Development of Lexicons Generation Tools for Arabic: Case of an Open Source Conjugator

Mourchid Mohammed
MIC search team, Laboratory MISC,
Faculty of sciences,
Ibn Tofail University in Kenitra- Morocco

EL Faddouli Nour-eddine
RIME search team, Laboratory LRIE,
MOHAMMADIA ENGINEERING SCHOOL,
Mohammed V University in Rabat-Morocco

Amali Said
OMEGA search team, Laboratory LERES,
Faculty FSJES,
Moulay Ismaïl University in Meknes-Morocco

ABSTRACT
The need for Automatic Natural Language Processing (NLP) in large dictionary resources continues to grow. The management of these linguistic knowledge should be taken into account because it is a fundamental element in the success and effectiveness of applications of NLP. Also, there is increasing interest for the development of reusable and independent lexical databases of a particular language application. Knowledge about a lexical database are complex, large sizes and various (ie, phonological, morphological, syntactic, semantic and pragmatic), which has negatively influenced many national and international projects for the development of lexical databases (monolingual or multilingual). Among such lexical database entries, we find conjugated verbs. To this end, we present in this paper, our open source conjugator application of arabic verbs that we have developed in Java under the Android platform. This conjugator developed within the MISC laboratory is structured into several modules whose core is a morphological generator Root-Pattern.

Keywords
Natural Language Processing, Lexical databases, Arabic verbs conjugator, Root.

1. INTRODUCTION
In the Arabic language, each word having a meaning consists of a root and a pattern. So we can represent all the Arabic words by a matrix in which the patterns are the columns and roots are the lines. A word is simply a point in this matrix [1] [3] [5].

Example:

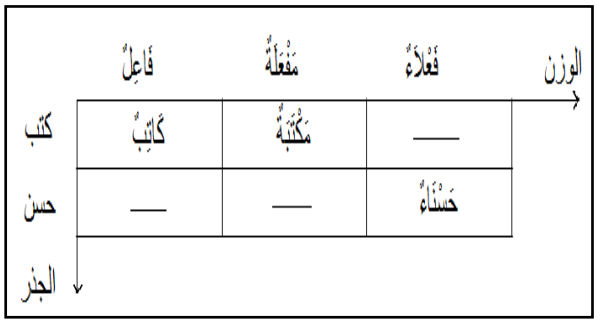

Fig 1: Matrix Root/Pattern

With this matrix, we can perform two operations [2][4][7]:

- Analysis: extract the pattern and the root of a given word.

- Generation: building a word from a column (pattern) and a line (root).

Figure 2 shows an example of using this matrix to extract the root and the pattern of the word مَكْتَبَةٌ (MaKtaba , Library) and to generate the word كَاتِبٌ from the root كتب (Root KTB, Write)and the pattern فَاعِلٌ.

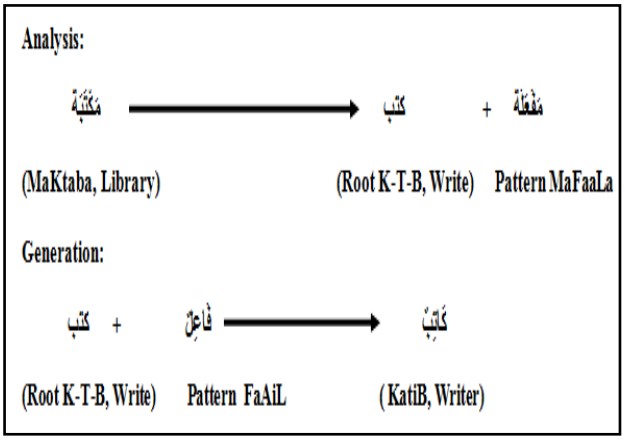

Fig 2: Operations: Analysis and Generation

In this paper, we present the approach of our conjugator developed in Java as an Android mobile application.

The principle of morphological generation used in our conjugator will be presented in the second section.

In the third section, we describe the different approach steps of our conjugator and the used repository.

The fourth section will be dedicated to description of the conjugator mobile aplication.

In conclusion, we will discuss the results and the main perspectives of our research.

