contains a word or words that share the same root with the search noun. The search-by-root option used in al-Idrisi produced these documents, and so far all 50 documents retrieved by each one of the nouns have been treated as relevant documents based on their containing a derivation of the root of the noun. What has not been considered yet is the validity of the assumption that all these documents should have been retrieved in the first place and, consequently, if the keywords that retrieved these documents are actually related to the original noun, that is, belong to the noun block.

In a traditional IR system, an inflectional variation of an English noun (a plural or feminine form) is usually the closest semantic variation of this noun. Dog and dogs, for example, are closely related, as are prince and princess. In Arabic, as

explained in more detail in Chapter 2, the morphological structure of the language creates clusters of words that are grouped under one root but are not necessarily related in terms of meaning. A searcher for the Arabic noun *jml* (camel) would be interested in a document that contains *jmal* (camels) or *alcml* (the camel), but would not be interested in a document that contains *jmlh* (phrase), even though *jml* and *jmlh* share the same root (*jml*).

To investigate the morphological reasons behind the MDs for each noun, the keywords that retrieved each of these documents were extracted and arranged to show their distribution in each document. Then, a table was created for each noun containing the Arabic keyword and its English translation. The following are systematic analyses of each of the keywords that retrieved an MD in al-Idrisi. For each noun, the analysis includes an assessment of the root factor in the success of the search (how the keyword is related to the original noun) and an explanation of how an MD, when appropriate, should be retrieved in AltaVista. MDs that were retrieved because of the occurrence of a keyword that is not related to the original noun (it does not belong to the noun block) are judged as false hits.

The analyses are arranged alphabetically by the English equivalent of the Arabic noun (included in parentheses). The root of each noun is given with an explanation of the general meanings (when appropriate) of the keywords that retrieved the MDs in al-Idrisi. The number of MDs is indicated, and an

explanation of which document/s should have been retrieved is given. Then, a two-column table is presented to show the exact meaning of these keywords and their relationship (if any) to the search noun. The first column lists the keywords, and the second column gives their English translations. In some cases, where two keywords in a table look identical, these keywords are homographs. A 'v.' included in parentheses after a keyword indicates that the keyword is a verb. If a keyword is from the same noun block as the search noun, the number of MDs that contain it is indicated in parenthesis to its right (the noun is in bold face). This number indicates the number of MDs that are relevant, and therefore ideally should have been retrieved by AltaVista.

agency (*wkalh*)

Root: *wkl*

MDs: 21

Although the six keywords listed in the table come from the root, it is clear that none of them is related to agency and they do not belong to its noun block. The 29 documents retrieved by AltaVista represent the ones that should be retrieved in actual searches. The 21 MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
awkl	delegate (v.)	*kyaly*	(proper name)
wkil	dealer	*kla*	both
atkl	rely on (v.)	*twkil*	commissioning

animal (*Hywan*)

Root: *Hyw*

MDs: 31

The keywords represent concepts related to life, civic divisions, greeting, and they do not relate to "animal" in any way. AltaVista retrieved all documents that contain keywords that relate to the noun, animal. The 31 MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
Hy	alive	*yHya*	(proper name)
aHyaay	biologist	*aHyaa*	resurrection
Hyah	life	*tHyh*	salutation
yHya	live (v.)	*Hya*	salute (v.)
Hy	neighbourhood	*Hywy*	vital

artist (*fnan*)

Root: *fnn*

MDs: 43

The first keyword (*kalfanan*) is the noun with the prefix combination *kal-* (a combination of *k* and a*l*), and occurs in one document. This document could be retrieved by AltaVista through including *kal-* in the search term. While artistic and art are conceptually related to artist, they are not derivatives of this noun (they are not from the same noun block). The remaining two keywords are not related to artist. Of the 43 MDs, 42 are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
kalfnan	(like the) artist (1 doc.)	*alfntyn*	(island's name)
fny	artistic	*fny*	technical
fn	art	*fny*	technician

birth (*wladh*)

Root: *wld.*

MDs: 23

A.D. (*mylady*) is used in Arabic to indicate the Christian year, while generation is related to the concept of generating power. All the 23 MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
mylady	A.D.	*wld*	was born (v.)
wlyd	(proper noun)	*wald*	parent
twlyd	generation	*mylad*	birthday
wld	boy	*wld*	(proper noun)

boy (*wld*)

Root: *wld*

MDs: 43

The keyword *awlad* is the irregular plural form of *wld* and occurs in five documents. These five documents should be retrieved by AltaVista, while the remaining 38 are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
mylady	A.D.	*wlyd*	infant
wladh	birth	*wald*	parent
mylad	birthday	*awld*	produce (v.)
awlad	boys (5 docs)	*wlyd*	(proper noun)
twlyd	generation	*twald*	reproduce (v.)
mwld	generator		

company (*Srkh*)

Root: *Srk*

MDs: 8

All MDs are false hits

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
mSrk	atheist	*Sark*	participate (v.)
mStrkh	joint	*mSarkh*	participation
mSark	participant		

connection (*wpl*)

Root: *wpl*

MDs: 35

All MDs are false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
wpwl	arrival	*mtwapl*	continuous
ypl	arrive (v.)	*wapl*	(proper name)
awpl	attach (v.)	*fypl*	(proper name)
atpalat	communication	*ypl*	reach (v.)
atpal	contact	*ytpl*	relate (v.)
wapl	continue (v.)	*plh*	relation

contestant (*mtsabq*)

Root: *sbq*

MDs: 35

All MDs are all false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
msbq	advance	*sbq*	precede (v.)
asbq	ancestor	*sabqh*	precedence
msabqh	competition	*sabqh*	precedent
sabq	former	*sabq*	previous
sabq	last	*sbaq*	race
sabq	past	*waplh*	suffix

control (*tHkm*)

Root: *Hkm*

MDs: 42

All MDs are false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
Hkm	rule	*mHkmh*	court
Hkwmh	government	*mHakmh*	trial
Hakm	governor	*Hkm*	rule (v.)
mHkm	tight	*Hkym*	wise

creation (*xlq*)

Root: *xlq*

MDs: 27

Four of the documents were not retrieved because of the occurrence of the *kSydh* between the characters of *xlq*. The remaining 23 MDs are false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
yxlq	create (v.)	*mxlwq*	creature
xl_q	creation (2 docs)	*xlaaq*	creatures
x_lq	creation (2 docs)	*mxlwqat*	creatures
xlaq	creative	*yxlq*	invent (v.)
xalq	creator	*axlaqy*	moral

dealer (*wkyl*)

Root: *wkl*

MDs: 38

The keyword *wklaa* is the irregular plural form of the noun and it occurs in 11 documents that should be retrieved by AltaVista. Another keyword that should be retrieved by AltaVista is *kwkyl* (the prefix *k-* attached to the noun) and it occurs in one document. The remaining 26 MDs are false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
wkalh	agency	***kwkyl***	(like a dealer) dealer (1 doc.)
kla	both	*kayaly*	(proper name)
wklaa	dealers (11 docs)	*mtwkl*	(proper name)
awkl	delegate (v.)	*atkl*	rely on (v.)
mwkl	delegator		

defense (*dfac*)

Root: *dfc*

MDs: 39

All MDs are false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
ydfc	force (v.)	*dfc*	push (v.)
andfc	run (v.)	*dafc*	incentive
adfc	prevent (v.)	*dfch*	payment
dfc	pay (v.)	*dfch*	instalment

department (*qsm*)

Root: *qsm*

MDs: 24

The keyword *aqsam* is the irregular plural form of the noun. The six documents that contain it should be retrieved by Altavista. The remaining 18 MDs are false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
aqsam	departments (6 docs)	*tqsym*	division
yqsm	divide (v.)	*qasm*	(proper noun)

download (*tnzyl*)

Root: *nzl*

MDs: 45

All MDs are false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
tnazl	compromise	*nzwl*	descent
ynzl	descend (v.)	*mnzl*	house
tnazly	descending	*mnzlh*	level

environment (*byah*)

Root: *bwa*

MDs: 2

Both MDs are false hits.

