Syntactic Parsing of Simple Arabic Nominal Sentence Using the NooJ Linguistic Platform

Said Bourahma, Samir Mbarki(✉), Mohammed Mourchid, and Abdelaziz Mouloudi

MISC Laboratory, Faculty of Science, Ibn Tofail University, Kenitra, Morocco
saidbrh@yahoo.fr, mbarkisamir@hotmail.com, mourchidm@hotmail.com, mouloudi_aziz@hotmail.com

Abstract. Natural Language Processing (NLP) applications such as machine translation, question answering, knowledge extraction, and information retrieval require parsing process as an essential step. In this paper, we present a parser to analyze simple Arabic nominal sentences using the NooJ platform. Hence, we propose a well-classified NooJ dictionary that includes most syntactic, and semantic features. We also present the rule describing the Arabic sentence. Then, we implement the parser that recognizes, and annotates all possible grammatical structures of simple Arabic nominal sentence. We implement a set of transducers modeling Arabic lexical, and syntactic constraints, these constraints reduce parsing ambiguity. Our parser is tested on many sentences extracted from real texts. These experimental results show the effectiveness of the proposed parser for analyzing simple Arabic nominal sentences.

Keywords: Natural Language Processing · Arabic language parser
Syntactic analysis · NooJ linguistic platform

1 Introduction

Natural Language is the language spoken by humans. Any language is based on a vocabulary which consists of a set of words. This group of words must match a set of grammatical rules. A sequence of words, from the vocabulary, form a text, and the set of all possible texts defines the language. NLP is a subfield of Artificial Intelligence and linguistic, devoted to make computers understand statements written in natural language [7, 13]. In fact, NLP employs computational techniques for the purpose of learning, understanding, and producing natural language content. Actually the natural language processing requires relevant information about the language at different levels. Therefore, we may be able to use four knowledge levels about the language: morpho-lexical, syntactic, semantic, and pragmatic. These levels overlay each other. Each level only focuses on a given issue related to that level.

In the NooJ linguistic platform, syntactic grammars are very useful to describe the words sequence which has a meaning [14]. Hence we can use them in order to focus on various kinds of simple nominal sentences. NooJ guarantees high integration of all levels of description thanks to compatible notations and a unified representation for all linguistic analysis results, enabling different analyzers at different linguistic levels to

A. Lachkar et al. (Eds.): ICALP 2017, CCIS 782, pp. 244–257, 2018.
https://doi.org/10.1007/978-3-319-73500-9_18

communicate with one another [13, 16]. The aim of our work is to develop a syntactic parser of simple Arabic nominal sentences. This parser is based on a set of structural grammars. These grammars are implemented in the NooJ platform. In general, Arabic texts are not diacritized. So these texts become ambiguous. That is why the disambiguation of the sentence components is also expected in this work.

The rest of this paper is organized as follows: Sect. 2 is dedicated to related work, Sect. 3 describes our contribution; we have three subsections in this part: the lexicon classification, the disambiguation, and mapping between the lexicon classes and the nominal sentence components. In Sect. 4, we present the main NooJ platform functionalities. Section 5 explains the implementation of our Simple Arabic nominal sentence syntactic parser. Section 6 is devoted to the running and the test of our parser on different sentences. Finally, the last part will present the conclusion and the future work.

2 Related Work

In literature, many approaches were applied to design and implement a syntactic analyzer for parsing Arabic sentences [1, 6, 9, 10]. Actually there are three main approaches: linguistic, statistical, and hybrid. The linguistic methods are based on lexicon and grammars. This approach lacks of resources, for instance, the Arabic grammars do not cover all sentences' types. It is often said that linguistic methods are costly to implement because they require the construction of dictionaries and grammars. However, statistical methods also require a great deal of work to manually construct their reference corpora. The hybrid approach incorporates linguistic rules and corpora-based statistics. So the strengths of both linguistic and statistical approaches to NLP can be combined in a single framework. The other shortcoming of statistical methods is that it relies on reference corpora. So, if the reference corpora contain so many errors, we cannot expect reliable results. Regarding the Arabic language, most of syntactic analyzer developed are based on statistical approach.

3 Methodology and Contribution

This section is devoted to our contribution. The aim of our work is to develop a syntactic parser for simple Arabic nominal sentences. As described in Fig. 1, our methodology is completely based on a linguistic approach. Therefore, we apply three main steps: lexicon classification, disambiguation, and grammar modeling regarding the simple Arabic nominal sentence structure. We have already defined a dictionary [4]. But these entries are not accurately classified. That is why we carry on a new lexicon classification. This classification is very helpful in the last step. In a previous work [8], we also implemented morpho-syntactic rules for processing agglutination using the NooJ platform. The morphological analysis result leads to multiple annotations for the same word. Hence, the disambiguation is required. Finally, from the simple Arabic nominal sentence structure, we map the sentence classes with the nominal sentence components. All these steps are described below:

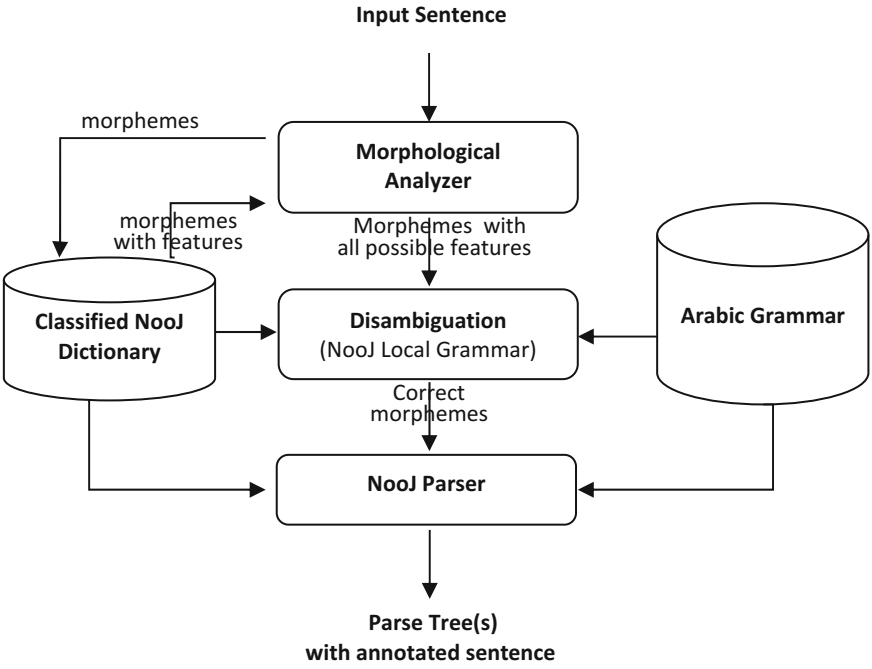

Fig. 1. Analysis steps of Arabic sentence.

3.1 Lexicon Classification

The Arabic language words are divided into three main classes: nouns, verbs, and particles [2, 3, 11, 12, 17, 20–22]. We can eventually add an extra class named "others" or "residuals". This class includes "borrowed words", "numbers", etc. Each class is in turn divided into sub-classes. For example, "complete noun" (الإسم التام, al-'ism al-ttām) and "incomplete noun" (الإسم الناقص, al-'ism al-nnāqiṣ) are two sub-classes for the noun.

A noun in Arabic language is defined as a word which has a meaning without being connected with the notion of time. This class includes pronouns. It also contains "adjectives", "verbal nouns", "noun of place", "proper nouns", etc. as sub-classes. The sub-class "adjectives" has also sub-classes: "resembling adjective" (الصفة المشبهة, al-ṣṣifah al-mušabbahah), "active participles" (اسم الفاعل,'ism al-fā'il), "passive participles" (اسم المفعول,'ism al-maf'ūl), etc.

A verb in Arabic language is a word with two features: action and time. When it is used in a specific context, the verb, prefixed by some particles such as futurity and interrogation particles, get more information about tense, form and meaning. In fact, the "verbs" class is divided into two main sub-classes: "complete verbs" (الفعل التام, al-fi'l al-ttām) and "incomplete verbs" (الفعل الناقص, al-fi'l al-nnāqiṣ). These features are

related to the flexion of the verb. We can also classify verbs regarding many features. One of them could be syntactic feature which involves deeming important property "verb transitivity". Hence a verb could be either "transitive" or "intransitive". The transitive verbs in Arabic handle from one to three accusative forms. Besides the syntactic classification, we can classify verbs regarding semantic features.

Unlike nouns and verbs, particles do not have a meaning regardless of nouns, verbs or particles. The particle can assume the role of a linker between sentences, a linker between words (verbs and nouns), a prefix or suffix, or a sentence modify. They can modify the sentence tense or the sentence meaning. Arabic grammarians divide the "particle" class into three sub-classes: those which are related to the verb (سوف, sawfa, will), "prepositions" are related to the noun, and those which are related to both of them such as "conjunctions" [18, 19]. We can also classify these sub-classes. The sub-sub-classes highlight the grammatical function of the particle.

3.2 Disambiguation

As result of the morphological analysis, a word can have many annotations. For example, in the sentence: silver and money are in the case (فضة و مال في الحقيبة, fiḍḍatun wa mālun fī al-haqībati). As the sentence is not diacritized, the third word could be annotated as the name money (مَال, māl) or as the verb tilt (مَالَ, māla). The disambiguation here is obvious because silver (فضة, fiḍḍah) is a name and the preposition "and" (و, wa) cannot link a name with a verb. In addition, the word placed after prepositions must be a noun. And the word placed after particles affecting verbs must be a verb. In the sentence: رجل قوية, the first word can be either the name man (رَجُل, raǧul) or the name feet (رِجْل, riğlun). Regarding the attributive and predicative adjectives agreement, the predicative adjective "strong" (قوية, qawiyaah) is feminine. So رجل must be feet (رِجْل, riğl) that is also feminine. The syntactic rules enable us to do automatic disambiguation [5, 15].

Our disambiguation approach is based on cooperation between the morphological analyzer and the parser. The morphological analyzer produces all possible interpretations of the textual Arabic word. Disambiguation would be resolved by applying certain types of constraints that are defined with the grammar rules (See Fig. 2). These constraints lead to a correct parse, it could resolve the ambiguity.

