Hindawi Publishing Corporation
Advances in Human-Computer Interaction
Volume 2011, Article ID 629305, 15 pages
doi:10.1155/2011/629305

Research Article

An XML Approach of Coding a Morphological Database for Arabic Language

Mourad Gridach and Noureddine Chenfour

Mathematics and Informatics Department, Faculty of Sciences of Fez, Fez, Morocco

Correspondence should be addressed to Mourad Gridach, mourad_i4@yahoo.fr

Received 18 June 2010; Revised 3 September 2010; Accepted 6 January 2011

Academic Editor: M. Tory

We present an XML approach for the production of an Arabic morphological database for Arabic language that will be used in morphological analysis for modern standard Arabic (MSA). Optimizing the production, maintenance, and extension of morphological database is one of the crucial aspects impacting natural language processing (NLP). For Arabic language, producing a morphological database is not an easy task, because this it has some particularities such as the phenomena of agglutination and a lot of morphological ambiguity phenomenon. The method presented can be exploited by NLP applications such as syntactic analysis, semantic analysis, information retrieval, and orthographical correction.

1. Introduction

Currently, Arabic faces many challenges due to a lot of reasons such as the increase of the Arabic web sites, Arabic media, and Arabic companies around the world using the Arabic language. For these reasons, a lot of research in the domain has been developed to satisfy the increasing demand of the applications using Arabic. Arabic morphology is one of the essential needs in this domain, and lots of morphological analyzers are available now, some of them have a commercial purpose, and the others are available for research and evaluation as discussed by Attia [1]. The development of a morphological analyzer requires the production of a morphological database, and many approaches has been developed. In morphological analysis, Buckwalter Arabic morphological analyzer and Xerox Arabic finite state morphology are two of the best known morphological analyzers for MSA, and they are available and well documented. Concerning the production of Arabic lexicon resource, the LMF approach is became as one of the most used for representing the lexicon resource.

The morphological analysis of Arabic is interested, as of other languages, in the structure of the word as discussed by El-Sadany and Hashish [2]. But being given the wealth of the Arabic word's structure and the problem of agglutination, the operation becomes more complex than in the other languages. We also note that diacritics are particularities for our language, and they are also considered as another source of difficulty for morphological analysis. For all these reasons seen so far, Arabic is conceived as one of the languages that present a big problem in the morphological analysis and make this process very complicated.

In this paper, we will be presenting a morphological analyzer based on morphological automaton developed using a new approach for the production of a morphological database. To develop the morphological analyzer and product the morphological database, we used the particularities of Arabic that is concretized on multilevel: verbs and nouns are also characterized by a specific representation named the matrix "root scheme". This representation will help us construct a morphological automaton for the Arabic language. We note that we have used an innovative language (XMODEL) to represent the Arabic morphological knowledge. The use of this new language helps us to reduce the number of the entries in the lexicon. It also makes our system very flexible and one of the best existing morphological analyzers for the Arabic language.

The structure of the paper is as follows. First, in this introduction, we discuss some challenges of Arabic language

and the importance of the production of morphological database and morphological analyzers in natural language processing. After that, we present some morphological analyzers for Arabic language related to our work in Section 2. Then, in Section 3, we explain our approach for the choice of the linguistic resource and compared it with the LMF approach. In Section 4, we present our Arabic Morphological Analyzer. In Section 5, we evaluate the proposed technique. In Section 6, we discuss the obtained results. Finally, in Section 7, we draw some conclusions and future works to be done.

2. Works in the Domain

Morphological processing involves two different tasks according to the operation type: generation and analysis. In generation, we produce correct forms using given morphemes, while in analysis, we try to identify morphemes for a given word. A lot of research has been done in the development of morphological analyzers for Arabic; some of them are available for research and evaluation, while the rest have a commercial purpose.

2.1. Buckwalter Arabic Morphological Analyzer (2004). This analyzer 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 as discussed by Atwell et al. [3]. 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 a result, the process of analyzing an Arabic text becomes long.

2.2. Xerox Arabic Morphological Analysis and Generation. Xerox Arabic morphological Analyzer is well known in the literature and available for evaluation and well documented. This analyzer is constructed using finite state technology (FST) as discussed by Beesley [4, 5]. 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 words that do not exist in the traditional Arabic dictionaries as discussed by Darwish [6], and we can consider the volume of the lexicon as another disadvantage of this analyzer which could affect the analysis process.

2.3. ElixirFM: An Arabic Morphological Analyzer by Otakar Smrz. ElixirFM is an online Arabic morphological analyzer for modern written Arabic developed by Otakar Smrz 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 and Ranta [7]) and initially relied on the reprocessed Buckwalter lexicon as discussed by Buckwalter [8]. It contains two main components: a multipurpose programming library and a linguistically morphological lexicon as discussed by Smrž [9]. 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 has limited coverage, because it analyzes only words in the modern written arabic.

3. Linguistic Resource

So as to develop a morphological analyzer of the Arabic language, representing the morphological knowledge becomes very crucial. Besides this, it is viewed as one of the central problems of the automatic processing of the Arabic morphology.

According to some works, in order to represent the morphological knowledge of the Arabic language, they have chosen to use the database concept as a basic support to store the morphological information as discussed elsewhere [5, 10]. To seek any information, they make use of requests consulting. Unfortunately, this method remains very limited to this type of challenges. Consequently, they do not give good results.

A second method of representation, which is widely used, is offered by the artificial intelligence. Accordingly, a morphological analyzer serves as an intelligent system able to infer the morphological nature of the analysed sentence from a certain knowledge-base which consists of data and morphological rules as discussed by Shaalan [11]. However, the artificial intelligence language is criticized for its being general and sequential search of information. The choice of the Lisp or Prolog language as a support of representing the morphological knowledge may not probably be the right option. This is due to the fact that the interpreter is not well adapted to this kind of problems.

A third method which will be used in the future works for representing, designing, and implementing the lexical resource is the method using lexical markup framework (LMF). It was used in lots of languages (Indo-European), but for Arabic language, this method still in progress towards a standard for representing the Arabic linguistic resource. In the next paragraph, we will present our approach to represent the linguistic resource for Arabic, and we finish by giving a comparison between our approach and the LMF approach for representing the linguistic resource.

3.1. Our Approach to Represent the Linguistics Resource. To achieve a better representation of the morphological knowledge of Arabic, we conceived an innovative language adapted for this specific situation: it is the XMODEL language (XML-based morphological definition Language). XMODEL is based on the XML language setting profits of its advantages and particularities. As a result, all morphological entries are gathered in an XMODEL files. Using the new language helps direct search for information and determinism. It also

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <package name="OrigineSchemesPackage">
  - <morphological_class name="OriginSchemeS">
    - <properties>
        <modifier>final</modifier>
        <is>FinalVerbS</is>
        <is>Number.NSg</is>
        <is>Person.Pr3</is>
        <is>Gender.GMa</is>
      </properties>
      <component name="facala" id="1"/>
      <component name="facila" id="2"/>
      <component name="facula" id="3"/>
      <component name="faclala" id="4"/>
    </morphological_class>
  </package>
```

(a)

```
<?xml version="1.0" encoding="UTF-8"?>
- <package name="OrigineSchemesPackage">
  - <morphological_class name="OriginSchemeS">
    - <properties>
        <modifier>final</modifier>
        <is>Number.NSg</is>
        <is>Person.Pr3</is>
        <is>Gender.GMa</is>
      </properties>
      <component name="فَعَلَ" id="1"/>
      <component name="فَعِلَ" id="2"/>
      <component name="فَعُلَ" id="3"/>
      <component name="فَعْلَلَ" id="4"/>
    </morphological_class>
  </package>
```

(b)

Figure 1: Representation of some verbs schemes using XMODEL language.

enables us to represent the whole components, properties, and morphological rules with a very optimal way. To clarify the last point, we note that our morphological database contains 960 lexicon items (morphological components) and 455 morphological rules to be applied to these morphological components which present a remarkable reduction in the number of entries in the lexicon compared to the existing systems (Xerox and Buckwalter). This representation helps us achieve the following goals:

(i) a symbolic definition, declarative and, therefore, progressive of the Arabic morphology,

(ii) a morphological database independent of processing that will be applied (see later),

(iii) a considerable reduction of the number of morphological entries,

(iv) the notion of scheme enables us to define the maximum morphological components by means of XMODEL language.

Our language makes it possible for us to represent the Arabic morphology as morphological classes and rules. Accordingly, our Arabic morphological database will be composed of three main parties: morphological classes, morphological properties, and morphological rules.

Now, let us first introduce the XMODEL language which permits to represent the morphological knowledge of Arabic and consists of the following three main parties.

3.1.1. Morphological Component Class. It enables us represent all morphological components of the Arabic language. It also permits to gather a set of morphological components having the same nature, the same morphological characteristics, and the same semantic actions. Relying on the notion of scheme /*ealwazn*/ (الوزن), this class allows us a better optimization, hence a considerable reduction of morphological entries. By doing so, we need not represent all the language items but only their schemes. We note that our lexicon

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <package name="PropertyPackage">
  - <morphological_properties>
    - <property name="Person" type="exclusive">
        <descriptor name="Pr1"/>
        <descriptor name="Pr2"/>
        <descriptor name="Pr3"/>
      </property>
    - <property name="Gender" type="additive">
        <descriptor name="GFe"/>
        <descriptor name="GMa"/>
      </property>
    </morphological_properties>
  </package>
```