Keyword	**Keyword's translation**
tbwa	occupy (v.)

fire (*nar*)

Root: *nwr*

MDs: 35

Four of the documents were not retrieved because of the occurrence of the *kSydh* between the characters of *nar*. The irregular plural form of the noun (*nyran*) occurs in one of the MDs and this document should be retrieved by AltaVista. The remaining 30 MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
mnyr	bright	*mnarh*	lit
n_ar	fire (4 docs)	*mnawrh*	military exercise
nyran	fires (1 doc)	*nyrwn*	(proper name)
nwr	light	*anwr*	(proper name)
mnarh	lighthouse	*nwry*	(proper name)

friend (*pdyq*)

Root: *pdq*

MDs: 34

One of the MDs was not retrieved because of the occurrence of the *kSydh* between the characters of the noun, while five others included the irregular plural form (*apdqaa*) of the noun and they should be retrieved by AltaVista. The remaining 28 MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
tpdyq	approval	***wpdy_qh***	(and his) friend (1 doc.)
padq	approve (v.)	***apdqaa***	friends (5 docs)
tpdyq	authentication	*pdaqh*	friendship
pdq	believe (v.)	*padq*	honest
mpdq	believer	*pdq*	honesty
qdqh	charity	*mpadqh*	signing
mpdaqyh	credibility	*apdq*	tell the truth (v.)
tqdq	donate (v.)		

game (*lcbh*)

Root: *lcb*

MDs: 33

Eight of the MDs were not retrieved because of the occurrence of the irregular plural form of the noun (*alcab*); they should be retrieved by AltaVista. The remaining 25 MDs are false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
alcab	games (8 docs)	*lacb*	player
alcb	play (v.)	*mlcb*	stadium

guide (*dlyl*)

Root: *dll*

MDs: 19

The irregular plural form of the noun (*adlh*) occurs in four of the MDs and these should be retrieved by AltaVista. The remaining 15 MDs are false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
istdl	conclude (v.)	*dl*	*show* (v.)
dl	Dell	*mdlwl*	*significance*
adlh	guides (4 docs)	*dlaly*	*significant*
dlalh	notion	*ydl*	*signify* (v.)
ydl	point (v.)		

history (*taryx*)

Root: *arx*

MDs: 3

All MDs are false hits.

Keyword	**Keyword's translation**
marx	dated

house (*byt*)

Root: *byt*

MDs: 13

The *kSydh* prevented the retrieval of four of the MDs, while the irregular plural form of the noun (*bywt*) prevented the retrieval of a further four. The remaining five MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
by_t	house (4 docs)	***bywt***	houses (4 docs)
ybyt	sleep (v.)	*bat*	become (v.)

industry (*pnach*)

Root: *pnc*

MDs: 11

All MDs are false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
mpnc	factory	*mpnwc*	product
pnc	make (v.)	*pnc*	production
tpnyc	manufacturing	*pnacy*	synthetic

information (*mclwmh*)

Root: *clm*

MDs: 29

All MDs are false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
tclym	education	*clm*	science
tclymyh	educational	*clmy*	scientific
clm	flag	*calm*	scientist
calmy	global	*clmanyh*	secularism
cwlmh	globalization	*clamh*	sign
tclymh	instruction	*yclm*	teach (v.)
aclm	know (v.)	*tclym*	teaching
tclm	learn (v.)	*calmyh*	universal
aclam	media	*calm*	world

inhabitant (*sakn*)

Root: *skn*

MDs: 5

All MDs are false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
skwn	quiet	*sknyh*	residential
yskn	inhabit (v.)	*askan*	habitation

institute (*mchd*)

Root: *chd*

MDs: 37

It is clear that the root-produced keywords are not related to institute. The only keyword that belongs to the noun block of *mchd* is *mcahd*, which is the irregular plural form of *mchd*. The seven documents containing this keyword should be retrieved by AltaVista, while the remaining 30 of the 37 MDs are false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
chdh	care	*cahd*	promise (v.)
mcahd	institutes (7 docs)	*chd*	reign
tchd	commit (v.)	*mcahdh*	treaty
tchd	commitment		

mail (*bryd*)

Root: *brd*

MDs: 9

One MD was missed by AltaVista because of the presence of the *kSydh*. The remaining MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
bard	cold	***bry_d***	mail (1 doc)
tbryd	cooling	*brwdi*	(proper name)

meal (*wjbh*)

Root: *wjb*

MDs: 49

All MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
bmwjb	according to	*mtwjb*	obligation
yjb	must (v.)	*ayjaby*	positive
wajbh	necessary	*ywjb*	require (v.)
ywjb	necessitate (v.)	*wajbat*	responsibilities
wjwb	necessity		

office (*mktb*)

Root: *ktb*

MDs: 22

Two of the MDs contain the keyword (*mkatb*), which is the irregular plural form of the noun, and should be retrieved by AltaVista. The remaining 20 MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
katb	author	*ktb*	write (v.)
ktab	book	*katbh*	writer
ktybh	brigade	*ktabh*	writing
mkatb	offices (2 docs)		

option (*xyar*)

Root: *xyr*

MDs: 34

All MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
xyr	bounty	*axtyaryh*	mayoral
xyryh	charitable	*axtyary*	optional
axtar	choose (v.)	*axtr*	select (v.)
axtyar	choosing	*axtyar*	selection
mxtar	mayor		

poem (*qpydh*)

Root: *qpd*

MDs: 42

All MDs are false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
aqtpady	economic	*mqpd*	intention
aqtpad	economy	*yqpd*	mean (v.)
qpd	intended (v.)		

pregnancy (*Hml*)

Root: *Hml*

MDs: 33

All MDs are false hits.

Keyword	Keyword's translation	Keyword	Keyword's translation
yHml	bear (v.)	*Haml*	holder
Hamlh	carrier	*aHtmal*	possibility
aHml	carry (v.)	*mHtml*	possible
Hmlt	force (v.)	*tHml*	suffer (v.)
ytHml	suffer (v.)	*tHmyl*	upload
aHml	hold (v.)	*Hmwlh*	weight

price (*vmn*)

Root: *vmn*

MDs: 45

All MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
vmanyh	eight	*vmyn*	expensive
vmanyn	eighty	*vamn*	eighth

reading (*qraah*)

Root: *qra*

MDs: 21

All MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
qran	Qur'an	*qra*	read (v.)
qrany	Qur'anic	*qara*	reader

recipe (*wpfh*)

Root: *wpf*

MDs: 26

All MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
pfh	adjective	*twpyf*	description
pfh	capacity	*pfh*	feature
ypf	describe (v.)	*mwapfh*	specification

result (*ntyjh*)

Root: *ntj*

MDs: 24

Six of the MDs contain the irregular plural form of the noun (*ntaaj*), and they should be retrieved by AltaVista. The remaining 18 are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
yntj	produce (v.)	*mntjh*	productive
mntj	producer	*natj*	resulting
mntwj	product	***ntaaj***	results (6 docs)
antaj	production		

service (*xdmh*)