2. MORPHOLOGICAL GENERATION PRINCIPLE
Morphological generation is a succession of morphological operations applied to an initial word accompanied by a set of attributes, to produce a final form of the word.

These attributes are for: [6]

Names: Number (singular, dual, plural); case (nominative, accusative, gerund); determination (defined, undefined), annexation (annexed)

Verbs: Aspect (perfect, imperfect, imperative); voice (passive, active); Number (singular, dual, plural); gender (male, female); person (first person, second person, third person); case (nominative, accusative, apocope); insistence.

There are three generation methods differentiated by their approaches [5].

2.1 Successive Transformations Method

This method relies on applying transformations progressively until obtaining of the final form.

These transformations are shown as follows:

Initial word + AT1====> word1

word1+ AT2 ====> word2

wordn-1 + ATn ====> final word

Where ATi is an attribute.

The figure 3 shows how to get final word الْعُلُومَ (the sciences) from the initial word عِلْمٌ (science) using the set of attributes(Plural , Defined , Accusative).

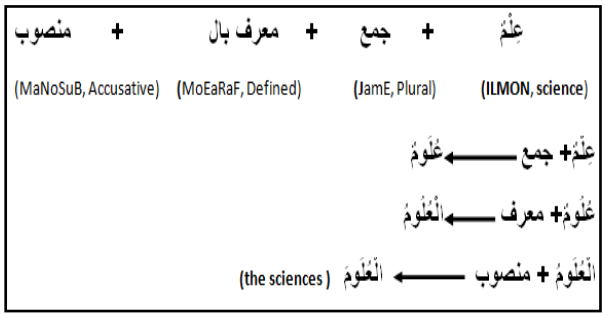

Fig 3: The final word is الْعُلُومَ (AL-OLOMA, the sciences)

2.2 Method of Pre-Established Models

This method is based on the use of a set of predetermined models each of which is associated with a class of words having the same characteristics.

Thereby the word generation is done in two steps:

- Determination of suitable model from the characteristics of the original word.
- Performing a substitution operation from the initial word and the corresponding model.

The figure 4 depicts the steps followed to obtain the mould from a set attributes.

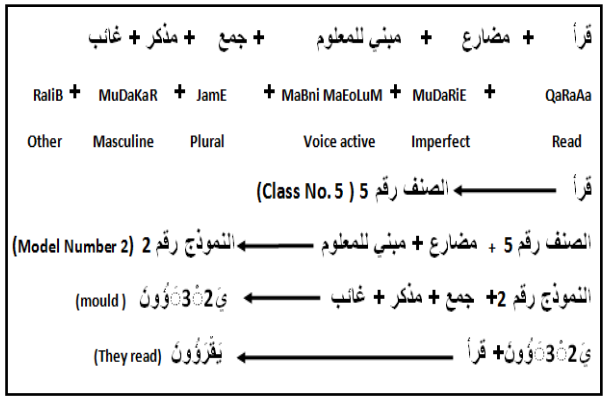

Fig 4: Obtaining the mould

2.3 Mixed Method

We apply, in a first step, the pre-established model method. Then we apply the successive transformations method on the result of the first step(See figure 5). .

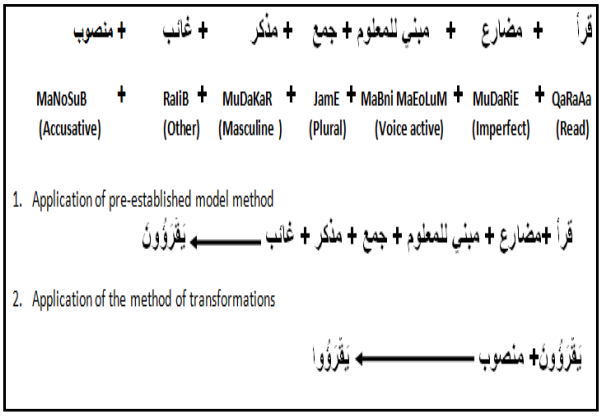

Fig 5: The final word is يَقْرَؤُوا (YAQRAO, They read)

3. CONJUGATOR OF VERBS [5]

The method adopted for conjugation of the verb is the mixed method, ie that we apply, in a first step, the pre-established model method, then we apply the successive transformations method on the result of the first step.

