Arabic NooJ Parser: Nominal Sentence Case

Nadia Ghezaiel Hammouda[1] and Kais Haddar[2(✉)]

[1] Miracl Laboratory, Higher Institute of Computer and Communication Technologies of Hammam Sousse, Sousse, Tunisia
ghezaielnadia.ing@gmail.com
[2] Miracl Laboratory, Faculty of Sciences of Sfax, University of Sfax, Sfax, Tunisia
kais.haddar@yahoo.fr

Abstract. Parsing Arabic corpora is an important task aiming to understand Arabic language, enrich and enhance the electronic resources, and increase the efficiency of natural language applications like translation or the recognition. In this paper, we propose a parsing approach for Arabic sentences especially for nominal ones. To do this, we first study the typology of the Arabic nominal sentence. Then, we develop a set of rules generating different nominal sentences. After that, we present our parsing approach based on transducers and on our tag set. In addition, we transform recursive graph of transducers into transducer cascade to reduce the complexity. Finally, we present the implementation and experimentation of our approach in NooJ platform. The obtained results are satisfactory.

Keywords: Arabic sentence · Recursive graph · Topic · Attribute
NooJ transducer

1 Introduction

Parsing Arabic corpora is an important task aiming to understand Arabic language, enrich and enhance the electronic resources, and increase the efficiency of naturel language applications like translation or the recognition. Arabic is considered as one of the difficult language to analyze due to its morphological, syntactic, phonetic and phonological characteristics. There are two types of sentences in Arabic: the verbal sentence and the nominal sentence.

There are different forms of the nominal sentence that can interact with verbal sentences. The formalization of rules requires much effort to guarantee several qualities like efficiency, robustness and extensibility. Transducers have proved their usefulness in a wide variety of applications in NLP [16]. Transducer cascades made possible to carry out robust and highly precise syntactic analysis on different corpora.

Transforming recursive graph of transducers into transducer cascade is very interesting. The transformation is a difficult task due to the difference between the application levels in every path and the interaction of the linguistic phenomena. For the cascade, the order of transducers should respect different constraints, which are deduced from observations done on Arabic corpora.

S. Mbarki et al. (Eds.): NooJ 2017, CCIS 811, pp. 69–80, 2018.
https://doi.org/10.1007/978-3-319-73420-0_6

Our objective is to construct an Arabic parser implemented in NooJ. To do this, we will study, essentially, the Arabic nominal sentences but also other sentence forms. Then we will establish a set of rules recognizing nominal sentences that can be generalized to treat any sentence type. Finally, we will implement the transducer cascade in NooJ.

In this paper, we begin by stating the different approaches, which allow the parsing and annotation of Arabic corpora. Then, we perform a study about the forms of Arabic nominal sentences. Next, we establish syntactic rules transformed in transducers. In addition, we implement and test all these rules in the NooJ platform respecting the cascade notion. Finally, we provide a concise conclusion and we give some future perspectives.

2 Previous Work

Many works aim to analyze Arabic corpora with different approaches: rule-based, statistical or hybrid approach. In [1], the authors have proposed a method for Arabic lexical disambiguation based on the hybrid approach. In [2], the author has developed a morphological syntactic analyzer for the Arabic language within Lexical Functional Grammar formalism. The developed parser is based on a cascade of finite-state transducers and a set of syntactic rules specified in Xerox Linguistics Environment. Also, in [3], the authors have proposed a rule-based approach for tagging non-vocalized Arabic words. In [4], the authors have designed an automatic tagging system by adding the part-of-speech tag in the Arabic text. In addition, in [5], the authors have presented an Arabic parser for Arabic nominal sentences. In this work, the HPSG formalism is used.

In addition, there are many other statistical and hybrid works. In [6], the proposed method of parsing dealt with the 'alif-nûn' sequence in a given sentence. This method is based on the context-sensitive linguistic analysis to select the correct sense of the word in a given sentence without doing a deep morpho-syntactic analysis.

Besides, in the last decades, many researchers have worked on systems, which aim to disambiguate Modern Standard Arabic. Among those systems, we mention MADA and TOKAN systems [7]. They are two complementary systems for the Arabic morphological analysis and disambiguation process. Their applications include high-accuracy part-of-speech tagging, discretization, lemmatization, disambiguation, stemming and glossing. In [8], the system AMIRA developed at Stanford University includes a tokenizer, a part of speech tagger (POS) and a Base Phrase Chunker (BPC). The model used by AMIRA is a supervised learning machine with no explicit dependence on knowledge of deep morphology. Concerning the finite state tools, we find the Xerox parser [9], which is based on finite state technology, tools (e.g. xfst, twolc, lexc,) for NLP. These tools have been used to develop the morphological analysis, tokenization, and shallow parsing of a wide variety of natural languages.