Root: *xdm*

MDs: 25

All MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
xadm	servant	*astxdm*	use (v.)
yxdm	serve (v.)	*mstxdm*	user

shopping (*tswq*)

Root: *swq*

MDs: 22

All MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
syaq	context	*saq*	leg
saaq	driver	*swq*	market
syaqh	driving		

show (*crD*)

Root: *crD*

MDs: 20

All MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
ytcarD	contrast (v.)	*carD*	oppose
ycrD	expose	*mcarDh*	opposition
yctrD	object (v.)	*astcrD*	review
actrD	object (v.)	*cryD*	wide

side (*jhh*)

Root: *wjh*

MDs: 31

One of the MDs contains the irregular plural form (*jhat*) of the noun and should be retrieved by AltaVista. The remaining 30 MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
mwajhh	confrontation	*wjh*	guide (v.)
awjh	direct (v.)	*atjh*	head (v.)
atjah	direction	*wajhh*	interface
wjh	face	***jhat***	sides (1 doc.)
twjyh	guidance	*wjhh*	view

university (*jamch*)

Root: *jmc*

MDs: 31

All MDs are false hits.

Keyword	**Keyword's translation**	**Keyword**	**Keyword's translation**
ajmc	agree (v.)	*ajmac*	consensus
jmyc	all	*mjmc*	council
jmcyh	assembly	*jmch*	Friday
jmc	bind (v.)	*jmach*	group
yjmc	collect (v.)	*ajtmac*	meeting
tjmic	collection	*ajtmacy*	social
jmacy	collective	*mjtmc*	society
mjmc	complex	*mjmwc*	sum

8.4 A summary

The previous sections showed the results of the various stages of searches in AltaVista and the analyses of the MDs identified after finishing these stages. The MDs were examined on a noun-by-noun basis, and the documents that were false hits were identified, as were the ones that should be retrieved by AltaVista. For the latter, explanations were given as to why AltaVista did not retrieve them. Identifying the false hits helps in determining the effect of root retrieval on precision, while identifying the documents that should be retrieved by AltaVista helps in identifying search features that were not used in the search stages and that can be used to retrieve these documents.

Table 8.7 summarizes the results of the analyses that were conducted on the MDs for each of the 40 nouns, and tabulates the numbers of MDs and false hits (FHs) based on the analysis. The fourth column (AVD) stands for AltaVista document and tabulates the number of documents that were missed by AltaVista but ideally should have been retrieved. The values in this column are obtained by subtracting the value of FHs from the value of MDs. For example, there are 19 MDs for the noun "guide", of which 15 are FHs, leaving the number of AVDs that were missed but nonetheless are morphologically related to the search noun as four. The table clearly shows the high number of false hits and, therefore, the adverse effect of root retrieval on precision. In 26 cases, all MDs are false hits, meaning that there are no documents that ideally should have been

retrieved by AltaVista. The remaining 14 nouns share among them 74 documents that should have been retrieved by AltaVista, for an average of less than six documents per noun, ranging from a low of one document to a high of 12 documents.

The cause of failure in AltaVista, and therefore the presence of AVDs in Table 8.7 is mostly related to keywords that represent the irregular plural forms of nouns. These are usually formed through the addition of infixes and cannot be retrieved through truncation or through manual attachment of prefixes (but can be retrieved by a root search). In addition, a character called *kSydh* presented by an underscore (_) prevented the retrieval of some documents. The use of this character is a peculiar aspect of presenting Arabic words in electronic format; it is used between two characters for the sole purpose of lengthening the distance between them, making the word more visually appealing. In two cases, the cause of failure is the presence of a prefix or a prefix combination that it was decided not to include in the prefixes/prefix combinations that were added to the nouns in the third and fourth stages of the searches (see Chapter 6). The prefix *k-* (like) and the prefix combination *kal-* (a combination of *k and al* (the)) occur in two documents that were not retrieved by AltaVista.

Table 8.7. AltaVista's performance record

Noun	MDs	FHs	AVD
dealer	38	26	12
game	33	25	8
house	13	5	8
institute	37	30	7
department	24	18	6
result	24	18	6
friend	34	28	6
boy	43	38	5
fire	35	30	5
guide	19	15	4
creation	27	23	4
office	22	20	2
side	31	30	1
artist	43	42	1
download	45	45	0
environment	2	2	0
animal	31	31	0
control	42	42	0
contestant	46	46	0
connection	35	35	0
information	29	29	0
company	8	8	0
industry	11	11	0
defense	39	39	0
mail	9	9	0
meal	49	49	0
birth	23	23	0
option	34	34	0
poem	42	42	0
inhabitant	5	5	0
pregnancy	33	33	0
price	45	45	0
reading	21	21	0
recipe	26	26	0
agency	21	21	0
service	25	25	0
shopping	22	22	0
show	20	20	0
history	3	3	0
university	31	31	0

Table 8.8 shows a breakdown of the numbers of AVDs among the 14 nouns that produced them (as shown in Table 8.7), and it relates them to the three causes of AltaVista's failure mentioned above. Irregular forms of the plural of nouns caused by far the highest number of failures, accounting for a total of 60 from 12 of the 14 nouns. Second comes the *kSydh*, which caused 13 failures in four nouns. Prefixes affected only two nouns, with a total number of two failures.

This chapter detailed the results of the search experiments, explaining the outcomes of each search stage and analysing the effect of root retrieval on precision. It also explored areas where AltaVista failed to retrieve documents and the causes of the failure. What implications do these results have for adapting AltaVista for use with Arabic text, and for abandoning root-based retrieval in favour of adopting stemming/truncation techniques that could be implemented in ELIR systems in order to handle Arabic nouns? The last chapter of the thesis discusses the results, their ramifications for research on Arabic IR in particular and for CLIR research in general, and outlines the limitations of the research plans for future work.

Table 8.8. Causes of failure in AltaVista

Noun	Irregular plural	*kSydh*	Prefix/prefix combination	Total
dealer	11		1	12
game	8			8
house	4	4		8
institute	7			7
department	6			6
result	6			6
friend	5	1		6
boy	5			5
fire	1	4		5
guide	4			4
creation		4		4
office	2			2
side	1			1
artist			1	1

9. Conclusions

It has been argued from the outset of the thesis that adapting ELIR systems for use with other languages in general, and for use with Arabic in particular, must be investigated at the word level. Investigating ways of enabling these systems to handle words must be undertaken as a preliminary to undertaking the traditional methods of evaluating the performance of IR systems: measuring recall and precision based on the relevance of retrieved documents to information needs expressed in queries submitted by actual searchers. When dealing with IR in a language for which a search engine was not designed (in this case, Arabic using an English-language search engine), the morphological structures of Arabic words (nouns) used in queries are of the utmost importance. If the morphological variations of Arabic nouns cannot be retrieved then documents will be missed in a search.

Chapters 2 and 3 demonstrated how different are the morphologies of Arabic and English, and how the morphological compositions of Arabic nouns can potentially make searching for and retrieving nouns a challenging task. To investigate this potentiality in an ELIR system environment, a methodology was developed using two Web-based IR systems (search engines): an ELIR system (AltaVista) and an Arabic IR system (al-Idrisi). A comparison of the performances of these two systems in finding documents using Arabic nouns would reveal how closely an ELIR system can approach the performance of an

Arabic-language system. Further, it would permit an examination of stemming versus root retrieval as possible techniques in the case of Arabic-language databases. Root retrieval has been advocated by al-Kharashi (1991), Abu Salem (1992), al-Kharashi and Evens (1994), Hmeidi, Kanaan and Evens (1997), and Abu Salem, al-Omari and Evens (1999).

