The use of Arabic linguistic resources to develop learning applications

Ilham Blanchete, Mohammed Mourchid
LaRI Laboratory, Department of Computer Science, Faculty of Science, Ibn Tofail University, Kenitra, Morocco

Article Info

Article history:

Received Sep 23, 2021
Revised Aug 3, 2022
Accepted Sep 6, 2022

Keywords:

Arabic broken plural
Arabic verbs
Linguistic resources
Natural language processing applications
NooJ platform
Root pattern morphology

ABSTRACT

This paper aims at explaining how Arabic linguistic resources are generated and exploited to enhance Arabic acquisition. We have adopted the root and pattern approach to generate our resources using the linguistic NooJ platform. This work has been carried out in two phases: generating the linguistic resources and developing the application that exploits the pre-built resources. First, we have generated three different resources: comprehensive verbs and masdar resources linked to each other. A nouns-and-adjectives resource, where nouns and adjectives are linked to their broken plural forms. NooJ calls these resources to apply linguistic analysis to a given corpus and returns detailed annotations, which provides accurate morphological, syntactical, and semantic properties of each analyzed word. We have also used the mixed nature of Arabic masdar to implement transformational rules, which generate nominal sentences from verbal ones and vice versa. Second, we have developed an application that provides valuable learning functionalities, like full/semi verb conjugation, the extraction of broken plural forms of a given singular form, the extraction of masdar forms of a given verb form, and return words that share the same root. The developed application can be used by teachers, students/learners, and computational linguists interested in Arabic acquisition.

Corresponding Author:

Ilham Blanchete
LaRI Laboratory, Department of Computer Science, Faculty of Science, Ibn Tofail University
B.P 242, Kénitra, Morocco
Email: ilham.blanchete@gmail.com

1. INTRODUCTION

The topic of this article is the generation of standard Arabic resources. These generated resources have been exploited to develop a learning application. For this purpose, we have adopted the linguistic approach rather than the statistical one- to build our linguistic resources. The linguistic approach has been adapted to describe the Arabic root and pattern morphology and build a resource that respects the structure of the language in question [1]. First, we have used root and pattern morphology to formalize the Arabic vocabulary and phenomena. Our linguistic resources contain verbs linked to their generated masdar forms, nouns and adjectives linked to their generated broken plural forms. Broken plural and masdar forms have been generated using their linguistic properties: roots, patterns, morphophonological, morphosyntactical, and semantic features, which have been tested to generate masdar and broken plural patterns. Second, we have used the linguistic NooJ platform, which adopts by its turn mathematical models to formalize any natural language [2]. NooJ's linguistic engine calls our resources to analyze Arabic texts and return accurate annotations. Third, to develop a learning application, we have exploited the returned annotations served by NooJ’s text annotation system [3]. This application provides useful functionalities, as we will detail in the contribution section.

Two approaches are adopted to make any natural language understandable by machines. Linguistic approach and statistic approach. The linguistic approach provides comprehensive linguistic resources, containing at least a dictionary linked to sets of lexical, morphological, syntactical, and semantic grammars. The dictionary describes the vocabulary, and grammars describe rules, which combine vocabulary elements to construct sentences [2]. Language phenomena could also be implemented using morphological or syntactical grammas, e.g., processing agglutination with morph-syntactic grammars or software applications, e.g., automatic conjugator that we have developed. Some patterns have been generated by applying conditions on dictionary entries. To clarify, conditions have been tested on the linguistic properties, which have been accurately assigned to dictionary entries. E.g., broken plural and masdar patterns [4]. Thus, we cannot avoid vocabulary roots, patterns, morphophonological, morphosyntactical, and semantic features during the formalization of the Arabic language. It is worth mentioning that a miscalculation of these properties leads to an incomplete linguistic resource, which makes it unable to overcome the complex analysis phases.

The second approach adopted to formalize natural languages is the statistical approach, which stands on the part of speech (POS) taggers. POS taggers reduce the linguistic properties and decrease analyzers' effectiveness. They refer to the reference corpus, like “Penn Treebank” [2], instead of dictionaries and grammars to statistically annotate texts, making them incapable of providing accurate language descriptions. Silberztein has argued in [2] that POS taggers cannot solve the ambiguity problem accurately, where words hold multiple linguistic descriptions. POS taggers also disregard the existence of multiword units and expressions, an example of the disregarded multiword in the peen treebank, where the compound noun “industrial managers”, the phrasal verb “to buck up”, the compound determiner “a boatload of”, the compound noun “samurai warrior”, the expression “to blow N ashore”, the adverb “from the beginning”, and the expression “it takes N to V-inf” have all been disregarded. Weaknesses like disregarding multiword units, phrasal verbs and expressions eliminate any possibility of conducting meaningful linguistic analyses on the resulting tagged text. Statistical taggers are not generalizable. Thus, the construction of a reference corpus or treebank including all potential uses of each word would be required [2]. These POS taggers' shortness can be easily solved using the linguistic approach by building dictionaries that assign accurate linguistic properties to each entry. Shortness also encouraged us to use the NooJ platform, which solves them by simply creating new grammars. Thus, this article discusses the Arabic linguistic resource. In particular, dictionaries linked to their grammars provide simple solutions for the weakness mentioned above. NooJ text annotation systems can easily refer to syntactical grammars to solve the previous-mentioned ambiguity problem [5]-[8]. Furthermore, adding local, morphological, and syntactical grammars to the linguistic resources solve problems that may appear during the resource testing; this process is much easier than modifying the reference corpus.

To represent the Arabic language, we need an appropriate descriptive structure that allows the representation of morphological, syntactical and semantic properties of each word. In addition to the semantic properties that characterize the Arabic language, there are other properties such as morphology and syntax. Using an ontology, we can represent the semantic relation between concepts. Therefore, the representation of language concepts will inevitably depend on a linguistic ontology that derives its vocabulary from the lexicon that we have built. Accordingly, we see that the structure that we have relied on represents the language's vocabulary in a way that allows it to represent all linguistic characteristics. An ontology can be added to the construction of the semantic analyzer.

