Improvement of Arabic NooJ Parser with Disambiguation Rules

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. Annotating sentences is important to exploit the different features of Arabic corpora. This annotation can be successful thanks to a robust analyzer. That is why in this paper we propose to mention the improvement of our previous analyzer. To do this, we propose a description of our previous analyzer, which presents advantages and gaps. Then, we choose a method of improvement, which is inspired by the former one. Finally, we put forward an idea about the implementation and experimentation of our new cascade of transducers in NooJ platform. The obtained results appear satisfactory.

Keywords: Arabic analyzer · Disambiguation rules · Disambiguation process Cascade of transducers · NooJ platform

1 Introduction

The parsing of Arabic corpora remains a challenge on account of the richness of the Arabic language. Due to its morphological, syntactic, phonetic and phonological characteristics, Arabic is considered one of the most difficult languages to analyze in NLP. Therefore, the use of computers and new visions can shed new light on the study of this language. Indeed, with the parsing process, syntactic information becomes clearer and this helps in such areas as the construction of stochastic parsers, automatic translators, and the automatic recognizer of named entities.

Despite all the work to date, having a good syntactic representation (i.e. annotation) of Arabic texts presents a challenge. Existing works suffer from several problems such as ambiguity and non-robustness. Then, automation, optimization and disambiguation are always desired and useful to improve a number of processes. In addition, the enrichment of the system of rules by various linguistic phenomena increases the quality of the annotation. The essential part in the annotation of a corpus is the constructed grammar. Furthermore, the tags attributed to Arabic words and phrases must be expressive to help in the understanding of syntactic information. All these proposed solutions are feasible by means of a cascade of finite transducers.

In previous work, we have already built a parser based on a cascade of transducers essentially processing the Arabic nominal sentences. The obtained results appear

I. M. Mirto et al. (Eds.): NooJ 2018, CCIS 987, pp. 204–216, 2019.
https://doi.org/10.1007/978-3-030-10868-7_18

satisfactory, but we have detected some problems in the annotation, especially in complex and embedded structures. To solve these problems we first added disambiguation and lexical rules allowing the complete annotation and secondly completed the cascade by the appropriate transducers. To improve the values of evaluation measures, we have also added other syntactic rules concerning linguistic phenomena untreated so far and dealing especially with the verbal sentence.

Therefore, the main objective of this paper is to improve an Arabic NooJ parser based on a transducer cascade with several disambiguation rules. To do this, we added a set of transducers treating Arabic linguistic phenomena by relying on a symbolic approach and based on linguistic studies. Linguistic resources will be constructed using finite transducers that are formalized in the NooJ linguistic platform. We use a corpus to test the effectiveness of our improved transducer cascade. The obtained results, evaluated on a large corpus, are ambitious.

In this paper, we begin by a state of the art presenting some existing works that are involved in the parsing and annotation of the Arabic language. Next, we perform a description about our previous analyzer to present advantages and disadvantages. Then, we present our proposed method to improve the Arabic analyzer and our experimentation. Finally, a conclusion and some perspectives conclude our paper.

2 Previous Work

Syntactic analysis is the result of a formalization of various lexical and syntactic rules. This formalization consists in the representation of various syntactic phenomena. Among the first systems created in syntactic analysis, we find Cass system (Cascaded Analysis of Syntactic Structure) for the parsing of English and German texts [1]. This system is based on a set of finite state automata applied in iterative order. In addition, the work of [9] presented the FSPar annotation system. The author proposed a cascade of transducers for the parsing of the German language. This work is based on a list of recursive transducers treating each group independently of the other forms. This cascade enriched the output text with the various syntactical features for each word. Also, in [4], the authors presented a tool for parsing Arabic text using recursive graphs. The used approach is based cascade of transducers implemented in NooJ platform. This tool is based on three steps: the segmentation phase, the preprocessing phase, and the annotation. Concerning the search for information, we mention the work of [7] which presented the ASRextractor system. The latter is a system for extracting and annotating semantic relations between Arabic named entities using the TEI formalism. ASRextractor is based on a transducer cascade for extraction and annotation.

