A Computational Feature-Based Morphological Analysis and Generation of Modern Standard Arabic

Mourad Gridach, Noureddine Chenfour
Computer Science Department
Sidi Mohamed Ben Abdellah University Faculty of Sciences
Fez, Morocco
{mourad_i4, chenfour}@yahoo.fr

ABSTRACT: *Arabic is a strongly structured and highly derivational and inflectional language. Arabic morphology and syntax provide the ability to add a large number of affixes to each word which makes combinatorial increment of possible words. Arabic morphological analysis has gained the focus of Arabic natural language processing research for a long time in order to achieve the automated understanding of Arabic. In this paper, we present a new approach for Arabic morphological analysis and generation based on morphological automata and using our lexicon. Examples of morphological analysis results from the proposed approach will be given and will illustrate how the system works. Our system will be very useful for NLP applications such as syntactic and semantic analysis, spell-checking, machine translation and information retrieval.*

Keywords: Morphological Automata, Feature, Arabic, Morphology, Analysis, Generation

Received: 12 December 2010, Revised 14 January 2011, Accepted 19 January 2011

1. Introduction

Arabic is the fourth most widely spoken language in the world (Nwesri et al., 2005). It is a morphologically rich language in which a single inflected word may correspond to a full sentence. Arabic morphological analysis has gained the focus of Arabic natural language processing research for a long time in order to achieve the automated understanding of Arabic. On the other hand, the Arabic morphological analysis is an important tool in all areas of scientific research and industry that require knowledge of the internal structure of the Arabic words.

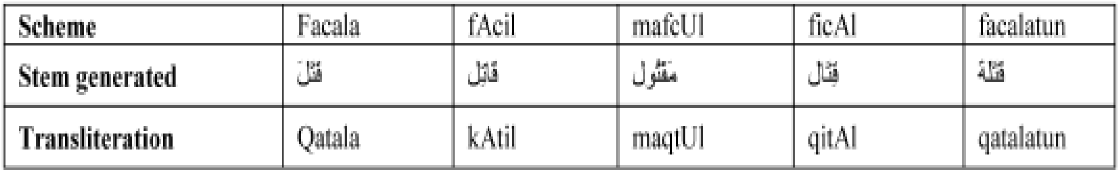
Nowadays, Arabic language faces many challenges. The first important challenge is the requirement to analyze Arabic morphology with high quality because it is considered as the essential stage in many NLP applications such as Information Retrieval and Machine Translation. The second challenge is concerning the use of morphology in machine translation systems. Koehn & Hoang (2007) have shown that factored translation models containing morphological information lead to better translation performance. Morphological analysis becomes more important when translating to or from morphologically rich languages such as Arabic. The third challenge is that morphological analysis is considered as the first step before syntactic analysis.

Recently, research work and development in natural language processing (NLP) systems has shown that morphological analysis of any word consists of determining the values of a large number of features, such as basic part-of-speech (i.e., noun, verb, etc.), gender, person, number, voice, information about the clitics, etc. (Habash, 2005). There has been much work on Arabic morphology (see Al-Sughaiyer & Al-Kharashi, 2004). Since, lots of morphological analysis approaches are available now, some of them have a commercial purpose and the others are available for research and evaluation (Attia, 2006).

Morphological analysis of Arabic sentences is a difficult task. The difficulty comes from several sources. One is that sentences are long and complex. The average length of a sentence is 20 to 30 words, and it often exceeds 100 words. Another difficultycomes from the word structure. The Arabic word is complex and morphologically ambiguous due to the frequent usage of a number of affixes. The agglutination phenomenon is another problem for analyzing Arabic texts. The main goal was to implement a computer system to analyze Arabic texts.

In this paper we present an approach for Arabic morphological analysis based on Arabic morphological automaton technique. To construct an Arabic morphological automaton, we used particularities of Arabic morphology that are concretized on multilevel: verbs and nouns are also characterized by a specific representation named the matrix "root – scheme". Arabic nouns and verbs are derived from roots by applying schemes to these roots to generate Arabic stems and then adding prefixes and suffixes to the stems to form a correct word in Arabic language. Table 1 show some schemes applied to the root "qtl" (قتل) . Our approach is used to develop a morphological analyzer for Arabic language.

The structure of the article is as follows. First, in the introduction we discuss the challenges of Arabic language and the importance of morphological analysis as an essential step in Natural Language Processing. We present the Arabic morphological system in the second section. Then, we discuss some challenges of the computational morphology of Arabic. We discuss some Arabic morphological analysis approaches related to the presented work in the fourth section. In the fifth section, we present our lexicon. In section six, we present our approach for Arabic morphological analysis. We present the application of the presented work in Arabic morphological analysis in section seven. In the eighth section, we evaluate the proposed technique. In section nine, we discuss the obtained results. Finally, in the last section, we draw some conclusions.



Scheme	Facala	fAcil	mafcUl	ficAl	facalatun
Stem generated	قتل	قاتل	مقتول	قتال	قتلة
Transliteration	Qatala	kAtil	maqtUl	qitAl	qatalatun

Table 1. Some examples of schemes to generate stems from the root "ktb"(كتب)

2. The Arabic morphological system

The morphological system in Arabic plays a major role in word formation. This in turn will affect the retrieval performance in Arabic. Therefore, a deep discussion of Arabic morphology is essential. This paragraph deals with the morphological system in Arabic, known as derivation and inflection systems, each of which has systematic rules for word formation. The present paragraph discusses issues most related to this study.

Arabic language is one of the Semitic languages defined as a diacritized language where the pronunciation of its words cannot be fully determined by their spelling characters only. It depends also on some special marks put above or below the spelling characters to determine the correct pronunciation; these marks are called diacritics. In addition, Arabic words are homographic: they have the same orthographic form, though the pronunciation and meaning is totally different (Ahmed 2000; Attia 2006).

On the other hand, Arabic is rich in derivational morphology. It is considered a non-concatenative language as it alters the stem of words according to the syntactic context. Arabic words are morphologically divided into three types: noun, verb, and particle. They are derived by applying a pattern to the root to generate a stem and then inflect into prefixes and suffixes (Diab et al., 2004). Consequently, the derivation process has made the Arabic language the richest in vocabulary ever found among all important natural languages (Ahmed 2000) although it has a relatively small number of derivative patterns. However, one of the most puzzling problems in the study of Arabic is its verbal system which is very rich in forms and meaning (Soudi et al., 2001).

The conjugation of verbs in different tenses, voices and mood is achieved using well behaved morphological rules. The irregularities are due to the phonological constraints of certain root consonants. The important irregularity issues are related to Arabic weak verbs that include one or more weak letter. Weak letters can be deleted or substituted by other letters because of Arabic phonological constraints (El-Sadany and Hashish 1989). For example, the replacement of the letter (و) /w/ by (ا) /ealif/ in taking the past (perfect) tense of the trilateral root قول /q-w-l/, using regular rules would generate قول /qawala/but as it is a hollow (middle weak) verb it should be generated according to special weak rules and thus it appears in written texts as قال /qAla/ (said).