Natural language processing (NLP) applications that adopt the statistical approach have proved their shortness since they use POS taggers rather than dictionaries, e.g., translation, summarization, text generation. More generally, systematically tagging texts without taking into account multiword units, phrasal verbs and expressions eliminates any possibility of conducting meaningful linguistic analyses on the resulting tagged text [2]. Many grammars have been developed to overcome the previous-mentioned shortness; by adopting the linguistic approach. They refer to the accurate annotations to perform their tasks. Like rules have been used for the implementation of Arabic phonological rules [9], formalization of the Arabic grammatical category (V-a) [7], lexicon-grammar tables development for Arabic psychological verbs, applying transformational grammars to recognize Arabic psychological verbs [9]. works mentioned in [4]-[9] have been realized with great attention to the Arabic root and pattern morphology. Hence, developers can rely on these works to develop NLP applications. It is worth mentioning that works have been implemented based on the linguistic approach but have some shortness. To name a few, the Arabic masdar generation mentioned in [7] has an unclear dictionary structure; it generates the dictionary entries from both the lemma and the root. Another work, “EL-DiCar” dictionary [3], which adopts the linguistic approach, but has shown several weaknesses in advanced analysis phases. Dictionaries that adopt the lemma approach to formalize the Arabic Semitic language, and ignore the root-pattern during the dictionary construction, are unable to: i) extract meaning using patterns [4], ii) extract words that share the same root [10], and iii) unable to generate the broken plurals and masdar forms from their singular linguistic properties; since no root and pattern has been assigned to dictionary entries [4]; instead, they assign the regular plural form for each singular one, if any. Besides root and pattern, some phonological, morphological, syntactical, and semantic features have been ignored during “EL-DiCar” construction, which is unacceptable since Arabic is Semitic [11]. It is worth mentioning that “EL-DiCar” has been implemented using the Nooj platform [12]. Another work has been realized based on the root and pattern approach, concerning pattern and root inflectional morphology (PRIM) [13], which is “an implemented model of arabic nouns inflectional morphology.” Each noun entry accepts only one plural form, even if the noun has more than one broken plural form. For instance, the entry (جَمَل-JaMaL/camel) that has 11 broken plural forms, the PRIM will insert 11 entries for the same singular form, which causes redundancy of morphsyntactic features in 11 lines that refer to the same singular form. Other works have been developed as NLP applications, kids’ learning games [14], decision-support tool of medical plants [15]. In previous stages, we have developed the learning application to enhance the educational process in the Moroccan mid-high stage using NooJ [16]. NooJ platform provides useful functionalities to developers, e.g., annotations reports in different formats and noojapply.exe functionality, which allows calling NooJ’s linguistic engine from any source code. Accordingly, developers can rely on these functionalities to develop multidisciplinary programs. The proposed application exploits our resources, built with special attention to the root and pattern morphology. We have used NooJ (a linguistic platform) and python (a programming language) to develop our application. The application provides useful functionalities for students/researchers/learners, linguists, and computational linguists. The application facilitates Arabic language acquisition by providing roots, patterns, morpho-phonological, morphosyntactical, and semantic features of verbs, nouns, adjectives, BPs, and masdar forms.

Our application allows for three main tasks: i) verb, noun, and masdar. Verb task enables us to make full/semi conjugation, extract masdar forms of a selected verb, return all verbs that share the same root with their linguistic properties, and return the possible meaning of a selected verb. It is worth mentioning that different meanings of the same entry may affect the linguistic properties, which leads to adding new entries to maintain these differences that make the resource rich; ii) noun task enables us to return all nouns/adjectives that share the same root, return different meanings of a selected noun/adjective, and extract their generated broken plural forms; and iii) masdar task returns masdar forms of the selected -unaugmented triliteral- verb if any. Two different uses of Arabic masdar can be distinguished. [5], verbal use and nominal use. Therefore, we have applied a transformational rule using our resources, which allows the transformation of the nominal phrases into verbal ones.

2. CONTRIBUTION

The complexity of Arabic morphology makes its learning challenging, especially for low levels. It is not easy to understand the complex phenomena that are applied by the inflectional morphology of this language. Low-level learners face this complexity and software developers who just started to work on Arabic NLP. Besides this, the application is of significant benefit to both of them. In addition, Arabic learning applications exhibit several problems related to the language structure, as they usually employ the rule-based and machine learning approach [17]. The complexity of resource building is that the Arabic language has fin linguistic properties like the root and pattern, which are unavoidable during this process. Besides these features, the root class must be defined for each dictionary entry [4] E.g., the root class for the verb (to say-KaALa-قَال) is CWC, which indicates that the root contains a short vowel and it is a (W-و). Accordingly, both its derivational and inflectional forms must be classified as (hollow-أجوف) even if the hollowed letter disappears affected by the morphophonemic phenomena may occur. A reason that obliges us to identify these fin linguistic properties is to provide the language features to be reused in advanced generating or analysis phases, to name a few. The infinitive form of the verb (to say- KaALa-قَال) is formed by applying an intersection between the root (KWL-قول) and the pattern (FaEaLa-فَعَل), the intersection between the second root letter, which is a long vowel (W-و) and the second pattern letter supposed to be a (Wa-و), but due to the Arabic complex morphology, some morphophonemic changes have been occurred and changed this letter to an (A-ا). However, the letter (Wa-و) appears again in other inflectional forms.

Similarly, the infinitive form of the verb (to sell-BaAaE-بَاعَ) that has the root (BYE-بيع) and the pattern (FaEaLa-فَعَل). Another problem explains the importance of the root and pattern, the generation of broken plural forms. Previous studies have extracted rules that restrict the generation of broken plural forms from their singular ones. Then, to apply the rule: [if the singular pattern is (فَعَل/FaEaL), and if it is a noun, and if its second letter is an (أ/A), which was a (و/W)-which means its second root letter is a (و/W) but according to morphophonemic changes during the intersection between the root and the pattern, the (و/W) letter has been substituted to an (أ/A)- then its broken plural form is (فُعْلَان/FuEoLaAN)], e.g., [(TaAJ-تَاج): its root letters (TWG-توج), (TaAJ-تَاج) is a noun, (Crown-TaAJ-تَاج) its second letter is an (أ/A)] then its broken plural form is (Crowns-TiIoGaAN-تِيْجَان), the same broken plural generation rule has been applied for the singular form (Neighbor-JaAR-جَار) (Neighbors-JiIoRaAN-جِيرَان)]. On the whole, roots that contain short vowels or hamza special letter (H-أ) are likely to change. The hamza may take one of the following changes during the inflection/derivation process: ى, ئ, ؤ, آ, إ, and ء. However, it is hard to apply these morphophonemic changes without adopting the root and pattern approach. These important linguistic features may fall from developers inadvertently during the resource representation. A miscalculation of these properties leads to an incomplete linguistic resource, which makes it unable to overcome the complex analysis phases like the morphological, syntactical, and semantic analysis.

