A lexicon of Arabic verbs constructed on the basis of Semitic taxonomy and using finite-state transducers

Alexis Amid Neme

Laboratoire d'informatique Gaspard-Monge – LIGM
Université Paris-Est, 77454 Marne-la-Vallée Cedex 2, France.

http://infolingu.univ-mlv.fr

E-mail: alexis.neme@gmail.com

Abstract

We describe a lexicon of Arabic verbs constructed on the basis of Semitic patterns and used in a resource-based method of morphological annotation of written Arabic text. The annotated output is a graph of morphemes with accurate linguistic information. An enhanced FST implementation for Semitic languages was created. This system is adapted also for generating inflected forms. The language resources can be easily updated. The lexicon is constituted of 15 400 verbal entries.

We propose an inflectional taxonomy that increases the lexicon readability and maintainability for Arabic speakers and linguists. Traditional grammar defines inflectional verbal classes by using verbal pattern-classes and root-classes, related to the nature of each of the triliteral root-consonants. Verbal pattern-classes are clearly defined but root-classes are complex. In our taxonomy, traditional pattern-classes are reused and root-classes are simply redefined.

Our taxonomy provides a straightforward encoding scheme for inflectional variations and orthographic adjustments due to assimilation and agglutination. We have tested and evaluated our resource against 10 000 diacriticized verb occurrences in the Nemlar corpus and compared it to Buckwalter resources. The lexical coverage is 99.9 % and a laptop needs two minutes in order to generate and compress the inflected lexicon of 2.5 million forms into 4 Megabytes.

1. Introduction

Arabic morphology can be described by many formal representations. However, Semitic morphology or *root-and-pattern* morphology (Kiraz, 2004) is a natural representation for Arabic [1]. The *root* represents a morphemic abstraction, usually for a verb a sequence of three consonants, like *ktb*. A *pattern is* a template of characters surrounding the root consonants, and in which the slots for the root consonants are shown by indices. The combination of a root with a pattern produces a surface form. For example, *kataba* and *yakotubu* are represented by the root *ktb* and the patterns *1a2a3a* or *ya1o2u3u*.

Root-and-pattern morphology is standard in Arabic and is learned in grammar text books. Arabic linguists use *root-and-pattern* representation in order to list verbal entries and related inflected forms. On the other hand, FSTs have shown their simplicity and efficiency in inflectional morphology for western languages. Computer scientists appoint FSTs as standard devices for inflection. Various formal representations for Arabic morphology have been created by computer scientists to avoid root-and-pattern representation. The point that motivated this trend is that FSTs formalism would not be fitted for Semitic morphology since FSTs are concatenative whereas Semitic morphology is not. In concatenative representation, the root-and-pattern representation is replaced by a stem- or lexeme-based representation. For these formalisms, a stem is a basic morpheme that undergoes affixations with other morphemes in order to form larger morphological or syntactic units. For root-and pattern morphology, a stem derives from a root and a particular pattern and subsequently undergoes affixations.

At the operational level, the lexical representation of the concatenative model is entirely concatenative in order to compel with the *[prefix][stem][suffix]* representation. However, these representations imply a manual stem precompilation based on a root-and-pattern representation. The concatenative models are generally composed of three components: lexicon, rewrite rules, and morphotactics. The lexicon consists of multiple sublexica, generally *prefix*, *stem*, and *suffix*. The rewrite rules map the multiple lexical representations to a surface representation. The morphotactics component aims with a subjacent representation to generate or to parse the surface form *[prefix][stem][suffix]* and performs alternation rules at morpheme boundaries such as deletion, epenthesis, and assimilation.

Any formal representation that is not adapted to Semitic morphology will be rejected by the majority of Arabic-speaking linguists. When linguists work in a newly created formalism, they continue to work with *root-and-pattern* representation on paper and subsequently, they unfold their descriptions for a specific formalism. Their contribution for updating and correcting lexical resources is complex and time-consuming, and therefore error-prone.

Our approach resorts to classical techniques of lexicon compression and lookup in an inflected full-form dictionary that includes orthographic variations related to morpheme agglutination. The formalization of all possible verbal tokens requires complex and interdependent rules. For these issues, we define a taxonomy for Arabic verbs composed of 460 inflectional

1 We would like to thank Eric Laporte and Sébastien Paumier for helpful discussions, contributions and for the adaptation of Unitex to Arabic. Unitex is an open source multilingual corpus processor. More than 12 European languages, Korean and Thai with their linguistic resources are operational in Unitex. http://www-igm.univ-mlv.fr/~unitex

classes. We demonstrate that FSTs are compatible with root-and-pattern representation. Our taxonomy encodes simultaneously in the lexical representation three variations at the surface level:
- inflectional classes of a lemma;
- inflectional subclasses related to morphophonemic assimilation;
- orthographic adjustments related to the agglutination of a pronoun.

In our orthographic representation, we use a fully diacriticized lexicon and we take advantage of the clear boundary, already defined in traditional grammar, between verbal inflection and verbal agglutination to describe these two levels independently. In order to satisfy both computer scientists and Arabic linguists, we have created in Unitex an enhanced version of FSTs adapted to root-and-pattern representation.

In Section 2, we outline the state-of-the-art approaches to Arabic morphological annotation. Section 3 describes the methodology and particularly the inflectional verbal taxonomy. Section 4 describes agglutination as morpheme combinatorics. Section 5 reports the construction of the lexicon. Section 6 reports the evaluation of the lexicon. A conclusion and perspectives are presented in Section 7.

2. State of the Art

Several morphological annotators of Arabic are available. The Buckwalter Arabic Morphological Analyzer (BAMA) is one of the best Arabic morphological analysers 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).