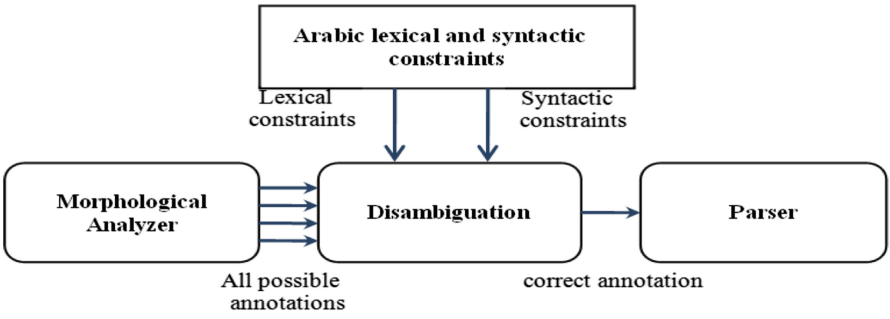

Fig. 2. Disambiguation schema.

3.3 Mapping Between the Lexicon Classes and the Nominal Sentence Components

All natural languages share the same structure which mainly consists of a nuclear predication with eventually extra elements (complement). The nuclear predication is mandatory. In Arabic, the attribution (الإسناد, al-'isnād) is the predication relation which holds between two syntagms in a sentence. The Arabic sentence is comprised of two required components: the predicate and the subject (المسند و المسند إليه, al-musnad wa al-musnad-'ilayh), which affect the sentence meaning [2, 3, 20, 21]. The predicate may precede or follow the subject whether in the nominal or the verbal sentence. Pre-position (التقديم, a-ttaqdīm) and post-position (التأخير, a-tta'ẖīr) are restricted by some conditions (أحكام التقديم و التأخير, aḥkām al-ttaqdīm wa al-tta'ẖīr). In the case of a nominal sentence, the predicate is the comment (الخبر, al-ẖabar) and the subject is the topic (المبتدأ, al-mubtada'). When the subject follows the predicate, we have to do with a pre-posed comment (خبرمقدم, ẖabar muqaddam) and post-posed topic (مبتدأ مؤخر, mubtada' mu'aḫẖar). Some modifiers may come before them regardless of their position, and consequently affect their diacritization. As example, let us consider the following sentence: the boy is assiduous (الولد مجتهد, al-waladu muğtahidun). If we begin the sentence with the particle (إنَّ,'inna, indeed), it changes the topic to accusative form. But if we replace the particle (إنَّ,'inna, indeed) with the particle (كَانَ, kāna, was) in the same sentence, it changes the comment to accusative form [7].

The Arabic grammarians have established the following rule describing the general structure of a sentence:

$$\textbf{الجملة} = [\text{ الصدر }] \ (\textbf{ المسند و المسند إليه }) \ [\text{ الفضلة }] \tag{1}$$

The sentence, al-ğumlah = [the head, al-ṣṣadr] **(the predicate, al-musnad and, wa the subject, al-musnad'ilayh)** [the complement, al-faḍlah]

Both of the predicate and the subject are mandatory in the Arabic sentence. In the other hand, the complement and the head are optional. In the context of simple Arabic Nominal sentence and regarding the lexicon classification, the head could be an incomplete verb, an interrogation particle, etc. The predicate could be an adjective, a prepositional phrase, etc. The subject is always a noun phrase. The complement is usually an incomplete noun or a prepositional/locative phrase. Therefore, a simple nominal sentence cover six kind of simple Arabic nominal sentence: when the comment could be either resembling adjective, derivative adjective (الصفة المشتقة, al-ṣṣefah al-mušttaqah(, verbal noun (المصدر, al-maṣdar), indefinite noun, prepositional phrase, or locative phrase (See Fig. 7). The Fig. 3 presents an example of simple Arabic nominal sentence.

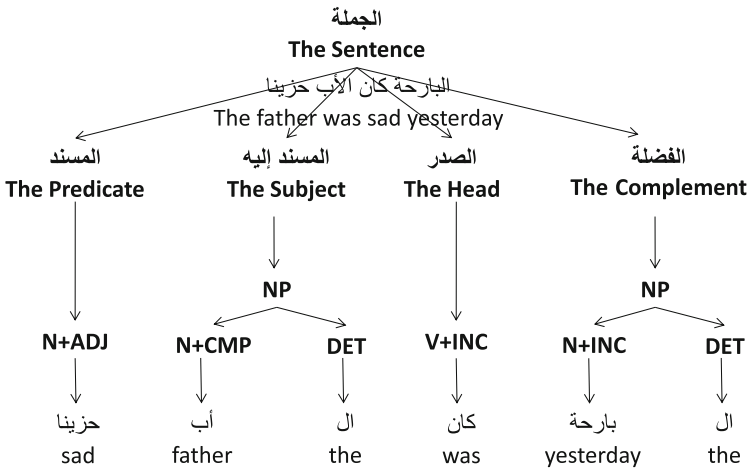

Fig. 3. Example of simple sentence structure.