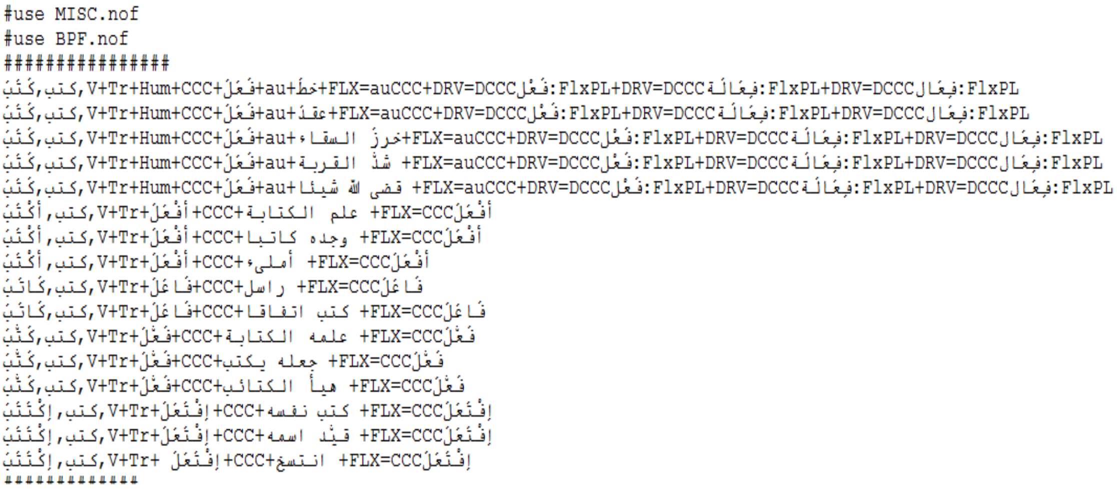
Another reason why the linguistic approach has been adopted; is that many Arabic learning applications show several shortages regarding the structure of this language. They usually employ the statistical approach or adopt the linguistic approach, but they neglige the Arabic language structure and adopt the lemma rather than root and pattern morphology, which makes extracting linguistic properties like roots and patterns hard to execute. Patterns also play an important role in Arabic morphology, e.g., we can extract words concepts from patterns [4], [10]. A case in point, the pattern (FiEaALaa-فِعَالَة) is employed to extract craft concept, generally. NLP applications that rely on meaning extraction may perform tasks using patterns defined in their linguistic resources.

We have adopted the root and pattern approach to building Arabic linguistic resources, implemented with special attention to the language structure. Especially the morpho-phonologic, morpho-syntactic, and semantic properties. The resource building process has been implemented over three steps:

- verbs resource: we have built a comprehensive verbs resource. It consists of a dictionary containing 295 possible verbs representative models [18]. Each verb is linked to its morpho-phonological, morpho-syntactical, and semantic features. The resource also contains lexical grammars. These grammars generate each dictionary entry's possible inflectional and derivational forms [10].
- broken plurals resource: contains nouns and adjectives linked to their possible broken plural form/forms. Broken plural forms have been generated based on their singular linguistic properties. We have extracted conditions restricting the generation of broken plurals from their singular forms [4].
- masdars resource: we have extracted conditions restricting the generation of masdar forms of the unaugmented triliteral verbs. This resource is linked to the first resource thanks to the linguistic relation that binds them.

Inflectional and derivational grammars have been implemented based on the finite state transducers approach using the NooJ platform. NooJ uses its linguistic engine to execute the linguistic analysis and return annotations using NooJ's text annotation system, which describes the linguistic properties of each analyzed word. NooJ also offers the functionality of "noojapply," a command-line used to call and employ the NooJ linguistic engine with our resources to analyze a given text. All these tasks have been provided in one command using noojapply.exe functionality. NooJ provides this command to make the generated linguistic resources useful and callable by any NLP application, making application development faster and easier. Now computational linguists can add linguistic resources separately without modifying their source code. They can process a new phenomenon by adding new grammars, testing the predefined linguistic properties, and exploiting the annotations to develop the desired application.

As an example of our resource, a dictionary entry has been detailed to clarify the importance of the linguistic properties that have been used, e.g., the verb (to write -KTB-كَتَبَ) that has the root (KTB-كتب), the pattern (FaEaLa-فَعَلَ) and the conjugational class (FaEaLa-YaFoEaLu/فَعَلَ-يَفْعُلُ) is represented as Figure 1 shows. Each declared line defines a new verb, e.g., verb (to write-KaTaBa-كَتَبَ) has five different meanings [4], which obliges us to insert five definitions for this verb.



```
#use MISC.nof
#use BPF.nof
################
كَتَبَ,كتب,V+Tr+Hum+CCC+فَعَلَ+au+خطّ+FLX=auCCC+DRV=DCCCفَعْل:FlxPL+DRV=DCCCفِعَالَة:FlxPL+DRV=DCCCفِعَال:FlxPL
كَتَبَ,كتب,V+Tr+Hum+CCC+فَعَلَ+au+عقدَ+FLX=auCCC+DRV=DCCCفَعْل:FlxPL+DRV=DCCCفِعَالَة:FlxPL+DRV=DCCCفِعَال:FlxPL
كَتَبَ,كتب,V+Tr+Hum+CCC+فَعَلَ+au+خرزَ السقاء+FLX=auCCC+DRV=DCCCفَعْل:FlxPL+DRV=DCCCفِعَالَة:FlxPL+DRV=DCCCفِعَال:FlxPL
كَتَبَ,كتب,V+Tr+Hum+CCC+فَعَلَ+au+شدَّ القربة +FLX=auCCC+DRV=DCCCفَعْل:FlxPL+DRV=DCCCفِعَالَة:FlxPL+DRV=DCCCفِعَال:FlxPL
كَتَبَ,كتب,V+Tr+Hum+CCC+فَعَلَ+au+قضى الله شيئا +FLX=auCCC+DRV=DCCCفَعْل:FlxPL+DRV=DCCCفِعَالَة:FlxPL+DRV=DCCCفِعَال:FlxPL
أكْتَبَ,كتب,V+Tr+أفْعَلَ+CCC+علم الكتابة +FLX=CCCأفْعَلَ
أكْتَبَ,كتب,V+Tr+أفْعَلَ+CCC+وجده كاتبا +FLX=CCCأفْعَلَ
أكْتَبَ,كتب,V+Tr+أفْعَلَ+CCC+أملىء +FLX=CCCأفْعَلَ
كَاتَبَ,كتب,V+Tr+فَاعَلَ+CCC+راسل +FLX=CCCفَاعَلَ
كَاتَبَ,كتب,V+Tr+فَاعَلَ+CCC+كتب اتفاقا +FLX=CCCفَاعَلَ
كَتَّبَ,كتب,V+Tr+فَعَّلَ+CCC+علمه الكتابة +FLX=CCCفَعَّلَ
كَتَّبَ,كتب,V+Tr+فَعَّلَ+CCC+جعله يكتب +FLX=CCCفَعَّلَ
كَتَّبَ,كتب,V+Tr+فَعَّلَ+CCC+هيأ الكتائب +FLX=CCCفَعَّلَ
إكْتَتَبَ,كتب,V+Tr+إفْتَعَلَ+CCC+كتب نفسه +FLX=CCCإفْتَعَلَ
إكْتَتَبَ,كتب,V+Tr+إفْتَعَلَ+CCC+قيّد اسمه +FLX=CCCإفْتَعَلَ
إكْتَتَبَ,كتب,V+Tr+ إفْتَعَلَ+CCC+انتسخ +FLX=CCCإفْتَعَلَ
##############
```