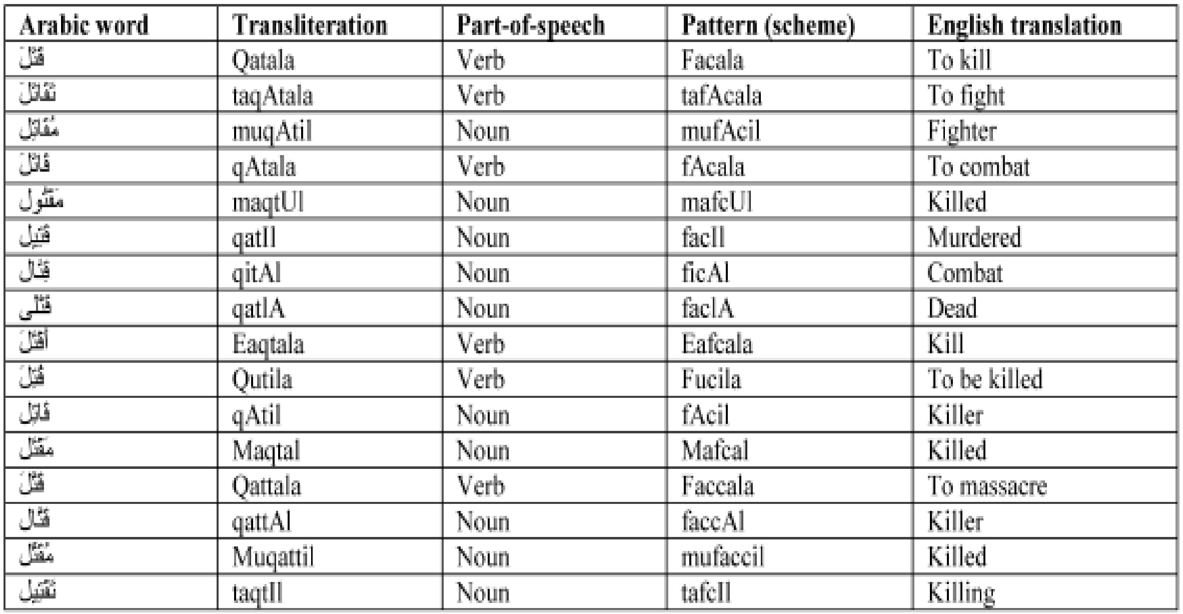
It is important here to differentiate between derivational and inflectional morphology in Arabic. The function of the inflection is to alter the form of the word in number, gender, mood, tense, aspect, person, and case (Klavans and Tzoukermann, 1992) while derivation may change the grammatical category of a word. There are a number of specific differences between derivation and inflection which may be summarized as follows:

- Inflection deals with syntactically determined affixation processes while derivational morphology is used to create new lexical items;
- Inflection is regular, while derivation is not;
- Inflection affixes are placed on the terminal point of the word, while the derivation affects the structure of the word;
- Inflectional morphology has a strong relationship with syntax.

3. Challenges

Arabic is a Semitic language which is different compared to the other languages (Indo-European languages). The difference occurs in many levels: morphology, syntax and the semantics. Its writing system contains twenty five consonants. Concerning the vowels, the writing system contains two types of vowels: short vowels and long vowels. For the short vowels, they are not a part of the alphabet; they are written as diacritics above or under a consonant (see Appendix 3). The writing of Arabic is made from right to left.

On the other hand, Arabic is a highly inflectional and derivational language and its vocabulary can be easily expanded using a framework that is latent in the creative use of roots and patterns. It should be noted that 85% of words derived from trilateral roots and there are around 10.000 independent roots (Al-Fedaghi and Al-Anzi (1989)). As a result, Arabic words are developed by applying different patterns (schemes) to roots. The root is an ordered sequence of valid three or four characters from the alphabet and rarely five characters. The root is not a valid Arabic word (for example /qtl/ /قتل /). The scheme is an ordered sequence of characters. Some of these characters are constants and some are variables. The variables characters are to be substituted with the characters of an Arabic root to generate a word called the “stem”. There are different schemes for the triliteral and tetraliteral roots. Note that the scheme is not a valid Arabic word, whereas the stem is a valid word. Table 2 shows an example of some Arabic words derived from the root ‘qtl’ (the notion of kill) by applying patterns.



Arabic word	Transliteration	Part-of-speech	Pattern (scheme)	English translation
قَتَلَ	Qatala	Verb	Facala	To kill
تَقَاتَلَ	taqAtala	Verb	tafAcala	To fight
مُقَاتِل	muqAtil	Noun	mufAcil	Fighter
قَاتَلَ	qAtala	Verb	fAcala	To combat
مَقْتُول	maqtUl	Noun	mafcUl	Killed
قَتِيل	qatIl	Noun	facIl	Murdered
قِتَال	qitAl	Noun	ficAl	Combat
قَتْلَى	qatlA	Noun	faclA	Dead
أَقْتَلَ	Eaqtala	Verb	Eafcala	Kill
قُتِلَ	Qutila	Verb	Fucila	To be killed
قَاتِل	qAtil	Noun	fAcil	Killer
مَقْتَل	Maqtal	Noun	Mafcal	Killed
قَتَّلَ	Qattala	Verb	Faccala	To massacre
قَتَّال	qattAl	Noun	faccAl	Killer
مُقَتِّل	Muqattil	Noun	mufaccil	Killed
تَقْتِيل	taqtIl	Noun	tafcIl	Killing

Table 2. Derivatives of root (qtl) /'ق ت ل/

Another point that characterizes Arabic language is the possibility to loan words from other languages to its system of derivational morphology to make them sound and behave like Arabic words as, for example, in the case of “aksadah” (oxidation), which is patterned on “faclalah” (Elkateb, 2005).

4. Related work

Much work has been done on Arabic computational morphology (Al-Sughaiyer and Al-Kharashi, 2004). Almost all the Arabic morphological systems focused on extracting roots or stems. Morphological systems are categorized as statistical driven methods (Al-Sahmsi and Guessoum, 2006; Mohamed et al., 2009; Ahmed and Nürnberger, 2007; Sinane et al., 2008), machine translation driven methods (Chen and Gey, 2002) and rule based methods (El-Hajar et al., 2010; Larkey et al., 2002, 2005; Buckwalter, 2002; Al-Ameed et al., 2005; Khoja and Garside,1999; Darwish, 2002).

In this paragraph, we will present the most refrenced systems. We will present works done by Habash et al., Smrz, Buckwalter and Beesley. They are available for research and evaluation and well documented. Firstly, we focus on MAGEAD. It’s a functional morphology systems compared to Buckwalter morphological analyzer which models form-based morphology (M. Altantawy et al., 2010). They use a morphemic representation for all morphemes and explicitly define morphophonemic and orthographic rules to derive the allomorphs. The lexicon is developed by extending Elixir-FM’s lexicon. The advantage of this analyzer is that it processes words from the morphology of the dialects which they considered as a novel work in this domain, but unfortunately this analyzer needs a complete lexicon for the dialects to make the evaluation more interesting and convincing, and to verify these claims.

Secondly, we present ElixirFM by Otakar Smrz. It is an online Arabic Morphological Analyzer for Modern Written Arabic developed available for evaluation and well documented. This morphological analyzer is written in Haskell, while the interfaces in Perl. ElixirFM is inspired by the methodology of Functional Morphology (Forsberg & Ranta, 2004) and initially relied on the re-processed Buckwalter lexicon (Buckwalter, 2002). It contains two main components: a multi- purpose programming library and a linguistically morphological lexicon (Smrz, 2007). The advantage of this analyzer is that it gives to the user four different modes of operation (Resolve, Inflect, Derive and Lookup) for analyzing an Arabic word or text. But the system is limited coverage because it analyzes only words in the Modern Written Arabic.

Thirdly, BAMA is considered as one of the most referenced in the literature, well documented and available for evaluation. It is also used by Linguistic Data Consortium (LDC) for POS tagging of Arabic texts, Penn Arabic Treebank, and the Prague Arabic Dependency Treebank (Atwell et al., 2004). It takes the stem as the base form and root information is provided. This analyzer contains over 77800 stem entries which represent 45000 lexical items. However, the number of lexical items and stems makes the lexicon voluminous and as result the process of analyzing an Arabic text becomes long.

Finally, Xerox Arabic morphological Analyzer is another well known Arabic morphological analyzer available for evaluation and well documented. This analyzer is constructed using Finite State Technology (FST) (Beesley, 1996; Beesley, 2000). It adopts the root and pattern approach. Besides this, it includes 4930 roots and 400 patterns, effectively generating 90000 stems. The advantages of this analyzer are, on the one hand, the ability of a large coverage. On the other hand, it is based on rules and also provides an English glossary for each word. But the system fails because of some problems such as the overgeneration in word derivation, production of words that do not exist in the traditional Arabic dictionaries (Darwish, 2002) and we can consider the volume of the lexicon as another disadvantage of this analyzer which could affect the analysis process.