The BAMA has three components: the lexicon subdivided in A, B, C sublexica, the compatibility tables (AB, BC, AC) and the analysis engine. An Arabic word is viewed as a concatenation of three regions, a prefix region (A), a stem region (B) and a suffix region (C). The prefix and suffix regions can be null. An entry in A may be the concatenation of proclitics and an inflectional prefix. An entry in C may be the concatenation of an inflectional suffix and an enclitic. The A and C lexica are composed of 561 and 989 entries which represent all possible combinations of inflectional and agglutinative morphemes for nouns and verbs. For each stem in B, a morphological compatibility category, an English gloss and part-of-speech (POS) data are specified. A list of stems is assigned to a lemma, and the lemma is not used in the analysis process. The B lexicon is composed of 82 000 stems which represent nearly 40 000 lemmas. Verbal stems are 33393[2] and represent 8709 verbal lemmas. A full ABC form must be allowed by the three compatibility tables AB, BC, AC.

2 Verbal stems are for perfect active (17008) stems, imperfect active (13241), perfect passive (403), imperfect passive (2611), and for imperative 130 stems. BAMA resource does not include all imperfect active stems, for instance.

qr>	qara>	PV->	qara>/VERB_PERFECT
qr|	qara|	PV-|	qara|/VERB_PERFECT
qr&	qara&	PV_w	qara&/VERB_PERFECT
qr>	qora>	IV	qora>/VERB_IMPERFECT
qr>	qora>	IV_wn	qora>/VERB_IMPERFECT
qr|	qora|	IV-|	qora|/VERB_IMPERFECT
qr&	qora&	IV_wn	qora&/VERB_IMPERFECT
qr}	qora}	IV_yn	qora}/VERB_IMPERFECT
qr>	qora>	IV_Pass	yuqora>/VERB_IMPERFECT

Table 1. BAMA stem lexicon using Buckwalter transliteration. A list of stems related to the lemma-identifier qara>-a_1 "to read". The 9 stems are related to the orthographic variants of the 3rd root consonant, here glottal stop (*hamza*), depending on the next inflectional suffix and the existence of an agglutinated pronoun.

The Buckwalter representation for the Arabic lexicon is not fitted for generation but only for text analysis. In ElixirFM (http://elixir-fm.sourceforge.net/), Smrz (2007) adapted the Buckwalter resources for generation and the project is implemented in Haskell, a functional programming language. In the ALMORGEANA project, Habash (2004) proposed also a version of Buckwalter resources adapted to generation and analysis. Below an example *lilkutubi* "books" :

lilkutubi⇔ [kitAb-1 POS: N l+ Al+ +PL +GEN]
li_l_kutub-i⇔
[lemma-ID NOUN PREP+DET+ (plustem) + Genitive]

Although the lexicon is an open linguistic resource, the procedure for updating it is complex. For instance, adding a new verb is an intricate operation. First, the A and C lexica are composed of 561 and 989 entries. Although the two disjoint sets of inflectional and agglutination suffix morphemes are clearly defined in Arabic, the *[prefixes] [stem][suffixes]* representation does not allow two suffix subsets to be defined. Second, the stem lexicon entries corresponding to a lemma are numerous and need to be subcategorized. In other words, a lemma is unfolded into many stems, and one uses a cumbersome subcategorization which mixes up inflectional and agglutinative features of verb stems in order to match with 3 compatibility tables, composed respectively of 2050, 1660, 1200 entries. Such composite data are complex and not transparent for Arabic linguists.
Mesfar (2008) adopts a "lemma-based lexicon" and FSTs for inflection. The project claims 10 000 verb lemmas. The framework is similar to ours since it resorts to classical techniques of lexicon compression and lookup in a full list of inflected -forms. The project does not use root-and-pattern representation. As far as we know, no figures on testing and evaluating the systems are available. The lemma lexicon is wordy such as the extract of the lexicon from Mesfar (2008):

```
ضَرَبَ,V+Tr+FLX=Vdaraba1+DRV=N_daraba1:FlxDRV+DRV=A_daraba1:FlxDRV
# le verbe "ذَكَرَ" et "كَتَبَ" se conjuguent et se dérivent selon les même modèles
ذَكَرَ,V+Tr+FLX=Vdakara2+DRV=N_dakara2:FlxDRV+DRV=A_dakara2:FlxDRV
كَتَبَ,V+Tr+FLX=Vdakara2+DRV=N_dakara2:FlxDRV+DRV=A_dakara2:FlxDRV.
```

FST are difficult to read and maintain (Mesfar, 2006, page 3):

> " أَلَّمَ ", V+Tr+FLX [8] = V_kallama (kallama – *to speak with someone*)
> Among the 122 inflectional transformations which are described in the flexional paradigm "V_kallama", here is one: (<LW> يُ <R4><S> <R><S> /ُ A+P+3+m+s). This NooJ transformation means: position the cursor (|) at the beginning of the form(<LW>) (|kallama), insert "يُ" (yu) into the head of the form (yu|kallama), skip four letters (<R4>) (yukall|ama), erase a letter (<S>) (yukall|ma), insert the vowel "ِ " (i) (yukalli|ma), skip a letter (<R>) (yukallim|a), delete of the following letter (<S>) (yukallim|)and finally insert the final vowel "ُ " (u) (yukallimu|)."

For their morpho-phonological system and in addition to concatenative rules, Carnegie Melon Univ. uses transformational rules to describe alternation of root letters (Cavalli-Sforza, et al., 2005). As far as we know, no figures on lexical coverage or evaluation are available.