Figure 1. Verb dictionary in NooJ

A question that might emerge here is why we must insert ten different definitions for the same verb while inserting the possible meanings in the same definition line is possible? The answer is that meaning may lead change the grammatical category or insert new morphosyntactic and semantic properties, making extracting these properties for a specific meaning hard to achieve. Each line in Figure 1 defines the same verb with different meanings and linguistic properties. Furthermore, this helps NLP applications process tasks that use word meaningn e.g. system that process Arabic sign language, which helps deaf people communicate with machines. To clarify, signs are changed according to the verb's meaning. Adding signs' codes to the NooJ dictionary may help deaf's sign system, which has been achieved in [19] to transmit the spoken text to deaf people.

The verb "to write" is defined as V: the grammatical category of verb. Tr: transitive, Hum: semantic field, which indicates the nature of the subject may be assigned to this verb. (FaEaLa-فَعَلَ): the pattern, which indicates action, (KTB-كتب): the root, CCC: root class, which means that root letters are consonants and, au: the conjugational class, which is (FaEaLa-YaFuEaLuفَعَلَ-يَفْعَلُ), (manuscript- KHaT- خَط): first meaning of this entry, FLX: the inflectional paradigm, and DRV: the derivational paradigm, which indicates possible masdar forms in this declaration. Possible meanings for the verb "to write" are [(to write-KHaTTa-خطَ),(write-EaKaDa-عقدَ), (write-CHaDDa AL-KuRoBaAa-شدَّ القربة),(write-KaDda A AL-LaHu CHaIEAean--قضى الله شيئا)]. Adding the possible meaning of each dictionary entry helps NLP applications to analyze corpora and return detailed annotations.

Figure 2 shows a local grammar that generates the possible inflection forms of the representative model auCCC (AuCCC is the representative model of all trilateral verbs that have only constants as root letters, and inflect according to the conjugational model FaEaLa-YaFoEuLu) in the perfect active voice, which means that we have to assign this new linguistic feature to each inflected form to be used in our application. NooJ has its operators to read, modify, and delete dictionary entries. This implementation can be realized by the computational linguists and linguists who can implement their models using this friendly platform.

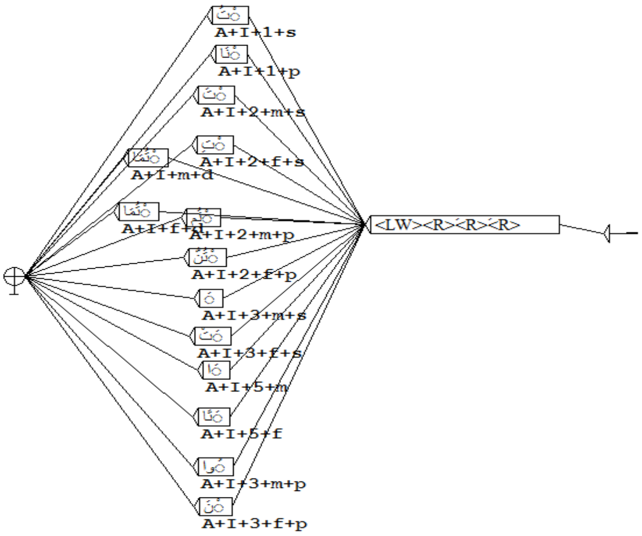


Figure 2. Inflectional forms of the verb to write

The operand <LW> left word places the cursor at the beginning of the root. The operand <R> reads the current letter [3]. Then, the instruction declared in the first node:<R>َ<R>َ<R>ْتُ modifies the entry, which is the root (KTB-كتب) to generate the inflectional form (i write-ANA KaTaBoTu-أنا كَتَبْتُ), and the linguistic properties have also been assigned to this node to be used as an annotation. The assigned annotation A+I+1+s means that this inflected form is conjugated in the perfect active voice and its morphological number feature is singular. As an application, these properties, a syntactic analyzer can easily use them to test sentence subject agreement.

The second resource is the broken plural one, representing singular forms of Arabic nouns and adjectives linked to their possible broken plural form/forms. We have extracted 108 rules and conditions that restrict the generation of broken plurals from their singular forms [10]. Rules and conditions have been extracted using singular's root, pattern, morphophonemic, morphosyntactic, and semantic features. Table 1 gives an example of the extracted conditions that restrict the generation of the broken plural pattern (فُعَلَاء/FuEaALaAE). This broken plural pattern is generated if and only if the: i) the singular pattern is (فَاعِل/FaEiL), ii) the singular form is an adjective, and iii) the singular morphological feature of gender is masculine and it must be rational [10].

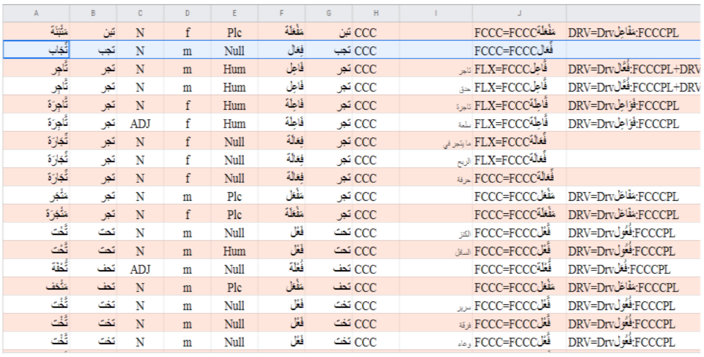
Table 1. The extracted condition

Singular form	Conditions	Broken plural form
Root ←CHER/شعر	Adjective	Pattern ←FuEaALaAX/فُعَلَاء
Pattern ←FaEiL/فَاعِل	Masculine	Form ← Poets/CHuEaRaAX/شُعَرَاء
Form←poet /CHAER/شَاعِر	Rational	
Root ←CHER/شعر	Adjective	Pattern ←FuEaALaAX/فُعَلَاء
Pattern ←FaEiL/فَاعِل	Masculine	Form ← Poets/CHuEaRaAX/شُعَرَاء
Form ←poet /CHAER/اعِرَ	Rational	