Thus, the conjugation operation of a verb proceeds in six steps:

- Determining the verb class (Cf Fig. 6) by consulting the lexicon t of triliteral verbs or make a treatment based on the length of the verb and the location of certain consonants.

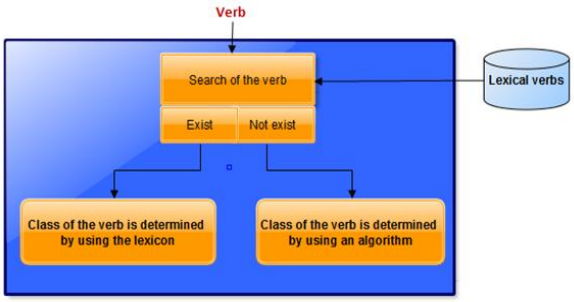

Fig 6: Determining of the verb class

- The adequate model is determined from the triple (class tense, time, voice). (Cf. Fig.7).

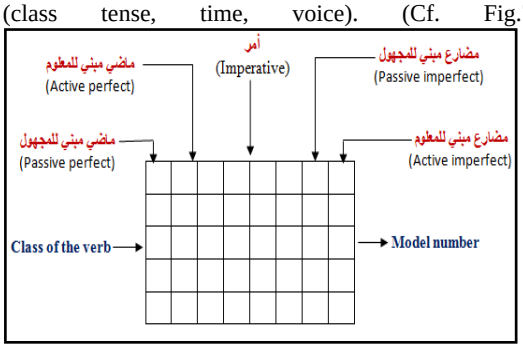

Fig 7: Determining the model.

- The triple (person, number, gender) and the model number allow to extract the desired mould (Cf Fig.8).

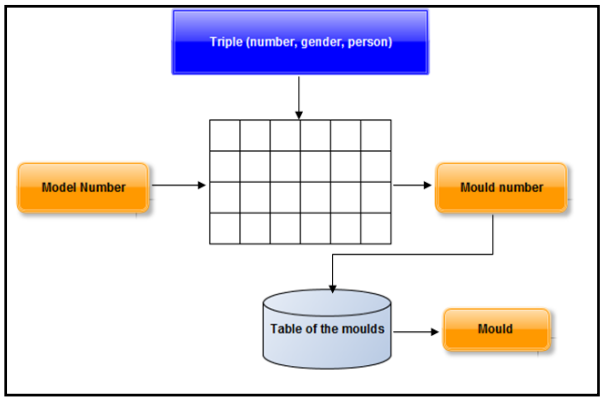

Fig 8: Determination of the mould

Substitution operation consists to replace the mould numbers with the corresponding consonants of the verb as shown in the example of table 1.

Table 1. The substitution operation.

Verb	Mould	Substitution
دَخَلَ (DaKaLa, Enter)	◌ُ3◌ُ2◌ْ1يَ	يَدْخُلُ (YaDoKuLu,Enter)

- Application of the transformation rules (T.R) to deal the case (Apocope, accusative, insistence)of imperfect tense .

For each case, we define the actions to perform on the consonants and vowels of the verb. Table 2 shows an example of the transformations actions:

Table 2. Some transformations actions

Action number	Action	Coding of actions
1	Change the last character of the vowel part by "◌َ"	M
2	remove the last character of the consonant part	S
3	remove the last character of the vowel part	S
4	Add the character 'ا' to the last position	A

Table 3 is an example of application of the transformation rules.

Table 3. Application of the RT.

Rule number	Transformations actions	Example (Before)	Example (After)
1	1	يَعْلَمُ	يَعْلَمَ
3	2,3	يَعْلَمَانِ	يَعْلَمَا
4	2,3,4	يَعْلَمُونَ	يَعْلَمُوا

- Spellchecking : This step involves performing the possible corrections on the verb, based on the lexicon of the corresponding anomalies and corrections, an example is given in table 4.

Table 4. Anomaly and correction.