The SARF project (Al-Bawab et al., 1994, http://sourceforge.net/projects/sarf/) is based on root-and-pattern representation. Starting from three- and four-consonant roots, it can generate Arabic verbs, derivative nouns, and gerunds, and inflect them. It has over 20 000 verb lemmas. The project uses conventional programming techniques with the Java language and roots encoded in XML files. It uses transformational rules in order to handle alternation of root letters in the Java programs. The patterns are hard-coded in the form of Java code. This work has the advantage of being clearly built on a strong linguistic basis that is the standard morphology in Arabic. However, it neither includes the use of a test collection nor reports a success rate; in addition, updating and correcting a language resource included in source code is complex since it involves two expertises: an Arabic linguist and a programmer; updating data and updating source code obey to different professional practices.

At Université de Lyon 2, the DIINAR project (Dichy & Ferghali, 2004) was developed for terminological and translation purposes. DIINAR.1 includes a total number of 119,693 lemmas, fully vowelled, among which 19,457 verb lemmas. A conventional programming framework and databases are used for generation and analysis with a lemma-based lexicon encoded according to this framework. As far as we know, no figures on testing and evaluating the system for morphological annotation are available.

For a complete survey of morphological parsers, readers should consider Al-Sughaiyer & Al-Kharashi (2004) and Habash (2010).

3. Method of description

3.1 A taxonomy for verb inflection

Our method is based on a precompiled diacriticized full-form dictionary with all possible inflected forms and their orthographic variations due to morphophonemic alternations. We exclude from this inflectional representation agglutinated prefixes and suffixes such as conjunctions and pronouns. We associate morphosyntactic feature values to each entry in the generated list of 2.43 million surface forms. In order to obtain this list, we provide a list of lemmas manually associated to codes defined by a taxonomy, each code representing a transducer. The full-form list is produced after inflecting each lemma by applying the encoded transducer (Silberztein, 1998).

Arabic and other Semitic languages have long been described in terms of a *root* interwoven with a *pattern*. The root is a sequence of consonants. Each Arabic verb contains 3 or 4 consonants that remain generally unchanged in all conjugated forms and make up the consonantal root; all the remaining information on a conjugated form is called 'pattern'. For example, *yakotubuwna = [ktb & ya1o2u3uwna]* is obtained through the interdigitation of the root *ktb* with the pattern of active-Perfect-3person-masculine-plural-indicative *ya1o2u3uwna*. Below some precisions:

- Some root consonants change. They are the glottal stop, noted *h* in the taxonomy, and glides, noted *w*, *y*; those that never change are written in patterns in the form of their position 1, 2, 3 or 4.
- At the surface level, the orthographic representation of glottal stop and glides can change. The glottal stop is represented by six allographs depending on the context. At phonological level, the glides become short vowels */ i, u/* or long vowels */a:*, *i:*, *u:/* or are omitted and transcripted as *zero-vowel*, *o*[3] (see also footnote 4).
- A pattern indicates the position of its letters relative to the root consonants. Generally, these letters are vowels and/or affixes related to derived verb form such as *IisotakotabuwA = [ktb & Iisota1o2a3uwA]*. The surface form may also be subdivided in *[prefix] [stem] [sufix]*. The *stem pattern* formalizes all infixation operations such as *kotub = [ktb & 1o2u3]*. Inflectional prefixes and suffixes can be concatenated subsequently to the stem form y*akotubuwna = [ya] [ktb & 1o2u3] [uwna]*.
- The third root consonant can be identical to the second one. In the root, it is represented by a gemination mark *G*, and in the pattern, by *2*, such as *madadota = [mdG & 1a2a2ota]*.
- By convention, the perfect-3rd person-masculine -singular is the form used as lemma. The corresponding pattern is called the canonical pattern. All patterns are defined in function of the canonical pattern.

Verbal pattern classes are clearly defined in Arabic grammar but root-classes are intricate and involve a complex terminology. Root-classes are defined according to the nature of some of the root consonants: regular, weak, geminated, with glottal stop, and to their position 1, 2, 3 or 4. In this terminology, *qaAla/yaquwlu* قال "say" is a *hollow verb of w kind*, with a weak consonant *w* at the second position; whereas *baAEa/yabiyEu* باع "sell" is a *hollow verb of y kind*. Moreover, two or three special values of the root consonants can appear at the same time. A verb like *OataY/yaOotiy* أتى "arrive" has a glottal stop at the first position and a weak consonant *y* at the third position. A classification with nature/position criteria and each with 4 sub-criteria yields to an intricate terminology and is not

3 The zero-vowel marks the absence of vowel between two consonants.

consensual in Arabic grammar.
Our classification is bi-dimensional like the traditional one and based on the traditional pattern-classes which are reused and root-classes which are redefined more simply. Traditional grammar defines an inflectional verbal class by a pattern-class and a root-class. Triliteral verbs are compatible with 16 possible canonical patterns and quadrilateral verbs with 4 canonical patterns. Our classification defines 31 root-classes. The root classes are defined according to the nature of the root consonants. The special values for the consonants are *w, y* and the glottal stop (h). An irregular root is a root with at least one special value in its consonants. The inflected forms of a verb are easily predictable on the basis of the features of the root. We revisited and simplified, with no loss of information, the root-based traditional classification by using three consonantic slots, noted *123*, except for special values: glottal stop (*h*), *w, y,* for each slot; and when the 3rd root consonant is identical to the 2nd, the slots are noted *122*. Thereby, the lemma *ktb* will be encoded *$V3au-123* where:

$ is the Semitic mode for FST which means the root consonants interdigitate into the pattern: *[ktb & ya1o2u3u] = yakotubu;*
V is the verbal POS;
3au is the class of triliteral verbs used with the patterns *1a2a3/ya1o2u3* for perfect/ imperfect;
123 is the class of roots in which no slot is occupied by a special value.