Figure 2: Representation of morphological properties "*Gender*" and "*Person*".

contains 960 items (morphological components) which is a remarkable reduction in the number of the items compared to the other dictionaries (Figure 1).

3.1.2. Morphological Properties Class. It permits to characterize the different morphological components represented by the morphological class: a morphological property class contains a set of morphological descriptors or morphological values of properties that would be assigned to the different morphological components. We mention, for example, the property "*Gender*" which will distinguish between masculine and feminine components. The morphological properties are not related to a specific morphological class, which makes it necessary to define them outside the morphological classes (Figure 2).

We have added the attribute "*type*" to work out the problem of the semantic of the morphological descriptors that might be exclusive (the morphological component cannot be characterized by the morphological descriptors of the same property as in the case of the "*Person*" property) or additive (the morphological component can be characterized

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <morphological_class name="NPEichArat">
  - <properties>
      <uses>Gender</uses>
      <uses>Number</uses>
      <uses>Place</uses>
    </properties>
  - <component name="hAvA">
      <md key="NSg"/>
      <md key="GMa"/>
      <md key="pro"/>
    </component>
  - <component name="vAlika">
      <md key="NSg"/>
      <md key="GMa"/>
      <md key="LOI"/>
    </component>
</morphological_class>
```

(a)

```
<?xml version="1.0" encoding="UTF-8"?>
- <morphological_class name="NPEichArat">
  - <properties>
      <uses>Gender</uses>
      <uses>Number</uses>
      <uses>Place</uses>
    </properties>
  - <component name="هَذَا">
      <md key="NSg"/>
      <md key="GMa"/>
      <md key="pro"/>
    </component>
  - <component name="ذَلِكَ">
      <md key="NSg"/>
      <md key="GMa"/>
      <md key="LOI"/>
    </component>
</morphological_class>
```

(b)

Figure 3: The property of components ("*Gender*" "*Number*", and "*Place*") characterizing the components "*hAvA*" and "*vAlika*".

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <morphological_class name="OriginSchemeS">
  - <properties>
      <is>Number.NSg</is>
      <is>Gender.GMa</is>
    </properties>
    <component name="facala" id="1"/>
    <component name="facila" id="2"/>
    <component name="facula" id="3"/>
    <component name="faclala" id="4"/>
    <component name="eafcala" id="5"/>
</morphological_class>
```

(a)

```
<?xml version="1.0" encoding="UTF-8"?>
- <morphological_class name="OriginSchemeS">
  - <properties>
      <modifier>final</modifier>
      <is>Number.NSg</is>
      <is>Gender.GMa</is>
    </properties>
    <component name="فَعَلَ" id="1"/>
    <component name="فَعِلَ" id="2"/>
    <component name="فَعُلَ" id="3"/>
    <component name="فَعْلَلَ" id="4"/>
    <component name="أَفْعَلَ" id="5"/>
</morphological_class>
```

(b)

Figure 4: Example of the property of classes.

by the morphological descriptors of the same property as it is the case in the "*Gender*" property).

There are two strategies to characterize the morphological components using the following properties.

Property of Components. A morphological class can use a list of morphological descriptors to define its components. Generally speaking, each morphological component can have its own morphological descriptors. As for the "*gender*" property, some components of this class can be masculine, while the others can be feminine. This type of properties is named the property of components. In order to put them into practice, we have introduced the "*uses*" tag. This means that the different morphological descriptors defined by the property of components can be used by the different morphological components of the morphological class (Figure 3).

Property of Classes. This one requires assigning a set of morphological components to the common morphological properties. For example, all components are masculine names. This type of property is known as property of classes. To realize this, we introduce the "*is*" tag (Figure 4). In the above example, all the schemes are singular components and masculine gender. It becomes evident to mention that the same class of the morphological components can use one combination of the tags "*uses*" and "*is*".

Property of Reference. Another strong point of the XMODEL language is the introducing of the notion of property of reference which has an important role to benefit from the specificities of the Arabic morphology. As for the Arabic language, some morphological components might be conjugated forms of other components which we call original components. An example of this is the case of the following components "*afcalu*", "*afcilu* ", and "*afculu*". These components are all conjugated forms of the component "*facala*". We have specified this link of reference between the components using the "*ref*" tag (Figures 5 and 6).

In order to concretize this reference between the components, we have opted the attribute "*id*" to the original component. This attribute is specified in the "*component*"

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <morphological_class name="OriginSchemeS">
    ...
    <component name="facala" id="1"/>
    <component name="facila" id="2"/>
    <component name="facula" id="3"/>
    <component name="faclala" id="4"/>
    <component name="eafcala" id="5"/>
    ...
  </morphological_class>
```

Figure 5: Example of some verbs schemes.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <morphological_class name="VerbSainMuDAric">
  - <properties>
      <ref>OriginSchemeS</ref>
    </properties>
    <component name="afcal" key="1"/>
    <component name="afcil" key="2"/>
    <component name="afcul" key="3"/>
    <component name="ufcil" key="5"/>
    ...
  </morphological_class>
```

Figure 6: The conjugated forms of some verbs.

tag. The components that are conjugated forms will use this code as an attribute of that tag (the "*key*" attribute) to indicate this reference.

3.1.3. Morphological Rules Class. Firstly, it should be noted that we developed 455 morphological rules for the Arabic language. They help us combine some morphological components (morphemes) together to generate correct language words. They use the different morphological components classes as well as the morphological properties classes. The morphological rules classes allow us the possibility to add new morphological descriptors which do not belong to the union of morphological descriptors of components of rules. As a result, they are considered as a generator of language words. The implementation of the morphological rules class permits to put into practice all the possible concatenations between components (Figure 7).

In the above example, this rule permits to generate the components which begin by the prefix "*la*" and the prefix "*bi*".

The structuring of our morphological database using XMODEL language allows us to generate the morphological automaton of the Arabic language. In the next section, we will be dealing with the notion of morphological automaton.

3.2. Comparison with LMF Approach. Lexical markup framework (LMF, ISO-24613) is the ISO standard which provides a common standardized framework for the construction of natural language processing 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 first step in developing LMF was to design an overall framework based on the general features of existing lexicons and to develop a consistent terminology to describe the components of those lexicons. The next step was the actual design of a comprehensive model that best represented all of the lexicons in detail. A large panel of 60 experts contributed a wide range of requirements for LMF that covered many types of NLP lexicons. The editors of LMF worked closely with the panel of experts to identify the best solutions and reach a consensus on the design of LMF. Special attention was paid to the morphology in order to provide powerful mechanisms for handling problems in several languages that were known as difficult to handle.

The aims of LMF are to provide a common model for the creation and use of lexical resources at all levels, 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. It allows instantiation of monolingual, bilingual, or multilingual lexical resources and works at a small scale or large scale. Finally, it tries to cover all natural languages (including languages with rich and complex morphology such as Arabic).

For Arabic language, this new method is still in progress, and some works try to standardize it as discussed elsewhere [12–14]. As we discussed before, LMF covered all levels of linguistic description (morphologic, syntactic, and semantic). To compare this method with our approach, we will be limited to the morphological level.