The third resource is the masdar one; we have extracted rules and conditions restricting the masdar generation for each unaugmented trilateral verb from Arabic grammarian books [20]-[24]. This resource has been linked to the verbs resource. To clarify, the verb (to write-KaTaBa-كَتَبَ) has been linked to three different masdars: [(KiTaABaa-كِتَابَة) that has the pattern (FiEaALAA-فِعَالَة), (KaToB-كَتْب) that has the pattern (FaEoL-فَعْل) and (KaATiB-كَاتِب) that has the pattern (FaAEIL- فَاعِل). They have different patterns, but they share the same root and meaning class. Masdar can replace Arabic syntax verbs, adverbs, and nouns [25]. Therefore, we have applied a transformation rule that substitutes a verb with one of its linked masdar forms; Since verbs are not randomly substituted, rules behind Arabic syntax must be applied. Our implemented transformational rule uses verb and masdar resources to transform using NooJ's morphological operations. NooJ also provides morphological operations to test conditions and filter entries that are not likely used in the implemented grammar.

3. IMPLEMENTATION

Computer assisted learning language aims to put the aspects of learning theories that respect the language structure and use linguistic resources to make computers and software programs capable of providing rich content [17]. Hence, we have used the above-detailed resources to enhance the acquisition of the Arabic language. The first step is to fill the dictionary manually using MS Excel files. Figure 3 gives an example of the filled dictionary.

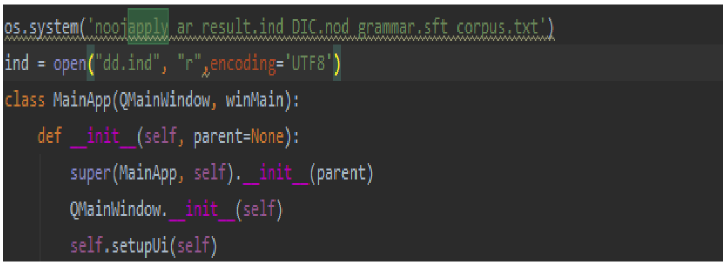


A	B	C	D	E	F	G	H	I	J	
مَتْبَنَة	تبن	N	f	Plc	مَفْعَلَة	تبن	CCC		FCCC=FCCCمَفْعَلَة	DRV=Drvمَفَاعِل:FCCCPL
تِجَاب	تجب	N	m	Null	فِعَال	تجب	CCC		FCCC=FCCCفِعَال	
تَاجِر	تجر	N	m	Hum	فَاعِل	تجر	CCC	تاجر	FLX=FCCCفَاعِل	DRV=Drvفُعَّال:FCCCPL+DRV
تَاجِر	تجر	N	m	Hum	فَاعِل	تجر	CCC	حذق	FLX=FCCCفَاعِل	DRV=Drvفُعَّال:FCCCPL+DRV
تَاجِرَة	تجر	N	f	Hum	فَاعِلَة	تجر	CCC	تاجرة	FLX=FCCCفَاعِلَة	DRV=Drvفَوَاعِل:FCCCPL
تَاجِرَة	تجر	ADJ	f	Hum	فَاعِلَة	تجر	CCC	سلعة	FLX=FCCCفَاعِلَة	DRV=Drvفَوَاعِل:FCCCPL
تِجَارَة	تجر	N	f	Null	فِعَالَة	تجر	CCC	ما يتجر في	FLX=FCCCفِعَالَة	
تِجَارَة	تجر	N	f	Null	فِعَالَة	تجر	CCC	الربح	FLX=FCCCفِعَالَة	
تِجَارَة	تجر	N	f	Null	فِعَالَة	تجر	CCC	حرفة	FCCC=FCCCفِعَالَة	
مَتْجَر	تجر	N	m	Plc	مَفْعَل	تجر	CCC		FCCC=FCCCمَفْعَل	DRV=Drvمَفَاعِل:FCCCPL
مَتْجَرَة	تجر	N	f	Plc	مَفْعَلَة	تجر	CCC		FCCC=FCCCمَفْعَلَة	DRV=Drvمَفَاعِل:FCCCPL
تُحْت	تحت	N	m	Null	فُعْل	تحت	CCC	الكثر	FCCC=FCCCفُعْل	DRV=Drvفُعُول:FCCCPL
تُحْت	تحت	N	m	Hum	فُعْل	تحت	CCC	السافل	FCCC=FCCCفُعْل	DRV=Drvفُعُول:FCCCPL
تُحْفَة	تحف	ADJ	m	Null	فُعْلَة	تحف	CCC		FCCC=FCCCفُعْلَة	DRV=Drvفُعَل:FCCCPL
مَتْحَف	تحف	N	m	Plc	مَفْعَل	تحف	CCC		FCCC=FCCCمَفْعَل	DRV=Drvمَفَاعِل:FCCCPL
تُخْت	تخت	N	m	Null	فُعْل	تخت	CCC	سرير	FCCC=FCCCفُعْل	DRV=Drvفُعُول:FCCCPL
تُخْت	تخت	N	m	Null	فُعْل	تخت	CCC	فرقة	FCCC=FCCCفُعْل	DRV=Drvفُعُول:FCCCPL
تُخْت	تخت	N	m	Null	فُعْل	تخت	CCC	وعاء	FCCC=FCCCفُعْل	DRV=Drvفُعُول:FCCCPL

Figure 3. Example of the filled dictionary

Column A←the inserted words, in this example this column contains (merchant-TaAJiR- تَاجِر), (merchant-TaJiRaa-تَاجِرَة), and (commerce-TiJaARaa-تِجَارَة). Column B←the root(TJR-تجر), column C←grammatical category, column D←morphological gender feature, column E ←semantic feature, column F←pattern, column H←root class, column I←entry meaning, column J←inflectional paradigm and column K←derivational paradigms. Second, we have developed a "mini-convertor" that converts excel files to NooJ dictionary format using python. Third, we have used NooJ platform to: i) create inflectional and derivational paradigms of each inserted entry, ii) compile the converted dictionary, iii) execute the linguistic analysis, and

iv) generate annotations that the application will use. Forth, we have used the Python programming language to develop the Arabic NLP application. The application calls NooJ using NooJ's functionality: Noojapply.exe, which executes a command to analyze a given text. Figure 4 shows how to call NooJ to analyze a given text using specified resources. The command "noojapplyarresult.indDIC.nodgrammar.sft corpus.txt" calls NooJ’s linguistic engine to apply a linguistic analysis on an "ar" Arabic language resource using the dictionary "DIC.nod" and the grammar "grammar.sft" to analyze the "corpus.txt" and returns the annotations to result.ind file.



```
os.system('noojapply ar result.ind DIC.nod grammar.sft corpus.txt')
ind = open("dd.ind", "r",encoding='UTF8')
class MainApp(QMainWindow, winMain):
    def __init__(self, parent=None):
        super(MainApp, self).__init__(parent)
        QMainWindow.__init__(self)
        self.setupUi(self)
```

Figure 4. Using NooJ’sfunctionalty

Finally, we have used QtDesigner to design our graphical user interface (GUI’s). Figure 5 shows the main interface that provides three functionalities: (nouns-AaSoMaAE-أسماء), (masdars-MaSaADiR-مصادر), and (verbs-AFoEaAL-أفعال). The verb section returns all possible verbs that share a given root (user entry), then the user can select a verb to be conjugated.

