Arabic Probabilistic Context Free Grammar Induction from a Treebank

Nabil Khoufi, Chafik Aloulou and Lamia Hadrich Belguith

ANLP Research Group, MIR@CL Lab,
Faculty of Economics and Management,
University of Sfax, Tunisia.
{nabil.khoufi, chafik.aloulou, l.belguith}@fsegs.rnu.tn

Abstract. Linguistic resources are very important to any natural language processing task. Unfortunately, the manual construction of these resources is laborious and time-consuming. The use of annotated corpora as a knowledge database might be a solution to a fast construction of a grammar for a given language. In this paper, we present our method to automatically induce a syntactic grammar from an Arabic annotated corpus (The Penn Arabic TreeBank), a probabilistic context free grammar in our case. To construct our resource, we first induce context free rules from the annotated corpus trees as a first step and then we calculate a specific probability for each induced rule. Finally, we present and discuss the obtained grammar.

Keywords: Linguistic Resource Construction, Syntactic Grammar, PCFG, Arabic language, PATB corpus.

1 Introduction

The development of effective NLP applications requires the use of reliable and large linguistic resources (knowledge) such as lexicons or grammars. For example, the parsing task requires, in addition to the input sentence, some knowledge about the kind of syntactic analysis that should be produced as output. One method to provide such knowledge to the parser is to write down manually a grammar of the language.

However, manual construction of such linguistic resources is a difficult task to undertake, and it is time consuming. Unlike a programming language, natural language is far too complex to simply list all the syntactic rules. Moreover, it is difficult to exhaustively list lexical properties of words in addition to validate the written grammar by some linguists of the treated language.

In this regard, describing the Arabic language needs special attention given its greater ambiguity when compared to other natural languages (especially the Indo-Europeans languages). Indeed, Arabic has several characteristics among which we mention the following:

- Vocalic ambiguity: the oversight of the vocalization marks increases the ambiguity of words' comprehension;
- Grammatical ambiguity: several words may have the same grammatical interpretation;
- Agglutination: articles, prepositions, pronouns, etc. can be affixed to adjectives, nouns, verbs and particles to which they are related;
- Problems related to the segmentation of texts into sentences;
- Abundant use of recursive structures in the Arabic texts;
- Elliptic and anaphoric structures.

An alternative method to build linguistic resources is to use treebanks as source of knowledge. Indeed, treebanks, as rich corpora with annotations, provide an easy way to build other linguistic resources, such as extensional and intentional lexicons, syntactic grammars, bilingual dictionaries, etc. This promotes their reuse as well as makes explicit their implicit information. Furthermore, treebanks have many advantages: they are not only developed and validated by linguists, but also submitted to consensus, which promotes their reliability. Having such resource makes it possible to generate automatically and in a very controlled basis, new and wide coverage resources on other formalisms. These resources inherit the original treebank qualities, while reducing construction time. Since we have access to the PATB corpus, currently, the largest Arabic corpus in modern standard Arabic, we have decided to use it in order to build our syntactic grammar.

This paper is organized as follows: Section 2 is devoted to presenting related works on the construction of grammars for Arabic language. Section 3 gives the probabilistic context free grammar's (PCFG) basic definitions. Section 4 explains our method. Experimental results are presented and discussed in section 5. Section 6 provides the conclusion and future work.

2 Grammars for Arabic Language

Grammar studies the way in which words/morphemes join to form meaningful sentences. It consists of a set of constraints on the possible sequences of symbols expressed as rules or principles. The construction of grammars is an important task to several NLP applications. Thus, for the Arabic language, many grammars were built following various grammar formalisms. The main purpose of a grammar in NLP is to serve the parsing task.

In this section, we present the different grammars that have been designed to represent Arabic syntax. All these works attempt to cover most of the syntactic rules of the Arabic language and its rich linguistic phenomena.

Habash and Rambow [1] have built a tree adjoining grammar by extracting elementary trees from an annotated corpus. In order to accomplish this, the authors reinterpreted the PATB as a dependency corpus and then extracted the Tree-Adjoining Grammar from the corpus. They used part 1, version 2.0 of the PATB, which comprises around 160,000 words of annotated Arabic text from newswire sources. Extracted structures have some variations in their constituent's positions because of the

semi-fixed order of the Arabic language. Thus, some obtained sentences have VSO (Verb Subject Object) structure and others have SVO structure.

The LFG (Lexical Functional Grammar) [2] formalism has been used to represent Arabic language syntax. Attia [3] used this formalism to build the grammar. Indeed, the developed grammar focuses on the specific features of Arabic, especially nominal sentences (sentences without verbs), sentences with a hidden subject, verbal sentences etc. Both LFG structures (c-structures and f-structures) are built to each sentence described by this grammar.