The morphological extension model allows representing the morphological information of lexicons. Given the inflectional and derivational aspect of Arabic, they have used a package of classes defined in the morphological extension of LMF to describe the scheme and the written lemmatized form and the root and the properties of inflected forms as discussed by Baccar et al. [12]. They defined two strategies to describe the morphology of a word. The first one is to represent explicitly all inflected forms, and the second strategy is to use an inflectional paradigm as discussed by Francopoulo and George [13]. Figure 8 shows an example of representing the lexical entry "kataba" (كَتَبَ).

Related to the example above and the description of our approach seen before, the main remark that we make when we compare the two approaches is concerning the representation of any Arabic word. Our approach permits a remarkable reduction of the lexical entries in the morphological database, because the main idea of our approach is to represent Arabic words by their schemes and not by their roots and schemes as we saw in the LMF model. So, this advantage will affect immediately the size of the morphological database and which was considered as one of the problems of morphological analysis. The future works which will be done on the construction of Arabic lexicon using LMF will prove this remark.

The second advantage of our approach concerns the grammatical features that characterize every lexical entry in

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <package name="RulesPackage">
  - <rules_class name="PrefixeSuffixes">
    - <rule>
        <morpheme key="PrefixeHJar.JarMaDmUr" component="la"/>
        <morpheme key="DamirMuttaSil.RDamirMuttaSil"/>
      </rule>
    - <rule>
        <morpheme key="PrefixeHJar.JarMaDmUr" component="bi"/>
        <morpheme key="DamirMuttaSil.JDamirMuttaSil"/>
      </rule>
    </rules_class>
  </package>
```

Figure 7: Class of rules which represent the components prefixed by the prefix "*la*" and "*bi*".

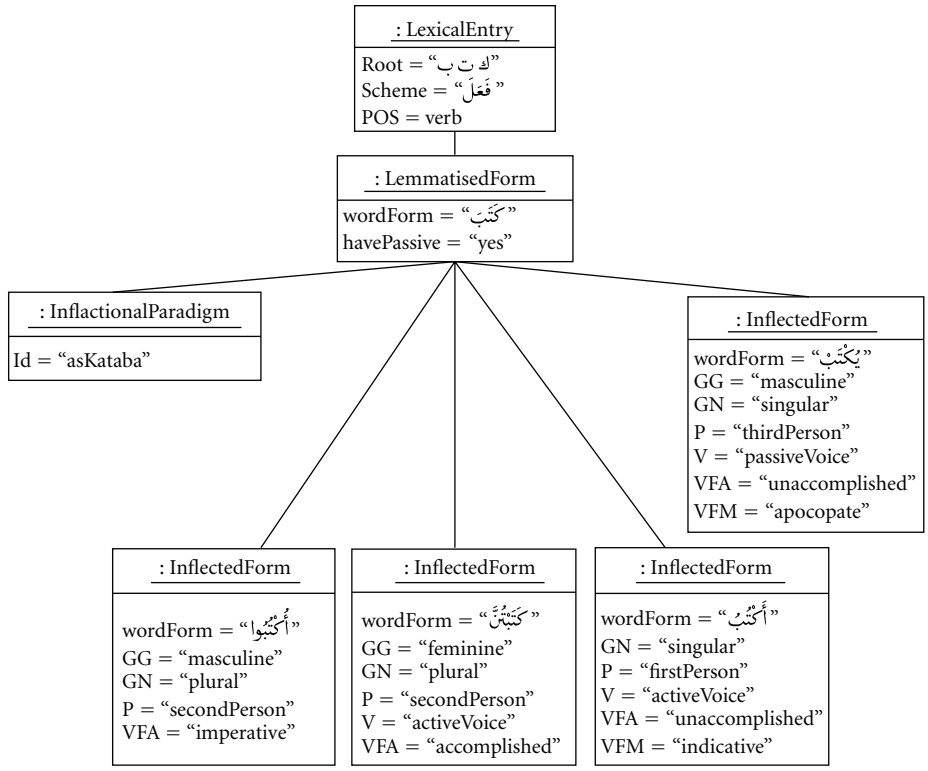


Figure 8: Representation of "kataba" using LMF approach.

the lexicon which will be useful especially in the future works to be done. This is due to the richness of the morphological database presented and the way to represent the lexical entries (Figure 8).

Finally, we can consider that using the new and innovative language (XMODEL) is one of the strengths of this work. We note that the two approaches are extensible, because they use XML to represent the lexical entry.

4. System Description

In this part, we describe the Arabic morphological analyzer. This latter is based on using morphological automaton technology. The implementation of each morphological analyzer for any language needs a main resource. This resource is the morphological database, so the first task is the conception and the realization of a morphological database. We also used a new language, XMODEL language, to create this database. After that, the second task is the development of a set of morphological automatons for Arabic language each of which represents a very definite category of morphology.

4.1. Arabic Morphological Automaton. To implement a morphological analyzer for any language, especially Arabic language, the use of morphological automaton is considered among the most efficient methods. It can be used for both analysis and generation. This latter is based on the notion of finite state automaton (DFA). A word is accepted by the

morphological automaton if it belongs to a correct word in Arabic and rejected in the contrary case. Generally, the Arabic morphological automaton will have the following features.

(i) Q is a finite set of states of the control unit which represents the states of a morphological automaton.

(ii) $\sum$ is a finite input tape alphabet symbols. Concerning morphological automaton for Arabic, it is constituted of the alphabets of Arabic language.

(iii) q_0 is the start state of the morphological automaton. It is constituted of only one start state in the case of a morphological automaton.

(iv) F is a subset of Q. It also represents the accepting states of the morphological automaton for Arabic. This latter has a very important role, because it permits to give us a set of information of the Arabic words analyzed. This information is called the morphological descriptors and they also characterize these words.

(v) The set τ also represents the transition function of the morphological automaton.

Consequently, the building of the morphological automaton of the Arabic language needs to use the XMODEL database discussed before. We have to extract all the morphological rules from this database and construct a morphological automaton of each rule. So, to realize that constructing, we have to use some automaton operations such as concatenation and union operation. Let us clarify how we can use these two operations to generate a morphological automaton for a definite morphological rule. So, we consider the following rule:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <package name="RulesPackage">
  - <rules_class name="CardNbCRules">
    - <rule id="rule_1">
        <morpheme key="CardNumber.CNAccepteSCID.CNAccepteSC"/>
        <morpheme key="CasSuffixe.SCD" component="u"/>
        <idp name="CNADefMarfUc"/>
      </rule>
      ...
    </rules_class>
  </package>
```

So, to generate the morphological automaton which represent this rule, we have to use the operation of concatenation to concatenate the first morpheme (key = "*CardNumber.CNAccepteSCID.CNAccepteSC*") with the second one (key = "*CasSuffixe.SCD*" component = "*u*"). Therefore, the morphological automaton that represents this morphological rule is the following shown in Figure 9.