Also, [2] ensured an analysis of the Arabic language, in particular the phenomenon of coordination. In this work, the author has developed a HPSG grammar based on a hierarchy of types in order to classify the different linguistic units for the Arabic language, essentially the coordinated forms. Besides, there are different researches linked to parsing Arabic language. In [6], the authors proposed a generator of TEI (Text Encoding Initiative) lexicons based on an Arabic word hierarchy.

Among disambiguation systems, we cite MADA, which is a tool providing multiple applications like POS tagging, diacritization, lemmatization, stemming and glossing.

This tool is based on a statistical approach by exploring SVM models (Support Vector Machines). The AMIRA toolkit improves the performance of both platforms. As a result, the MADAAMIRA [8] system explored Arabic and the Egyptian language. In addition to these systems, we find the Stanford parser, which is a system, developed at Stanford University. This parser and the used pos-tagger are based essentially on the Maximum Entropy Model (MEM) called Conditional Markov Model (CMM). The tagger used by the Stanford parser was inspired by the Arabic Penn Treebank (ATB), which provides a large set of tags.

All the cited works show that the parsing of the Arabic language is a difficult task and several issues remain unsolved because the full formalization is always a challenge.

3 Description of Our Previous Parser

In our first version of the parser, we focused on the annotation of Arabic nominal sentences. Indeed, the automatic annotation of Arabic nominal sentences is not an easy task. In fact, the disambiguation process becomes more difficult due to the specific typology of nominal sentences. In the previous work, we proposed an approach to disambiguation. This approach consists of three main phases: the segmentation, the preprocessing and the disambiguation. The segmentation phase consists of the identification of sentences based on punctuation signs. An XML tag delimits each identified sentence. The second phase consists of the agglutination's resolution by using morphological grammars. The last phase aims to identify the adequate lexical category of each word in a given sentence and 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 to the least one. Also, a high level of granularity is used for lexical categories which help to distinguish between nominative, accusative and genitive cases for nouns and can resolve the absence of vocalization.

After that, we implement our Arabic analyzer for the case of nominal sentence in the NooJ platform using a transducer cascade. The implemented parser also permits the automatic annotation of sentences. In addition, to experiment our implemented prototype, we used an Arabic corpus that contains 200 meaningful nominal sentences mainly from stories. Also, we used dictionaries containing 24732 nouns, 10375 verbs and 1234 particles. Besides, in our experimentation we used a list of morphological grammars containing 113 inflected verb patterns, 10 broken plural patterns and one agglutination grammar with 40 subgraphs. In addition, we used 70 graphs representing lexical rules, and a set of 10 constraints describing the execution of the rule's application. Then, we calculate the precision, the recall and the F-measure. As a result, we obtained, in case of precision, the measure of 60%, in case of recall, we obtained 80%, and for the F-measure, we obtained 72%. Those results are ambitious but not satisfactory. In fact, the automation, optimization and disambiguation are needed. Also, the enrichment of the system of rules by the various linguistic phenomena existed increases the quality. The essential part in the annotation is the improvement of the constructed cascade. Furthermore, the tags attributed to Arabic words and phrases must be more expressive in order to facilitate the understanding of syntactic information. All these

proposed solutions are feasible to perform our analyzer and to generalize annotations in case of nominal and verbal sentence. To improve our analyzer, we need to know the different issues in the previous version. In fact, we recognize those issues especially thanks to the disambiguation process of our parser.

In the following, we present our proposed method that is based on a previous work, new disambiguation rules and the method of optimization.

4 Proposed Method for the Improvement of Arabic NooJ Parser

To improve our syntactical analyzer, we need to know the different problems in the previous version. In fact, we have recognized some problems especially in the disambiguation process of our parser.

As we have already indicated, the proposed method is based on previous work, that is presented in Fig. 1.