The HPSG (Head driven Phrase Structure Grammar) [4] was also used to describe the syntax of Arabic in the work of Haddar [5]. The main objective of this work is to construct an Arabic HPSG grammar based on a proposed type hierarchy that categorizes Arabic words. In fact, some adaptations were introduced to HPSG at the level of features and ID schemata. All linguistic resources (e.g., lexicon, type hierarchy, syntactic rules) are specified in the Type Description Language (TDL). The experimentation of the constructed grammar was achieved using the Linguistic Knowledge Building (LKB) platform which contains generation tools. The authors justified the use of the IDL by its syntax similarity to HPSG representation.

In the present work, we chose to induce a PCFG from a treebank. The grammar choice is justified by its capacity to give a partial solution for any grammar ambiguity; indeed a PCFG gives some idea of the plausibility of a sentence. Likewise, a PCFG is a robust grammar that can admit everything with low probability. In the following section, we present the basic definitions of the PCFG grammar and then we present our method.

3 PCFG Basic Definitions

A probabilistic context-free grammar (PCFG) also called stochastic CFG (SCFG), is an extension of the famous context-free grammar, where a certain probability is assigned to each rule. Probabilistic context-free grammars are defined by a 5-uplet $< N, T, R, S, P >$ as follows:

- N is a finite set of non-terminal symbols.
- T is a finite set of terminal symbols.
- R is a finite set of rules Ri of the form $X \rightarrow Y1Y2 \ldots Yn$, where $X \in N$, $n \geq 0$, and $Yi \in (N \cup T)$ for $i = 1 \ldots n$.
- $S \in N$ is a distinguished start symbol.
- P is the set of probabilities Pi associated with rules Ri where: $\sum P(X \rightarrow Yi) = 1, \forall X \in N$ and $Yi \in (N \cup T)$ for $i = 1 \ldots n$.

Note that some sentences may have more than one underlying derivation in case of the use of a classic CFG and therefore generates several parse-trees. So the probabilities P in a PCFG are used to determine which parse-tree is the most likely to be the best parse for the given sentence. The probability of a parse-tree is obtained by multiplying the probability of each used rule at each node of the tree.

4 Proposed Method

Our objective is to automatically induce a PCFG from an annotated corpus. This mechanism consists of two steps: The first one is to induce a CFG rules from the annotated corpus. The second step is to allocate a specific probability for each induced rule. The application of these two steps allows us to obtain a PCFG. Figure 1 illustrates the workflow of our method for deriving a PCFG.

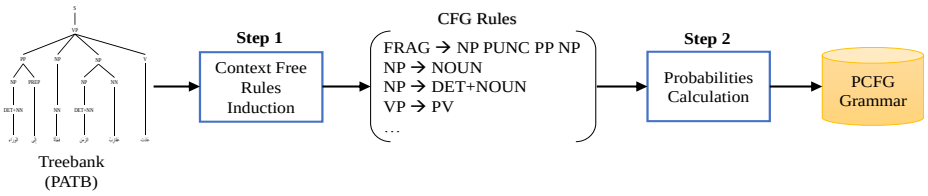


Fig. 1. Steps of the PCFG induction method

Given that our method is based on an annotated corpus to build the PCFG grammar, we present in the next subsection the used corpus then we detail the steps of the method.

4.1 The Used Corpus

In our work, we chose to use the well-known corpus, Penn Arabic Treebank (PATB). This choice was motivated not only by the richness, the reliability and professionalism with which it was developed but also by the syntactic relevance of its source documents (converted to several other Treebank representations). Indeed, its annotations have the advantage of being reliable. This is proven by its efficiency in a large number of works in various fields of NLP [6]. The good quality of the text and its annotations are proven by their pertinence for the creation of other Arabic Treebanks such as the PADT [7] and the CATiB [8], which converted the PATB to its syntactic representations in addition to other annotated texts.

Indeed, these annotations were manually elaborated and validated by linguists. Moreover, this treebank is composed of data from linguistic sources written in Modern Standard Arabic. This corpus is also the largest Arabic corpus which integrates syntactic tree files. The use of a large amount of annotated data in a construction process of a grammar increases the quality of the generated linguistic resource.

The PATB was developed in the Linguistic Data Consortium (LDC) at the University of Pennsylvania [9]. Texts in the corpus do not contain any vowels as it is typically in use in most texts written in Arabic.