These earlier researchers had employed the traditional measures of recall and precision in their experimental evaluations of Arabic retrieval systems. They did not consider problems of adapting ELIR systems for use with Arabic, however, and they did not attack the problem at the fundamental noun level. The methodology in this dissertation, in contrast, has adopted the Arabic noun and its variants as the single most important aspect for investigation; measures of effectiveness, therefore, are based on a linguistic measure rather than on relevance decisions of retrieved documents based upon real or hypothetical users. A document is considered relevant to a query if it contains a noun or nouns that are morphological variants of the noun used to formulate that query.

Employing this notion of relevance, searches were conducted using AltaVista to investigate how well this example of an ELIR system performs in an Arabic-language environment, and what search features might be added to better adapt it for use with Arabic nouns. These searches also allowed an assessment of the performance of root retrieval versus stemming retrieval and improved the understanding of the effect of root retrieval on precision. The following four

sections discuss the types of retrieval problems created by the morphological composition of Arabic nouns in an ELIR system, and suggest searching and indexing techniques that can be implemented to overcome these problems. They also look at the adverse effect of root searching on the retrieval of Arabic nouns and discuss the implications for CLIR of the methodological approach adopted in the thesis.

9.1 Arabic nouns in an ELIR system

The first two stages of searching in AltaVista – using only the original noun, and then the noun plus right-hand suffix truncation – are easy to implement on a typical ELIR, but showed how the engine as a consequence produced low recall levels, missing a high number of documents. The performance of the engine in these two stages was affected by the absence of left-hand prefix truncation that allows truncation at the beginning of an Arabic noun, and therefore can take account of the presence of prefixes in these nouns. Once the prefixes were added to the search terms (a manual simulation of left-hand truncation) in the last two stages of the searches, the recall levels increased dramatically (Figure 8.5). In this experimental environment, we can conclude that the biggest obstacle facing effective retrieval in Arabic is the occurrence of prefixes. These are very commonly used with Arabic nouns, and it is extremely important that an ELIR system should be able to accommodate them. Unfortunately, the manual addition of such prefixes is both very time consuming and prone to spelling

mistakes at the query input stage; it also requires a very good knowledge of Arabic morphology.

While the manual additions of prefixes enabled the ELIR system to handle the problem of prefixes, other problems arose because of the presence of morphological variants in the Arabic nouns. As Chapter 2 indicated, many of the plural forms of Arabic nouns are irregular; unlike regular plural forms, which are formed by adding suffixes to the singular nouns (easily retrieved by right-hand truncation), irregular plurals are usually formed by the addition of infixes to the stem forming the basic noun. Theoretically, this can be handled with middle truncation, but the user has to be well versed in the language to know where to place the truncation symbol. For example, the plural form of *mlcb* (playground) is *mlacb*; the user must know this in order to place the truncation symbol between the "*l*" and the "*c*"" and retrieve both forms of the noun (*ml*cb* will retrieve occurrences of the singular and plural forms).

The last retrieval problem identified in the search experiments is the occurrence of the special character "__" (*kSydh*). This character is used to lengthen the distance between two Arabic characters for aesthetic purposes and is indexed by the IR system as a separate character. If the *kSydh* is present in an Arabic noun, that noun cannot be retrieved unless entered with the *kSydh*. The user must know the position of this character in the noun and enter it accordingly.

9.2 Adapting an ELIR system for use with Arabic

AltaVista's failure to undertake left-hand truncation constituted a major drawback in using it as a search engine for Arabic retrieval. One solution to the prefix problem is offered in existing Arabic IR systems, including al-Idrisi. Advanced stemming is applied to the words to strip them of prefixes and suffixes. For example, if a user enters the noun *jrydh* (newspaper) as a search term, al-Idrisi will retrieve documents that contain the exact match of this word and any other forms of it containing prefixes, suffixes, or a combination of the two. This is accomplished through the implementation of algorithms that isolate the prefixes and suffixes and allow the entry of index terms under the stemmed noun (the noun stripped of prefixes and suffixes). In an ELIR system, it might be difficult to implement such algorithms if the system does not recognize the language being indexed. The ELIR system has to have a mechanism by which it identifies the Arabic words at the indexing stage and applies the prefix/suffix-stripping algorithms to them. Otherwise, this system will not know when to apply the algorithms and when to ignore them.

An alternative to automatic stripping at the indexing stage is automatic inclusion of prefixes at the search stage. In these search experiments, seven prefixes/prefix combinations were used to improve recall levels. The ELIR system might be modified to automate what was done here manually. Again, the system must have a mechanism to identify the word being entered in a search as an Arabic

word. Once the word is identified, the system must then include the word in its original form in addition to the other seven forms, thereby generating a query that would retrieve documents containing any of the eight noun forms. For example, if a user is searching for documents containing the noun *dftr* (notebook), the system would search for all occurrences of *dftr* alone or attached to the automatically entered prefixes.

In the experiments, the irregular plurals of Arabic nouns presented the most challenging problem for retrieval using the ELIR system. Some ELIR systems, especially Web search engines, provide automatic stemming of regular English plural forms (see 5.2.2). As explained in chapters 2 and 3, irregular plural nouns are not as common in English as they are in Arabic. Some ELIR systems handle the irregular English forms through stemming algorithms (Porter 1980), while others, including Web search engines, assume that the user knows the forms and the engines themselves do not provide any indexing capabilities to handle them.

Retrieval by the root of the noun can solve this problem in an Arabic IR system because the singular and plural forms share the same root: both are retrieved when either one is entered as a search term. In an ELIR system, a possible solution for this problem could be the inclusion of a list of irregular plural forms along with their singular forms in the indexing algorithms. At the indexing stage, whenever a document containing either form is countered, it is indexed in a way that allows its retrieval no matter which one of the forms is entered in a

query. This is analogous in English to including the singular noun "tooth" plus its corresponding irregular plural form "teeth" as a linked pair in the index. A document including the noun "teeth" would then be retrieved whenever a query contains either the nouns "tooth" or "teeth".

For obvious reasons, AltaVista failed to retrieve nouns that contained the *kSydh* between their characters. These nouns could have been retrieved only if the exact position of the *kSydh* had been known. But there is no way for the user to know this; the use of this character is arbitrary, and even if it exists in a noun in one document, it may not exist in the same noun in another document. The best way for an ELIR system to deal with *kSydh* is to ignore it altogether at the indexing stage. AltaVista does ignore certain special characters in indexing, such as %, $, / and #; it could be modified to ignore the *kSydh* as well. For example, the word *n_ar* (fire) would be indexed as *nar*.