4 The NooJ Linguistic Platform

NooJ is a development environment used to construct large-coverage formalized descriptions of natural languages, and apply them to large corpora, in real time [13, 16]. The descriptions of natural languages are formalized as electronic dictionaries, as grammars represented by organized sets of graphs. NooJ supplies tools to describe inflectional and derivational morphology, terminological and spelling variations, vocabulary (simple words, multi-word units and frozen expressions), semi-frozen phenomena (local grammars), syntax (grammars for phrases and full sentences) and semantics (named entity recognition, transformational analysis). In fact, NooJ allows linguists to combine in one unified framework Finite-State descriptions such as in XFST, Context-Free grammars such as in GPSG, Context-Sensitive grammars such as in LFG and unrestricted grammars such as the ones developed in HPSG.

NooJ is also used as a corpora processing system: it allows users to process sets of (thousands of) text files. Typical operations include indexing morpho-syntactic patterns, frozen or semi-frozen expressions (e.g. technical expressions), lemmatized concordances and performing various statistical studies of the results. NooJ is a freeware, linguistic engineering development environment used to formalize various types of textual phenomena (orthography, lexical and productive morphology, local, structural and transformational syntax) using a large gamut of computational devices (from Finite-State Automata to Augmented Recursive Transition Networks). NooJ includes tools to construct, test, debug, maintain and accumulate large sets linguistic resources, and can apply them to large texts [16].

5 Implementation

In a previous work, we have already implemented an Arabic dictionary in the NooJ platform. It is based on root and pattern properties. This dictionary consists of 160.000 lexical entries which are also obtained from flexional and derivational models [4]. However, this dictionary lacks a fine-grained classification allowing a correct syntactic analysis. Therefore, before the implementation in the NooJ platform, we must add new syntactic and semantic properties allowing a successful Arabic sentence parsing. Table 1 summarizes new defined properties holding the lexicon classification discussed in Sect. 3.1.

Table 1. Lexicon classification.

Property/ sub-property	Code	Example
- Noun, إسم, ’ism	N	
- Complete Noun, الإسم التام	N+CMP	قلم, pen, qalam
- Incomplete Noun, الإسم الناقص	N+INC	يوم, day , yawm
- Pronoun, الضمير, al-ddamīr	N+PRO	هو, he, howa
- Adjective, الصفة, al-ṣṣifah	N+ADJ	
- Resembling adjective الصفة المشبهة, al-ṣṣifah al-mušabbahah	N+ADJ +ARP	عظيم, great, ‘aẓīm
- Active Participle إسم الفاعل, ’ism al-fā´il	N+ADJ +AAP	كاتب, writer, kātib
- Passive Participle إسم المفعول. ’ism al-maf´ūl	N+ADJ +APP	مكتوب, wroten, maktūb
- Noun of Place, إسم المكان, ’ism al-makān	N +PLC	مطبخ, kitchen, maṭbaḫ
- Noun of Time, إسم الزمان, ’ism al-zzamān	N +TIM	مغرب, sunset, maḡrib
- Verb, فعل, fi‘l	V	
- Complete Verb, الفعل التام, al-fi‘l al-tām	V+CMP	
- Transitive 1	V+TR1	طلب, ṭalaba, to request
- Transitive 2	V+TR2	أعطى, ’aʿṭā, to give
- Transitive 3	V+TR3	أرى, ’arā, to show
- Intransitive	V+ITR	مات, to dead, māta
- Incomplete Verb, الفعل الناقص	V+INC	كان, to be, kāna
- Particle, حرف, ḥarf	PART	
- Annulling Particle	PART+ANN	لعل, la´alla, might
- Vocative Particle	PART+VOC	أيا, ’ayā, وا, wā: oh

After that, we implement some disambiguation rules that include three constraint types: lexical constraints, syntactic constraints, and agreement constraints. Theses constraints are implements as local grammars using the NooJ platform. Each analyzed sentence is matched with these grammars in a sequential mode in order to overcome

meaningless tags. The following local grammars (Figs. 4, 5 and 6) respectively summarize lexical, syntactic, and agreement constraints.

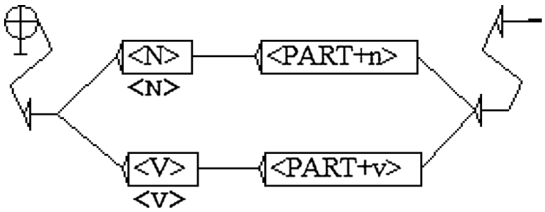

Fig. 4. Lexical constraint.

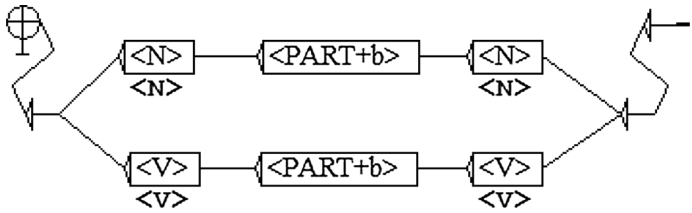

Fig. 5. Syntactic constraint.