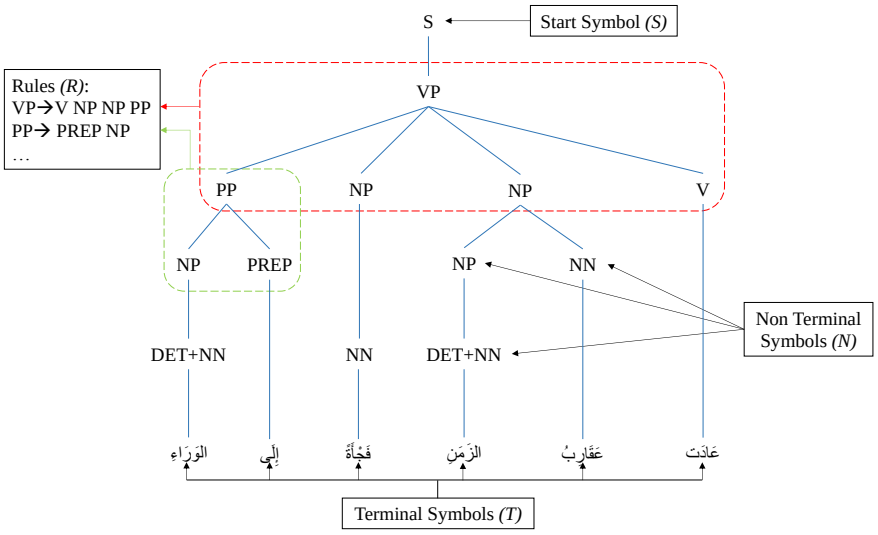
4.2 Description of the Method

As shown in Figure 1, the first step of the method is the induction of CFG rules, including duplicates which will be used in the second step. A deep study of the PATB

allows us to identify rules which guide the CFG rules induction process. Indeed, we focused on the morpho-syntactic trees of the PATB and we identified the following rules:

R1: Tree root → Start symbol
R2: Internal tree node → Non terminal symbol
R3: Tree word → Terminal symbol
R4: Tree fragment → CFG rules

In fact, we noticed that each parse tree is a sequence of context-free rules and each one has the same symbol "S" at its root. Thus, the root symbol "S" is taken to be the start symbol (S) of the grammar. Non-terminal (N) symbols consist of the set of internal nodes of the whole parse tree. The set of all words seen in the trees (leaves) represent the terminal symbols (T). Edges between the nodes of the trees are used to induce the CFG rules(R). The following figure presents the process of induction of the PCFG elements (S, N, T, R) from a PATB parse tree.

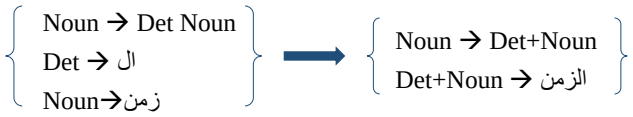


(The hands of time turned suddenly backwards.)

Fig. 2. Induction of PCFG elements from a PATB parse tree

Note that the tags on the right hand-side of the induced rules are in reversed order compared to the parse tree. This is due to the reading orientation of the Arabic language which is from right to left.

In the Arabic language, the word and its determinant are agglutinated as we can see in the word الزَمَنِ (Noun). We chose to keep these elements together in one rule to reduce the grammar size. This choice does not influence the analysis quality since there is no loss of information.



After applying the first step of our method to the example of Figure 2, we obtain 12 CFG rules composed by 6 contextual rules and 6 lexical rules as presented in the following table:

Table 1.CFG Rules induced from the example of Figure2

Contextual Rules	**Lexical Rules**
S→VP	V→عَادَت (turned)
VP→V NP NP PP	NN→عَقَارِبُ (the hands)
NP→NN NP	DET+NN→الزَمَنِ (time)
NP→DET+NN	NN→فَجْأَةً (suddenly)
NP→NN	PREP→إِلَى (to)
PP→PREP NP	DET+NN→الوَرَاءِ (backwards)

Once the CFG rules had been induced from the PATB with all duplicates, we move to the second step, which consists in the rule probability calculation *(P)* to finally obtain the PCFG grammar. Each rule probability is estimated using the following formula:

$$P(X \rightarrow Y) = \frac{Count(X \rightarrow Y)}{Count\ (X)}$$

Where Count (X→Y) is the number of times the rule X→Y is seen in the Treebank and Count (X) is the count of rules that have the non-terminal X on the left-hand side.

For example, the rule VP→V NP PP is seen109 times in the PATB and we have counted 1311 rules that have VP on the left-hand side, thus:

$$P(VP \rightarrow V\ NP\ PP) = \frac{109}{1311}$$

In the following section, we describe our experiment and expose some interesting information about the obtained grammar.