Each root/canonical-pattern pair corresponds to a lemma. This representation seems well-founded and also well-established in Arabic morphology. Above all, it is ubiquitous in the Arabic-speaking world. Below, some examples from the lexicon:

```
/Lemma,encoding/ canonical-patt. Special values
-----------------------------------------------
/ simple forms
نقش,$V3au-123 / 1a2a3a/ya1o2u3u no special values
جرّ,$V3au-122 /    third root identical to second
عاد,$V3au-1w3 /      with waw as a second root
غفا,$V3au-12w /      with waw as a third root
فتح,$V3aa-123/ 1a2a3a/ya1o2a3u
لمز,$V3ai-123 / 1a2a3a/ya1o2i1u
حاك,$V3ai-1y3 /   with yeh as a second root
سرى,$V3ai-12y /   with yeh as a third root
أوى,$V3ai-hwy /   with hamza, waw and yeh
علم,$V3ia-123 / 1a2i3a/ya1o2a3u
وطئ,$V3ia-w2h /   waw and hamza as 1rst and 3rd
كرُم,$V3uu-123 / 1a2u3a/ya1o2u3u
حسب,$V3ii-123 / 1a2i3a/ya1o2i3u
/ Derived forms
أقبل,$V61-123  / Aa1o2a3a
دشّن,$V62-123   / 1a2Ga3a
داهم,$V63-123  / 1aA2a3a
إنشغل,$V64-123 / Iino1a2a3a
إنطلى,$V64-12y /   with yeh as a third root
إختنق,$V65-123 / Ii1ota2a3a
إزهرّ,$V66-123  / Ii1o2a3Ga
تهاجن,$V67-123 / ta1aA2a3a
تآكل,$V67-h23  /   with  hamza as a first root
تحدّد,$V68-122  / ta1a2Ga2a with identical 3rd root
تلكّأ,$V68-12h  /   with hamza as a third root
إستبسل,$V69-123 / Iisota12a3a
اعشوشب,$V70-123 / Ii1o2aw2a3a
/ Quadriliteral roots
بعثر,$V40-1234  / 1a2o3a4a a quadriliteral root
طمأن,$V40-12h4  /   with hamza as a third root
دمدم,$V40-1212  /   a geminated quadriliteral root
تبعثر,$V41-1234 / ta1a2o3a4a
تلألأ,$V41-1h1h  /    a geminated root with 2 hamzas
```

Below, some of the 31 possible combinations of root-classes related to class-pattern V3ia. Some root-classes are empty which means that there is no verb with such root-classes for class-pattern V3ia:

```
/Lemma,encoding/  /lemma-transliteration
علم ,$V3ia-123   /Elm
ظل  ,$V3ia-122   /ZlG
أمَ  ,$V3ia-h22   /OmG
ألف ,$V3ia-h23   /Olf
رئف ,$V3ia-1h3   /ref
ظمئ ,$V3ia-12h   /Zme
//First weak root consonant
ودَ  ,$V3ia-w22   /wdG
    ,$V3ia-wh3
وطىء,$V3ia-w2h   /wTe
وجع ,$V3ia-w23   /wjE
    ,$V3ia-y22
يئس ,$V3ia-yh3   /yes
يقظ ,$V3ia-y23   /yqZ
```

The format of the lexicon is a list of lemma entries. In our format, the string before comma transcribes plain letters and the gemination mark but no short vowel diacritics. The pattern includes the encoding of short vowels (*a, i, u).* This transcript choice is consistent with usual practice in traditional paper dictionaries.
Our full-form lexicon is produced by FSTs. The FST output format is *surface-form,lemma.V:feature-values* such as *:*
كتب,تَـكْـتُـبُ.V:aI3fsN
/active-Imperfect-3rdpers-fem-sing-iNdicative

The *feature values* are :
- Voice: active (a), passive (b);
- Tense: Perfect, Imperfect, Imperative (Y);
- Person: 1, 2, 3;
- Gender: masculine, feminine;
- Number: singular, dual, plural;
- Mode: indicative (N), Subjunctive, Jussive, Energetic.

In the following two sub-sections, we present first inflectional transducers and then inflection-related orthographic adjustments.

3.2 The inflection transducers

An inflection transducer specifies the inflectional variations of a word. It is shared by the class of words that inflect in the same way. The input parts of the transducer encode the modifications that have to be applied to the canonical forms. The corresponding output parts contain the codes for the inflectional features. A transducer is represented by a graph and can include subgraphs. The transducers are displayed in Unitex style, i.e. input parts are displayed in the nodes, and output parts below the nodes.

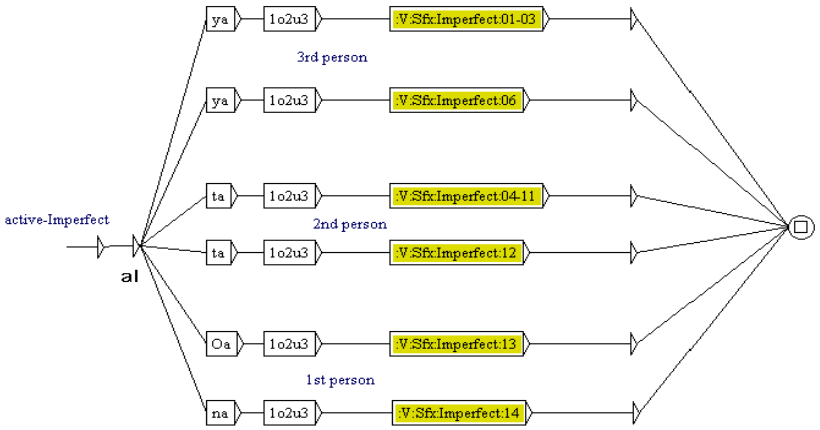

Fig 1. The active imperfect (aI) subgraph. Each path contains a prefix, a stem-pattern and a subgraph of suffixes. The Person-Gender-Number variations are numbered from 01 to 14.