9.3 Root retrieval

Previous research on Arabic IR mainly has compared root retrieval with stemming as indexing methods for effective IR. Such research was conducted on experimental IR systems designed specifically for Arabic and including both stemming and root indexing capabilities. While not disregarding the important role of stemming, al-Kharashi (1991) advocates the use of root retrieval, as do Abu-Salem (1992) and Hmeidi, Kanaan and Evens (1997). At the outset of the

research presented in this dissertation, this earlier work formed its starting point and therefore employed a root-stemming algorithm available on al-Idrisi to create a test set of documents. The results now strongly suggest, however, that in order to adapt ELIR systems to operate effectively with Arabic, stemming in fact is a better approach than root retrieval. This conclusion has been drawn after matching the performance of the root-retrieval based al-Idrisi search engine with an ELIR system using search techniques that are equivalent to stemming. That is to say, the ends of nouns were truncated and prefixes added to them manually. These two procedures, when performed together, have an effect equivalent to the advanced stemming of an Arabic noun (stripping it both of prefixes and suffixes). When the documents that had not been retrieved by the ELIR system after both suffix and prefix stemming had been applied were judged for morphological relevance, a majority were found to be irrelevant. In other words, the root retrieval capabilities available on al-Idrisi were generating irrelevant hits. This finding strongly supports the case for stemming rather than root retrieval as an effective means to retrieve Arabic documents.

While root-based retrieval has appeared to be a logical choice for researchers and developers of Arabic IR systems (Hmeidi, Kanaan and Evens 1997), it is clear that the number of false hits produced by root retrieval in the experiments conducted in this research is great. The search experiments show that an ELIR system can be modified to handle Arabic nouns without the need for developing root-retrieval capabilities. Equipping an ELIR system with root-retrieval

capabilities is not necessary to handle the morphology of the Arabic language, and indeed is likely to prove counter-productive. Although the development of such capabilities may be feasible in small experimental systems for research purposes, implementation in operational systems will be a costly and time-consuming endeavour. Worse still, however, it is likely to reduce precision by retrieving documents that contain nouns sharing a root with the noun or nouns used in the search query, but which semantically are irrelevant to those query nouns. In other words, a more complex and expensive system will have been developed to perform less effectively.

9.4 Language-dependent investigation methods and CLIR

Each language has its own linguistic properties that affect IR. In an age where information is being integrated in multilingual environments, and where the Web has introduced the languages of the world to users of different linguistic backgrounds, IR research should focus on integrating the findings of traditional research with the individual linguistic properties of each language. This dissertation has dealt with one language—Arabic—and investigated the way it is handled by an ELIR system.

An area of IR related to the topic of the dissertation is CLIR; the main occupation of this research area is to solve the problem of matching queries and documents across different languages. Researchers in CLIR have focussed on

finding the best approach to translation, and the problems that it creates depending on the two or more languages used (Sperer and Oard 2000, Turid, Pirkola and Jarvelin 2001). The language-dependent aspect of CLIR is, then, translation. Obviously, each language creates its own translation problems, and these problems have to be dealt with accordingly in a CLIR environment. In our research, we have focussed on the morphological structure of languages and their impact on the effectiveness of IR. We believe that morphology should also be a concern for CLIR. Translation is only a tool to create documents and queries in CLIR systems; finding the words of the queries in documents is a problem in Arabic for an ELIR system, and there is no reason to believe that Arabic is unique in this respect. CLIR has not yet focussed greatly on the problems that remain even when the query has been translated accurately into the language of the document collection: can the search engine, if intended to work across two languages with widely differing morphologies, perform equally well in both?

9.5 Limitations of the research

As is the case with any research undertaking, this thesis has its limitations. It set out to investigate the adaptation of an ELIR system for use with Arabic texts and to explore the issue of root retrieval and stemming in Arabic IR. One limitation of this approach is that only one ELIR system (search engine) was used to conduct search experiments and therefore to be evaluated in terms of its

performance compared with the performance of an Arabic IR system. One possibility would have been to select two ELIR systems with different features and to conduct a comparative study of their performance. However, for logistical reasons and given the scope of the thesis, it was not practical to use more than one system in the present research.

Another limitation of the research is the choice of the search engine. It was explained in Chapter 5 that search engines have different features and they are constantly changing. The search engine used in this research (AltaVista), has undergone many changes in the last few years, and other search engines have added features that were not available at the time when the research was initiated. How these new features affect the findings of the research remains to be seen.

9.6 Future work

This research has investigated the use of an ELIR system to retrieve Arabic nouns and identified search/indexing features that could be implemented in this system to improve retrieval effectiveness and adapt the system for use with Arabic text. An IR system specifically designed for Arabic was used as a test benchmark to gauge the performance of the ELIR system. Its retrieve-by-root search/indexing feature was also evaluated based on the morphological

relationship between the words grouped under one root and the search terms used in queries.

A special concept of morphological relevance has been defined and used in the evaluation of retrieval effectiveness. The search experiments utilized methods and techniques to match query nouns with documents containing morphological variants of these nouns. The research findings clarify IR issues concerning individual word matching, and such word matching is the starting point of any search query/document collection matching process in any language. The next stage is to investigate the implications of these findings when applied to actual queries that express actual information needs of real users.

Plans for future work include a study of genuine user queries in a system that implements the improved search/indexing techniques suggested by the findings. This study will employ traditional measures of recall and precision to evaluate the effectiveness of the system. In another work, it is planned to evaluate the performance of ArabVista, the Arabic version of AltaVista. This search engine combines the features of a typical ELIR system and those of a specialized Arabic one. A study of its features will contribute to an understanding of the emerging problems associated with IR in different languages, and will advance ideas on the investigation of language-dependent aspects of IR.

Appendix A

Prefixes and prefix combinations in the Arabic script

	Prefixes and prefix combinations
A1	ال
A2	ب
A3	ف
A4	ك
A5	ل
A6	و
A7	بال
A8	فال
A9	كال
A10	لل
A11	وال
A12	وب
A13	وبال
A14	ول
A15	ولل

Appendix B

Test nouns in the Arabic script

	Noun
N1	رسول
N2	أولاد
N3	قرآن
N4	طلاق
N5	زوج
N6	كذب
N7	دين
N8	تجاره
N9	خمر
N10	شعر
N11	لباس
N12	تحريم
N13	زواج
N14	أسره
N15	حق
N16	حب
N17	أرواح
N18	حكم
N19	مسلم
N20	شرع
N21	حلال
N22	حرام
N23	لحم
N24	دم
N25	فائده
N26	طاعه
N27	طعام
N28	إيمان
N29	موت
N30	شرب
N31	بنت
N32	نفقه
N33	أرض
N34	أمانه
N35	رشوه

Appendix C

Noun data set in the Arabic script

Noun	Root
بريد	برد
بيئه	بوء
بيت	بيت
تاريخ	أرخ
تحكم	حكم
تسوق	سوق
تنزيل	نزل
ثمن	ثمن
جامعه	جمع
جهه	وجه
حمل	حمل
حيوان	حيو
خدمه	خدم
خلق	خلق
خيار	خير
دليل	دلل
دفاع	دفع
ساكن	سكن
شركه	شرك
صديق	صدق
صناعه	صنع
عرض	عرض
فنان	فنن
قراءه	قرء
قسم	قسم
قصيده	قصد
لعبه	لعب
متسابق	سبق
معلومه	علم
معهد	عهد
مكتب	كتب
نار	نور
نتيجه	نتج
وجبه	وجب
وصفه	وصف
وصل	وصل
وكاله	وكل
وكيل	وكل
ولاده	ولد
ولد	ولد