5. Lexicon

The lexicon of a language is the set of its valid lexical forms. As in any morphological analysis system, developing a high quality lexicon is often the first step towards building a robust morphological analyzer, which is in turn the front-end to many NLP systems. There are two aspects that contribute to this enhancement level. The first aspect concerns the number of lexicon entries contained in the lexicon. Second aspect concerns the richness in linguistics information contained by the lexicon entries.

Several Arabic lexicons are available now. The Buckwalter Arabic Morphological Analyzer (BAMA) is one of the best Arabic morphological analyzers and is available as open source. The BAMA uses a concatenative lexicon-driven approach where morphotactics and orthographic adjustment rules are partially applied into the lexicon itself instead of being specified in terms of general rules that interact to realize the output (Buckwalter, 2002). It used by large Arabic morphological analyzers (Elixir-FM by Otakar Smrz and MAGEAD by Nizar Habash).For an overview of the existing Arabic lexicon see (Al-Sughaiyer and Al-Kharashi, 2004).

Nowadays, a new method was been implemented to represent, design and implement the lexicons. It is based on the Lexical Markup Framework (LMF). LMF is the ISO-24613 standard for natural language processing (NLP) and lexicons. The US delegation is the first which started the work on LMF in 2003. In early 2004, the ISO/TC37 committee decided to form a common ISO project with Nicoletta Calzolari (Italy) as convenor and Gil Francopoulo (France) and Monte George (US) as editors. The aims of LMF are to provide a common model for the creation and use of lexical resources, to manage the exchange of data between and among these resources, and to enable the merging of large number of individual electronic resources to form extensive global electronic resources. This method for representing lexical resource covers all the natural languages. We note that for Arabic language, lexicons based on LMF are still in progress towards a standard for representing the Arabic linguistic resource.

Our approach for representing the lexicon is based on XMODEL (XML-based Morphological Definition Language). In this approach, the Arabic lexicon contains morphological classes, morphological properties and morphological rules. Morphological classes allow gathering a set of morphological components having the same nature, the same morphological characteristics and the same semantic actions. For the morphological properties, they allow characterizing the different morphological components represented by the morphological classes; they contain morphological descriptors (the features) that would be assigned to different morphological components (the property “Gender” distinguishes between masculine and feminine components). Finally, morphological rules allow combining the morphological components to generate correct language words. They are considered as a generator of language words. The use of XMODEL allows representing the morphological database independent of processing which will be applied and allows a considerable reduction of morphological entries.

The lexicon entry is implemented as an XMODEL file. Every morphological entry (verb, noun or particle) is described by a set of morphological descriptors or morpho-syntactic features. The next three paragraphs show some possible morpho-syntactic features of nous and verbs.

A. Nouns: A noun has the following features:

An Arabic noun will be described by: Type, Gender, Number, Case and State. Where:

- Type: Particular | Derived
- Gender: Feminine | Masculine
- Number: Singular | Dual | Plural
- Case: ManSUb | MarfUc | MajrUr | MajzUm
- State: Definite | Indefinite

B. Verbs: A verb has the following form:

An Arabic-verb will be described by: Gender, Number, Tense, Voice, Person and Case. Where:

- Gender: Feminine | Masculine
- Number: Singular | Dual | Plural
- Tense: MADI (الماضي) | MuDAric (المضارع) | eamr (الأمر)
- Voice: Active | Passive
- Person: First person | Second Person | Third person
- Case: ManSUb | MarfUc | MajrUr | MajzUm

It should be noted that there are more morpho-syntactic features that characterize verbs, nouns and particles.

6. The proposed method

There has been much work on Arabic morphological analysis where lots of approaches are implemented to satisfy that area of research. For an overview of the approaches of Arabic morphological analysis, see (Al-Sughaiyer & Al-Kharashi, 2004).

The proposed method is based on Arabic Morphological Automata. It is considered among the most efficient methods. The Arabic morphological automata is responsible for both analysis and generation tasks. A word is accepted by morphological automata if it belongs to a correct word in Arabic. Consequently, implementing morphological automata needs to use the lexicon discussed in the previous section. We have to extract all the morphological rules from the lexicon and implement morphological automata for each rule. So to realize that implementing, we have to use some operations such as concatenation and union. In the next paragraphs, we explain how we can use these two operations to generate morphological automata for a definite morphological rule. The following morphological rule is responsible to product Arabic numbers that accept the suffix "an" (tanwIn).

```
<package name = "RulesPackage">
<rules_class name="cardNbCRules">
          <rule id="2">
                    <morpheme key = "CardNumber.CNAccepteSCID"/>
                    <morpheme key = "CasSuffixe.SCID" component = "an"/>

                    <idp name = "CNIndefManSUb"/>
          </rule>
          ...
</rules_class>
</package>
```

So to generate the morphological automata representing this morphological rule, we concatenate the first morpheme (key = "*CardNumber.CNAccepteSCID*"), which represents Arabic numbers that accept suffixes (like "wAHid" /واحد/, "ca^arat"/) عشرة /, "~amAn" / ثمان /, etc.), with the second one (key = "*CasSuffixe.SCID*" component = "*an*"), which represents the suffix "an". Figure 1 shows the resulting morphological automata obtained from this rule.

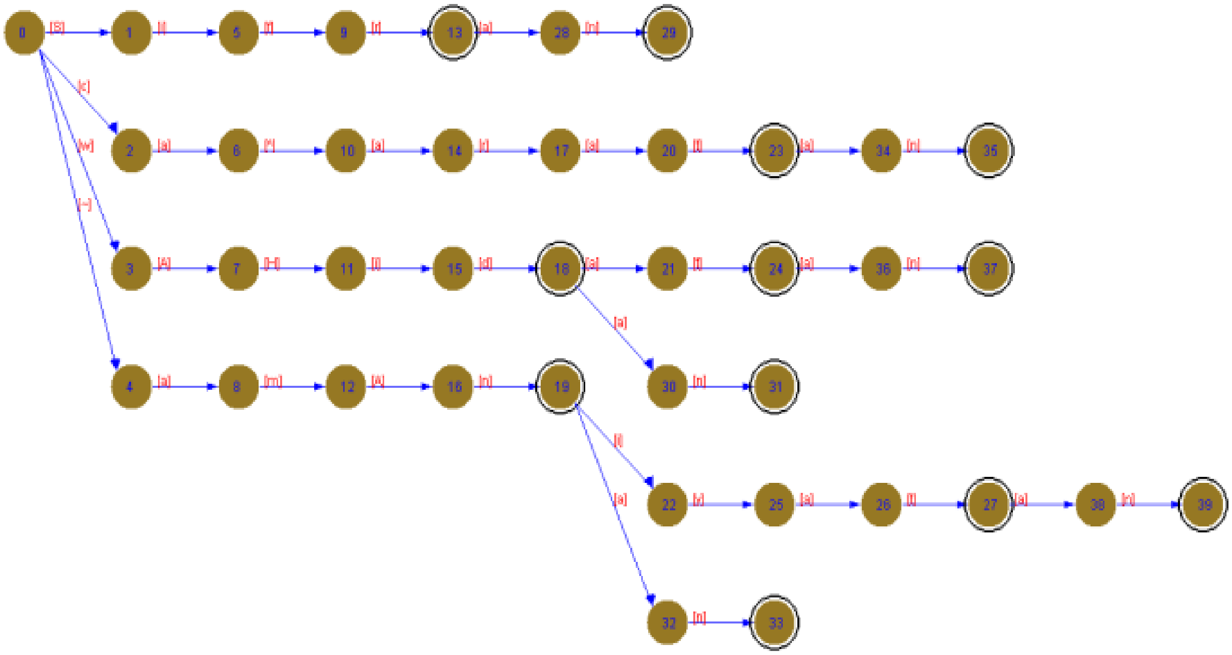


Figure 1. A morphological automaton representing the above morphological rule

In the following paragraphs, we present a detail of how to implement all the morphological automata and the technique used in the implementation.