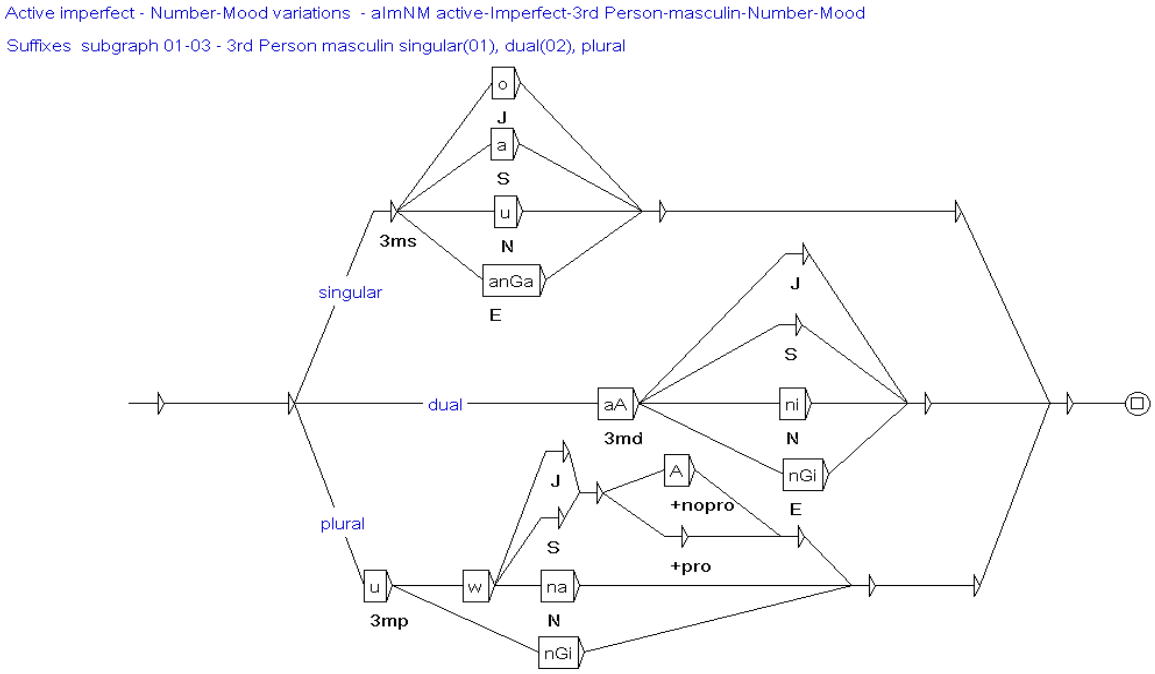

Fig 2. The 01-03 subgraph represents Number-Mode suffix variations for active Imperfect 3rd Person masculine, related to Person-Gender-Number-Mode variations.

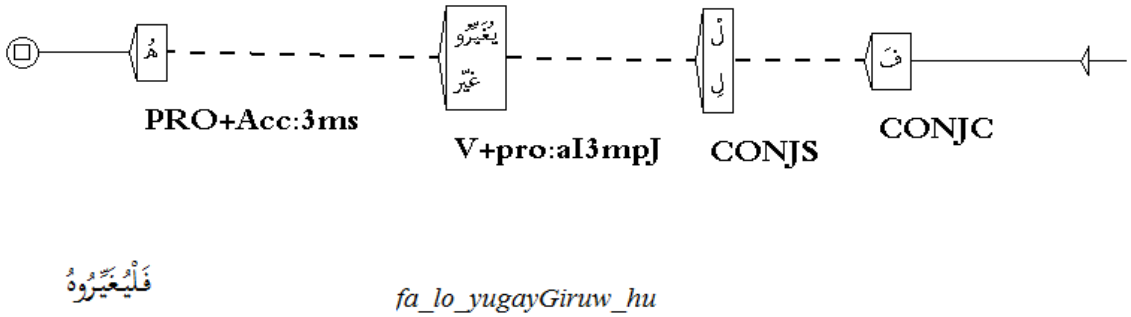

Fig 3. Text automaton as output of the application of a graph dictionary. Here a morphological analysis of *faloyugayGirohu (and_to_change-they_it)*. The morphological dictionary graph restricts the selection to V+pro agglutinated variant only. Dashed lines connect segments in the same token.

A Buckwalter transliteration is used as a standard to map Arabic characters into Latin ones. An XML version of this transliteration was created in order to handle this format. We create a modified version of the XML version where all special characters such as (', |, * , $, ~) are respectively replaced by (c, C, J, M, G) [4]. Many systems use special characters in a special way.
In order to generate the full-form dictionary, the following steps are accomplished.
- The lemma lexicon is transliterated.
- The FSTs are applied to the list and produces a transliterated full-form dictionary output.
- The output is transliterated into Arabic script.
So, both the lemma lexicon and the full-form dictionary are in Arabic script which is handier to read for Arabic linguists.