Appendix D

Simple and advanced searches in AltaVista in the Arabic script

Noun	Root	SS	AS
بريد	برد	بريد	بريد*
بيئه	بوء	بيئه	بيئ*
بيت	بيت	بيت	بيت*
تاريخ	أرخ	تاريخ	تاريخ*
تحكم	حكم	تحكم	تحكم*
تسوق	سوق	تسوق	تسوق*
تنزيل	نزل	تنزيل	تنزيل*
ثمن	ثمن	ثمن	ثمن*
جامعه	جمع	جامعه	جامع*
جهه	وجه	جهه	جهه*
حمل	حمل	حمل	حمل*
حيوان	حيو	حيوان	حيو*
خدمه	خدم	خدمه	خدم*
خلق	خلق	خلق	خلق*
خيار	خير	خيار	خيار*
دليل	دلل	دليل	دليل*
دفاع	دفع	دفاع	دفاع*
ساكن	سكن	ساكن	ساكن*
شركه	شرك	شركه	شرك*
صديق	صدق	صديق	صديق*
صناعه	صنع	صناعه	صناع*
عرض	عرض	عرض	عرض*
فنان	فنن	فنان	فنان*
قراءه	قرء	قراءه	قراء*
قسم	قسم	قسم	قسم*
قصيده	قصد	قصيده	قصيد*
لعبه	لعب	لعبه	لعب*
متسابق	سبق	متسابق	متسابق*
معلومه	علم	معلومه	معلوم*
معهد	عهد	معهد	معهد*
مكتب	كتب	مكتب	مكتب*
نار	نور	نار	نار*
نتيجه	نتج	نتيجه	نتيج*
وجبه	وجب	وجبه	وجب*
وصفه	وصف	وصفه	وصف*
وصل	وصل	وصل	وصل*
وكاله	وكل	وكاله	وكال*
وكيل	وكل	وكيل	وكيل*
ولاده	ولد	ولاده	ولاد*
ولد	ولد	ولد	ولد*

Appendix E

Samples of manually modified and advanced manually-modified searches in the Arabic script

Noun	MMS	AMMS
وصل	وصل الوصل و الوصل للوصل بالوصل ووصل لوصل بوصل	وصل* الوصل* و الوصل* للوصل* بالوصل* ووصل* لوصل* بوصل*
خلق	خلق الخلق و الخلق للخلق بالخلق وخلق لخلق بخلق	خلق* الخلق* و الخلق* للخلق* بالخلق* وخلق* لخلق* بخلق*
وكيل	وكيل الوكيل و الوكيل للوكيل بالوكيل ووكيل لوكيل بوكيل	وكيل* الوكيل* و الوكيل* للوكيل* بالوكيل* ووكيل* لوكيل* بوكيل*
تنزيل	تنزيل التنزيل و التنزيل للتنزيل بالتنزيل وتنزيل لتنزيل بتنزيل	تنزيل* التنزيل* و التنزيل* للتنزيل* بالتنزيل* وتنزيل* لتنزيل* بتنزيل*
بيئه	بيئه البيئه و البيئه للبيئه بالبيئه وبيئه لبيئه ببيئه	بيئ* البيئ* و البيئ* للبيئ* بالبيئ* وبيئ* لبيئ* ببيئ*

References

Abbott, N. (1938). *The rise of the northern Arabic script and its kuranic development*. Oriental Institute Publications, Vol. L, Chicago: University of Chicago Press.

Abu Salem, H. (1992). *A microcomputer based Arabic bibliographic information retrieval system with relational thesauri*. Unpublished doctoral dissertation, Computer Science department, Illinois Institute of Technology, Chicago.

Abu Salem, H., M. al-Omari and M. Evens. (1999). Stemming methodologies over individual query words for an Arabic information retrieval system. *Journal of the American Society for Information Science*, 50 (6): 524-529.

Adriani, M. (2000). Using statistical term similarity for sense disambiguation in cross - language information retrieval. *Information Retrieval*, 2 (1): 69-80.

Ahmad, F., M. Yussof and T. Sembok. (1996). Experiments with a stemming algorithms for Malay words. *Journal of the American Society for Information Science*, 47 (12): 7-15.

al-Anzi, K. and M. Collier. (1994). Arabisation of library and information systems. *Program*, 28 (4): 395-403.

Aissing, A. (1995). Cyrillic transliteration and its users. *College & Research Libraries*, 56 (3): 207-219.

Aliprand, J. (Spring 1989-Winter 1990). Hebrew on RLIN--an update. *Judaica Librarianship,* 12-20.

Aliprand, J. (1992). Arabic script on RLIN. *Library Hi Tech*, 10 (4): 59-80.

Aman, M. (1984). Use of Arabic in computerized information interchange. *Journal of the American Society for Information Science*, 35 (4): 204-210.

Bachir, I. and A. Buxton. (1993). The use of topic sentences for evaluating the representativeness of Arabic article titles. *Journal of Information Science*, 19 (6): 455-65.

Barraclough, E. (1981). Opportunities for testing with online systems. In Sparck Jones, K. (ed.), *Information retrieval experiment*. Boston: Butterworths, 128-135.

Bauer, L. (1983). *English word-formation*. Cambridge: Cambridge University Press.

Bawden, D. (1990). *User-oriented evaluation of information systems and services*. Aldershot, England: Gower.

Beesley, K. (1996). Arabic finite-state morphological analysis and generation. In *COLING-96 Proceedings*, Volume 1. Copenhagen: Center for Sprogteknologi, The 16th International Conference on Computational Linguistics, 89-94.

Beesley, K. (2000). Romanization, transcription and transliteration. [http://www.rxrc.xerox.com/research/mltt/arabic/info/romanization.html]

Beeston, A. (1970). *The Arabic language today*. Hutchinson & Co. Ltd.

Blair, D. (1990). *Language and representation in information retrieval*. New York: Elsevier Science Publishers.

Borko, H. (1962). *Evaluating the effectiveness of information retrieval systems*. Santa Monica: System development Corp.

Bryant, E. (1968). *Procedural guide for the evaluation of document retrieval systems*. Bethesda, Md.: Westat Research, Inc.

Buckwalter, T. (2000). Arabic Transliteration/Encoding Chart. 2000. http://www.xrce.xerox.com/research/mltt/arabic/info/translit-chart.html

Bush, V. (1949). As we may think. *Atlantic Monthly*, 176 (7): 101-108.

Carbonell, J. et al. (1997). Translingual information retrieval: A comparative evaluation. In: *Proceedings of the 15th International Joint Conference on Artificial Intelligence; 1997 August 23-29; Nagoya, Japan*. San Francisco, CA: Morgan Kaufmann, 708-715.

Chaudhry, A. and M. Ashoor. (1990). Potential of DOBIS/LIBIS and MINISIS for automating library functions: a comparative study. *Program*, 24 (2): 109-128.

Chejne, A. (1969). *The Arabic language: its role in history*. Minneapolis: University of Minnesota Press.

Cleverdon, C. (1964). *Evaluation of operational information retrieval systems. Part 1: Identification of criteria.* Cranfield, UK: Aslib Cranfield Research Project, College of Aeronautics.

Cleverdon, C. et al. (1966). *Factors determining the performance of indexing systems*. Cranfield, UK: Aslib Cranfield Research Project, College of Aeronautics. 2 volumes. (Volume 1: Design, Volume 2: results).

Cohen, D. (1970). *Études de linguistique sémitique et arabe*. The Hague: Mouton.

Cowan, D. (1958). *An introduction to modern literary Arabic.* Cambridge: Cambridge University Press.

Crystal, D. (1985). *A dictionary of linguistics and phonetics* (2nd edition updated and enlarged). Oxford: Blackwell/André Deutsch.

De Guzman, V. and W. O'Grady. (1987). Morphology: the study of word structure. In O'Grady, W. and M. Dobrovolsky (eds.) *Contemporary linguistic analysis*. Toronto: Copp Clark Pitman Ltd, 127-155.