So as to implement morphological automata, we have classified Arabic words in to two categories: the first category is that which submits to the derivation process, while the second one doesn't. This derivation process is generated by a set of morphological rules known in the Arabic grammar under the name "*qawAcidu eaSSarfi*" /قواعد الصرف/. They repose on the manipulation of a set of very determined schemes (measures) named "*ealeawzAn*" /الأوزان/.

The scheme (measure or form) is a general mould composed of an ordered sequence of characters. Some of these characters are constants (instantiated) and some are variables (uninstantiated). The uninstantiated characters are to be substituted (instantiated) with the characters of an Arabic root to generate a word called the "stem." There are different schemes for the trilateral and tetraliteral roots. Note that the scheme is not a valid Arabic word, whereas the stem is a valid word.

Finally, after generating a series of morphological automata, their size is about 130 MB. Concerning the number of the entries generated, it's about 6000 entries, which represent a remarkable reduction of the entries number and makes our approach as one of the best existing approaches in the literature. We could mention that using XMODEL to implement the lexicon could be another advantage that explains the obtained results. We note that developing the Arabic morphological automata is the proposed approach to develop an Arabic morphological analyzer. In the following paragraph we will present the application of our approach in Arabic morphological analysis.

7. Application in Arabic Morphological Analysis

In this section, we operate our approach in Arabic morphological analysis. It is based on the Arabic Morphological Automata method presented in the previous section. The implementation of our approach has been done using an oriented object framework. It is developed using Java Programming Language and based on a reduced lexicon built using XMODEL language.

The use of morphological automata technology makes our system usable as a generator as well as an analyzer, unlike some morphological analyzers which cannot be converted to generators in a straightforward manner (Sforza, 2000; Buckwalter, 2004; Habash, 2004).

So as to develop an Arabic morphological analyzer and generator, firstly, we used our lexicon. It regroups three packages: morphological components package that contains verbs, nouns, particles and affixes. The second package includes the morphological rules and the last package is concerned with the morphological properties. Secondly, we used a set of Arabic morphological automata each one represents a very specific morphological category. It is considered as the main idea to develop an Arabic morphological analyzer. Finally, we developed a framework handling the lexicon and the morphological automata.

The presented method involves five steps. In this paragraph, we provide a brief description of the principle of this method. As input, the proposed technique accepts an Arabic text. The first step is to apply a tokenization process to the text given. Then, a set of morphological automata are loaded, in a second step. The part-of-speech is determined in the third step. After that, the method determines all possible affixes. Then the next step consists of extracting the morpho-syntactic features according to the valid affixes.

The tokenization process consists of extracting all the words from the text given. A set of Arabic morphological automata are loaded from a package that contains all the implemented Arabic morphological automata. Then, the approach determines which morphological automata are suitable for that word. The result may be one or more morphological automata loaded. Then, the method determines the part-of-speech. If the word analyzed is a noun or a verb, the method determines if it contains a scheme. Then, if it is a verb, the method determines the type of the verb (strong, weak, or incomplete), its tense ("mADI"/ ماضي /, "muDAric" / مضارع / or "eamr" / أمر /), its voice (active or passive), etc. If it is a noun, we determine if it is a derived noun or particular noun. If it is a particle, the method determines if it is a preposition particle / حروف الجر /, conjunction particle / حروف العطف /, etc. After that, the method applied a process of extracting the possible affixes attached to the word analyzed. The next step consists of extracting the morpho-syntactic features according to the valid affixes and the scheme. Additional information is extracted called in our approach morphological descriptors. They describe the word analyzed and they are very table where each row contains the word analyzed and all the data characterizing this word (see figure 2).

To concretize these obtained results, we analyze some examples of Arabic words using the proposed technique. These examples are taken from a standard input text provided by ALECSO (Arab League, Educational, Cultural and Scientific Organization) which organized a competition in April 2009 of the Arabic Analyzers in Damascus. The standard input text provided by ALECSO is unvocalized, in this test, we used a vocalized version. This standard input text is provided in this file: http://www.alecso.org.tn/images/stories/ OULOUM/MOHALLILAT%20SARFIADAMAS2009/020%20 NIZAR.html.

Figure 2 shows the morphological analysis results of some Arabic words analyzed using the proposed technique. As discussed before, the analyzer displays the Part-of-speech (verb, noun or particle), the original scheme is displayed in column B because Arabic has this particularity which is summarized in that some words might be conjugated forms of other words like "*afcalu*", "*afcilu* ", "*afculu*", these three words are all conjugated forms of "*facala*". The gender (masculine or feminine) is displayed in column D, the person (first, second or third person) is displayed in column E, the number (singular, dual or plural) is displayed in column F. For the column G, it concerns some properties that characterize the word analyzed and they are very useful to the user. Some morphological descriptors are displayed in column H. Finally, the column I and J show the affixes attached to the word.

Finally, the proposed technique for Arabic morphological analysis has many advantages such as:

- The separation between the linguist and the developer task.
- We can also reuse our programs in future works.
- Development standardization means in our application that we have build all the applications with the same standards.
- The facility of maintenance: it's easy to add some new features or morphological characteristics to the presented system if the user or the linguist needs them for his Arabic morphological analysis. It's also easy to extend our system to include some new works related to Arabic NLP such as information retrieval, syntactic and semantic analyzers, correction and generation of Arabic texts.

8. Evaluation

To evaluate our system, we select two of the best known morphological analyzers in the literature: ElixirFM by Otakar Smr• (Otakar Smr• and Viktor Bielický, 2010) and Xerox Arabic Morphological Analyzer. We note that the corpus used for the evaluation is taken from a standard input text provided by ALECSO (Arab League, Educational, Cultural and Scientific Organization) which organized a competition in April 2009 of the Arabic Morphological Analyzers in Damascus.