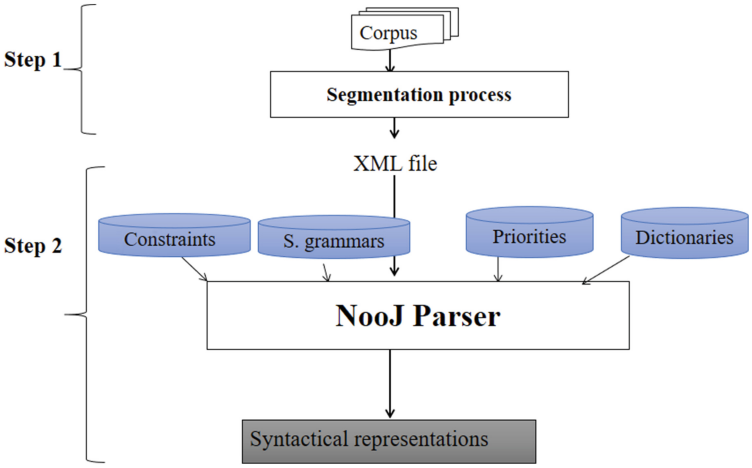

Fig. 1. Proposed method

The improvement of the parsing method is proportional to the improvement of the four entries. Those entries are constraints, syntactic grammars, priorities and dictionaries.

In fact, we elaborate new disambiguation rules and constraints. These rules and constraints are designed in form of transducers and are executed in a cascade with appropriate priorities. To do that, we firstly identify all structures of the Arabic sentences; we then make a study through a corpus for verbal and nominal sentences because a close relationship between these two forms exists. From this study, we achieve a system of syntactical rules allowing the annotation of Arabic sentences. A similar effort was given for the formalization step. In fact, the formalization of

elaborated rules requires much effort to guarantee several qualities (i.e., optimization, recursion, without rule's explosion and ambiguities). Secondly, we study the nature of each sub-graph to add new syntactic graphs in the adequate location compared previous cascade. Note that each graph is based on the graph's complexity, the number of sub-graphs and their depth. After this study, we classify graphs by levels: from low complex one to high complex one. Thirdly, we studied the relation between graphs: elementary, embedded and recursive graphs. Finally, we created the transducer cascade, which respects the extracted graph hierarchy. In fact, the cascade executes the established transducers in an adequate ranking. This ranking is identified thanks to multiple experimentations. In the following, we will present some disambiguation rules and their transducers in the new cascade for Arabic sentences starting from specific phrases to entire sentences.

5 Linguistic Studies Allowing the Improvement

To improve the quality of our parser, we have studied several linguistic phenomena, e.g. relative clauses, coordination and ellipsis. In what follows, we will focus on these matters.

5.1 Elliptical Forms

By definition, elliptical forms are characterized by the absence of one or more words in a sentence. By referring to a part of the previous discourse, the general meaning of the sentence becomes understandable. There are two principal types of elliptical forms: nominal elliptical forms and verbal elliptical forms. The nominal elliptical form can exist in a nominal or verbal sentence. In this type of ellipsis, the noun phrase is omitted and can be found in the previous part of the sentence. Therefore, a sentence containing an incomplete nominal phrase contains a nominal ellipsis form. The words in brackets are the omitted words.

كان الثعلب يذهب إلى كروم القرية، ويقطف (الثعلب) العناقيد الحمراء والصفراء والسوداء

The fox went to the vineyards of the village, and the (fox) gather the red, yellow, and black clusters

In the example, the elliptical form is a single word representing the subject in the elliptical sentence. This is an example of interaction between the ellipsis phenomenon and the coordination phenomenon in one sentence.

Verbal ellipsis exists generally in verbal sentences. In this type of ellipsis, the verbal phrase is omitted and can be identified in the previous part of the sentence.

أكل الثعلب الكروم الحمراء ثم (أكل الثعلب الكروم) السوداء

The fox eats the red clusters then (the fox eats the clusters) black

In the example, the elliptical form is a single word representing the subject in the elliptical sentence. This is an example of interaction between the ellipsis phenomenon and the coordination phenomenon in one sentence.