De Young, T. (2000). Arabic: A general introduction. [http://www.arabicstudies.edu/arabiclangrev.html]

Ekmekcioglu, F. and P. Willett. (2000). Effectiveness of stemming for Turkish text retrieval. *Program*, 34 (2): 195-200.

al-Fedaghi, S. and H. al-Sadoun. (1990). Morphological compression of Arabic text. *Information Processing & Management*, 26 (2): 303-316.

Funredes. (2001). The fifth study of languages and the Internet: 3. Overview of the study and its results. [http://www.funredes.org/LC/english/L5/L5overview.html]

Ghani, A. (1987). Arabic literature: Uniterm indexing system for storage and retrieval. *International Library Review*, 19 (4): 321-333.

Gibb, H. (1963). *Arabic Literature: an introduction.* Oxford: At the Clarendon Press.

Grefenstette, G., (ed.). (1998). *Cross-language information retrieval.* Boston: Kluwer.

Hannon, H. (1992). *Discovering RLIN: An introduction to the Research Libraries Information Network database*. Mountain View, CA: The Research Libraries Group, Inc.

Harman, D. (1987). A failure analysis on the limitation of suffixing in online environments. In *Proceedings of the 10th Annual International ACM SIGIR Conference*, New York: Association of Computer Machinery, 102-108.

Harman, D. (1991). How effective is suffixing? *Journal of the American Society for Information Science*, 42 (1): 7-15.

Harman, D., (ed.). (1996). *The 4th Text REtrieval Conference (TREC-4); 1995 November 1-3; Gaithersburg, MD*. Gaithersburg, MD: National Institute of Standards and Technology.

Harter, S. (1986). *Online information retrieval: concepts, principles, and techniques*. Orlando: Academic press, Inc.

Haywood, J. (1960). *Arabic lexicography*. Leiden: E. J. Brill.

Hedlund, T., A. Pirkola, and K. Jarvelin. (2001). Aspects of Swedish morphology and semantics from the perspective of mono- and cross-language information retrieval. *Information Processing & Management,* 37 (1): 147-161.

Hegazi, M. and A. Elsharkawi. (1985). An approach to a computerized lexical analyzer of natural Arabic. *Computer processing of the Arabic Language, Workshop Papers*, (Vol. 1). Kuwait: Kuwait Institute for Scientific Research (KISR).

Hegazi, N., N. Ali and E. Abed. (1987). Information content in textual data: Revisited for Arabic text. *Journal of the American Society for Information Science*, 38 (2): 133-137.

Hildreth, C. (1989). *Intelligent interfaces and retrieval methods for subject searching in bibliographic retrieval systems*. Washington, D.C.: Cataloging Distribution service, Library of Congress.

Hitti, P. (1963). *History of the Arabs*. 8th edition. London: Macmillan.

Hlal, Y. (1987). Information systems and Arabic: the use of Arabic in information systems. In Descout, R. (ed.) *Applied Arabic linguistics and signal & information processing*. Washington: Hemisphere Pub. Corp., 191-197.

Hmeidi, I., G. Kanaan and M. Evens. (1997). Design and implementation of automatic indexing for information retrieval with Arabic documents. *Journal of the American Society for Information Science*, 48 (10): 867-881.

Hock, R. (2000). Web search engines: (more) features and commands. *Online*, 24 (3): 17-18, 20, 22-24, 26.

Hudson, G. (1986). Arabic root and pattern morphology without tiers. *Journal of Linguistics*. 22, Mar.: 85-122.

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

Hull, D. and G. Grefenstette. (1996). Querying across languages: A dictionary-based approach to multilingual information retrieval. In: *SIGIR '96: Proceedings of the Association for Computing Machinery Special Interest Group on Information retrieval (ACM/SIGIR) 19th Annual International Conference on Research and Development in Information Retrieval; 1996 August 18-22; Zurich, Switzerland.* New York, NY: ACM, 792-798.

Inktomi. (2000). Inktomi Webmap. [http://www.inktomi.com/webmap/]

Janes, J. (1994). Other people's judgments: A comparison of users' and others' judgments of document relevance, topicality, and utility. *Journal of the American Society for Information Science*, 45 (3): 160-171.

Jackson, D. and K. Sparck Jones. (1970). The use of automatically-obtained keyword classifications for information retrieval. *Information storage and retrieval*, 5 (4): 175-201.

Jones, G. et al. (1999). A comparison of query translation methods for English-Japanese cross-language information retrieval. *Proceedings of SIGIR: International Conference on R&D in Information Retrieval.* 22: 269-270.

Kalamboukis, T. (1995). Suffix stripping for Modern Greek. *Program*, 29 (3): 313-321.

Kantor, P. (1994). Information retrieval techniques. In Williams. M. (ed.), *Annual Review of Information Science and Technology*, Washington, D.C.: American Society for Information Science, 29: 53-90.

al-Kharashi, I. (1991). *Micro-Airs: Microcomputer based Arabic information retrieval system, comparing words, stem, and roots as index terms.* Unpublished doctoral dissertation, Computer Science department, Illinois Institute of Technology, Chicago.

al-Kharashi, I. and M. Evens. (1994). Comparing words, stems, and roots as index terms in an Arabic information retrieval system. *Journal of the American Society for Information Science*, 45 (8): 548-560.

al-Kharashi, I. (2000). A Web search engine for indexing, searching and publishing Arabic bibliographic databases. [http://www.isoc.org/inet99/posters/085/]

Khurshid, Z. (1992). Arabic online catalog. *Information Technology and Libraries*, 11 (3): 244-51.

Kiewitt, E. (1979). *Evaluating information retrieval systems: The PROBE program.* London: Greenwood Press.

King, D. (1971). *The evaluation of information services and products.* Washington, D.C.: Information Resources Press.

Kowalski, G. (1997). *Information retrieval systems: Theory and implementation.* Boston: Kluwer Academic Publishers.

Lancaster, F. (1968a). *Evaluation of the MEDLARS Demand Search Service.* Washington, D.C.: National Library of Medicine.

Lancaster, F. (1968b). *Information retrieval systems: characteristics, testing, and evaluation.* New York: Wiley.

Lancaster, F. (1979). *Information retrieval systems: characteristics, testing, and evaluation*, 2nd ed. New York: John Wiley & Sons.

Lancaster, F. (1981). Evaluation within the environment of an operating information service. In Sparck Jones, K. (ed.), *Information retrieval experiment.* Boston: Butterworths, 9-31.

Lancaster, F. and A. Warner. (1993). *Information retrieval today.* Arlington, VA: Information Resources Press.

Lancaster, F. and E. Fayen. (1973). *Information retrieval on-line.* Los Angeles: Melville Publishing Company.

Large, A. and H. Moukdad. (2000). Multilingual access to Web resources: an overview. *Program*, 34 (1): 43-58.

Lawrence, S. and L. Giles. (1999). Accessibility of information on the Web. *Nature*, 400 (6740): 107-110.

Lazinger, S. and J. Levi. (1996). Multiple non-roman scripts in ALEPH-Israel's research Library Network. *Library Hi Tech*, 14 (1): 111-116.

Lovins, J. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11: 22-31.

Mace, J. (1998). *Arabic grammar: a revision guide.* Edinburgh: Edinburgh University Press.

Maron, M. (1977). On indexing, retrieval and the meaning of about. *Journal of the American Society for Information Science*, 28 (1): 38-43.

Matthews, P. (1974). *Morphology*. London: Cambridge University Press.

Meadow, C. (1992). *Text information retrieval systems*. Toronto: Academic Press, Inc.