In addition to the operation of concatenation used to concatenate morphemes or morphological automatons together, we used the union operation to associate two or several morphological automatons generated by the first operation, each one represent a definite morphological rule. To concretize the use of this second operation, let us consider the following class of morphological rules:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <package name="RulesPackage">
  - <rules_class name="CardNbCRules">
    - <rule id="rule_1">
        <morpheme key="CardNumber.CNAccepteSCID.CNAccepteSC"/>
        <morpheme key="CasSuffixe.SCD" component="u"/>
        <idp name="CNADefMarfUc"/>
      </rule>
    - <rule id="rule_2">
        <morpheme key="CardNumber.CNAccepteSCID.CNAccepteSC"/>
        <morpheme key="CasSuffixe.SCD" component="a"/>
        <idp name="CNADefManSUb"/>
      </rule>
      ...
    </rules_class>
  </package>
```

In the above example, we have two morphological rules; each one generates a morphological automaton. We used the union operation to associate the first automaton which represents the rule identified by "*rule_1*" with the second automaton which represents the rule identified by "*rule_2*". The result morphological automaton is shown in Figure 10.

In the following paragraphs, we present a detail of how to construct all the morphological automatons of the Arabic language and technique used in this constructing.

So as to build a morphological automaton of the Arabic language, we have classified words of the Arabic language in to two categories: the first category is that which submits to the derivation process, while the second one does not. This process of derivation 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 named "*ealeawzAn*" /الأوزان/.

A scheme "*ealwazn*" /الوزن/ is an abstract linguistics term that represents a family of varied derived words; these words might be verbs or derived nouns which share the same linguistic features as discussed by Tahir et al. [15]). At the graphical level, a scheme generally constitutes of the following.

(i) three main consonants that are represented by the letters "f" /ف/, "c" /ع/, and "l" /ل/ with a possibility to duplicate the last letter "l" as in the case of schemes that correspond to a four letters root like "*faclala*" /فعلل/,

(ii) some consonants that serve as tools to extend the root like "*stafcala*" /استفعل/ and "*tafAcala*" /تفاعل/,

(iii) a group of adequate vowels.

We have grouped in the first category of words the following items:

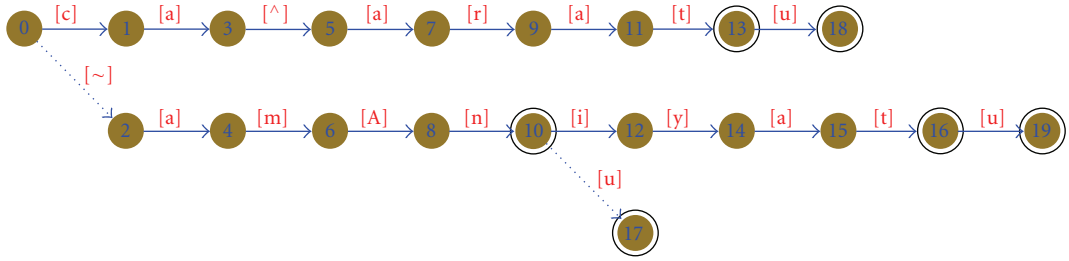


Figure 9: A morphological automaton representing the above morphological rule.

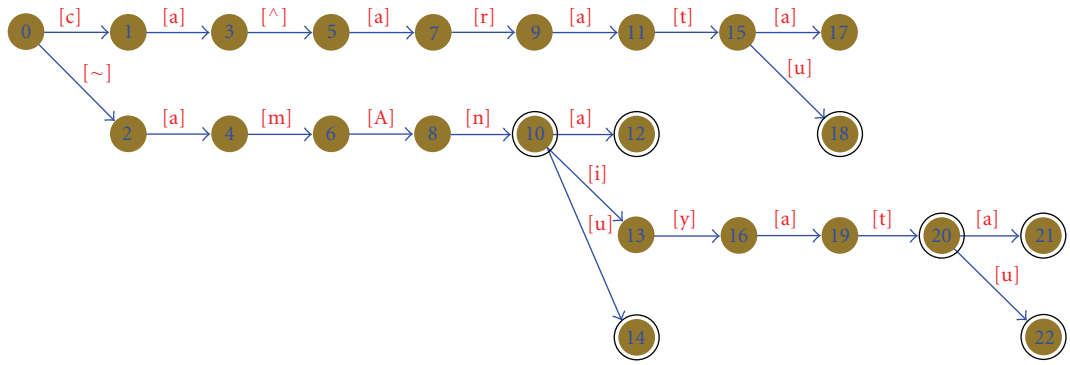


Figure 10: A morphological automaton representing the above rule.

(i) derived nouns "*ealaSmAe ealmu^taqqa*" /الأسماء المشتقة/,

(ii) strong verbs "*ealeafcAl eaSSaHIHa*" /الأفعال الصحيحة/: these are the verbs that contain no weak letters. In the Arabic language, there are three weak letters: "w" /و/, "A" /ا/, and "y" /ي/,

(iii) weak verbs "*ealeafcAl ealmuctalla*" /الأفعال المعتلة/: these are the verbs that contain a weak letter. Weak verbs are also classified into three categories as discussed by Attia [10]:

(a) assimilated "*ealmi~al*" /المثال/: a verb that contains an initial weak letter,

(b) hollow "*ealajwaf*" /الأجوف/: a verb that contains a middle weak letter,

(c) defective "*eannAqiS*" /الناقص/: a verb that contains a final weak letter.

While the second category of words contains three families of words:

(i) the particular nouns "*ealasmAe ealxASSa*" /الأسماء الخاصة/: these nouns comprise proper nouns, names of cities, names of countries, and so forth. It also regroups the exclusive nouns "*easmAe ealeisti~nAe*" /أسماء الاستثناء/, the interrogative nouns "*easmAe ealeistifhAm*" /أسماء الاستفهام/, the demonstrative nouns "*easmAe ealei^Ara*" /أسماء الإشارة/, the conditional nouns "*easmAe ea^^art*" /أسماء الإشارة/, and so forth,

(ii) the particles "*ealHurUf*" /الحروف/ like for example "*HurUfu ealjarri*" /حروف الجر/, "*HurUfu ealjazmi*" /حروف الجزم/, "*HurUfu ealcaTfi*" /حروف العطف/, and so forth,

(iii) the incomplete verbs "*ealeafcAl eannAqiSa*" /الأفعال الناقصة/: this family of verbs contains the family of verb "*kAda*" /كاد/, the family of verb "*kAna*" /كان/ and the family of verb "*Zanna*" /ظن/.

Finally, after the generation the morphological automaton which contains the Arabic vocalized, its size is about 120 MB. Concerning the number of the entries generated by our system, it is about 5961 entries, which represent a remarkable reduction of the number of the entries and makes our system as one of the best existing systems. These obtained results confirm what we explained before about our new language (XMODEL) to represent the Arabic morphological knowledge with an optimal way. To concretize this representation, let us take an example of how to generate a morphological automaton of verbs that have these schemes: "*facala*", "*facila*", "*facula*", and "*faclala*".

So, related to the example in Figure 11, the automaton contains 19 states including 4 accepting states ("19", "16", "17", and "18") which represent the four schemes. This representation permits a remarkable reduction of the number of the morphological entries and explains the results seen before.

We note that developing the morphological automatons of the Arabic language is the main idea of constructing a

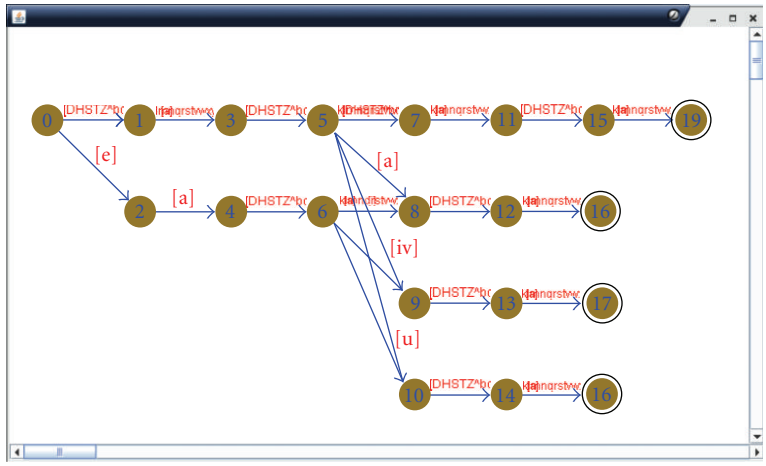


Figure 11: An automaton representing the schemes "*facala*", "*facila*", "*facula*", and "*faclala*".

morphological analyzer for our language. So, in the following paragraph, we will present the proposed technique for the Arabic language (Figure 11).

4.2. The Arabic Morphological Analyzer. First of all, because our morphological analyzer is based on the morphological automaton which is the main idea of this work, so this part it will be shorter than the part of the Arabic morphological automaton. So, in this section, we describe our morphological analyzer for Arabic language. This analyzer is developed using three principal components.