Arabic verbs inflect according to the voice, mood, and tens [10]. Voices are: active and passive voices, moods are: indicative المرفوع, subjunctive المنصوب, jussive المجزوم, tenses are: perfect ماضي and imperfect-مضارع, imperativeالأمر, long energetic المؤكد الثقيل and imperative of long energetic-الامر المؤكد. The user can choose full/semi conjugation. Figure 6 shows the full conjugation of the verb “to open- FaTaHa-فَتَحَ” in the active voice.

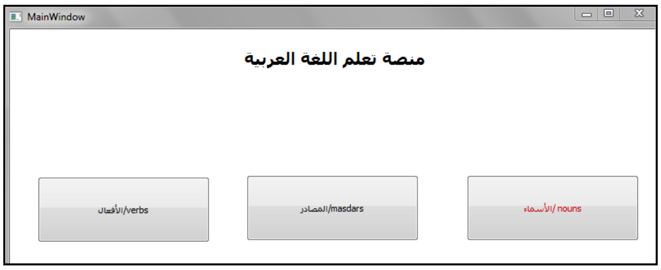
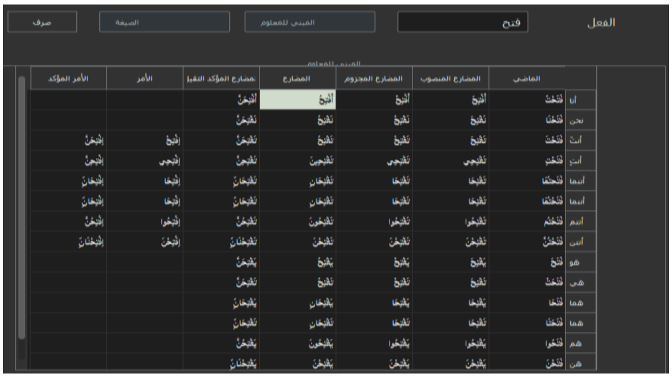


Figure 5. Main GUI



الفعل فتح

المبني للمعلوم

الصيغة

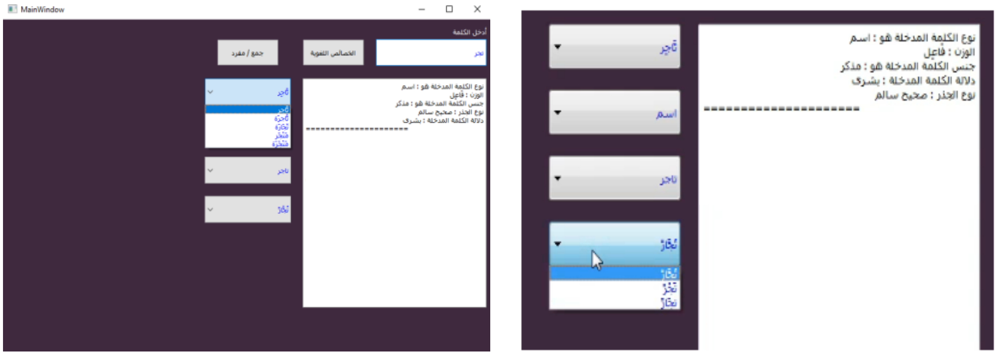
صرف

المبني للمعلوم

	الماضي	المضارع المنصوب	المضارع المجزوم	المضارع	مضارع المؤكد الثقيل	الأمر	الأمر المؤكد
أنا	فَتَحْتُ	أَفْتَحَ	أَفْتَحْ	أَفْتَحُ	أَفْتَحَنَّ		
نحن	فَتَحْنَا	نَفْتَحَ	نَفْتَحْ	نَفْتَحُ	نَفْتَحَنَّ		
أنتَ	فَتَحْتَ	تَفْتَحَ	تَفْتَحْ	تَفْتَحُ	تَفْتَحَنَّ	إِفْتَحْ	إِفْتَحَنَّ
أنتِ	فَتَحْتِ	تَفْتَحِي	تَفْتَحِي	تَفْتَحِينَ	تَفْتَحِنَّ	إِفْتَحِي	إِفْتَحِنَّ
أنتما	فَتَحْتُمَا	تَفْتَحَا	تَفْتَحَا	تَفْتَحَانِ	تَفْتَحَانِّ	إِفْتَحَا	إِفْتَحَانِّ
أنتما	فَتَحْتُمَا	تَفْتَحَا	تَفْتَحَا	تَفْتَحَانِ	تَفْتَحَانِّ	إِفْتَحَا	إِفْتَحَانِّ
أنتم	فَتَحْتُم	تَفْتَحُوا	تَفْتَحُوا	تَفْتَحُونَ	تَفْتَحُنَّ	إِفْتَحُوا	إِفْتَحُنَّ
أنتن	فَتَحْتُنَّ	تَفْتَحْنَ	تَفْتَحْنَ	تَفْتَحْنَ	تَفْتَحْنَانِّ	إِفْتَحْنَ	إِفْتَحْنَانِّ
هو	فَتَحَ	يَفْتَحَ	يَفْتَحْ	يَفْتَحُ	يَفْتَحَنَّ		
هى	فَتَحَتْ	تَفْتَحَ	تَفْتَحْ	تَفْتَحُ	تَفْتَحَنَّ		
هما	فَتَحَا	يَفْتَحَا	يَفْتَحَا	يَفْتَحَانِ	يَفْتَحَانِّ		
هما	فَتَحَتَا	تَفْتَحَا	تَفْتَحَا	تَفْتَحَانِ	تَفْتَحَانِّ		
هم	فَتَحُوا	يَفْتَحُوا	يَفْتَحُوا	يَفْتَحُونَ	يَفْتَحُنَّ		
هن	فَتَحْنَ	يَفْتَحْنَ	يَفْتَحْنَ	يَفْتَحْنَ	يَفْتَحْنَانِّ		