Moreover, there are several parsing works performed with the NooJ platform. In [10], the authors proposed a method to identify all possible syntactic representations of the Arabic relative sentences. The authors explain the different forms of relative clauses and the interaction of relatives with other linguistic phenomena such as ellipsis and

coordination. We can cite also the work described in [11] to analyze the Arabic broken plural. This work is based on a set of morphological grammars used for the detection of the broken plural in Arabic texts. Arabic broken plural analysis can facilitate the parsing because we can distinguish between different types of nouns.

3 Arabic Lexical Ambiguity

The Arabic language is written and read from right to left. The alphabet has 28 consonants, adopting different spellings according to their position (at the beginning, middle or end of a lexical unit). Arabic token is written with consonants and vowels. The vowels are added above or below the letters. The presence of vowels allows us to understand text and disambiguate different words. In Arabic, the word should respect a well-defined type hierarchy. Indeed, a word can be either a verb or a name or a particle. Each type itself is detailed in several subtypes. Thus, any specific linguistic information to the Arabic language should be represented through this hierarchy. Before beginning our study of lexical ambiguity, we give an overview about some specificities of Arabic language. Indeed, the Arabic sentence is characterized by a great variability in the order of its words. In general, in Arabic, we put at the beginning of the sentence the word (noun or verb) on which we want to attract the attention and at the end the richest term to keep the meaning of the sentence. This variability in the order of words causes artificial syntactic ambiguities. So in the grammar, we should give all possible combinations of inversion rules for the word order in the sentence. Note that the Arabic sentence can be either verbal or nominal.

Arabic lexical ambiguity has several causes, but we focus mainly on five of them.

Unvocalization: It can cause lexical ambiguities because a word in Arabic language can be read differently in a sentence, depending on its context. For example, the word *kaataba* can refer to the noun (the writer), or the verb *to write* in English.

Emphasis sign (Shadda ّ): In Arabic, the emphasis sign Shadda is equivalent to writing the same letter twice. The insertion of Shadda can change the meaning of the word. For example, the word *darasa* means *lessons* (noun) while *darrasa* means *he taught* (verb).

Hamza sign: The presence of Hamza sign (hamzah) reduces ambiguity. If we add the Hamza to a word then the number of ambiguities decreases. As an example, the word Faas can be a city or an ax.

Agglutination: In Arabic, particles, prepositions, pronouns, can be attached to adjectives, nouns, verbs and particles. This characteristic can generate many types of lexical ambiguity. For example, the letter *faa'* in the word *fa-slun* (season) is part of the root while in the word *fasala* (then he prayed) is a prefix.

Compound words: Lexical ambiguity sometimes derives from compound words. For example, the compound noun "الحاسوب المحمول" hassub mahmul can be interpreted as a laptop or a portable pc.

4 Typology of Nominal Sentence

As we have mentioned, the Arabic language has two types of sentences: the nominal sentence and the verbal one. In the following section, we will present the typology of the nominal sentence. The nominal sentence is any sentence beginning with a noun and can contain a verbal sentence as a component. Also, each nominal sentence is composed of a topic (*Mubtada '*) and an attribute (*Khabar*) and the attribute is compatible with the topic in gender and number. From this definition, we can identify several types of the Arabic nominal sentence.

4.1 Structure of Nominal Sentence

The topic and the attribute can be presented in many forms. In what follows, we detail these forms. In our study, we concluded that the topic could have many forms. It can be a single word, a phrase or a sentence.

(a) The case of a single noun: In this case, the topic can be a proper noun (name of person, geographical name, etc.) or a common noun. Also, it can be a personal pronoun, a demonstrative pronoun or an interrogative pronoun. Examples from (1) to (4) illustrate this case.

(1) **مريم** جميلة
Mariam [is] beautiful
(2) **الطاولة** مستديرة
The table [is] round
(3) **أنت** جميلة
You are beautiful
(4) **هذا صديقي**
This is my friend

(b) The case of a nominal phrase: In this case, the topic can be a phrase of annexation, an adjectival phrase, a relative clause or a phrase of conjunction. Also, each one of those phrases can be recursive or contain one of the other. To illustrate this case we present the following examples:

(6) **باب الحديقة** جميل
The door of the garden is beautiful
(7) **باب حديقة المنزل** جميل
The door of the garden's home is beautiful

The example (6) presents a phrase of annexation which is composed of an indefinite noun (باب) and a definite noun (الحديقة), but the example (7) presents a recursive phrase which contains another phrase of annexation (حديقة المنزل) and (حديقته).