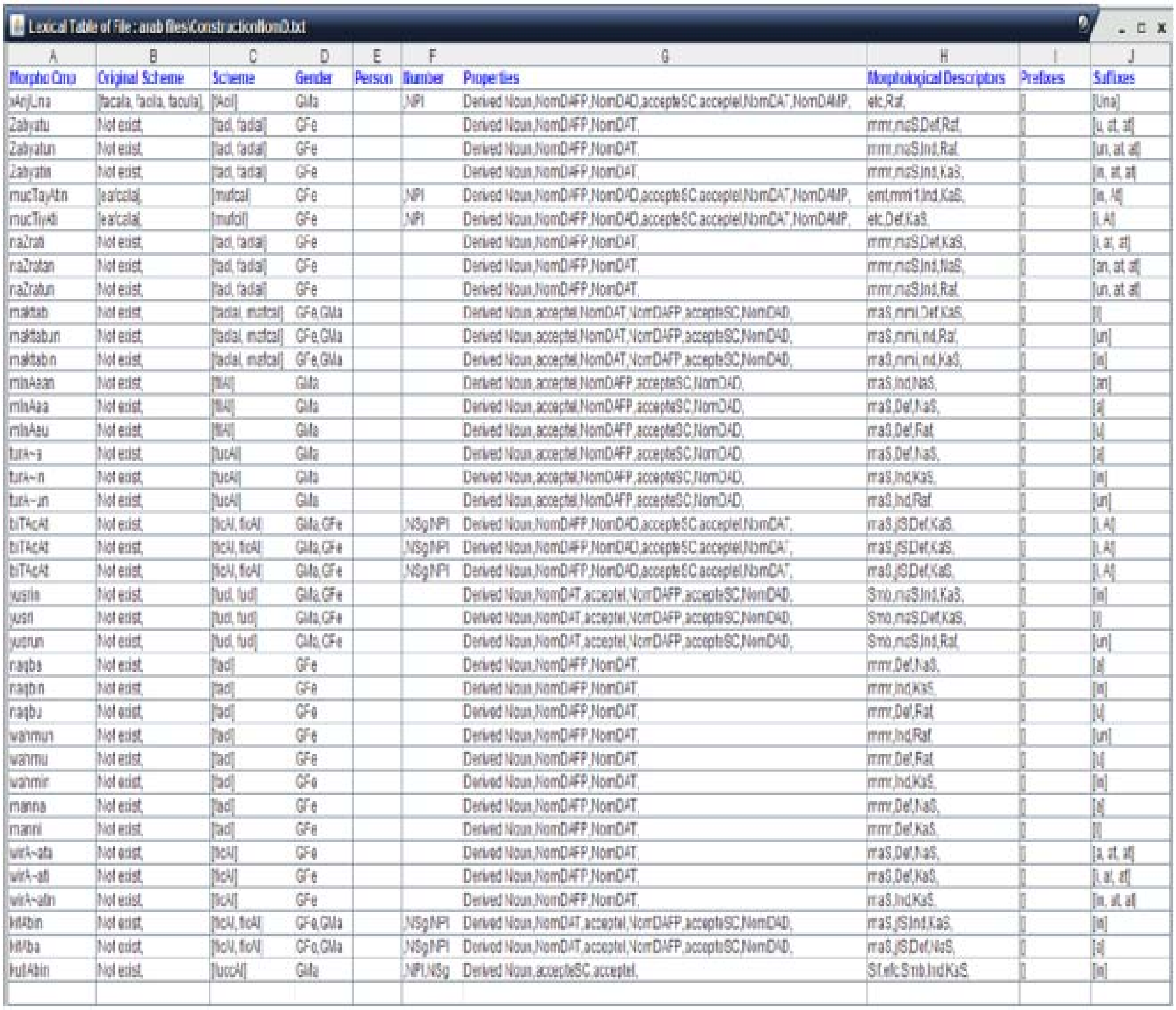
The evaluation process shows that our morphological analyzer is strong concerning the features given by each analyzer which makes our system useful for the most of NLP applications unlike the others; they are destined for specific applications. In addition, the presented morphological analyzer gives more additional information about each word analyzed and more precision.

In the evaluation done we process words in a corpus selected from ALECSO input text containing different part-of-speech (verbs, nouns and particles), then, we calculate success of each analyzer as: S = number of words with solutions / number of words. Table 1 provides the evaluation results of the three analyzers. Note that Table 1 contains in each column of the analyzers the number of words (nouns, verbs and particles) with no solution.

The analyzer presented in this paper reaches a success of 95.08% which will make it one of the best existing morphological analyzers for Arabic language and it will be very useful for the next future works to be done in NLP applications such as syntactic and semantic analysis, machine translation, information retrieval, etc.

9. Conclusion

As Arabic is a highly inflected and derivational language, there will always be possibilities for improving its natural language processing approaches and tools. In this paper, we have described an approach for Arabic morphological analysis. It is called the Arabic Morphological Automata. We have evaluated the presented approach using Xerox Arabic Morphological Analyzer and Arabic Morphological Analyzer by Otakar Smrz because they are considered as the most referenced approaches for Arabic morphological analysis and they are available for research and evaluation. The use of the Arabic morphological useful especially