(i) A morphological database constructed using the XMODEL language based on XML language integrating all the data suitable for Arabic language. Its regroups three packages: package of morphological components that contains verbs, nouns, particles, and affixes. The second package includes the morphological rules and the last package is concerned with the morphological properties.

(ii) A set of morphological automatons for the Arabic language each of which represents a very specific morphological category.

(iii) A program handling the morphological database and the morphological automaton. It is developed through the use of Java language.

In addition, the method presented is meant to give a set of information about any Arabic word given to it. This set of information is about the following.

(i) The gender of the word: masculine or feminine.

(ii) The person of the word: first, second, or third person.

(iii) The number of the word: singular, dual, or plural.

(iv) The case of the word: "*marfUc*" (مرفوع), "*manSUb*" (منصوب), "*majrUr*" (مجرور), or "*majzUm*" (مجزوم).

(v) The type of the word: verb, noun or particle.

(vi) If the word is a verb, we give its tense: present ("*ealmuDAric*": المضارع), past ("*ealmADI*": الماضي), or imperative ("*ealeamr*": الأمر). We also give its voice: active or passive.

(vii) The origin scheme of the word is given if available.

Let us analyze some examples of Arabic verbs and nouns using our system. 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 our test, we used a vocalized version. This standard input text is provided in this file: http://www.alecso.org.tn/images/stories/OULOUM/MOHALLILAT%20SARFIA_DAMAS_2009/020%20NIZAR.html (Figure 12).

The example above shows the results of the morphological analysis of some verbs using our system. The example below will show the results of the morphological analysis of some nouns.

We note that this work is very rich for the information giving about every noun analyzed which makes it one of the best Arabic morphological analyzers. So, let us take the "properties" column in the case of nouns, we conclude that our system gives the maximum of information about each word analyzed which will be very useful for the future works to be done (Figure 13).

We note that this set of information has an important role especially in future works like for example the building of a syntactic analyzer, a semantic analyzer, machine translation, and so forth.

Finally, the development of our morphological analyzer for Arabic language has many advantages such as the following.

(i) The separation between the task of the linguist and the developer.

(ii) We can also reuse our programs in future works.

Lexical Table of File : arab files\dfa01.txt

A	B	C	D	E	F	G	H	I	J
Morphological	Original Scheme	Scheme	Gender	Person	Number	Properties	Morphological Descriptors	Prefixes	Suffixes
yatadaHrajAni	[tafaclala],	[]	GMa	,Pr3	,NDI	Strong Verb,MOD,ACT,	Raf,	[y]	[Ani]
eaSTaffa	[eifcalla],	[]	GFe,GMa	,Pr1	,NSg	Strong Verb,ACT,MOD,	Def,NaS,	[e]	[a]
eaSTaffa	[eifcalla],	[]	GFe,GMa	,Pr1	,NSg	Strong Verb,ACT,MOD,	Def,NaS,	[e]	[a]
eaSTaffu	[eifcalla],	[]	GFe,GMa	,Pr1	,NSg	Strong Verb,ACT,MOD,	Def,Raf,	[e]	[u]
yacuddu	[cadda],	[]	GMa	,Pr3	,NSg	Incomplete Verb,MOD,	Def,Raf,	[y]	[u]
yacidu	[wacala, wacila, wacula],	[]	GMa	,Pr3	,NSg	Weak Verb,ACT,MOD,	Def,Raf,	[y]	[u]
yucda	[facA, faciya],	[]	GMa	,Pr3	,NSg	Weak Verb,PAS,MOD,	NaS,Jaz,	[y]	[a]
yari~a	[wacala, wacila, wacula],	[]	GMa	,Pr3	,NSg	Weak Verb,ACT,MOD,	Def,NaS,	[y]	[a]
yar~i	[eafcA],	[]	GMa	,Pr3	,NSg	Weak Verb,ACT,MOD,	NaS,Jaz,	[y]	[i]
yur~a	[facA, faciya],	[]	GMa	,Pr3	,NSg	Weak Verb,PAS,MOD,	NaS,Jaz,	[y]	[a]
yari~u	[wacala, wacila, wacula],	[]	GMa	,Pr3	,NSg	Weak Verb,ACT,MOD,	Def,Raf,	[y]	[u]
eastaqbila	[eistafcala],	[]	GFe,GMa	,Pr1	,NSg	Strong Verb,ACT,MOD,	Def,NaS,	[e]	[a]
eastaqbilu	[eistafcala],	[]	GFe,GMa	,Pr1	,NSg	Strong Verb,ACT,MOD,	Def,Raf,	[e]	[u]
tuSAfiHa	[fAcala],	[]	GMa,GFe	,Pr2,Pr3	,NSg	Strong Verb,ACT,MOD,	Def,NaS,	[t]	[a]
tuSAfiHa	[fAcala],	[]	GMa,GFe	,Pr2,Pr3	,NSg	Strong Verb,ACT,MOD,	Def,NaS,	[t]	[a]
tuSAfiHu	[fAcala],	[]	GMa,GFe	,Pr2,Pr3	,NSg	Strong Verb,ACT,MOD,	Def,Raf,	[t]	[u]
eabtacidu	[eiftacala],	[]	GFe,GMa	,Pr1	,NSg	Strong Verb,ACT,MOD,	Def,Raf,	[e]	[u]
eabtacida	[eiftacala],	[]	GFe,GMa	,Pr1	,NSg	Strong Verb,ACT,MOD,	Def,NaS,	[e]	[a]
yankami^a	[einfacala]	[]	GMa	,Pr3	,NSg	Strong Verb,ACT,MOD,	Def,NaS,	[y]	[a]
yankami^u	[einfacala]	[]	GMa	,Pr3	,NSg	Strong Verb,ACT,MOD,	Def,Raf,	[y]	[u]
tusarbila	[faclala],	[]	GMa,GFe	,Pr2,Pr3	,NSg	Strong Verb,ACT,MOD,	Def,NaS,	[t]	[a]
tusarbila	[faclala],	[]	GMa,GFe	,Pr2,Pr3	,NSg	Strong Verb,ACT,MOD,	Def,NaS,	[t]	[a]
tusarbilu	[faclala],	[]	GMa,GFe	,Pr2,Pr3	,NSg	Strong Verb,ACT,MOD,	Def,Raf,	[t]	[u]
tusarbilu	[faclala],	[]	GMa,GFe	,Pr2,Pr3	,NSg	Strong Verb,ACT,MOD,	Def,Raf,	[t]	[u]
lAHaZat	[fAcala],	[]	GFe	, Pr3	,NSg	Strong Verb,ACT,MAD		[]	[at]
lAHaZtu	[fAcala],	[]	GFe,GMa	,Pr1	,NSg	Strong Verb,MAD,ACT		[]	[tu]
lAHaZta	[fAcala],	[]	GMa	,Pr2	,NSg	Strong Verb,MAD,ACT		[]	[ta]
lAHaZti	[fAcala],	[]	GFe	,Pr2	,NSg	Strong Verb,MAD,ACT		[]	[ti]
Darrabat	[faclala, faccala],	[]	GFe	, Pr3	,NSg	Strong Verb,ACT,MAD		[]	[at]
Darrabti	[faclala, faccala],	[]	GFe	,Pr2	,NSg	Strong Verb,MAD,ACT		[]	[ti]
Duribat	[facula, facala, facila],	[]	GFe	, Pr3	,NSg	Strong Verb,MAD,PAS		[]	[at]
Duribtu	[facula, facala, facila],	[]	GFe,GMa	,Pr1	,NSg	Strong Verb,MAD,PAS		[]	[tu]
Duribti	[facula, facala, facila],	[]	GFe	,Pr2	,NSg	Strong Verb,MAD,PAS		[]	[ti]
Darabtu	[facala],	[]	GFe,GMa	,Pr1	,NSg	Strong Verb,MAD,ACT		[]	[tu]
kabkabat	[faclala],	[]	GFe	, Pr3	,NSg	Strong Verb,ACT,MAD		[]	[at]
faramtu	[facala],	[]	GFe,GMa	,Pr1	,NSg	Strong Verb,MAD,ACT		[]	[tu]
faramti	[facala],	[]	GFe	,Pr2	,NSg	Strong Verb,MAD,ACT		[]	[ti]
farramti	[faclala, faccala],	[]	GFe	,Pr2	,NSg	Strong Verb,MAD,ACT		[]	[ti]
faramat	[facala],	[]	GFe	, Pr3	,NSg	Strong Verb,ACT,MAD		[]	[at]
farramtu	[faclala, faccala],	[]	GFe,GMa	,Pr1	,NSg	Strong Verb,MAD,ACT,		[]	[tu]

Figure 12: A morphological analysis of some verbs using our system.

(iii) Development standardization means in our application that we have build all the applications with the same standards technologies (Java language, XML technologies, SAX, DOM, etc.).

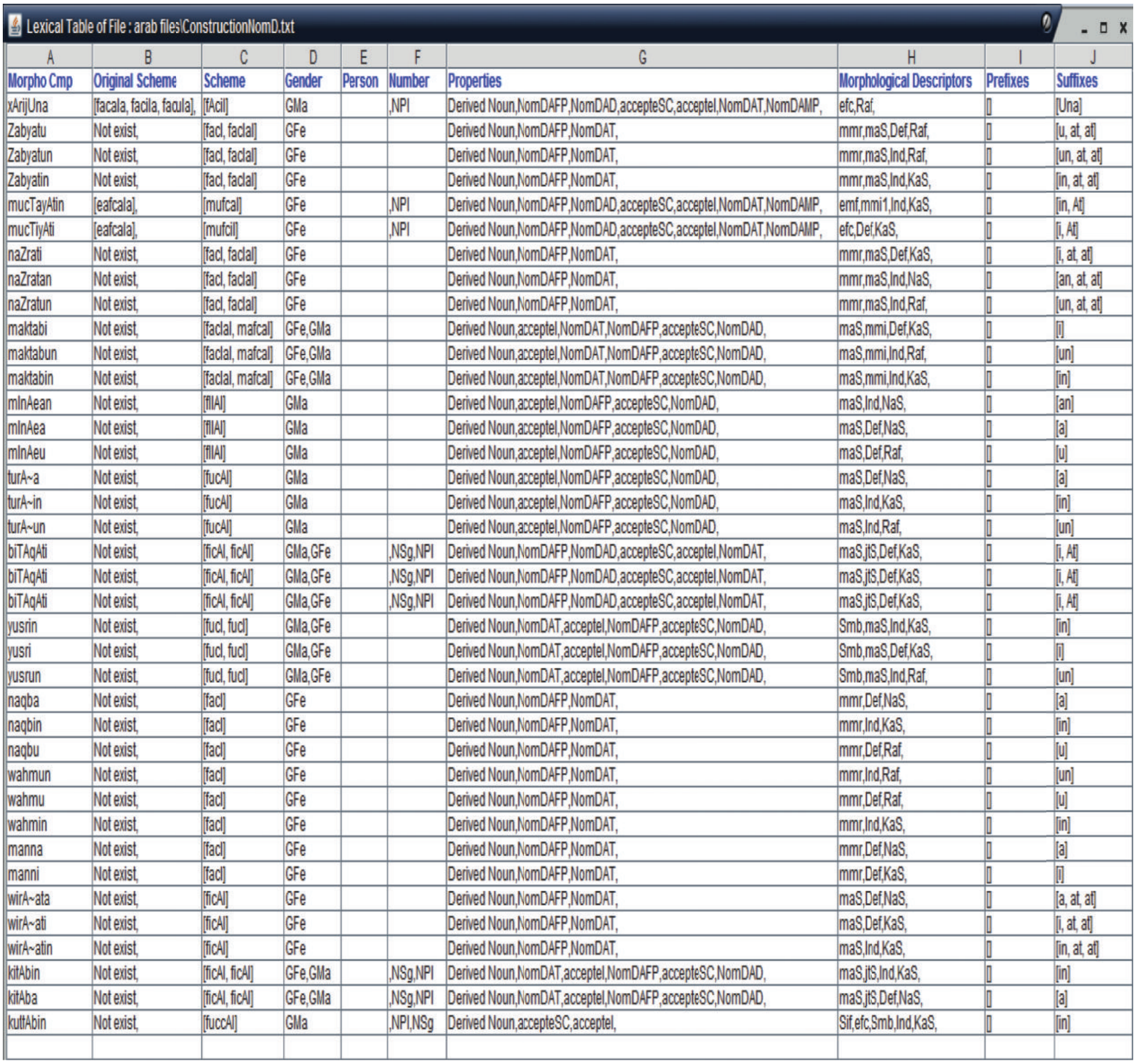
(iv) The work presented is developed using Java language. Therefore, our analyzer can be run in any platform such as Windows, Linux, UNIX, and Mac OS.