5.2 Relative Clauses

Relative clauses are the second of the most frequent phenomena in Arabic language. This phenomenon represents a subordinated phrase and it appears in any component of the sentence: either the subject or the object. The relative phrase is introduced by a relative particle, which can be followed by a verbal sentence, a nominal sentence or simple phrases. We illustrate this phenomenon with the following example:

إشترى الرجل الدار [التي على الربوة]
'ishtara alrajulu aldaara allati taka'u 'ala alrabwati
The man bought the house which is located on the hill

The example presents a verbal sentence and the relative form appears in the component الدار (*Al daara*), which is an accusative noun phrase.

5.3 Coordination Phenomena

The coordination phenomenon aims to bring together of at least two linguistic units. The coordinated units can be simple words, phrases, clauses or even sentences. This phenomenon appears in a sentence in form of deleted verb, deleted subject or deleted the object.

[مريم و منال] أختان جميلتان
Maryamu wa manaalu 'ukhtaani jamiilataani
Mariam and Manel are two beautiful sisters

[نسمة فرحانة] ف [اليوم عطلة نهاية الأسبوع] و [جدّها ينتظر زيارتها]
NismatN farhaanatN falyawma àuTlatu nihaayati al'usbuài wa jadduhaa yantathiru ziyaaratahaa
Nesma is happy, the weekend begins today and her grand-father is waiting for her visit

The two previous examples mention two types of coordination: the first one exists inside the phrase, whilst the second connects three nominal sentences.

6 Disambiguation Rules

We carried out a linguistic study which allows us to identify lexical rules resolving several forms of ambiguity. The identified rules are classified with the mechanism of subcategorization.

6.1 Rules for Ellipsis

Elliptical forms can appear in two different sentence types in the text. The first type is the verbal sentence, it depends on the transitivity criteria of the verb, and the second one is the nominal sentence. Several particles connect the original sentence to elliptical forms especially particles of coordination:

الواو ، الفاء ، ثم ، حتى ، أم ، أو ، لا ، بل ، ولكن

Table 1. Table summarizing rules for elliptical phenomena

Type of the sentence	Verb valency	Followed structures
Verbal	Intransitive	VP (NPNOM) (adverb) (PP)*
	Transitive	VP NPACC NPACC (adverb) (PP)*
	Double transitive	VP NPACC NPACC NPACC (adverb) (PP)*
	Triple transitive	VP NPACC NPACC NPACC NPACC (adverb) (PP)*
Nominal	–	TopicNom AttNom

Table 1 shows rules for the ellipsis phenomenon. As we see, there is a dependency between the transitivity criteria and the number of the constituents in the followed structure. We formalize and optimize those rules to obtain a transducer recognizing verbal elliptical sentence (Fig. 2) and nominal elliptical sentence (Fig. 3).

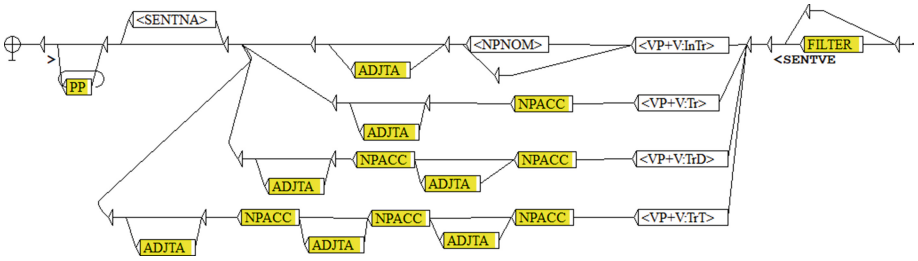

Fig. 2. Transducer for verbal elliptical sentence

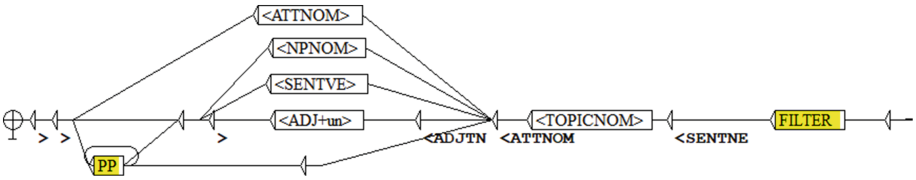

Fig. 3. Transducer for nominal elliptical sentences

6.2 Rules for Relative Clauses

Relative clauses are always introduced by elliptical particles. Those particles can be classified into two parts: particles followed by a verbal structure and particle followed by a nominal structure.