Lexical Table of File : arab files\ConstructionNomD.txt

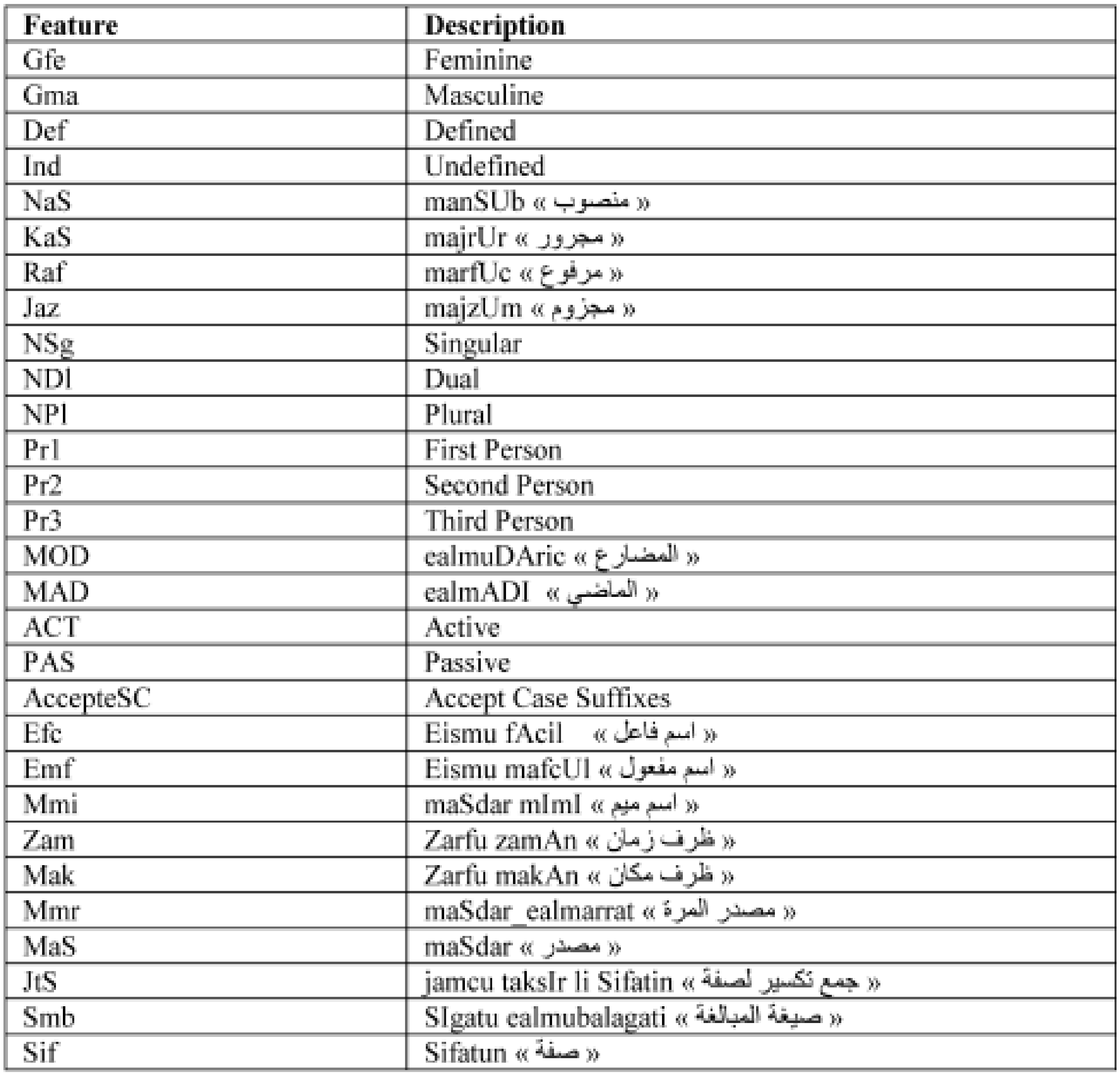
A	B	C	D	E	F	G	H	I	J
Morpho Cmp	Original Scheme	Scheme	Gender	Person	Number	Properties	Morphological Descriptors	Prefixes	Suffixes
>AnjiLna	[facala, facila, facula],	[fAcil]	GMa		,NPl	Derived Noun,NomD4FP,NomD4D,accepteSC,acceptel,NomDAT,NomD4MP,	etc,Raf,	[]	[Una]
Zabyatu	Not exist,	[fad, fadal]	GFe			Derived Noun,NomD4FP,NomDAT,	mmr,maS,Def,Raf,	[]	[u, at, at]
Zabyatun	Not exist,	[fad, fadal]	GFe			Derived Noun,NomD4FP,NomDAT,	mmr,maS,Ind,Raf,	[]	[un, at, at]
Zabyatin	Not exist,	[fad, fadal]	GFe			Derived Noun,NomD4FP,NomDAT,	mmr,maS,Ind,KaS,	[]	[in, at, at]
mucTayAtin	[eafcalal,	[mufcal]	GFe		,NPl	Derived Noun,NomD4FP,NomD4D,accepteSC,acceptel,NomDAT,NomD4MP,	emt,mmi1,Ind,KaS,	[]	[in, At]
mucTiyAti	[eafcalal,	[mufcil]	GFe		,NPl	Derived Noun,NomD4FP,NomD4D,accepteSC,acceptel,NomDAT,NomD4MP,	etc,Def,KaS,	[]	[i, At]
naZrati	Not exist,	[fad, fadal]	GFe			Derived Noun,NomD4FP,NomDAT,	mmr,maS,Def,KaS,	[]	[i, at, at]
naZratan	Not exist,	[fad, fadal]	GFe			Derived Noun,NomD4FP,NomDAT,	mmr,maS,Ind,NaS,	[]	[an, at, at]
naZratun	Not exist,	[fad, fadal]	GFe			Derived Noun,NomD4FP,NomDAT,	mmr,maS,Ind,Raf,	[]	[un, at, at]
maktab	Not exist,	[fadal, mufcal]	GFe,GMa			Derived Noun,acceptel,NomDAT,NomD4FP,accepteSC,NomD4D,	maS,mmi,Def,KaS,	[]	[i]
maktabun	Not exist,	[fadal, mufcal]	GFe,GMa			Derived Noun,acceptel,NomDAT,NomD4FP,accepteSC,NomD4D,	maS,mmi,Ind,Raf,	[]	[un]
maktabin	Not exist,	[fadal, mufcal]	GFe,GMa			Derived Noun,acceptel,NomDAT,NomD4FP,accepteSC,NomD4D,	maS,mmi,Ind,KaS,	[]	[in]
minAean	Not exist,	[fiAl]	GMa			Derived Noun,acceptel,NomD4FP,accepteSC,NomD4D,	maS,Ind,NaS,	[]	[an]
minAaa	Not exist,	[fiAl]	GMa			Derived Noun,acceptel,NomD4FP,accepteSC,NomD4D,	maS,Def,NaS,	[]	[a]
minAeu	Not exist,	[fiAl]	GMa			Derived Noun,acceptel,NomD4FP,accepteSC,NomD4D,	maS,Def,Raf,	[]	[u]
turA~a	Not exist,	[fucAl]	GMa			Derived Noun,acceptel,NomD4FP,accepteSC,NomD4D,	maS,Def,NaS,	[]	[a]
turA~in	Not exist,	[fucAl]	GMa			Derived Noun,acceptel,NomD4FP,accepteSC,NomD4D,	maS,Ind,KaS,	[]	[in]
turA~un	Not exist,	[fucAl]	GMa			Derived Noun,acceptel,NomD4FP,accepteSC,NomD4D,	maS,Ind,Raf,	[]	[un]
biTAcAt	Not exist,	[ficAl, ficAl]	GMa,GFe		,NSg,NPl	Derived Noun,NomD4FP,NomD4D,accepteSC,acceptel,NomDAT,	maS,jS,Def,KaS,	[]	[i, At]
biTAcAt	Not exist,	[ficAl, ficAl]	GMa,GFe		,NSg,NPl	Derived Noun,NomD4FP,NomD4D,accepteSC,acceptel,NomDAT,	maS,jS,Def,KaS,	[]	[i, At]
biTAcAt	Not exist,	[ficAl, ficAl]	GMa,GFe		,NSg,NPl	Derived Noun,NomD4FP,NomD4D,accepteSC,acceptel,NomDAT,	maS,jS,Def,KaS,	[]	[i, At]
yusrin	Not exist,	[fucl, fucl]	GMa,GFe			Derived Noun,NomDAT,acceptel,NomD4FP,accepteSC,NomD4D,	Smb,maS,Ind,KaS,	[]	[in]
yusri	Not exist,	[fucl, fucl]	GMa,GFe			Derived Noun,NomDAT,acceptel,NomD4FP,accepteSC,NomD4D,	Smb,maS,Def,KaS,	[]	[i]
yusrun	Not exist,	[fucl, fucl]	GMa,GFe			Derived Noun,NomDAT,acceptel,NomD4FP,accepteSC,NomD4D,	Smb,maS,Ind,Raf,	[]	[un]
naqba	Not exist,	[facl]	GFe			Derived Noun,NomD4FP,NomDAT,	mmr,Def,NaS,	[]	[a]
naqbin	Not exist,	[facl]	GFe			Derived Noun,NomD4FP,NomDAT,	mmr,Ind,KaS,	[]	[in]
naqbu	Not exist,	[facl]	GFe			Derived Noun,NomD4FP,NomDAT,	mmr,Def,Raf,	[]	[u]
wahmun	Not exist,	[facl]	GFe			Derived Noun,NomD4FP,NomDAT,	mmr,Ind,Raf,	[]	[un]
wahmu	Not exist,	[facl]	GFe			Derived Noun,NomD4FP,NomDAT,	mmr,Def,Raf,	[]	[u]
wahmin	Not exist,	[facl]	GFe			Derived Noun,NomD4FP,NomDAT,	mmr,Ind,KaS,	[]	[in]
manna	Not exist,	[facl]	GFe			Derived Noun,NomD4FP,NomDAT,	mmr,Def,NaS,	[]	[a]
manni	Not exist,	[facl]	GFe			Derived Noun,NomD4FP,NomDAT,	mmr,Def,KaS,	[]	[i]
wirA~ata	Not exist,	[ficAl]	GFe			Derived Noun,NomD4FP,NomDAT,	maS,Def,NaS,	[]	[a, at, at]
wirA~ati	Not exist,	[ficAl]	GFe			Derived Noun,NomD4FP,NomDAT,	maS,Def,KaS,	[]	[i, at, at]
wirA~atin	Not exist,	[ficAl]	GFe			Derived Noun,NomD4FP,NomDAT,	maS,Ind,KaS,	[]	[in, at, at]
kitAbin	Not exist,	[ficAl, ficAl]	GFe,GMa		,NSg,NPl	Derived Noun,NomDAT,acceptel,NomD4FP,accepteSC,NomD4D,	maS,jS,Ind,KaS,	[]	[in]
kitAba	Not exist,	[ficAl, ficAl]	GFe,GMa		,NSg,NPl	Derived Noun,NomDAT,acceptel,NomD4FP,accepteSC,NomD4D,	maS,jS,Def,NaS,	[]	[a]
kullAbin	Not exist,	[fuccAl]	GMa		,NPl,NSg	Derived Noun,accepteSC,acceptel,	St,etc,Smb,Ind,KaS,	[]	[in]

Figure 2. Morphological analysis of some nouns using the proposed technique

in Natural Language Processing applications. Finally, the morphological analyzer displays the results in aautomaton makes the morphological analyzer efficient and very fast. Our system could be used in NLP applications as syntactic and semantic analysis, spell-checking, information retrieval and machine translation.