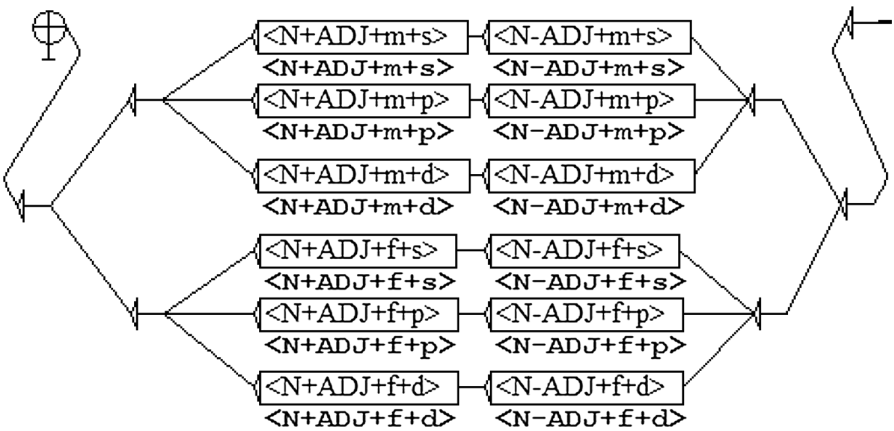

Fig. 6. Agreement constraint.

The final step in our contribution is the implementation of the simple Arabic nominal sentence structure. All in all, thirty structural grammar graphs, with seven levels of nesting, were implemented. All of them cover six kind of simple Arabic nominal sentence: Nominal sentence when the comment could be resembling and derivative adjective, a verbal noun, an indefinite noun, a prepositional phrase, or a locative phrase. Some of these grammar graphs are presented in Figs. 7 and 8.

In the NooJ linguistic platform, we implement a set of syntactic rules. These rules are based on the formula (1) describing the simple Arabic sentence structure.

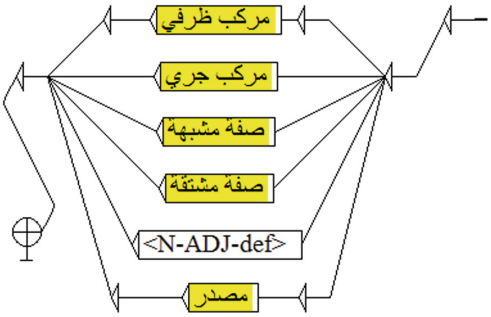

Fig. 7. Simple Arabic nominal sentence predicate values.

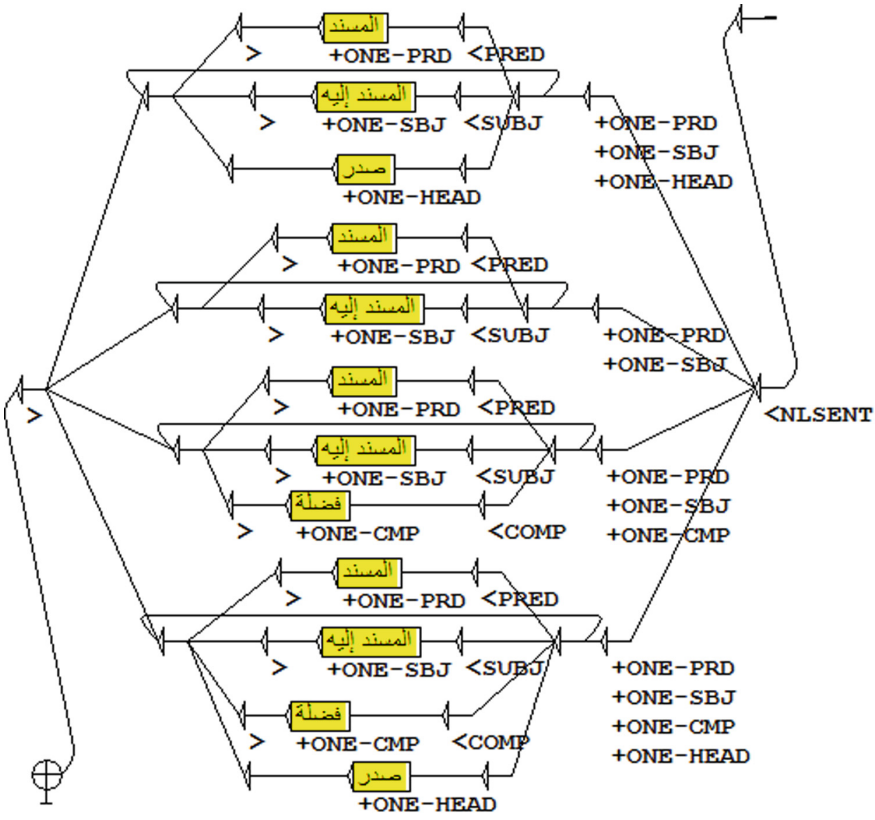

Fig. 8. First level of our grammar.

Our implementation takes advantage of the lexicon classification presented above. Figure 8 shows the first level of our grammar.

The graph shown in Fig. 8 handle and annotate the main components of the input sentence, and produce as output annotated parse tree(s) (see Sect. 3.3). our grammar is able to parse any nominal sentence regardless of the order of its components.

Table 2. Sentence components annotations

Abbreviation	Full form
COMP	Complement
INC. VERB	Incomplete verb
NLSENT	Nominal sentence
PRED	Predicate
SUBJ	Subject

Table 2 presents the list of abbreviations used in the sentence annotations produced by the first level of our grammar.