For example, the lexical entry `ktb,$V3au-123` is processed by the transducer named *V3au-123* in order to get all inflected forms. The main graph contains five subgraphs referring to the five voice-tense variations. In turn, each subgraph (Fig. 1) contains suffixes of Person, Gender, Number for the perfect and Person, Gender, Number, Mode for the Imperfect (Fig. 2).

3.3 Inflection-related adjustments

The inflectional taxonomy takes into account variations due to orthographic adjustment and morphophonemic assimilations. The phoneme involved in the variation is replaced by a gemination mark or by another phoneme. At morpheme boundaries between a stem and a suffix, the first letter *n* and *t* of the perfect suffix is changed to gemination mark like in *daxGan+naA => daxGanGaA* , "smoked-we"; *Oavobat + tu => OavobatGu* "demonstrated-I". Our taxonomy includes the inflectional classes Vpp-12n, Vpp-12t in order to take into account such phenomena. In our resource, we have counted 614 entries in Vpp-12n and 154 in Vpp-12t root-classes.

Due to morphophonemic variations, the *t* in the canonical pattern V65 or *Ii1ota2a3a* (اِفْتَعَلَ) has an orthographic variation depending on the value of the first root consonant. It is replaced by emphatic *T*, or *d*, or by gemination mark *G*. The subclasses V65T, V65d, V65G encode the *t* variation, we have counted: 46 entries with V65T-rrr such as `ISTfY,$V65T-12y` إصطفى; 31 entries with V65d-rrr such as `Izdwj,$V65d-1w3` إزدوج; and 114 entries with V65G-rrr such as `ItGbE,$V65G-123` إتّبع or `ItGSl,$V65G-w23` إتّصل.

4. Agglutination and omission of diacritic

4.1 Orthographic adjustments and agglutination

In Arabic, a token delimited by spaces or punctuation symbols is composed of a sequence of segments. Each segment in a token is a morpheme. In Unitex, this segmentation is formalized via a morphological dictionary graph. Such graphs introduce morphological analyses in the text automaton (Fig 3) where dashed lines connect segments.

The combination of a sequence of morphemes obeys a number of constraints. Checking these constraints is necessary to discard wrong segmentations. In Arabic, a verbal token is composed by one morpheme <*V*> or the concatenation of up to 4 morphemes such as:
<*CONJC*> <*CONJS*> <*V*> <*PRO+accusative*>
where <CONJC> is a coordinating conjunction, <CONJS> is a subordinating conjunction and <*PRO+accusative*> an agglutinated object pronoun.

<*CONJC*> combines freely with any inflected verb. The <*CONJS*> constraints the verb to the Imperfect Subjunctive or Jussive. Finally, an inflected verb form is often insensitive to the agglutinated pronoun but some forms are sensitive like forms with a glottal stop as the third root consonant.

The subgraph selects only V+pro variants from the full-form dictionary (cf. Fig 3). When followed by a pronoun, a verbal segment may have an orthographic adjustment. This is often the case when the verbal segment ends with a long */a:/ A*, its allograph *Y*, or a glottal stop which has 6 allographs depending on its position and the surrounding vowels. For verbs, the roots with a glottal stop as the third consonant change their graphemic representation. A suffix subgraph related to classes Vpp-rrh represents the orthographic variations of an ending glottal stop due to pronoun agglutination.

The generation of the agglutinable variants of an inflected verb is performed directly with a lexicon of words, which is another way to implement a rule. In fact, the dictionary graph links each morphological variant to the correct context, which also expresses a rule. The variants are generated during the compilation of the resources, not at analysis time as in rule-based systems in which a rule should compute each morphological variant at run time, then link each variant to the correct context. The advantage of our method is that it simplifies and speeds up the process of annotation.

4.2 Diacritics

Diacritics are often omitted in Arabic written text. According to our corpus study of 6930 tokens from Annahar newspaper, 209 tokens (3%) include at least a diacritic. 140 tokens (2 %) are with the *F* diacritic *(–an)* and 57 (1 %) are with gemination mark *G*, in which nearly 0.8 % is related to a verbal form. 9 are with the short vowel *u*. For the *u* diacritic, 7/9 involve a passive verbal form. For the gemination diacritic, 49/57 involve a verbal form and are the following.
- 41 to V62 refer to *1a2**G**a3a* derived form (فَعَّلَ).
- 5 to V68 refer to *ta1a2**G**a3a* derived form (تَفَعَّلَ).
- 2 to V65G refer to *Ii1**G**a2a3a* derived form (اِفْتَعَلَ).
- 1 to V3au refers to *ya1o2**u**lu* a triliteral simple form (فَعَل يَفعُل).

4 The Transliteration in Unitex Arabic <=> Latin: ء, c; ا, C; أ, O; ؤ, W; إ, I; ئ, e; ا, A; ب, B; ة, P; ت, T; ث, V; ج, J; ح, H; خ, X; د, d; ذ, J; ر, r; ز, z; س, s; ش, M; ص, S; ض, D; ط, T; ظ, Z; ع, E; غ, g; ف, f; ق, q; ك, k; ل, l; م, m; ن, n; ه, h; و, w; ى, Y; ي, y; ً, F; ّ , N; ْ , K; َ, a; ُ, u; ِ, i; ً, G; ْ, o;

Editors generally display diacritics for unusual forms such as passive verb forms. When some are displayed, they can avoid misinterpretations to the reader. For verbs, diacritics are the short vowels *(a, i, u)* or the gemination mark followed by a short vowel. Arabic verbs can include a sequence of two diacritics: the gemination mark followed by a short vowel. In the case of two diacritics, diacritics omission is not totally free. One can omit the two diacritics or the last diacritic but never the gemination mark alone.