The attribute is manifested in several forms. It can be a unique word, a phrase or a verbal sentence.

(a) The case of a unique word: In this case, the attribute can be a noun, a personal pronoun, an intransitive verb, or an adjective. We illustrate this case by examples (8) and (9).

(8) العلم **نور**
Knowledge is the light
(9) الولد **يضحك**
The boy laughs

(b) The case of a phrase: generally the attribute is in the form of a phrase. It can be a nominal phrase (example (10)), a prepositional phrase (example (11)) or a relative phrase (example (12)).

(10) الهدهد **طائر جميل**
Hoopoe is a beautiful bird
(11) الأولاد **في المدرسة**
Boys are at school
(12) محمد **الذي نجح في الإمتحان لمثابرته**
Mohammed who passed the exam, due to his perseverance

(c) The case of a sentence: the attribute can be a verbal sentence or a nominal sentence. To illustrate this case, we present the following examples:

(13) المديرة **تمنح التلاميذ المميزين الجوائز**
The director gives presents to distinguished students
(14) الله **هو النور** الأعظم
Allah is the greatest light

In the example of (13), the attribute is a verbal sentence. On the other hand, the attribute of example (14) is a nominal sentence.

4.2 Other Types of Nominal Sentence

In Arabic, the nominal sentence can be introduced by particles such as the particle *Inna* or defective verbs such as the verb *Kaana*. The insertion of defective verbs or particles in a nominal sentence can change the joint of the topic and the attribute. In fact, the particle *Inna* accepts a subject and a predicate through dependencies called *Ism*

inna (اسم إن) and *khabar inna* (خبر إن). The subject *ism inna* is always in the accusative case *manṣūb* (منصوب) and the predicate *khabar inna* is always in the nominative case *marfū*. The example of sentence (15) uses the particle *Inna* the topic becomes accusative but the attribute stays nominative. The same sentence of (16) without the particle *Inna* keeps its characteristics.

(15) إنّ الإمتحانَ صعبٌ
The exam is difficult
(16) الإمتحانُ صعبٌ
The exam is difficult

5 Formalization of Lexical Rules

We carried out a linguistic study, which allowed us to identify lexical rules and resolve several forms of ambiguity. The identified rules were classified through the mechanism of subcategorization for verbs, nouns and particles [12].

Particles can be subdivided into three categories: particles acting on nouns, particles acting on verbs and particles acting on both nouns and verbs. There are Arabic particles which must be followed by a noun like prepositions and particles of restriction {مِنْ، إلى، عنْ، على، في، ب، لِ، ك، حتّى، رُبَّ، واو القسم، ت، حاشا، خلا،عدا}.

Particles can also be followed by a verb, like subjunctive particles, apocopate particles, prohibition particles. As an example, if we find a subjunctive particle like {لن/ كي/ حتى/ لام التعليل/ إذن/ فاء السببية/ وأو المعية/ لام الجحودأن/}, then, it should be followed by a verb. A noun or a verb can follow some particles. To solve this ambiguity, we studied the context of the sentence.

We can apply the principle of sub-categorization to resolve the ambiguity related to verbs. We based essentially on the transitivity feature of verbs. In Arabic, a verb can be intransitive, transitive, di-transitive and tri-transitive. Either transitive or intransitive verbs can be transformed to transitive verbs with prepositions. The mechanism of transitivity is explained by the above sentences. Note that these examples respect the VSO order.

We can also apply the principle of sub-categorization to resolve the ambiguity linked to nouns. We based essentially on successors feature of nouns. In Arabic, a noun can be defined with 'ال' '*alifLam'* or be indefinite. Each one of these types has its followers. The defined noun can be followed by a noun phrase (NP), a defined adjectival phrase (AP), a prepositional phrase (PP), a relative phrase (RP) or an empty set (∅). Besides, the non-defined noun can have the same followers but the AP should be non-defined. Note that, the nominative, accusative or genitive criteria will be inherited by the nominal group.

To implement our rules, we use the linguistic platform NooJ which is a linguistic environment to build and manage electronic dictionaries and grammars with wide coverage and to formalize various language levels: spelling, inflectional and derivational morphology, lexicon of simple words, compound words and idioms, local syntax and

disambiguation, structural and transformational syntax, semantics and ontologies. Also, formalized descriptions can then be used to process and analyze texts and large corpus.