6 Results

To test the parser and the disambiguation local grammar on corpora, we have to segment corpora text into sentences. This task requires a particular processing which is not the aim of this work. So our test is applied on a text containing one hundred and

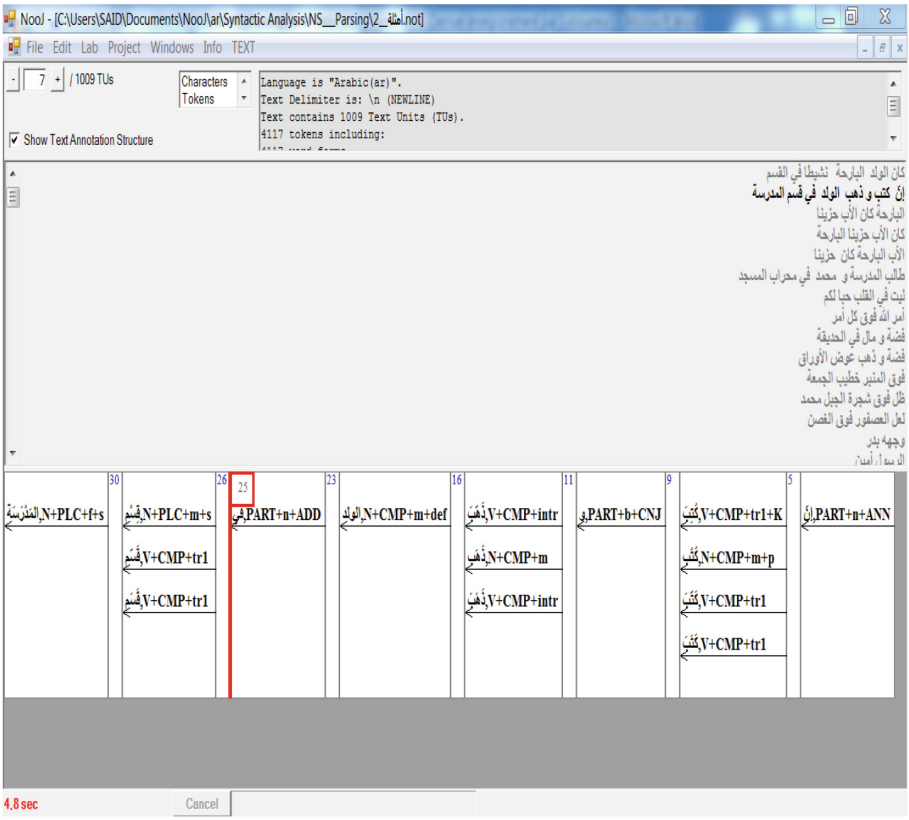

Fig. 9. Sentence annotations before disambiguation

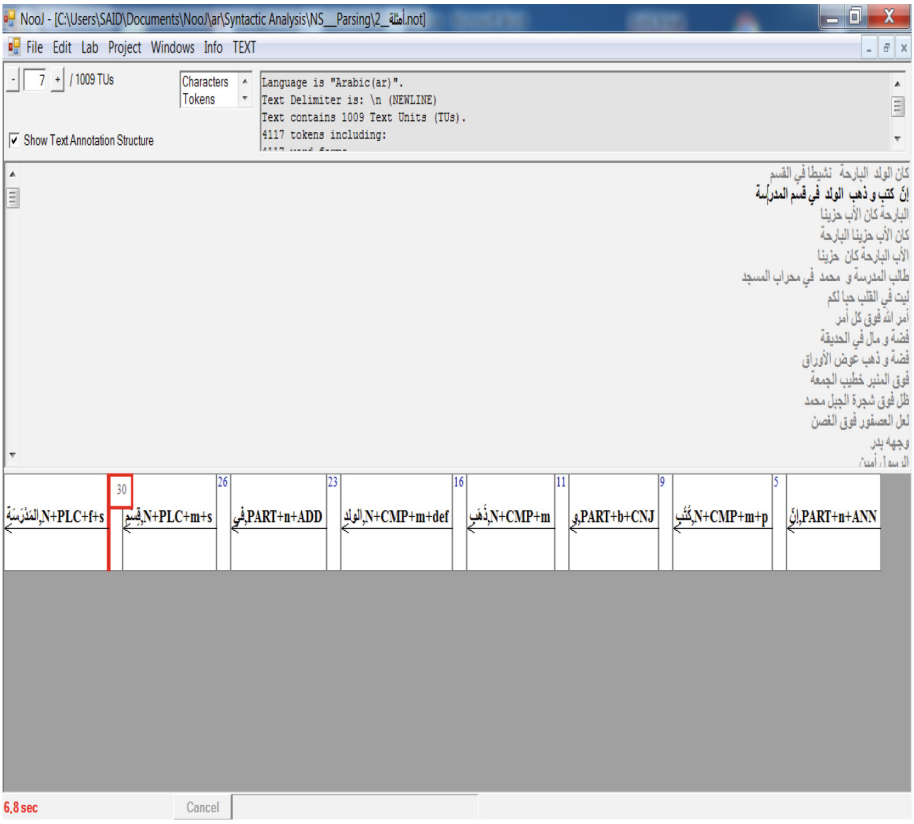

Fig. 10. Sentence annotations after disambiguation.

twenty nominal sentences having various structures. The rate of disambiguation is around eighty-six percent. This can be explained because some constraints are not yet implemented. However, we still have ambiguity in some sentences due to ambiguity feature of Arabic language. This issue could be solved once we implement a semantic analyzer beside the syntactic analyzer. Regarding the parsing task, the success rate of our analyzer is around ninety-five percent. This is obvious since some grammars are not yet implemented. Figure 9 presents the tagging of a sentence just before the disambiguation step. Figure 10 presents the tagging of the same sentence just after the disambiguation step and before the parsing step. Figures 11 and 12 contain the syntactic analysis result for the same sentence. We notice the success of the analysis even though the words order is not the same in each of them.