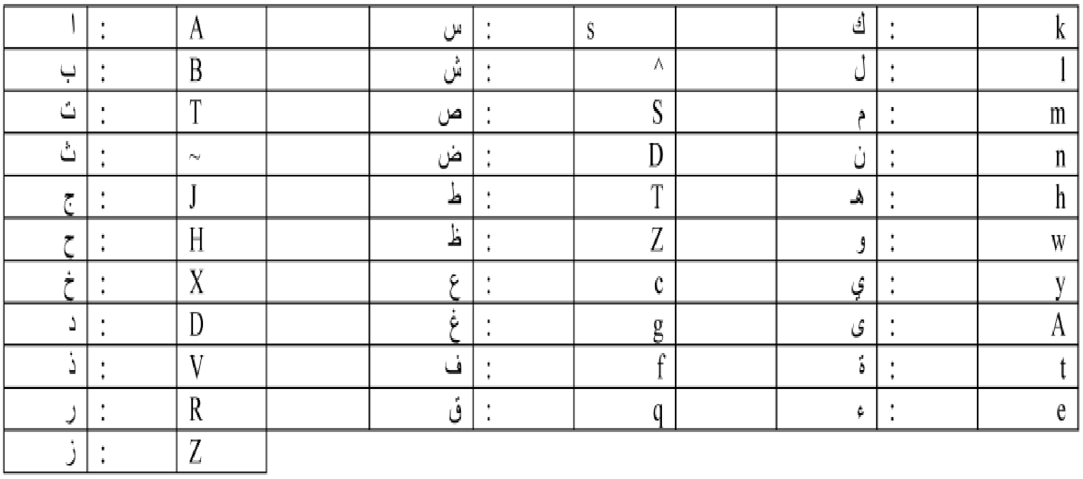
POS	The number	Xerox Morphological Analyzer	ElixirFM	Our System
Nouns	576	60	56	40
Verbs	457	31	24	19
Particles	167	42	45	-
Total	1200	133	125	59
Success (%)		88.91%	89.58%	95.08%

Table 1. The evaluation process results



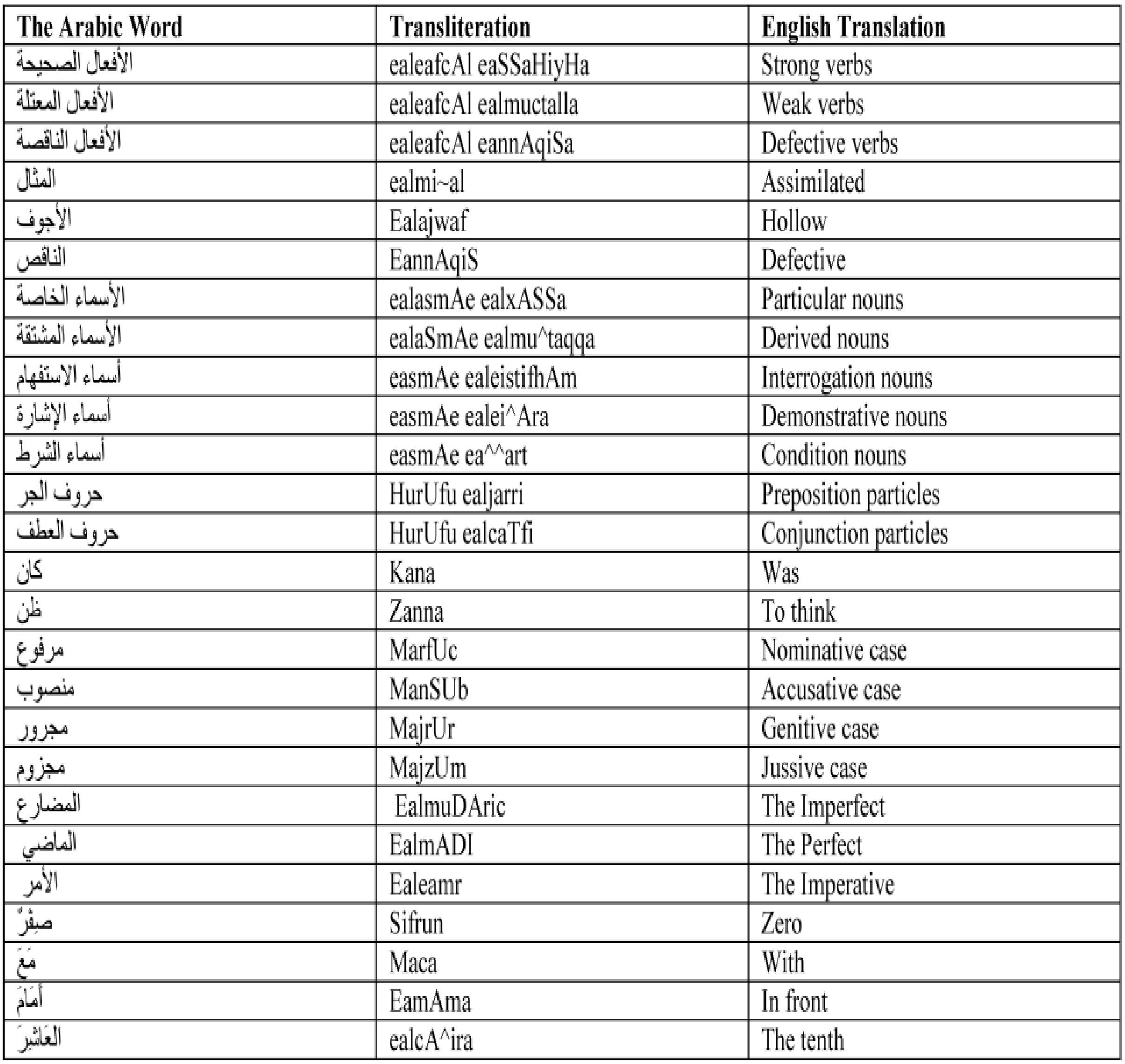
Feature	Description
Gfe	Feminine
Gma	Masculine
Def	Defined
Ind	Undefined
NaS	manSUb « منصوب »
KaS	majrUr « مجرور »
Raf	marfUc « مرفوع »
Jaz	majzUm « مجزوم »
NSg	Singular
NDl	Dual
NPl	Plural
Pr1	First Person
Pr2	Second Person
Pr3	Third Person
MOD	ealmuDAric « المضارع »
MAD	ealmADI « الماضي »
ACT	Active
PAS	Passive
AccepteSC	Accept Case Suffixes
Efc	Eismu fAcil « اسم فاعل »
Emf	Eismu mafcUl « اسم مفعول »
Mmi	maSdar mImI « اسم ميم »
Zam	Zarfu zamAn « ظرف زمان »
Mak	Zarfu makAn « ظرف مكان »
Mmr	maSdar_ealmarrat « مصدر المرة »
MaS	maSdar « مصدر »
JtS	jamcu taksIr li Sifatin « جمع تكسير لصفة »
Smb	SIgatu ealmubalagati « صيغة المبالغة »
Sif	Sifatun « صفة »

Appendix 1. Features signification in morphological analysis



ا	:	A		س	:	s		ك	:	k
ب	:	B		ش	:	^		ل	:	l
ت	:	T		ص	:	S		م	:	m
ث	:	~		ض	:	D		ن	:	n
ج	:	J		ط	:	T		هـ	:	h
ح	:	H		ظ	:	Z		و	:	w
خ	:	X		ع	:	c		ي	:	y
د	:	D		غ	:	g		ى	:	A
ذ	:	V		ف	:	f		ة	:	t
ر	:	R		ق	:	q		ء	:	e
ز	:	Z								