(v) The facility of maintenance: it is easy to add some new features to our system if the user or the linguist needs them for his Arabic morphological analysis. It is also easy to extend our system to include some new works related to NLP of Arabic such as information retrieval, syntactic and semantic analyzers, correction, and generation of Arabic texts.

5. Evaluation

In this section, we are going to evaluate Xerox Arabic morphological analyzer, the Arabic morphological analyzer by Otakar Smrz, and our system. We note that a standard annotated corpus for Arabic language is not yet available, and for this reason, the process of evaluation will be difficult. So, we have chosen Xerox morphological analyzer and the morphological analyzer by Otakar Smrz, because they are one of the best known morphological analyzers for MSA, and they are also available and well documented as shown in Tables 1, 2, and 3.

On the one hand, the first remark when we compare the three morphological analyzers is about the information giving by each one. Used an innovative language (XMODEL) for



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
xArijUna	[facala, facila, facula],	[fAcil]	GMa		,NPl	Derived Noun,NomDAFP,NomDAD,accepteSC,acceptel,NomDAT,NomDAMP,	efc,Raf,	[]	[Una]
Zabyatu	Not exist,	[facl, faclal]	GFe			Derived Noun,NomDAFP,NomDAT,	mmr,maS,Def,Raf,	[]	[u, at, af]
Zabyatun	Not exist,	[facl, faclal]	GFe			Derived Noun,NomDAFP,NomDAT,	mmr,maS,Ind,Raf,	[]	[un, at, af]
Zabyatin	Not exist,	[facl, faclal]	GFe			Derived Noun,NomDAFP,NomDAT,	mmr,maS,Ind,KaS,	[]	[in, at, af]
mucTayAtin	[eafcala],	[mufcal]	GFe		,NPl	Derived Noun,NomDAFP,NomDAD,accepteSC,acceptel,NomDAT,NomDAMP,	emf,mmi1,Ind,KaS,	[]	[in, At]
mucTiyAti	[eafcala],	[mufcil]	GFe		,NPl	Derived Noun,NomDAFP,NomDAD,accepteSC,acceptel,NomDAT,NomDAMP,	efc,Def,KaS,	[]	[i, At]
naZrati	Not exist,	[facl, faclal]	GFe			Derived Noun,NomDAFP,NomDAT,	mmr,maS,Def,KaS,	[]	[i, at, af]
naZratan	Not exist,	[facl, faclal]	GFe			Derived Noun,NomDAFP,NomDAT,	mmr,maS,Ind,NaS,	[]	[an, at, af]
naZratun	Not exist,	[facl, faclal]	GFe			Derived Noun,NomDAFP,NomDAT,	mmr,maS,Ind,Raf,	[]	[un, at, af]
maktabi	Not exist,	[faclal, mafcal]	GFe,GMa			Derived Noun,acceptel,NomDAT,NomDAFP,accepteSC,NomDAD,	maS,mmi,Def,KaS,	[]	[i]
maktabun	Not exist,	[faclal, mafcal]	GFe,GMa			Derived Noun,acceptel,NomDAT,NomDAFP,accepteSC,NomDAD,	maS,mmi,Ind,Raf,	[]	[un]
maktabin	Not exist,	[faclal, mafcal]	GFe,GMa			Derived Noun,acceptel,NomDAT,NomDAFP,accepteSC,NomDAD,	maS,mmi,Ind,KaS,	[]	[in]
mlnAean	Not exist,	[flIAl]	GMa			Derived Noun,acceptel,NomDAFP,accepteSC,NomDAD,	maS,Ind,NaS,	[]	[an]
mlnAea	Not exist,	[flIAl]	GMa			Derived Noun,acceptel,NomDAFP,accepteSC,NomDAD,	maS,Def,NaS,	[]	[a]
mlnAeu	Not exist,	[flIAl]	GMa			Derived Noun,acceptel,NomDAFP,accepteSC,NomDAD,	maS,Def,Raf,	[]	[u]
turA~a	Not exist,	[fucAl]	GMa			Derived Noun,acceptel,NomDAFP,accepteSC,NomDAD,	maS,Def,NaS,	[]	[a]
turA~in	Not exist,	[fucAl]	GMa			Derived Noun,acceptel,NomDAFP,accepteSC,NomDAD,	maS,Ind,KaS,	[]	[in]
turA~un	Not exist,	[fucAl]	GMa			Derived Noun,acceptel,NomDAFP,accepteSC,NomDAD,	maS,Ind,Raf,	[]	[un]
biTAqAti	Not exist,	[ficAl, ficAl]	GMa,GFe		,NSg,NPl	Derived Noun,NomDAFP,NomDAD,accepteSC,acceptel,NomDAT,	maS,jtS,Def,KaS,	[]	[i, At]
biTAqAti	Not exist,	[ficAl, ficAl]	GMa,GFe		,NSg,NPl	Derived Noun,NomDAFP,NomDAD,accepteSC,acceptel,NomDAT,	maS,jtS,Def,KaS,	[]	[i, At]
biTAqAti	Not exist,	[ficAl, ficAl]	GMa,GFe		,NSg,NPl	Derived Noun,NomDAFP,NomDAD,accepteSC,acceptel,NomDAT,	maS,jtS,Def,KaS,	[]	[i, At]
yusrin	Not exist,	[fucl, fucl]	GMa,GFe			Derived Noun,NomDAT,acceptel,NomDAFP,accepteSC,NomDAD,	Smb,maS,Ind,KaS,	[]	[in]
yusri	Not exist,	[fucl, fucl]	GMa,GFe			Derived Noun,NomDAT,acceptel,NomDAFP,accepteSC,NomDAD,	Smb,maS,Def,KaS,	[]	[i]
yusrun	Not exist,	[fucl, fucl]	GMa,GFe			Derived Noun,NomDAT,acceptel,NomDAFP,accepteSC,NomDAD,	Smb,maS,Ind,Raf,	[]	[un]
naqba	Not exist,	[facl]	GFe			Derived Noun,NomDAFP,NomDAT,	mmr,Def,NaS,	[]	[a]
naqbin	Not exist,	[facl]	GFe			Derived Noun,NomDAFP,NomDAT,	mmr,Ind,KaS,	[]	[in]
naqbu	Not exist,	[facl]	GFe			Derived Noun,NomDAFP,NomDAT,	mmr,Def,Raf,	[]	[u]
wahmun	Not exist,	[facl]	GFe			Derived Noun,NomDAFP,NomDAT,	mmr,Ind,Raf,	[]	[un]
wahmu	Not exist,	[facl]	GFe			Derived Noun,NomDAFP,NomDAT,	mmr,Def,Raf,	[]	[u]
wahmin	Not exist,	[facl]	GFe			Derived Noun,NomDAFP,NomDAT,	mmr,Ind,KaS,	[]	[in]
manna	Not exist,	[facl]	GFe			Derived Noun,NomDAFP,NomDAT,	mmr,Def,NaS,	[]	[a]
manni	Not exist,	[facl]	GFe			Derived Noun,NomDAFP,NomDAT,	mmr,Def,KaS,	[]	[i]
wirA~ata	Not exist,	[ficAl]	GFe			Derived Noun,NomDAFP,NomDAT,	maS,Def,NaS,	[]	[a, at, af]
wirA~ati	Not exist,	[ficAl]	GFe			Derived Noun,NomDAFP,NomDAT,	maS,Def,KaS,	[]	[i, at, af]
wirA~atin	Not exist,	[ficAl]	GFe			Derived Noun,NomDAFP,NomDAT,	maS,Ind,KaS,	[]	[in, at, af]
kitAbin	Not exist,	[ficAl, ficAl]	GFe,GMa		,NSg,NPl	Derived Noun,NomDAT,acceptel,NomDAFP,accepteSC,NomDAD,	maS,jtS,Ind,KaS,	[]	[in]
kitAba	Not exist,	[ficAl, ficAl]	GFe,GMa		,NSg,NPl	Derived Noun,NomDAT,acceptel,NomDAFP,accepteSC,NomDAD,	maS,jtS,Def,NaS,	[]	[a]
kuttAbin	Not exist,	[fuccAl]	GMa		,NPl,NSg	Derived Noun,accepteSC,acceptel,	Sif,efc,Smb,Ind,KaS,	[]	[in]

Figure 13: A morphological analysis of some nouns using our system.

representing the morphological knowledge and the notion of the morphological automaton for Arabic language, our morphological analyzer gives more information about each word analyzed and more precision compared to Xerox Arabic morphological analyzer and Smrz's analyzer. To clarify this point, let us take some Arabic words and try to analyze them using the three morphological analyzers.

Related to Tables 1, 2, and 3 which represented the results of ten different Arabic words analyzed using the three morphological analyzers, we note that our morphological analyzer provides more information and more precision about the word analyzed compared to the others. Thanks to our new innovative language (XMODEL) which permits to represent the morphological knowledge in an optimal way and the power of the morphological automaton for Arabic, this advantage will be very useful especially in the future works which will be done later. It should be noted that our system could provide more information about the word analyzed according to the user needs.