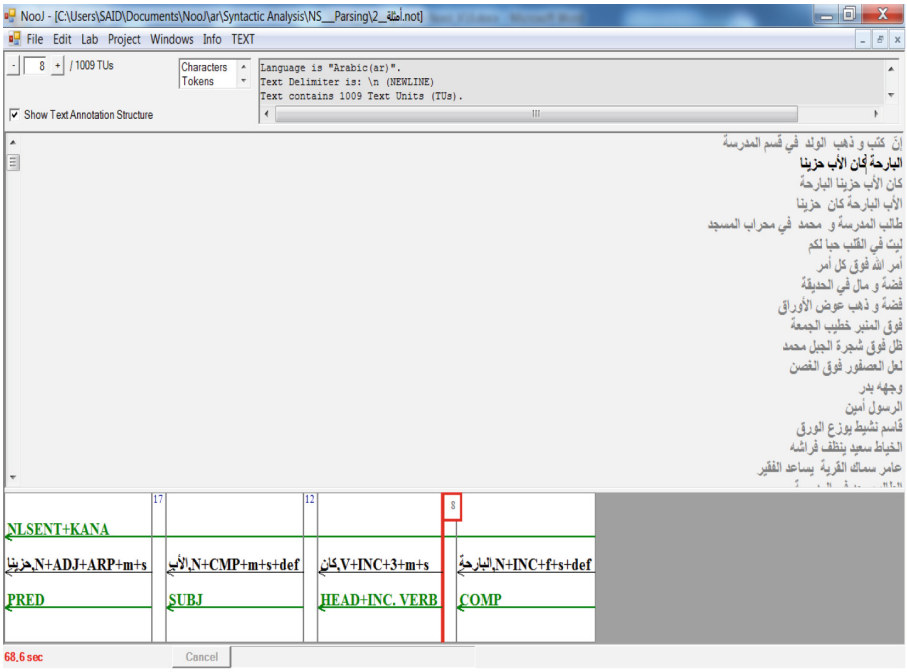

Fig. 11. Sentence annotations after the parsing step (1).

The NooJ Linguistic engine analyzes all possible interpretations for a not diacritized word, and add all possible morphological annotations in the NooJ Table Annotation Structure (TAS). Figure 9 shows an ambiguous NooJ TAS, result of the NooJ linguistic analysis, before applying our disambiguation local grammar.

After applying disambiguation local grammar, all impossible morphological annotations are filtered out from the NooJ TAS. Figure 10 presents a disambiguated NooJ TAS, which can be used for an efficient syntactic parsing and generation.

After the disambiguation step, we obtain a disambiguated sentence which is the input of our parser. The parser has to match parse tree(s) to the input sentence. Figures 11 and 12 show the NooJ TAS after the parsing step applied on two sentences. These sentences are similar but the order of their components is different. The parser returns the same syntactic annotations of the sentence components in the two proposed cases.

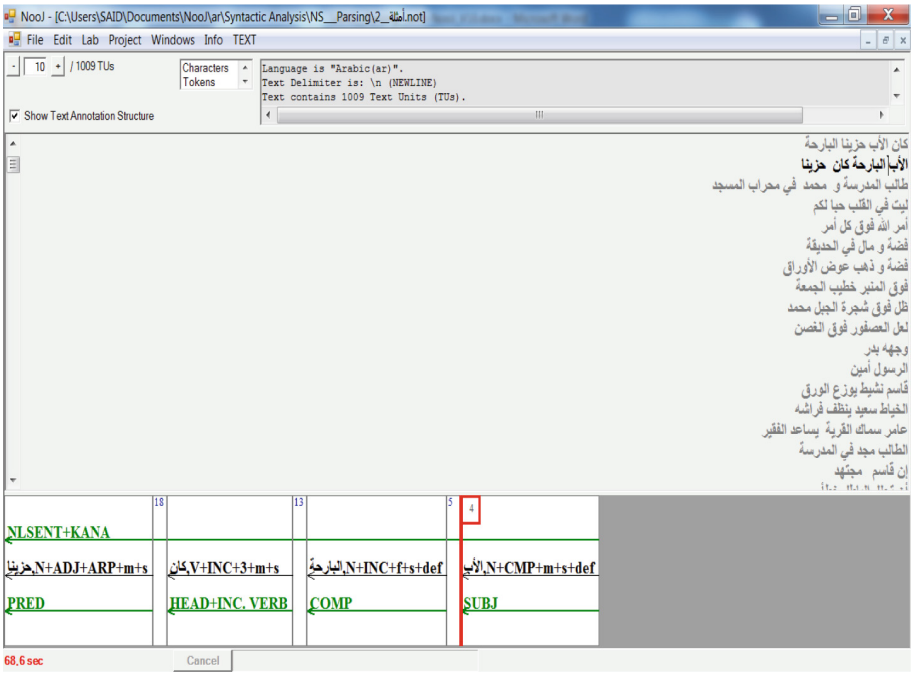

Fig. 12. Sentence annotations after the parsing step (2).