6 Proposed Method

Our proposed approach of analysis consists of two main phases: the segmentation and the parsing.

The segmentation phase [13] consists of the identification of sentences based on punctuation signs. Each identified sentence is delimited by an XML tag. As an output of this phase, we obtain an XML document for the corpus, and it will be the input for the pre-processing phase. The second phase consists of the agglutination's resolution using morphological grammars. As an output of this phase, we obtain a Text Annotation Structure (TAS) containing all possible annotations for corpus's sentences. The obtained TAS is the input of the third phase. Then, we identify the appropriate lexical category of each word in the sentence to construct different sentence phrases. This identification is based on several syntactic grammars specified with NooJ transducers. Transducer's applications respect a certain priority from the most evident and intuitive transducer until arriving at the least one (Fig. 1). The output of the parsing phase will be a disambiguated TAS containing right paths and right annotations. Note that we used a high granularity's level for lexical categories. This distinction between nominative, accusative and genitive modes for nouns can resolve the absence of vocalization. Another remark, we have tested two methods to analyze Arabic nominal sentences.

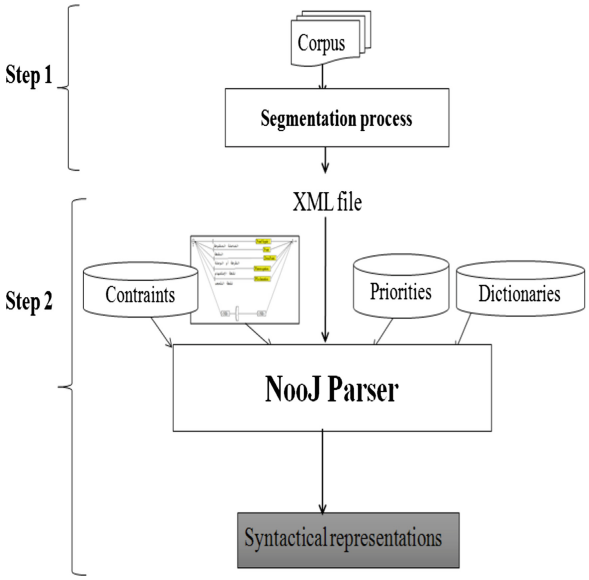

Fig. 1. Proposed method

7 Implementation

The extracted rules have been implemented in the NooJ platform [14]. In fact, the process of parsing is based on the set of the developed NooJ transducers and a tag set that is indicated in the following (Table 1).

Table 1. Used tag set

NN	Indefinite Nominative Noun u
NTN	Indefinite Nominative Noun un
NND	Definite Nominative Noun u
NA	Indefinite Accusative Noun a
NTA	Indefinite Accusative Noun an
NAD	Definite Accusative Noun a
NG	Indefinite Genitive Noun i
NTG	Indefinite Genitive Noun in
NGD	Definite Genitive Noun i

In this part, we will explain different stages in our cascade approach by giving an idea about the recursive approach.

7.1 Segmentation Phase

The implementation of the segmentation phase is based on a set of developed transducers in the NooJ linguistic platform. This set contains 9 graphs representing contextual rules. The main transducer adds an XML tags <S> to delimit the frontiers of a sentence.

7.2 Preprocessing Phase

The implementation of the preprocessing phase is based on a set of morphological grammars and dictionaries [15] existing in the NooJ linguistic platform (Table 2). This implementation resolves all forms of agglutination. The outputs contain all possible lexical categories of each word in sentences.

Table 2. Table summarizing morphological grammars

Morphological grammars	Numbers
Verb inflected form patterns	113
Inflected relative pronoun patterns	8
Broken plural patterns	10
Agglutination's grammars	3

7.3 Analysis Phase with Recurisve Graphs

Figure 2 illustrates the NooJ implementation of rules for nominal sentences.

In fact, Fig. 2 shows different forms of topics and attributes. A nominal sentence can be formed by a nominative topic followed by a nominative attribute. Also, we can find the modal verb "KANA" followed by a nominative topic and an accusative attribute. In addition, we find the modal verb "INNA" followed by an accusative topic and a nominal attribute. In the case of a simple nominal phrase, the topic and the attribute should have the same joint. They respect the nominative form.