On the other hands, let us see the evaluation process from another view. So, we have selected a corpus of 975 words containing different type of the word in Arabic (verbs, nouns, and particles). Then, we tested them on each morphological analyzer, after that, we draw a detailed analysis for the three analyzers. Our corpus contains 975 words divided into 481 nouns, 362 verbs, and 132 particles. Table 4 shows the number of words which are not found when they are analyzed using the three morphological analyzers.

To conclude this part of evaluation, using a new innovative language (XMODEL) and the notion of morphological automaton, our morphological analyzer can reach an average of performance around 90% which will make it one of the best existing morphological analyzers for the Arabic language, and it will be very useful for the next future

Table 1: Words analyzed using Xerox Arabic morphological analyzer.

The word	Morphological analysis using Xerox Arabic morphological analyzer
صِفْرٌ [Sifrun]	CiCoC Noun + N Indef Nom
خَارِجُونَ [xArijUna]	CACiC participle Active + U3na Masc Plur Nom
مُرْتَدِّي [murtaddI]	muCtaCaC Participle Passive + I3 Ma Plur Acc/Gen Possessive
فُصِلْتُ [fuSiltu]	+tu 1stPer Masc/Fem Sing CuCiC Verb
أُخْرِجْتُمَا [euxrijtumA]	uCoCiC Verb + tumA 2ndPer Masc/Fem Dual
مَعَ [maca]	maEa Funcwa
أَمَامَ [eamAma]	CaCAC Noun + a Def Acc
العَاشِرَ [ealcA^ira]	al Article CACiC Noun + a Def Acc
بِهِمَا [bihimA]	bi + himA Funcwa
يُجَادِلُونَ [yujAdilUna]	yu Imperfect Prefix CACiC Verb + Una Indicative 3rdPer Masc Plur

Table 2: Words analyzed using ElixirFM.

The word	Morphological analysis using ElixirFM
صِفْرٌ [Sifrun]	FiCL-un noun, singular, nominative, indefinite
خَارِجُونَ [xArijUna]	FāCiL-ūna adjective, masculine, plural, nominative, indefinite
مُرْتَدِّي [murtaddI]	muFtaCL-ī noun, plural, genitive, reduced/construct
فُصِلْتُ [fuSiltu]	FuCiL-tu perfective verb, passive, first person, singular
أُخْرِجْتُمَا [euxrijtumA]	uFCiL-tumā perfective verb, passive, second person, dual
مَعَ [maca]	inflected preposition, accusative
أَمَامَ [eamAma]	FaCāL-a inflected preposition, accusative
العَاشِرَ [ealcA^ira]	al-FāCiL-a adjective, masculine, singular, accusative, definite
بِهِمَا [bihimA]	This word is divided into “bi” and “himA”
يُجَادِلُونَ [yujAdilUna]	yu-FāCiL-ūna imperfective verb, indicative, active, third person, masculine, plural

Table 3: Words analyzed using our morphological analyzer.

The word	Morphological analysis using our Arabic morphological analyzer
صِفْرٌ [Sifrun]	Gma Particular Noun V0 Ind Raf [un]
خَارِجُونَ [xArijUna]	facala facila facula fAcil Gma NPl Derived Noun accepteSC accepteI efc Raf [Una]
مُرْتَدِّي [murtaddI]	eifcalla mufcallI Gma Pr1 NDl NSg Derived Noun accepteSC accepteI emf mmi8 KaS [I]
فُصِلْتُ [fuSiltu]	facula facala facila Gfe Gma Pr1 NSg Strong Verb MAD PAS [tu]
أُخْرِجْتُمَا [euxrijtumA]	eafcala Gfe Gma Pr2 NDl Strong Verb MAD PAS [tumA]
مَعَ [maca]	Particle accepteI zam mak Def NaS [a]
أَمَامَ [eamAma]	Particle accepteI mak Def NaS [a]
العَاشِرَ [ealcA^ira]	Particular Noun V10 Def NaS [eal] [a]
بِهِمَا [bihimA]	Gfe Gma Pr3 NPl KaS [bi] [himA]
يُجَادِلُونَ [yujAdilUna]	fAcala Gma Pr3 NPl Strong Verb MOD ACT Raf [y] [Una]

Table 4: Results of the evaluation process.

Type of the word	The number	Xerox morphological analyzer	ElixirFM	Our system
Nouns	481	39	76	21
Verbs	362	16	21	—
Particles	132	29	55	—
Total	975	84	152	21

Table 5: The acronyms signification in morphological analysis.

The Acronym	Signification
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 "الماضى"
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"

works to be done in NLP. We note that an update of our morphological database could resolve these errors seen in Table 4. Represented as a set of XMODEL files, the process of updating the morphological database becomes very easy which make our innovative language one of the best languages to represent the Arabic morphological knowledge.

Table 6: The english translation of some Arabic words.

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
الماضى	ealmADI	The Perfect
الأمر	Ealeamr	The Imperative
صِفْرٌ	Sifrun	Zero
مَعَ	Maca	With
أَمَامَ	eamAma	In front
العَاشِرَ	ealcA^ira	The tenth

6. Discussion

To compare our morphological analyzer for the Arabic language to the other existing systems, the task is difficult to do, because there is no standard to make this comparison, and every system has its own target. For this reason, each analyzer has some advantages and disadvantages compared to the others.

Our morphological analyzer for the Arabic language has some advantages compared to the others analyzers. These advantages are the following.

(i) Our morphological analyzer can be used in both analysis and generation.

(ii) It handles diacritized texts which permit to reduce the rate of ambiguity.


Lexical Table of File : arab files\dfa01.txt

A	B	C	D	E	F	G	H	I	J
Morpho Cmp	Original Scheme	Scheme	Gender	Person	Number	Properties	Morphological	Prefixes	Suffixes
faqaltum	[facala],	[]	GMa	,Pr2	,NPl	Strong Verb,MAD,ACT,		[]	[tum]
babiltu	[facila],	[]	GFe,GMa	,Pr1	,NSg	Strong Verb,MAD,ACT,		[]	[tu]
tatalta!ti	[tafaCiala],	[]	GFe	,Pr2	,NSg	Strong Verb,MAD,ACT,		[]	[ti]
wawiltumA	[facila, wacila],	[]	GFe,GMa	,Pr2	,NDl	Weak Verb,ACT,MAD,		[]	[tumA]
yasila	[wacala, wacila, wacula],	[]	GMa	,Pr3	,NSg	Weak Verb,ACT,MOD,	DefNaS,	[y]	[a]

Figure 14: Some invalid words analyzed by our system.

(iii) Our new and innovative language (XMODEL) used for the representation of the morphological knowledge and the use of morphological automaton for the Arabic language permit to avoid a huge problems of ambiguity in the Arabic language which the most analyzers cannot resolve.

(iv) The use of XMODEL language permit to reduce the number of the entries in the morphological database which present a big problem of the other morphological analyzers.

(v) Represented as a set of XMODEL files, the process of updating the Arabic morphological database is very easy to develop. This advantage makes our system very flexible and one of the best existing morphological analyzers.