Table 2 displays rules for relative clauses. There are rules acting on verbal sentences and other acting on nominal sentences. We implement those rules through transducers which are illustrated by Figs. 4 and 5.

Table 2. Table summarizing rules for relative phenomenon

List of particles	Followed structures
اللَّذَانِ الَّتِي الَّذِي الَّذِينَ اللَّتَانِ اللَّوَاتِي اللَّاتِي اللَّائِي	Elliptical verbal sentence
مَنْ, مَا, أَيّ	Elliptical verbal or nominal sentence

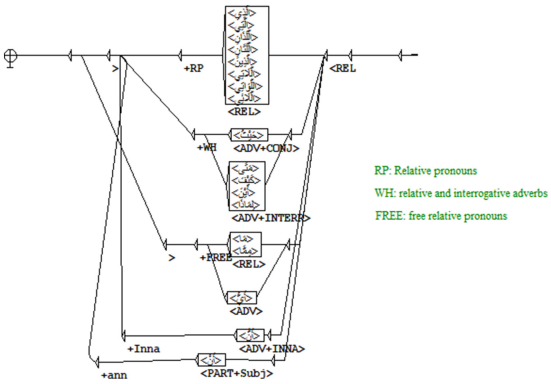

Fig. 4. Transducer for relative particles

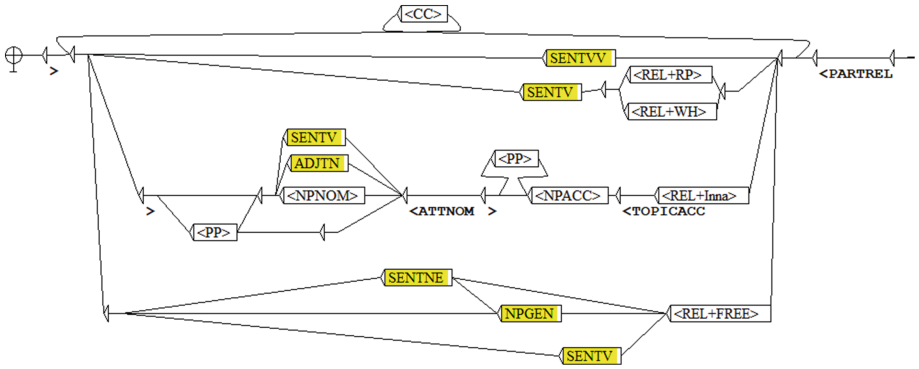

Fig. 5. Transducer for relative forms

6.3 Rules for the Coordination Phenomenon

The coordination phenomenon can link between different units or sentences. And generally the second part of the coordination is an elliptical clause. It can be introduced by the particles of coordination (Table 3).

Table 3. Table summarizing rules for coordination phenomena

List of coordination	Linked structures
Inside units	(NP)* (PP)* NP CONJ NP PP CONJ PP
Inside sentences	SENTV CONJ SENTV SENTN CONJ SENTN SENTV CONJ SENTN SENTN CONJ SENTV

Note that the coordinated forms can appear once or several times and it can be multiple to the same phrase. That is why the conjunction appears in all transducers, especially in filters.