Consequently, processing written Arabic text should take into account undiacriticized and partially diacriticized text. A lookup procedure in Unitex[5] has been adjusted to deal with omission of diacritics in Arabic. This procedure finds in the diacriticized full-form dictionary all possible diacriticized candidate forms compatible with a given undiacriticized or partially diacriticized form. When a diacritic is present in a surface form, the lookup procedure excludes the candidates in the lexicon which do not have that diacritic at the same position.

5. Some figures

Our lexicon is composed of 15 400 entries. Each entry is inflected into 144 surface forms and in average 158 forms if we include orthographic variations due to agglutination. The size of the full-form dictionary is 2.43 million surface forms. The size of the full-form dictionary in plain text is 132 Megabytes in Unicode little Endian and is compressed and minimized into 4 Megabytes which is loaded to memory for fast retrieval. The generation, compression and minimization of the full-form lexicon lasts two minutes[6] on a Windows laptop.
The number of main inflectional graphs is 460. Each main graph is composed of 5 subgraphs for voice-tense features variations, that is 2300 subgraphs. These subgraphs use also 540 suffix subgraphs related to person-gender-number-mode features. In all, the number of graphs and subgraphs is 3300 (460+2300+540), to be compared with nearly 100 graphs and subgraphs dedicated to the verbal inflection system for Brazilian Portuguese constructed also for Unitex (Muniz et al. 2005). A sample will be freely available from the time of the workshop.

We have noticed that many simple triliteral verbs may have orthographical variants related to the variation of the vowel after the second root consonant. However, these variations may correspond to meaning differences; therefore we should have different entries. In order to facilitate the encoding scheme, all orthographic variants of verbs are encoded in separate entries. In our lexicon, a verb may have several inflectional codes. These codes can correspond to different lexical items or to orthographic variants of the same item. In the future, we plan to encode different lemmas if the different inflectional behaviours are correlated to differences at other levels, e.g. semantic, which is the case of `Hsb,$V3au-123` "count", and `Hsb, $V3ii-123` "think". One should also encode a single lemma if the inflectional behaviours are a free variation, such as for `kfl,$V3au-123` and `kfl,$V3ai-123` "grant". Out of a total 4135 simple triliteral root in the lexicon, 1278 triliteral root have several inflection al codes.

Some inflectional classes are redundant such as V62-122, which is identical to V62-123, whereas V65-122 is different from V65-123. In order to make the encoding scheme easier to handle for Arabic linguists, we have duplicated the inflectional graph V62-122. The 122 root-class delimits two classes in nearly all other cases. We estimate such redundancy at 15%. We offer a simple encoding scheme with duplicated inflectional classes in order to make it unnecessary for Arabic linguists to memorize in which cases some features have to be marked.

6. Evaluation

We have chosen the NEMLAR Arabic Written Corpus (Attia et al., 2005), first to improve our lexicon of verbs, and then to constitute our test collection. The Nemlar data consists of about 500 thousand words of Arabic text from 13 different genres. The text is provided in 4 different versions: raw text, fully diacriticized text, text with Arabic lexical analysis, and text with Arabic POS-tags. The database was produced and annotated by RDI, Egypt, for the Nemlar Consortium.
The extraction of occurrences of verbs from "text with Arabic POS-tags" provided 50 000 occurrences of verbs. These occurrences were split in two disjoint parts: nearly 40 000 token occurrences (11050 token types) for correcting the resource and a test collection of 10 000 token occurrences (5222 token types) for testing it after the correction stage.
The test collection shows that 10 verbs lemmas were missing in our lexicon[7]. Hence, the fault rate of the resource is 0.1% in this corpus. Let us assume that a page is composed of 50 lines/page, 10 tokens/line, 1 verb/10 tokens. In other words, in 20 pages of real corpus, our resource fails to recognize 1 verb.
In order to compare our lexicon with the Buckwalter resource, we ran BAMA on the first 550 occurrences of verbs of the same test collection. 14 occurrences of verbs were unrecognized, which represents a 2.5 % error rate, i.e. 25 times the error rate of our resource. The unrecognized tokens involve: 10 missing passive stems, 2 imperative stems and 2 missing verb lemmas.
Morphosyntactic tagging is generally part of a pipeline of written text processing. In a common undiacriticized Arabic corpus, most verbs have two possible analyses, one as active and one as passive. The lack of passive stems in the Buckwalter resource leads to assign only the active tag to verbs, which can jeopardize a subsequent deep syntactic parsing of a sentence.
A fallback procedure in order to assign morphosyntactic

5 The lookup procedure was adjusted by Sébastien Paumier.

6 At Columbia University, MAGEAD Project constructs an Arabic resource according to Buckwalter's Prefixes-Stem-Suffixes representation. They describe an Arabic lexicon based on root-and-pattern representation and rules dedicated to orthographic variations due morphophonemic alternations; and other rules dedicated to orthographic adjustment due to agglutinations (Habash & Rambow, 2006). The program needs more than 15 hours to generate such resource (Owen Rambow, personal communication).

7 jzm,$V32-123; qrGZ,$V62-123; thrGb,$V68-123; rDb,$V33-123; kfl,$V34-123; tnAqM,$V67-123; sAb,$V32-1y3; zEq,$V33-123; DnG,$V32-1nn; tAh,$V32-1y3

features to unrecognized tokens is often included in a language processing pipeline. Since our fault rate is 0.1 %, it might be useless to construct a fallback procedure for unrecognized verbs when this resource is used.