(vi) The major advantage of our system is that it permits, on the one hand, to give the affixes, the stem and the scheme (if the word is a noun or a verb and if it has a scheme) for any word given. On the other hands, it gives the information about the word analyzed using a list of morphological descriptors which permits to characterize every Arabic word.

Our system has some disadvantages compared to some other systems. Firstly, it cannot handle undiacritized texts. Secondly, it does not provide an English glossary, and finally, it handles words which do not exist in the Arabic language. To clarify the last disadvantage, let us take an example of some invalid words analyzed with our system (Figure 14).

The example above shows five invalid words analyzed as they are valid words in the Arabic language. So, the next works to be done is to solve this problem. To solve it, we will keep the current design of morphological analysis. Our idea is to use a lexicon of any existing morphological analyzer (e.g., Buckwalter's analyzer, Xerox analyzer, etc.) and try to eliminate the invalid words generated by the morphological automaton and reconstruct it without these errors. This operation will reduce the number of the invalid words. But, this kind of problems is not very serious to take into consideration, because Arabic Language is very rich in words, and every year, there is some news words added to the language.

7. Conclusion and Future Works

The production of a morphological database is the first step in morphological analysis and most natural language processing. In this work, we presented an approach for representing the lexicon resource for Arabic language constructed using a flexible and extensible language (XMODEL). This morphological database used to develop a morphological analyzer for Arabic language. The strength of our morphological analyzer is the portability and the reusability because we have used Java for the development and the XML technology.

To extend our platform, we can also think to develop some works in the future such as the following.

(i) Develop the syntactic analyzer: our morphological database holds a lot of syntactic information, which makes it very useful for syntactic analysis.

(ii) Develop the semantic analyzer: our morphological database holds semantic information, which makes it very useful in semantic analysis.

(iii) Include the undiacritized texts.

(iv) Develop a system for Arabic learning.

(v) The help of the correction and the generation of texts.

(vi) Automatic understanding of the texts: classification of texts, automatic summary, and automatic extraction of the key words.

(vii) Realize some specific applications such as machine translation, *Q/R* systems, information retrieval systems; that is, these applications need morphological database and morphological analysis to be developed.

The next step is to use this morphological database and morphological analyzer to develop a syntactic analyzer for Arabic language using especially the syntactic information given by our system.

Appendix

See Tables 5 and 6.

References

[1] M. Attia, "An ambiguity-controlled morphological analyzer for modern standard Arabic modelling finite state networks," in *The Challenge of Arabic for NLP/MT Conference*, The British Computer Society, London, UK, 2006.

[2] T. A. El-Sadany and M. A. Hashish, "An Arabic morphological system," *IBM System Journal*, vol. 28, no. 4, pp. 600–612, 1989.

[3] E. Atwell, L. Al-Sulaiti, S. Al-Osaimi, and B. Abu Shawar, "A review of Arabic corpus analysis tools," in *Proceedings of the 11eme Conference Annuelle sur le Traitement Automatique des Langues Naturelles (TALN '04)*, pp. 229–234, Fez, Morocco, May 2004.

[4] K. R. Beesley, "Arabic finite-state morphological analysis and generation," in *Proceedings of the 16th Conference on Computational Linguistics*, vol. 1, pp. 89–94, Association for Computational Linguistics, Copenhagen, Denmark, 1996.

[5] K. R. Beesley, "Finite-state non-concatenative morphotactics," in *Proceedings of the 5th Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON '00)*, pp. 1–12, Luxembourg, Luxembourg, August 2000.

[6] K. Darwish, "Building a shallow morphological analyzer in one day," in *Proceedings of the Workshop on Computational Approaches to Semitic Languages in the 40th Annual Meeting of the Association for Computational Linguistics (ACL '02)*, pp. 47–54, Philadelphia, Pa, USA, 2002.

[7] M. Forsberg and A. Ranta, "Functional morphology," in *Proceedings of the 9th ACM SIGPLAN International Conference of Functional Programming (ICFP'04)*, pp. 213–223, Snowbird, Utah, USA, September 2004.

[8] T. Buckwalter, "Buckwalter Arabic morphological analyzer version 1.0. linguistic data consortium," University of Pennsylvania, LDC Catalog No.: LDC2002L49, 2002.

[9] O. Smrž, "ElixirFM—implementation of functional Arabic morphology," in *Proceedings of the Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources (ACL '07)*, pp. 1–8, Prague, Czech Republic, 2007.

[10] M. Attia, "Developing a robust Arabic morphological transducer using finite state technology," in *Proceedings of the 8th Annual CLUK Research Colloquium*, Manchester, UK, 2005.

[11] K. Shaalan, "Extending prolog for better natural language analysis," in *Proceedings of the 1st Conference on Language Engineering, Egyptian Society of Language Engineering (ELSE '98)*, pp. 225–236, Cairo, Egypt, March 1998.

[12] F. Baccar, A. Khemakhem, B. Gargouri, K. Haddar, and A. Ben Hamadou, "Modélisation normalisée LMF des dictionnaires électroniques éditoriaux de l'arabe," in *Proceedings of the 15eme Conference Annuelle sur le Traitement Automatique des Langues Naturelles (TALN '08)*, Avignon, France, June 2008.

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

[14] A. Khemakhem, B. Gargouri, A. Abdelwahed, and G. Francopoulo, "Modelisation des paradigmes de flexion des verbes arabes selon la norme LMF—ISO 24613," Traitement Automatique des Langues Naturelles, 2007.

[15] Y. Tahir, N. Chenfour, and M. Harti, "Modélisation à objets d'une base de données morphologique pour la langue arabe," in *Proceedings of the 11eme Conference Annuelle sur le Traitement Automatique des Langues Naturelles (TALN '04)*, Fez, Morocco, May 2004.