Figure 6. Full conjugation model of the verb "to open" in the active voice

The second functionality, nouns, returns all nouns/adjectives that share the same root with their linguistic properties as it is shown in Figure 7, the user gives a root as entry as Figure 7(a) shows, the user entry (TJR-تجر), - which is a root- then the application returns: i) the possible nouns/ adjectives that share this root, (merchant- TaAJiR- تَاجِر), (merchant- TaAJiR-تَاجِرَة), (trade- TiJaARaa- تِجَارَة), (store, MaToJaR-مَتْجَر) and (store, MaToJaR-مَتْجَرَة); ii) linguistic features as gender, root class, pattern, and the grammatical category; and iii) different meanings of the selected noun/adjective if any, and the possible broken plural form/forms of the selected noun/adjectiv. Figure 7(b) shows the possible broken plural forms of the selected singular form (merchant-TaAJiR-تَاجِر). Broken plural forms are: (merchants-TeJaR-تُجَّار), (merchants-TaJoR-تَجْر) and (merchants-TiJaAR-تِجَار).



نوع الكلمة المدخلة هو : اسم
الوزن : فاعِل
جنس الكلمة المدخلة هو : مذكر
نوع الجذر : صحيح سالم
دلالة الكلمة المدخلة : بشري
====================

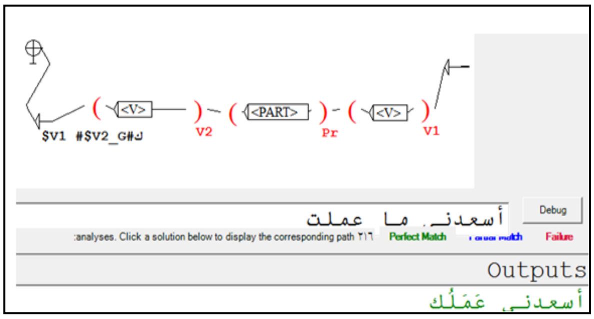
نوع الكلمة المدخلة هو : اسم
الوزن : فاعِل
جنس الكلمة المدخلة هو : مذكر
دلالة الكلمة المدخلة : بشري
نوع الجذر : صحيح سالم
====================

(a) (b)

Figure 7. Possible inflected forms of the root TJR (a) possible nouns and adjectives forms, and (b) possible broken plural forms of the selected noun (merchant TaAJiR-تَاجِر)

We have also applied a transformational rule using the masdar resource in the NooJ platform. The rule converts a verbal sentence to a nominal one. The sentence: i) transformed to ii) using the transformational rule in Figure 8. Both of these two sentences have the same meaning. We have used masdar resource with verb resource to apply the transformational rule, the verb (you did-عملت) has been converted to the masdar form (your work -عملك).

i) *أسعدني ما عملت.* I like what you did.
ii) *أسعدني عملك.* I like what you have done.



أسعدنـ. مـا عملت

:analyses. Click a solution below to display the corresponding path ٢١٦

Outputs

أسعدنـي عَمَلُك

Figure 8. Transformational grammar in NooJ

NooJ has the possibility to implement transformational rules using syntactic grammar and morphological operations [2], [3]. As we have mentioned before, we have implemented a transformational grammar that transforms a verbal phrase into a nominal one using masdar moawal-المصدر المؤول. The morphological operation $V1 #V2_G#ك takes the value of the V1, which is the first verb (I like-أسعدني) and concatenate it with the masdar form of the second verb. #V2_G returns the MASDAR form which has been generated and assigned as "G". The value of V2 is the inflected form of the verb "to do", which is (you did-عَمِلْتَ) and the linked masdar is (the work - عَمَل).

4. CONCLUSION

We have exploited our resources to developed an application that facilitates Arabic language learning. The application has three main tasks: verb, noun/adjectives and masdar. Verb task allows the user to return verbs that share the entered root. The user will choose a verb to apply the full/semi conjugation function, extract the masdar form/forms of a chosen verb, or extract the verb's linguistic properties. It is worth mentioning that the possible meanings of the chosen verb have also been returned in this task. The noun task returns possible nouns/adjectives that share the entered root; the user will be able to extract the different meanings of the chosen noun/adjective, extract the possible broken plural forms, and extract the linguistic properties of the chosen noun/adjective. masdar task returns masdar forms that share the entered root, return the possible meanings of the chosen masdar, return possible verbs linked to the chosen verb. We have called NooJ-from the application using “noojapply”-to apply a linguistic analysis using our prebuilt resources. We have used the annotations to achieve the previous tasks. We have also implemented transformational grammar using masdar resource and NooJ’s morphological operations. Our perspectives come over: i) convert our dictionary to an ontology based on root and pattern approach and ii) implement additional functions to the application like: produce paraphrasing using our transformational rules.

REFERENCES

[1] M. El-Hannach, "Syntaxe des verbes psychologiques en arabe," Ph.D. dissertation, Paris VI university, Paris, 1988.

[2] M. Silberztein, *Formalizing natural languages: The NooJ approach*. UK: Hoboken, ISTE Ltd and John Wiley & Sons Inc, 2016.

[3] M. Silberztein, *NooJ Manual*. 2003. [Online]. Available : http://www.nooj4nlp.org/files/NooJManual.pdf.

[4] I. Blanchete, M. Mourchid, S. Mbarki, and A. Mouloudi, "Formalizing Arabic Inflectional and derivational verbs based on root and pattern approach using NooJ platform," in *Formalizing Natural Languages with NooJ and Its Natural Language Processing Applications. NooJ 2017. Communications in Computer and Information Science*, S. Mbarki, M. Mourchid, and M. Silberztein, Eds. Cham: Springer, 2018, pp. 52-65, doi: 10.1007/978-3-319-73420-0_5.

[5] S. Bourahma, S. Mbarki, M. Mourchid, and A. Mouloudi, "Syntactic parsing of simple Arabic nominal sentence using the NooJ linguistic platform," in *Arabic Language Processing: From Theory to Practice. ICALP 2017. Communications in Computer and Information Science*, A. Lachkar, K. Bouzoubaa, A. Mazroui, A. Hamdani, and A. Lekhouaja, Eds. Springer, 2018, pp. 244-257, doi: 10.1007/978-3-319-73500-9_18.

[6] A. Al-Taani, M. Msallam, and S. Wedian, "A top-down chart parser for analyzing arabic sentences," *The International Arab Journal of Information Technology (IAJIT)*, vol. 9, no. 2, pp.109-116, 2012.