7. A conclusion and perspectives

We elaborated a model for Arabic verbs with the following features. A detailed and simple taxonomy is based on Semitic morphology. Lemma-based verbs are used as entries in the lexicon. FSTs are used to produce inflected forms. Agglutination is described independently from inflection.Our experimentation shows that the method outperforms state-of-the-art systems of Arabic morphological annotation.
We made language resources the central point of the problem. All complex operations were integrated among resource management operations. The output of our system is accurate and informative; the language resources used by the system can be easily updated by an expert of Arabic independently from computational linguistics experts, which allows users to control the evolution of the accuracy of the system. Morphological annotation of Arabic text is performed directly with a lexicon of words and without morphological rules, which simplifies and speeds up the process. The undiacriticized, partially and fully diacriticized Arabic text can be annotated excluding incompatible analyses.

We reuse traditional Semitic patterns and we provide a clear scheme for root-class encoding by avoiding intricate terms. Root-and-pattern representation facilitates our task in encoding the lexicon since it is a standard but also it helps to debug our transducers quickly which is not the case of a rule-based system.
This work opens several perspectives. The resources can be extended by running the annotator and analysing output. Another perspective is to extend this methodology to inflection of noun and adjective, mainly to encode singular and the plural under the same lemma entry using Semitic patterns فَـعِـيل فُـعَـلاء. For example, the pair *raeiys*, *ruWasaAc* (رَئِيس رُؤَسَاء) "president" will be represented by one entry:

```
raeiys,$N3_1a2iy3-1u2a3Ac-1h3
nabiyl,$N3_1a2iy3-1u2a3Ac-123
```

where number *3* denotes a triliteral root; *1a2iy3-1u2a3aAc* is a pattern pair that represents singular-plural variations; and *1h3* (vs *123*) encode the glottal stop variations of the 2nd consonant root *(e =>W)*.

8. References

Al-Bawab, M., Mrayati, M., Alam, Y.M., Al-Tayyan, M.H. (1994). A computerized morpho-syntactic system of Arabic. In *The Arabian Journal of Science and Engineering*, 19, 461-480. Published by KFUPM, Saudi Arabia.

Attia., M., Yaseen., M., Choukri., K. (2005). Specifications of the Arabic Written Corpus produced within the NEMLAR project, www.NEMLAR.org.

Beesley, Kenneth R. (1996). Arabic finite state morphological analysis and generation. *In COLING'96, volume 1*, pages 89– 94, Copenhagen, August 5-9. Center for Sprogteknologi. The 16th International Conference on Computational Linguistics, 1996.

Buckwalter, T. (2004). Issues in Arabic Orthography and Morphology Analysis. *In Proceedings of the COLING 2004*. Workshop on Computational Approaches to Arabic Script-based Languages, pages 31–34.

Buckwalter Arabic Morphological Analyzer Version 1.0. (2002). LDC Catalog No.: LDC2002349.

Cavalli-Sforza, Soudi, Mitamura (2000). Arabic Morphology Generation Using a Concatenative Strategy. In *Proceedings of the 6th Applied Natural Language ProcessingConference (ANLP 2000)*, pages 86–93, Seattle, Washington, USA.

Dichy, J., Farghaly, A. (2003). Roots & Patterns vs. Stems plus Grammar-Lexis Specifications: there what basis should is multilingual lexical database centred on Arabic be built? *In Workshop on Machine Translation for Semitic Languages*, New Orleans, USA.

Habash, N., Rambow, O. (2005). Arabic Tokenization, Morphological Analysis, and Part-of-Speech Tagging in One Fell Swoop. *In Proceedings of the Conference of American Association for Computational Linguistics* (ACL05).

Habash, N., Rambow, O. (2006). MAGEAD: A Morphological Analyzer and Generator for the Arabic Dialects. *In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 681–688, Sydney, Australia, July.*

Habash, N. (2010). *Introduction to Arabic Natural Language Processing*. Morgan & Claypoll Publishers.

Huh, H.-G. Laporte É. (2005). A resource-based Korean morphological annotation system. *In Proc. Int. Joint Conf. on Natural Language Processing*, Jeju, Korea, 2005.

Kiraz, A. (2004): http://www.scribd.com/doc/46443095/Computational-Nonlinear-Morphology-With-Emphasis-on-Semitic-Languages-Studies-in-Natural-Language-Processing-9780521631969-41686

Mesfar, S. (2008). Analyse morpho-syntaxique automatique et reconnaissance des entités nommées en arabe standard. Thèse, novembre 2008, Université de Franche-Comté.

Mesfar, Slim. (2006). Standard Arabic formalization and linguistic platform for its analysis in *Proceedings of Arabic NLP/MT conference*, London, 2006

Marcelo C.M. Muniz, Maria das Graças V. Nunes, and Éric Laporte (2005). UNITEX-PB, a set of flexible language resources for Brazilian Portuguese. *Workshop TIL'05*. pp. 2059–2068.

Paumier, Sébastien. (2011). *Unitex - manuel d'utilisation 2.1* , University of Marne-la-Vallée.

Silberztein, Max. (1998). INTEX: An integrated FST toolbox, in Derick WOOD, Sheng YU (éd.), *Automata Implementation*, p. 185-197, Lecture Notes in Computer Science, vol. 1436. Second International Workshop on Implementing Automata, Berlin/Heidelberg: Springer.

Smrz, Otakar. (2007). ElixirFM — Implementation of Functional Arabic Morphology. *In Computational Approaches to Semitic Languages*, ACL 2007, Prague.

Al-Sughaiyer, Imad A., Al-Kharashi, Ibrahim A. (2004). Arabic Morphological Analysis Techniques: A Comprehensive Survey. *In Journal of the American Society for Information Science and Technology*, 55(3):189–213.