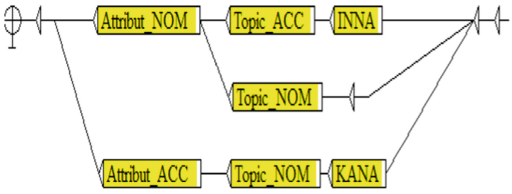

Fig. 2. Transducer representing a lexical rule for a nominal sentence

Figure 3 shows that the topic can be a unique word, a unique noun phrase or recursive one. Note that the subgraph PP represents the different forms of the prepositional phrases and the subgraph NP_NOM represents the noun phrase. For a nominative attribute, the appropriate transducer is given in Fig. 4.

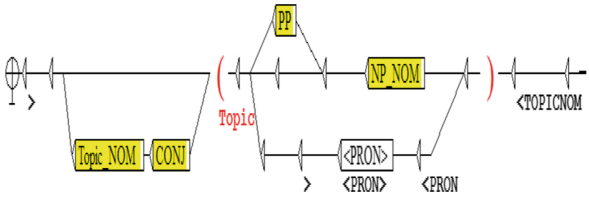

Fig. 3. Transducer representing a rule for a nominative topic

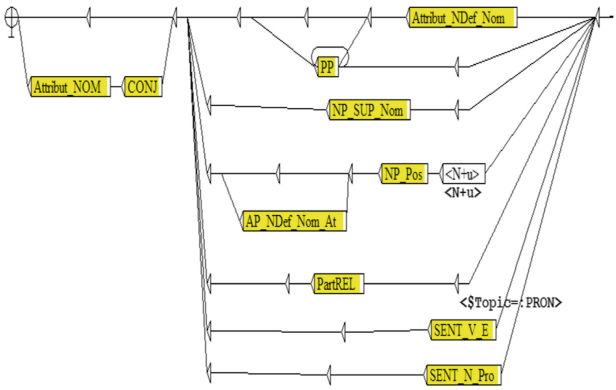

Fig. 4. Transducer representing a lexical rule for a nominative attribute

7.4 Cascade for Parsing

A separated transducer implements each nominal sentence component. In what follows, some transducers respecting the proposed approach are given.

Figures 5, 6 and 7 show how our cascade works and show that the different transducers use automatically the calculated output.

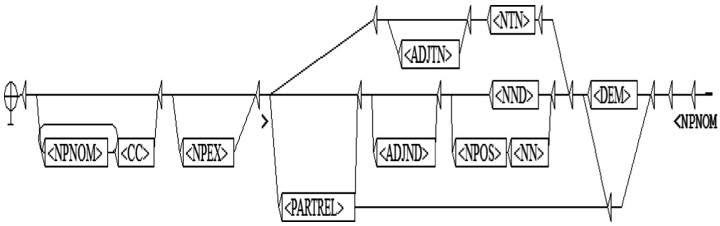

Fig. 5. Transducer for nominative NP

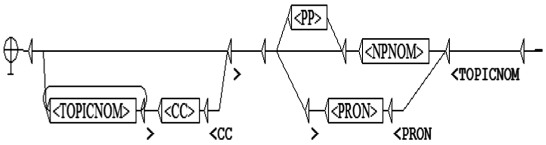

Fig. 6. Transducer for a topic

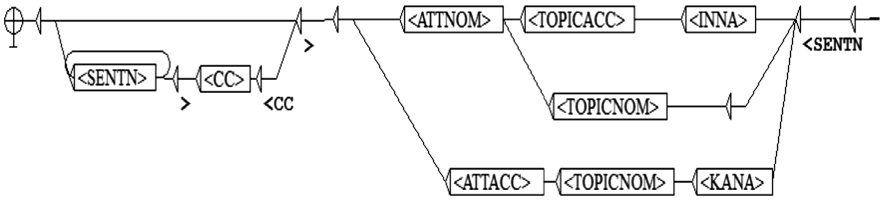

Fig. 7. Transducer for nominal sentence

8 Experimentation and Evaluation

To experiment our approach, we implemented our cascade of transducers in NooJ platform. Then, we compared the cascade with recursive transducers in the case of the nominal sentence. In fact, fixing the call order of transducers was inspired by our study. To be more specific, the idea consists of starting with phrases until gathering the sentence entirely: Particles → Phrases → Sentences. The implemented syntactic transducer cascade contains in total 50 graphs called in a fixed order. This is illustrated in the following (Fig. 8).