Appendix 3. Letter mappings



The Arabic Word	Transliteration	English Translation
الأفعال الصحيحة	ealeafcAl eaSSaHiyHa	Strong verbs
الأفعال المعتلة	ealeafcAl ealmuctalla	Weak verbs
الأفعال الناقصة	ealeafcAl eannAqiSa	Defective verbs
المثال	ealmi~al	Assimilated
الأجوف	Ealajwaf	Hollow
الناقص	EannAqiS	Defective
الأسماء الخاصة	ealasmAe ealxASSa	Particular nouns
الأسماء المشتقة	ealaSmAe ealmu^taqqa	Derived nouns
أسماء الاستفهام	easmAe ealeistifhAm	Interrogation nouns
أسماء الإشارة	easmAe ealei^Ara	Demonstrative nouns
أسماء الشرط	easmAe ea^^art	Condition nouns
حروف الجر	HurUfu ealjarri	Preposition particles
حروف العطف	HurUfu ealcaTfi	Conjunction particles
كان	Kana	Was
ظن	Zanna	To think
مرفوع	MarfUc	Nominative case
منصوب	ManSUb	Accusative case
مجرور	MajrUr	Genitive case
مجزوم	MajzUm	Jussive case
المضارع	EalmuDAric	The Imperfect
الماضي	EalmADI	The Perfect
الأمر	Ealeamr	The Imperative
صِفْرٌ	Sifrun	Zero
مَعَ	Maca	With
أَمَامَ	EamAma	In front
العَاشِرَ	ealcA^ira	The tenth

Appendix 2. The English translation of Arabic words

References

[1] Abdusalam F,A., Nwesri, S,M,M.,Tahaghoghi, Falk Scholer. (2005). String Processing and Information Retrieval: 12th International Conference, SPIRE 2005, Buenos Aires, Argentina, LNCS, Springer Berlin / Heidelberg, p. 206 – 217.

[2] Ahmed, M, A. (2000). A Large Scale Computational Processor of the Arabic Morphology, and Applications. Master thesis, Cairo University, Egypt.

[3] Ahmed, F., Nürnberger A (2007). N-grams Conflation Approach for Arabic, ACM SIGIR Conference, Amsterdam.

[4] Al-Ameed H, Al-Ketbi S, Al-Kaabi K, Al-Shebli K, Al-Shamsi N, Al-Nuaimi N, Al-Muhairi S (2005) Arabic Light Stemmer: A new Enhanced Approach. The Second International Conference on Innovations in Information Technology (IIT'05).

[5] Al-Fedaghi S., Al-Anzi F. (1989). A New Algorithm to Generate Arabic RootYPattern Forms, *In: Proceedings of the 11th National Computer Conference*, King Fahd University of Petroleum & Minerals, Dhahran, p. 04Y07, Saudi Arabia.

[6] Al-Sahmsi, F., Guessoum, A . (2006). A hidden Markov Model – Based POS Tagger for Arabic. 8es Journees internationals d'Analyse statistique des Donnees Textuelles.

[7]Al-Sughaiyer Imad, A., Al-Kharashi Ibrahim, A. (2004). Arabic morphological analysis techniques: A comprehensive survey, *Journal of the American Society for Information Science and Technology*, 55 (3) 189–213.

[8] Altantawy Mohamed, Nizar Habash, Owen Rambow & Ibrahim Saleh (2010). Morphological Analysis and Generation of Arabic Nouns: A Morphemic Functional Approach. *In: Proceedings of the Language Resource and Evaluation Conference*, Malta.

[9] Attia, M. (2006). An Ambiguity-Controlled Morphological Analyzer for Modern Standard Arabic Modelling Finite State Networks. The Challenge of Arabic for NLP/MT Conference, the British Computer Society, London.

[10] Atwell E., Al-Sulaiti L., Al-Osaimi S., Abu Shawar B. (2004, April). A Review of Arabic Corpus Analysis Tools, JEP-TALN 04,Arabic Language Processing, Fès, 19-22.

[11] Beesley, K.R. (1996). Arabic Finite-State Morphological Analysis and Generation, *Proceedings of the 16th conference on Computational linguistics*, V 1. Copenhagen, Denmark: Association for Computational Linguistics, p 89-94.

[12] Beesley KR. (2000, August). Finite-State Non-Concatenative Morphotactics SIGPHON-2000, *In*: *Proceedings of the Fifth Workshop of the ACL Special Interest Group in Computational Phonology*, p. 1-12, Luxembourg.

[13] Buckwalter, T. (2002). Buckwalter Arabic Morphological Analyzer Version 1.0. Linguistic Data Consortium, University of Pennsylvania, LDC Catalog No.: LDC2002L49.

[14] Buckwalter, T. (2004). Buckwalter Arabic morphological analyzer version 2.0.

[15] Cavalli-Sforza, V., Soudi, A., Teruko, M. (2000). Arabic Morphology Generation Using a Concatenative Strategy. *In: Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics* (NAACL 2000), Seattle, USA.

[16] Chen, A., Gey, F. (2002). Building an Arabic stemmer for information retrieval.

[17] Darwish K. (2002). Building a Shallow Morphological Analyzer in One Day, Proceedings of the workshop on Computational Approaches to Semitic Languages in the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02). Philadelphia, PA, USA.

[18] Diab, M., Hacioglu, K., Jurafsky, D. (2004). Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. In Proceedings of HLT NAACL 2004, Boston, MA, p 149 152.

[19] El-Hajar, A., Hajar, M., Zreik, K. (2010). A System for Evaluation of Arabic Root Extraction Methods, *In*: *Fifth international Conference on Internet and Web Applications and Services*.

[20] El-Sadany & Hashish (1989). An Arabic Morphological System. *IBM SYSTEM JOURNAL* 28 (4).

[21] Elkateb, S. (2005). Design and implementation of an English Arabic dictionary/editor. PhD thesis, The University of Manchester, United Kingdom.

[22] Forsberg , M., Ranta A. (2004). Functional Morphology ICFP'04, Proceedings of the Ninth ACM SIGPLAN International Conference of Functional Programming, 19-21. Snowbird, Utah.

[23] Francopoulo, G., & George, M. (2008). ISO/TC 37/SC 4 N453 (N330 Rev.16). Language resource management — Lexical markup framework (LMF).

[24] Habash Nizar (2004). Large scale lexeme based Arabic Morphological generation. *In*: Proceedings of Traitement Automatique du Langage Naturel (TALN-04). Fez, Morocco.

[25] Habash Nizar, Rambow Owen (2005). Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. *In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics* ACL 2005, p. 573–580, Ann Arbor.

[26] Koehn, P., Hoang, H. (2007, June). Factored translation models. *In: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (EMNLP-CoNLL), p. 868–876, Prague, Czech Republic.

[27] Khoja, S, Garside R. (1999). Stemming Arabic Text. Technical report, Lancaster University, Lancaster, U.K.

[28] Klavans, J. L., Tzoukermann, L. (1992). Morphology. *In*: Stuart C. Shapiro, ed. Encyclopaedia of artificial intelligence. NewYork: John Wiley & sons.

[29] Larkey L., Ballesteros, L., Connel, M.E. (2002). Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. *Proc. of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 275 – 282.

[30]Larkey L., Ballesteros, L., Connell, M. (2005). Light Stemming for Arabic IR Arabic Computational Morphology: Knowledgebased and Empirical Methods, A.Soudi, A. van en Bosch, and Neumann, G., Editors. Kluwer/Springer's serieson Text, Speech, and Language Technology.

[31] Mohamed El-Hadj, Al-Sughayeir IA, Al-Ansari AM. (2009). Arabic Part of Speech Tagging Using the Sentence Structure 2nd international Conference on Arabic Language Resources & Tools. Cairo.

[32] Otakar Smrz (2007). ElixirFM. Implementation of Functional Arabic Morphology. In ACL Proceedings of the Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources, pages 1–8, Prague, Czech Republic. Otakar Smr• and Viktor Bielický. 2010. ElixirFM. Functional Arabic Morphology, http://sourceforge.net/projects/elixir-fm/.

[33] Sinane, M., Rammal, M., Zreik, K. (2008). Arabic documents classification using N-gram, Conference ICHSL6, Toulouse.

[34] Soudi, A., Cavalli Sforza, V., Jamari, A. (2001). A Computational Lexeme based Treatment of Arabic Morphology. *In*: *Proceedings of the Workshop on Arabic Language Processing: Status and Prospects*, (ACL 2001), Toulouse, France, p: 155 162.