7 Experimentation and Evaluation

To experiment our transducer cascade parser on the test corpus, we have used firstly our segmentation tool presented in [4]. Secondly, we have used our proper tagset inspired by Stanford's tagset (Table 4), a set of morphological grammars (113 infected verb patterns, 10 broken plural patterns and one agglutination transducer) and dictionaries that already exist in the NooJ linguistic platform (24732 nouns, 10375 verbs

and 1234 particles). However, the resources that exist in NooJ are not complete and sufficient. For this raison, we have added two other dictionaries. The first dictionary named "verbeintr.nod" contains 91 entries (intransitive, double transitive and triple transitive verbs). The second one named "prenom.nod" for proper nouns contains 105 entries. This dictionary recognizes the missing proper nouns. Thus, all these resources have the same priority except the "verbeintr.nod"; it has a high priority of a single level "H1" compared to others. All lexical resources are presented in Fig. 6.

Table 4. Syntactic resources

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

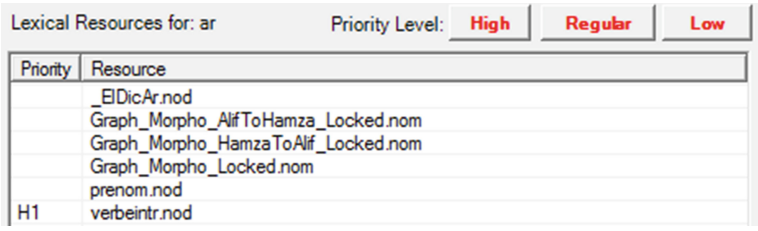

Fig. 6. Lexical resources

Thirdly, the proposed cascade executes 23 transducers including 100 sub-graphs. The transducers that are edited in NooJ linguistic platform called in a fixed ranking. Figure 6 illustrates the chosen order.

Fourthly, we tested our parser on two corpora. The first corpus is extracted from the Arabic Treebank (ATB) containing 836 sentences and the second is extracted from Arabic stories containing 5900 sentences. Concerning the ATB corpus, we deleted the sentence annotations in order to obtain a raw corpus or the test. In addition, the number of words in the sentences of the two corpora varies between 4 and 83 words. These sentences have different forms of verb phrases (with and without tools) with different tenses and modes. Also they have several noun phrase structures (one word, adjectival compound, annexation compound, relative compound, conjunctive compound and adjectival annexation compound).

The test result is a set of XML annotated sentences. As example of a sentence annotation after applying our parser on the following sentence:

لن تريد الطفلة أن تسلم على هذا الزهر

Lan turida altuflatu 'an tussalima 'ala hatha alzahri

The girl would not want to greet these flowers

```
<SENTV><VP><TOOLSUB>لن</TOOLSUB> <VSUB>تريد</VSUB></VP>
<NPNOM><NPDNOM><NND>الطفلة</NND></NPDNOM></NPNOM> <NPACC><PARTREL><VP><REL>أن</REL>
<VSUB>تسلم</VSUB></VP> <PP><PREP>على</PREP> <NPGEN><NPIGEN><DEM>هذا</DEM>
<NGD>الزهر</NGD></NPIGEN></NPGEN></PP></PARTREL></NPACC></SENTV>
```

Fig. 7. Sentence annotation example

Figure 7 presents the annotation result of this sentence. The treated verbal sentence SENTV has « لن تريد » "lan turida" (would not want) which is a verbal phrase VP where « لن » "lan" (would not) is a subjunctive verbal tool TOOLSUB and « تريد » "turida" (want) is a subjunctive verb VSUB. « الطفلة » "altuflatu" (the girl) is a subject that is a nominative noun phrase NPNOM and more specifically, it is a definite nominative noun NND. Whereas « أن تسلم على هذا الزهر » "altuflatu 'an tussalima 'ala hatha alzahri" (to greet these flowers) is an accusative noun phrase NPACC and more precisely it is a relative part PARTREL where « أن » "'an" is a relative REL and « تسلم » "tussalima" (to greet) is a subjunctive verb VSUB and « على هذا الزهر » "'ala hatha alzahri" (these flowers) is an indirect object PP. « على » "'ala" is a PREP preposition and « هذا الزهر » "hatha alzahri" (these flowers) is a genitive noun phrase NPGEN where « هذا » "hatha" (these) is a demonstrative pronoun DEM and « الزهر » "alzahri" (flowers) is a definite genitive noun NGD.