[7] A. Bounoua, A. Zinedine, M. El Hannach, and R. Kasmi, "Formalization of the Arabic grammatical category (V-a) using the NooJ platform," *Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications*, 2018, pp. 1-6. doi: 10.1145/3230905.3230928.

[8] S. Vietri, "The formalization of Italian lexicon-grammar tables in a NooJ pair dictionary/grammar," *Proceedings of the 2008 International NooJ Conference, Cambridge Scholars Publishing*, 2010, pp. 138-147.

[9] A. Amzali, M. Mourchid, A. Mouloudi, and S. Mbarki, "Arabic psychological verb recognition through NooJ transformational grammars," in *Formalising Natural Languages: Applications to Natural Language Processing and Digital Humanities. NooJ 2020. Communications in Computer and Information Science*, B. Bekavac, K. Kocijan, M. Silberztein, and K. Šojat, Eds. Cham: Springer, 2021, pp. 74–84, doi: 10.1007/978-3-030-70629-6_7.

[10] I. Blanchete, M. Mourchid, S. Mbarki, and A. Mouloudi, "Arabic broken plural generation using the extracted linguistic conditions based on root and pattern approach in the NooJ platform," in *Formalizing Natural Languages with NooJ 2018 and Its Natural Language Processing Applications. NooJ 2018. Communications in Computer and Information Science*, I. Mirto, M. Monteleone, and M. Silberztein, Eds. Cham: Springer, 2019, pp. 27–37, doi: 10.1007/978-3-030-10868-7_3.

[11] A. Z. Al-Khamayseh, "The definite articles in the semitic languages: a comparative study," *Journal of Literature, Languages and Linguistics*, vol. 46, pp. 7–17, 2018.

[12] S. Mesfar, "Analyse morpho-syntaxique automatique et reconnaissance des entités nommées en arabe standard," Ph.D. dissertation, Université de Franche-Comté, France, 2008.

[13] A. A. Neme and E. Laporte, "Pattern-and-root inflectional morphology: the Arabic broken plural," *Language Sciences*, vol. 40, pp. 221-250, 2013, doi: 10.1016/j.langsci.2013.06.002.

[14] H. Fehri and I. Ben Messaoud, "Construction of educational games with NooJ," in *Formalizing Natural Languages with NooJ 2019 and Its Natural Language Processing Applications. NooJ 2019. Communications in Computer and Information Science*, H. Fehri, S. Mesfar, and M. Silberztein, Eds. Cham: Springer, 2020, pp. 173–184, doi: 10.1007/978-3-030-38833-1_15.

[15] H. Fehri, M. A. F. Seideh, and S. Dardour, "A decision-support tool of medicinal plants using NooJ platform," in *Automatic Processing of Natural-Language Electronic Texts with NooJ. NooJ 2016. Communications in Computer and Information Science*, L. Barone, M. Monteleone, and M. Silberztein, Eds. Cham: Springer, 2016, pp. 246–257, doi: 10.1007/978-3-319-55002-2_21.

[16] I. Blanchete, M. Mourchid, S. Mbarki, and A. Mouloudi, "Arabic learning application to enhance the educational process in Moroccan mid-high stage using NooJ platform," in *Formalizing Natural Languages with NooJ 2019 and Its Natural Language Processing Applications. NooJ 2019. Communications in Computer and Information Science*, H. Fehri, S. Mesfar, and M. Silberztein, Eds. Cham: Springer, 2020, pp. 149–160, doi: 10.1007/978-3-030-38833-1_13.

[17] A. El Kah, I. Zeroual, and A. Lakhouaja, "Application of Arabic language processing in language learning," in *Proceedings of the 2nd international Conference on Big Data, Cloud and Applications*, 2017, pp. 1-6, doi: 10.1145/3090354.3090390.

[18] B. Azman, "Root identification tool for Arabic verbs," *IEEE Accessed*, vol. 7, pp. 45866-45871, 2019, doi: 10.1109/ACCESS.2019.2908177.

[19] A. H. Aliwy and A. A. Alethary, "Development of Arabic sign language dictionary using 3D avatar technologies," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 21, no. 1, pp. 609-616, Jan. 2021, doi: 10.11591/ijeecs.v21.i1.pp609-616.

[20] S. El-Beltagy, A. Rafea, "A corpus based approach for the automatic creation of Arabic broken plural dictionaries," *2013, Computational Linguistics and Intelligent Text Processing, CICLing. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2013, vol 7816, pp 89-97, doi: 10.1007/978-3-642-37247-6_8.

[21] N. Habash, R. Marzouk, C. Khairallah, and S. Khalifa, "Morphotactic modeling in an open-source multi-dialectal Arabic morphological analyzer and generator," *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 2022, pp. 92-102. doi: 10.18653/v1/2022.sigmorphon-1.10

[22] N. Mia, S. Sopian, S. Nalahuddin, and S. Syihabuddin, "Why is masdar important? an investigating of masdar and its translation," A*LSUNIYAT Jurnal Penelitian Bahasa Sastra dan Budaya Arab*, vol. 5 no. 1, pp. 82-93, 2022, doi: 10.17509/alsuniyat.v5i1.44843.

[23] L. Laks and E. Saiegh-Haddad, "Between varieties and modalities in the production of narrative texts in Arabic," in *Developing Language and Literacy. Literacy Studies*, R. Levie, A. Bar-On, O. Ashkenazi, E. Dattner, and G. Brandes, Eds. Cham: Springer, 2022, pp. 247–271, doi: 10.1007/978-3-030-99891-2_15.

[24] H. Abdul Sattar, *Fundamentals of classical Arabic*, Chicago: Sacred Learning, 2002.

[25] J. Kremers, "The formation of deverbal nouns in Arabic draft," 2007, Accessed: Jan. 24, 2022. [Online]. Available: https://www.academia.edu/2271061/The_formation_of_deverbal_nouns_in_Arabic_draft. (Accessed Sep. 01, 2019).

BIOGRAPHIES OF AUTHORS

Ilham Blanchete was born in Russia, she received the B.Sc. of Engineering in software engineering and information systems from the faculty of Information Technology, Damascus, Syria. She is currently in the final stage to get her Ph.D. on computer science field of computational linguistic, project of Arabic Natural Language Processing. Her research interests include software engineering, computational linguistic and ANLP. She can be contacted at email: ilham.blanchete@gmail.com.

Mohammed Mourchid Doctorate Degree in Computer Science In 1999; Associate Professor at The Computer Science Department at The Faculty of Sciences, Ibn Tofail University in Kenitra Morocco He is a supervisor of computational linguistic projects. He has been working on an Arabic dictionary since 2017 in the same department. On going research interests: natural language processing, web semantic, and information systems. He can be contacted at email: mourchidm@hotmail.com.