Syntactic Resources:

Order	Grammar
1	Cas_CONJ.nog
2	Cas_Daref.nog
3	Cas_PrepZamen.nog
4	Cas_PrepPART.nog
5	Cas_DEM.nog
6	Cas_KANA.nog
7	Cas_INNA.nog
8	Cas_ProREL.nog
9	Cas_TOOL.nog
10	Cas_NG.nog

Fig. 8. Syntactic resources

To evaluate our prototype, we calculate also the precision, the recall and the F-measure for two approaches using respectively recursive graphs and cascade, as illustrated in Tables 3 and 4.

Table 3. Table summarizing the precision and recall measures for recursive graphs

Corpus	Precision	Recall	F-measure
5900 sentences	0.6	0.7	0.64

Table 4. Table summarizing the precision and recall measures for the proposed cascade

Corpus	Precision	Recall	F-measure
5900 sentences	0.74	0.84	0.77

The obtained values of these measures are interesting and show that a cascade method is better than a recursive one. This result can be improved by adding other rules and heuristics.

9 Conclusion

In this paper, we have proposed a parsing method dealing with the Arabic nominal sentences. This method is based on a set of transducers and a high level of granularity. This method is implemented in the NooJ platform and used a cascade instead of recursive graph. The elaborated parser can annotate the Arabic corpora. So, we did a study on different forms of Arabic nominal sentences. This study allowed us to establish a set of rules for parsing Arabic nominal sentences. The established rules are specified with NooJ transducers. The proposed cascade of transducers reduces the parsing complexity. Thus, an experiment is performed on a set of nominal sentences, mainly from stories. The obtained results are satisfactory, which is proved by the calculated measures. Concerning the future works, we want to enrich our linguistic resources by improving our dictionaries and transducers.

References

1. Shaalan, K., Othman, E., Rafea, A.: Towards resolving ambiguity in understanding Arabic sentence. In: The Proceedings of the International Conference on Arabic Language Resources and Tools, NEMLAR, 22nd–23rd September, Cairo, Egypt, pp. 118–122 (2004)
2. Attia, M.: Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machine translation (2008)
3. Al-Taani, A.T., Al-Rub, S.A.: A rule-based approach for tagging non-vocalized Arabic words. Int. Arab J. Inf. Technol. (IAJIT) **6**(3), 320–328 (2009). 4 Diagrams, 5 Charts, 1 Graph
4. Diab, M., Hacioglu, K., Jurafsky, D.: Automatic tagging of Arabic text: from raw text to base phrase chunks. Linguistics Department, Stanford University (2004)
5. Hadddar, K., Abdelkarim, A., Ben Hamadou, A.: Étude et analyse de la phrase nominale arabe en HPSG. In: TALN 2006 (2006)
6. Dichy, J., Alrahabi, M.: Levée d'ambiguité par la methode d'exploration contextuelle: la sequence 'alif-nûn' en arabic. In: Second International Conference (SIIE) (2009)
7. Habash, N., Rambow, O., Roth, R.: MADA+TOKAN: a toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming ald lemmatization. In: Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt (2009)
8. Diab, M.: Second generation tools (AMIRA 2.0): fast and robust tokenization, POS tagging, and base phrase chunking. In: MEDAR 2nd International Conference on Arabic Language Resources and Tools, April, Cairo, Egypt (2009)
9. Beesley, K.: Finite-state morphological analysis and generation of Arabic at xerox research: status and plans. In: ACL/EACL 2001, 6th July, Toulouse, France (2001)
10. Zalila, I., Haddar, K.: Construction of an HPSG grammar for the Arabic relative sentences. In: Natural Language Processing, RANLP 2011, 12–14 September 2011, Hissar, Bulgaria (2011)
11. Ellouze, S., Haddar, K., Abdelhamid, A.: Etude et analyse du pluriel brisé arabe avec la plateforme NooJ (2009)
12. Hammouda, N.G., Haddar, K.: Toward the resolution of Arabic lexical ambiguities with transduction on text's automaton. In: CICLing (2015)
13. Hammouda, N.G., Haddar, K.: Integration of a segmentation tool for Arabic corpora in NooJ platform to build an automatic annotation tool. In: Will appears in NooJ (2016)
14. Silberztein, M.: Disambiguation tools for NooJ. In: Proceedings of the 2008 International NooJ Conference, pp. 158–171. Cambridge Scholars Publishing, Newcastle (2010)
15. Mesfar, S.: Analyse morpho-syntaxique automatique et reconnaissance des entités nommées en arabe standard. University of Franche Comté, p. 235 (2008). Thesis
16. Gross, M.: The Construction of Local Grammar. Finite-State Language Processing, pp. 329–354. MIT Press, England (1997)