Mehdi, S. (1986). Arabic language parser. *International Journal of Man-Machine Studies*, 25: 593-611.

Moukdad, H. (1999). An investigation of the necessity of information retrieval algorithms for full-text Arabic databases. *Information Science: Where has it Been, Where is it Going? Proceedings of the 27th Annual Conference of the Canadian Association for Information Science, Université de Sherbrooke, June 1999*. [Toronto]: CAIS, 207-227.

Moutaouakil, A. (1987). Lexical derivation in Arabic: roots and patterns. In Descout, R. (ed.) *Applied Arabic linguistics and signal & information processing*. Washington: Hemisphere Pub. Corp., 93-97.

Murtonen, A. (1964). *Broken plurals: origin and development of the system*. Leiden: E. J. Brill

Musa, F. (1986). A system for processing bilingual Arabic/English text. *Journal of the American Society for Information Science*, 37 (5): 288-293.

Nie, J. et al. (1999). Cross -language information retrieval based on parallel texts and automatic mining of parallel texts from the Web. *Proceedings of SIGIR: International Conference on R&D in Information Retrieval*. 22: 74-81.

Oddy, R. (1981). Laboratory tests: automatic systems. In Sparck Jones, K. ed. *Information retrieval experiment*. Boston: Butterworths, 156-78.

Park, T. (1994). Toward a theory of user-based relevance: A call for a new paradigm of inquiry. *Journal of the American Society for Information Science*, 45 (3): 135-141.

Pasterczyk, C. (1985). Russian transliteration variations for searchers. *Database*, 8 (1): 68-75.

Pevzner, B. (1969). Automatic translation of English text to the language of the Pusto-Nepusto-2 system. *Automatic Documentation and Mathematical Linguistics*, 3 (4): 40-48.

Pigur, V. (1979). Multilanguage information-retrieval systems: Integration levels and language support. *Automatic Documentation and Mathematical Linguistics*, 13 (1): 36-46.

Popoviç, M. and P. Willett. (1990). Processing of documents and queries in a Slovene language free text retrieval system. *Literary and Linguistic Computing*, 5: 182-190.

Popoviç, M. and P. Willett. (1992). The effectiveness of stemming for natural-language access to Slovene textual data. *Journal of the American Society for Information Science*, 43: 384-390.

Porter, M. (1980). An algorithm for suffix stripping. *Program*, 14, 130-137.

Rafea, A. and F. Shaalan. (1993). Lexical analysis of inflected Arabic words using exhaustive search of an augmented transition network. *Software-Practice and Experience*, 23 (6): 567-588.

Reig, D. (1983). *La conjugaison arabe*. G. P. Maisonneuve et Larose.

Robertson, S. (1981). The methodology of information retrieval experiment. In K. Sparck Jones, (ed.), *Information retrieval experiment*. Boston: Butterworths, 9-31.

Robertson, S. and N. Belkin (1978). Ranking in principle. *Journal of Documentation*, 34: 93-100.

Ruiz, M. and P. Srinivasan. (1998). Cross-language information retrieval: an analysis of errors. *Proceedings of the ASIS Annual Meeting*, 35, 153-165.

Saffady, W. (1989). *Text storage and retrieval systems: A technology survey and product directory*. London: Meckler.

Sakai, Y., Y. Terashita and K. Takenmoto. (1986). An experimental system for creating and managing Arabic Bibliographic Database--a step toward effective international information exchange. *Libri*, 36 (4): 259-275.

Salton, G. (1962-70). *Information storage and retrieval*. ISR Report nos. 1-17. Ithaca: Cornell University.

Salton, G. (1970). Automatic processing of foreign language documents. *Journal of the American Society for Information Science*, 21 (3): 187-194.

Salton, G., (ed.). (1971). *The SMART retrieval system experiments in automatic document processing*. Englewood Cliffs, NJ: Prentice Hall.

Saracevic, T. (1995). Evaluation of evaluation in information retrieval. *Proceeding of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 138-145.

Savoy, J. and J. Picard. (2001). Retrieval effectiveness on the Web. *Information Processing & Management*, 37: 543-569.

Schwartz, C. (1998). Web search engines. *Journal of the American Society for Information Science*, 49 (11): 973-982.

Shaw, R. (1949). The rapid selector. *Journal of Documentation*, 5: 164-171.

Shi, Y. and R. Larson. (1989). Facilitating Chinese character entry and information retrieval through regular expression searching. *Library and Information Science Research*, 11: 335-355.

Soergel, D. (1994). Indexing and retrieval performance: The logical evidence. *Journal of the American Society for Information Science*, 44 (6): 589-599.

Sparck Jones, K., (ed.). (1981). *Information retrieval experiment*. Boston: Butterworths.

Sparck Jones, K. and M. Kay. (1973). *Linguistics and information science*. New York: Academic Press.

Sperer, R. and D. Oard. (2000). Structured translation for cross-language information retrieval. In: *SIGIR 2000, Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 120-127.

Stetkevych, J. (1970). *The modern Arabic literary language: lexical and stylistic development*. Chicago: University of Chicago Press.

Su, L. (1994). The relevance of recall and precision in user evaluation. *Journal of the American Society for Information Science*, 45 (3): 207-217.

Swanson, D. (1988). Historical note: Information retrieval and the future of an illusion. *Journal of the American Society for Information Science*, 39 (2): 92-98.

Tague-Sutcliffe, J. (1996). Some perspectives on the evaluation of information retrieval systems. *Journal of the American Society for Information Science*, 47 (1): 1-3.

Taube, M. (1953). Evaluation of information systems for report utilization. *Studies in Coordinate Indexing*, 1: 96-110.

Taubes, G. (1995). Indexing the Internet. *Science*, 269: 1354-1356.

van Rijsbergen, C. (1979). *Information Retrieval, Second Edition*. London: Butter Worths.

van Rijsbergen, C. (1981). Retrieval effectiveness. In Sparck Jones. (ed.) *Information retrieval experiment*. Boston: Butterworths, 32-43.

Vernon, E. (1991). Hebrew and Arabic scripts materials in the automated library: The United States scene. *Cataloging and Classification Quarterly*, 14 (1): 49-67.

Versteegh, K. (1997). *The Arabic language*. Edinburgh: Edinburgh University Press.

Wellisch, H. (1975). *Transcription and transliteration: an annotated bibliography on conversions of scripts*. Silver Spring, MD: Institute of Modern Language.

Wightwick, J. and M. Gaafar. (1998). *Arabic verbs and essentials of grammar: a practical guide to the mastery of Arabic*. Lincolnwood, Ill.: Passport Books.

Wood, F., N. Ford and C. Walsh. (1994). The effect of postings information on searching behaviour. *Journal of Information Science*, 20 (1): 29-40.

Wu, Z. and G. Tseng. (1995). ACTS: An automatic Chinese text segmentation system for full text retrieval. *Journal of the American Society for Information Science*, 46 (2): 83-96.

Yahya, A. (1989). On the complexity of the initial stage of Arabic text processing. Paper presented at the *First Great Lakes Computer Conference*, Kalamazoo, MI.

Young, D. (1984). *Introducing English grammar*. London: Hutchinson.

Zheng, Y. and C. He. (1986). A new Chinese character coding system for computers. In *Proceedings, National Online Meeting, 1986*. Medford, NJ: Learned Information, Inc.

Ziadeh, F. and R. Winder. (1957). *An introduction to modern Arabic*. Princeton: Princeton University Press.