For the evaluation, we used the known metrics: recall, precision and f-measure. We obtain the following results for the ATB corpus presented in Table 5.

Table 5. Table summarizing the metrics obtained for ATB corpus

	ATB corpus	Recall	Precision	F-measure
Cascade parser	836 sentences	0.9	0.94	0.91

The major advantage of the new parser compared with Stanford parser is the great reduction of the execution time (parsing 836 sentences in 59.2 s). However, there is no improvement in the measure values. Concerning the stories corpus, we obtained the following evaluation metric values illustrated in Table 6.

Table 6. Table summarizing the metrics obtained for stories corpus

	Stories corpus	Recall	Precision	F-measure
Cascade parser	5900 sentences	0.74	0.82	0.77
Previous parser	5900 sentences	0,6	0,8	0,72

The measure values of Table 6 show the efficiency of the cascade parser compared to the previous parser. But some parsing problems are detected especially in complex and embedded structures. To solve these problems, we must add some constraints. Also to improve the measure values, we must add other syntactical rules concerning untreated linguistic phenomena and specific to Arabic language.

8 Conclusion

In the present paper, we have improved an Arabic NooJ parser. This improvement is made on a transducer cascade by using several disambiguation rules. This tool is based on an improved method inspired by our previous method. We focused especially in lexical resource and the priorities criteria of each one. In addition, we have shown the efficiency of our new transducer cascade when compared with the previous one. Thus, the evaluation is performed on a set of sentences belonging to two corpora. The results obtained are ambitious and show that our parser can efficiently treat different sentence forms. As for perspectives, we would like to increase the coverage of our designed dictionaries. We will also improve our parser by adding other syntactic rules recognizing frozen forms of sentences.

References

1. Abney, S.: Partial parsing via finite-state cascades. Nat. Lang. Eng. **2**(4), 337–344 (1996)
2. Boukedi, S., Haddar, K.: HPSG grammar for Arabic coordination experimented with LKB system. In: Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference, FLAIRS 2014, Pensacola Beach, Florida, 21–23 May 2014, pp. 166–169 (2014)
3. Hammouda, N.G., Haddar, K.: Parsing Arabic nominal sentences with transducers to annotate corpora. Computación y Sistemas, vol. 21, no. 4: Advances in Human Language Technologies (Guest Editor: A. Gelbukh), pp. 647–656 (2017)
4. Hammouda, N.G., Haddar, K.: Integration of a segmentation tool for Arabic corpora in NooJ Platform to build an automatic annotation tool. In: Barone, L., Monteleone, M., Silberztein, M. (eds.) NooJ 2016. CCIS, vol. 667, pp. 89–100. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-55002-2_8
5. Hammouda, N.G., Haddar, K.: Arabic NooJ parser: nominal sentence case. In: Mbarki, S., Mourchid, M., Silberztein, M. (eds.) NooJ 2017. CCIS, vol. 811, pp. 69–80. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73420-0_6
6. Maamouri, M., Bies, A., Buckwalter, T., Mekki, W.: The Penn Arabic Treebank: building a large-scale annotated Arabic corpus. In: NEMLAR Conference on Arabic Language Resources and Tools, vol. 27, pp. 466–467 (2004)
7. Mesmia, F.B., Zid, F., Haddar, K., Maurel, D.: ASRextractor: a tool extracting semantic relations between Arabic named entities. In: 3rd International Conference on Arabic Computational Linguistics, ACLing 2017, 5–6 November 2017, Dubai (2017)
8. Pasha, A., et al.: MADAMIRA: a fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In: Proceedings of LREC, Reykjavik, vol. 14, pp. 1094–1101 (2014)

9. Schiehlen, M.: A cascaded finite-state parser for German. In: Proceedings of EACL 2003, vol. 2, pp. 163–166 (2003)
10. Silberztein, M.: A new linguistic engine for NooJ: parsing context-sensitive grammars with finite-state machines. In: Mbarki, S., Mourchid, M., Silberztein, M. (eds.) NooJ 2017. CCIS, vol. 811, pp. 240–250. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73420-0_20