Verb anomaly	Verb after correcting
أَأَمَنُ	آمَنَ

4. IMPLEMENTATION AND TEST OF OUR CONJUGATOR

We realised our conjugator based on the rules outlined in the previous sections. It consists of four modules (Cf Figure 9).

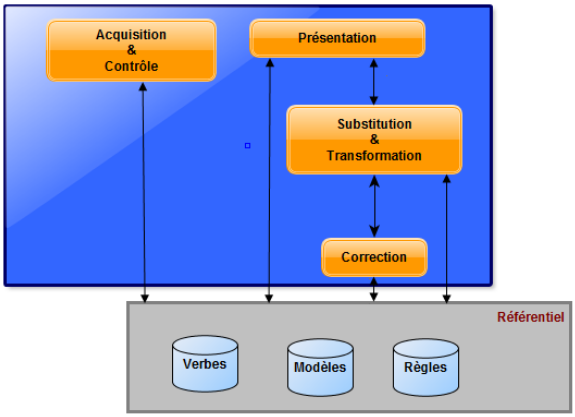

Fig 9: Conjugator Modules.

- The presentation module enables communication with the user, which can be a person or an application in which our conjugator can be integrated. This module makes the necessary checks on the verb to conjugate and determines the pattern to apply.
- The substitution and transformation module enables to apply the transformation rules on the pattern determined by the previous module.
- The correction module applies correction rules on the result provided by the transformation module to obtain the final shape.
- A linguist to feed the repository by the verbs, the models and the correction rules missing will use the acquisition module and control.

The repository of our conjugator contains all relevant data: verbs, models and correction rules. These data can be stored in several forms (relational DB, XML, Jason ...).

We implemented our conjugator, in its first version, as an Android mobile application using the Java language for coding, the DBMS SQLite for managing the repository and XML to generate mobile interfaces.

Figure 10 shows the interface that allows the user to enter the verb, choose the aspect, the voice, and the mode of conjugation.

Fig 10: data entry

These data will be recovered by the presentation module that will determine the model to apply. The transformation rules for the latter will be applied by the substitution module whose result will be processed by the correction module. The final result obtained will be communicated to the user (Cf Figure 11).

Fig 11: Conjugation of the verb 'كتب' at the accomplied tense

Statistics on the repository are presented in Table 5.

Table 5. Statistics

Number of roots	6413
Number of triliteral roots	5635
Number of roots quadrilitaires	592
Number of canonical schemas	199
Number of irreducible trilitaires verbs (مجردة)	6138
Number of reducible trilitaires verbs (مزيدة)	11832
Number of irreducible quadrilitaires verbs (مجردة)	452
Number of reducible quadrilitaire verbs (مزيدة)	245

5. CONCLUSION

In this paper, we presented the architecture of a system of conjugation of verbs Arab. It operates according to a five-step process: determining the class of verb , determining the model, the substitution operation, applying transformation rules, and spell correcting. The results of the performed tests are very satisfactory.

We intend to use this conjugator to enrich the lexical database by the verbs for morphological analysis using the dictionary-based approach.

Our conjugator can also be used as the core of a learning environment of Arabic and especially conjugation.

6. REFERENCES

[1] Ali Nabil (1988) "Arabic Language and Computer", [in Arabic], Ta'areeb.

[2] Hlal Yahya, (1979) "Learning Methods for Morphosyntactic Analysis (Experienced in the case of Arabic and French)", thesis doctoral degree, University Paris.

[3] Hlal Yahya, (1987) "Generation from the root and pattern", Conference on the progress of linguistics in Arab countries.

[4] Hlal Yahya, (1990) "Morphology and syntax of the Arabic language", In Proceedings of the Arab School of Science and Technology Applied Arabic Linguistics for Informatics (pp.201).

[5] Mourchid Mohamed, (1999) "Generation Morphological and Applications", Specialty thesis of 3rd round, Mohammed V University in Rabat-Morocco.

[6] Nizar Y. Habash, (2010) Introduction to Arabic Natural Language Processing, Morgan & Claypool Publishers series.

[7] Abdelhadi Soudi, Antal van den BoschNizar, & G¨unter Neumann (2007) Arabic Computational Morphology, Knowledge-based and Empirical Methods, Springer