7 Conclusion and Perspectives

In this paper we presented our methodology of simple Arabic sentence parsing. This methodology consists in classifying the entries of an Arabic electronic dictionary regarding the category, implementing disambiguation rules, and creating syntactic grammars. The implementation is performed using the NooJ platform. If the platform NooJ allows to process all the stages of analysis (morphological, syntactic and semantic), our work is focused on the stage of syntactic analysis, with a preliminary stage of disambiguation.

Thus, we have implemented many transducers modeling a set of lexical and syntactic constraints in Arabic language. These transducers are applied sequentially. After that, with our structural grammars, we have analyzed several simple Arabic nominal sentences, disambiguated it automatically and generate their syntactic trees. These graphs add syntactic annotations.

Our method will not be limited to the simple nominal sentence, we will extend it to other types of Arabic sentence. So we will be able to syntactically analyze different text and corpora thereafter.

Once the Arabic analyzer is done, many issues could be solved such as automatic diacritics, Arabic sentences correction, and accurate translation. Also, other disambiguation rules could be implemented when the semantic analysis can be used.

References

1. Al Daoud, E., Basata, A.: A framework to automate the parsing of arabic language sentences. Int. Arab J. Inf. Technol. **6**(2), 191–195 (2009)
2. Assamirai, S.F.: Composition and Types of Arabic Sentence, 2nd edn. dar al kitab, Bagdad (2007)
3. Alsuhaibani, S.O.: The Verbal Sentence in Written Arabic. Thesis for the degree of Doctor of philosophy, University of Exeter, Ukraine (2012)
4. Blanchete, I., Mourchid, M., Mouloudi, A., Mbarki, S.: Formalizing Arabic inflectional and derivational verbs based on root and pattern approach using NooJ platform. In: Proceedings of the International NooJ Conference, NooJ 2017, Kenitra-Rabat, Morocco (2017)
5. Bourahma, S., Mbarki, S., Mourchid, M., Mouloudi, A.: Disambiguation and annotation of Arabic simple nominal sentences using NooJ platform. In: Proceedings of the International NooJ Conference, NooJ 2017, Kenitra-Rabat, Morocco (2017)
6. Fashwan, A., Alansary, S.: SHAKKIL: an automatic diacritization system for modern standard Arabic texts. In: The Third Arabic Natural Language Processing Workshop, Valencia, Spain (2017)
7. Habash, N.: Introduction to Arabic Natural Language Processing. Morgan & Claypool Publishers Series, San Rafael (2010)
8. Kassmi, R., Mourchid, M., Mouloudi, A., Mbarki, S.: Processing agglutination with a morpho-syntactic graph using NooJ. In: Proceedings of the International NooJ Conference, NooJ 2017, Kenitra-Rabat, Morocco (2017)
9. Khoufi, N., Aloulou, C., Belguith, L.H.: ARSYPAR: a tool for parsing the Arabic language based on supervised learning. In: The International Arab Conference on Information Technology, ACIT, University of Science & Technology, Sudan (2013)
10. Marton, Y., Habash, N., Rambow, O.: Improving Arabic dependency parsing with lexical and inflectional morphological features. In: Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages, Los Angeles, California, pp. 13–21 (2010)
11. Mourchid, M., Elfaddouli, N., Amali S.: Development of lexicons generation tools for Arabic: case of an open source conjugator. Int. J. Nat. Lang. Comput. IJNLC **5**(2), 13–25 (2016)
12. Mourchid, M.: Génération morphologique et applications. Thèse de Spécialité de 3ème Cycle, Université Mohammed V (1999)
13. Silberztein, M.: Formalizing Natural Languages The NooJ Approach. ISTE Editions, London (2016)
14. Silberztein, M.: Syntactic parsing with NooJ. In: The International NooJ Conference, Tozeur, Tunisia (2009)
15. Silberztein, M.: Disambiguation tools for NooJ. In: The International Nooj Conference, Budapest, Hungary (2008)
16. Silberztein, M.: NooJ Manual (2003). www.nooj4nlp.net
17. ابن الناظم : شر ح ابن عقيل على ألفية ابن مالك,الجزء الأول, الجزء الثاني, دار التراث, القاهرة, مصر (1980)
18. أبو زيد المقرئ الإدريسي : حروف المعاني في اللغة العربية دراسة تركيبية ودلالية, مؤسسة الإدريسي, الدار البيضاء (2016)
19. الحسين بن قاسم, الجنى : الداني في حروف المعاني, دار الآفاق الجديدة, بيروت (1983)
20. الرازي فخر الدين : نهاية الإيجاز في دراية الإعجاز, مطبعة الآداب ، القاهرة (1317 ه)
21. سيبويه: الكتاب, بولاق, القاهرة, (1316 ه)
22. نبيل علي : اللغة العربية والحاسوب, تعريب, القاهرة (1988)