5 Experimentation and Results

After applying our method, we obtained the PCFG grammar. We used the PATB 3 version 3.2 of this corpus, which consists of 599 files, and includes POS tags, morpho-syntactic structures at many levels and glosses. It comprises 402,291 tokens and 12,624 sentences. It is available in various formats: The "sgm" format refers to source documents. The "pos" format gives information about each token as fields before and

after clitic separation. The "xml" format contains the "tree token" annotation after clitic separation. The "penntree" format generates a Penn Treebanking style. And finally the "integrated" format brings together information about the source tokens, the tree tokens, and the mapping between them and the tree structure.

As mentioned earlier, the PATB is a very rich corpus and it contains many annotations such as mood, gender, number, etc. The PATB corpus is annotated using a large set of annotations which gives a high level of granularity. For example, the Part of speech annotation tag set contains 498 tags that provide extensive information: gender, the mood, etc. [9]. There are also 22 syntactic category tags and 20 tags that describe semantic relations between tokens. In addition to that, stop words, which are very numerous in the Arabic language, are also annotated with specific tags.

The incorporation of all this information within the grammar increases its complexity and its size. The size depends on the granularity level of the categories it describes: the higher this level, the more these grammars are complex, but the more they respect the language specificity. For instance, this tag set {ADJ+NSUFF_FEM_DU_NOM, ADJ+CASE_DEF_ACC ADJ+CASE_DEF_GEN ADJ+CASE_DEF_NOM} describes adjectives with a high level of granularity. If the granularity level is reduced to the minimum, these tags will be reduced into one tag, ADJ. This reduction influences the number of grammar rules. We chose to reduce POS tags to the basic tags which are about 70 tags for the Arabic language to facilitate the use of the grammars for NLP application.

There is also other tags in the PATB that are generated during the initial tagging and parsing process like *ICH*, *O*, *RNR*, NONE and NAC. Those tags, if considered in a parsing task, will increase the number of rejected parses because they describe morphologic tagging errors and empty categories. Therefore, those tags were removed from our tag set to maintain a better consistency of our grammar.

We present below some statistics about our PCFG grammar. Table2 presents the most frequent PCFG syntactic rules generated from the PATB corpus after applying our method. Table 3 presents the overall count of rules (contextual and lexical rules).

Table 2. Most frequent LHS rules

Left-Hand symbol (LHS)	NP	VP	S	FRAG	ADJP	UCP	PP
Rule count	1821	1311	1154	360	330	196	150

Table 3. Rule count extracted from the PATB

Contextual Rules	5757
Lexical Rules	38 901
Total	44 658

5.1 Arabic PCFG Induction System

We developed a tool to automatically induce the PCFG from PATB tree files. The system allows the user to:

- Induce CFG rules with full morphological annotations,
- reduce the morphological annotations of the induced rules,
- and generate PCFG from the CFG rules

The following figure 3 shows the induction interface of our system.

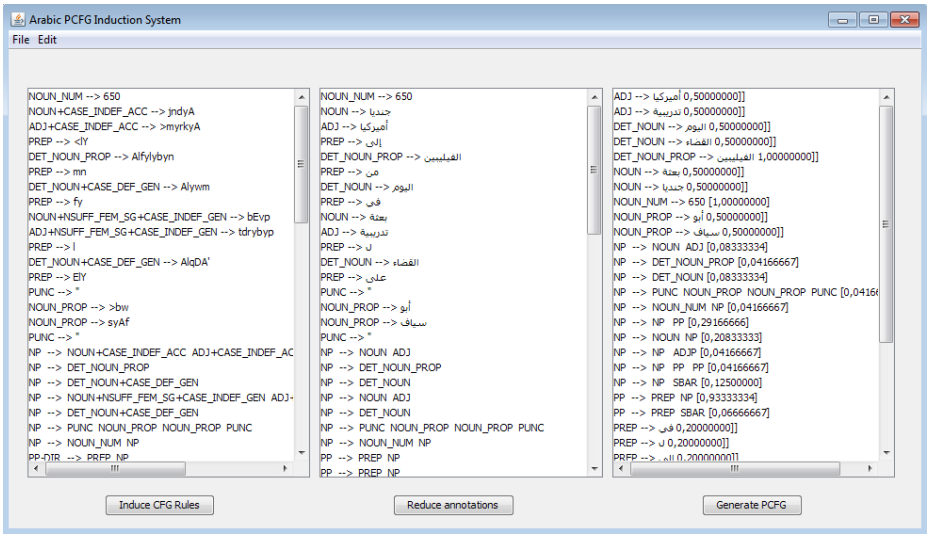


Fig. 3. Induction interface

Besides the PCFG induction, the system allows the user to browse the grammar and give some information about it. Indeed, the system can sort grammar rules by the left hand symbol to facilitate the consultation of rules. The system also shows useful information for each selected rule such as the number of rules with the same left hand symbol, the number of occurrences of the selected rule, its probability and the number of right hand symbols in addition to some statistics of the induced grammar.

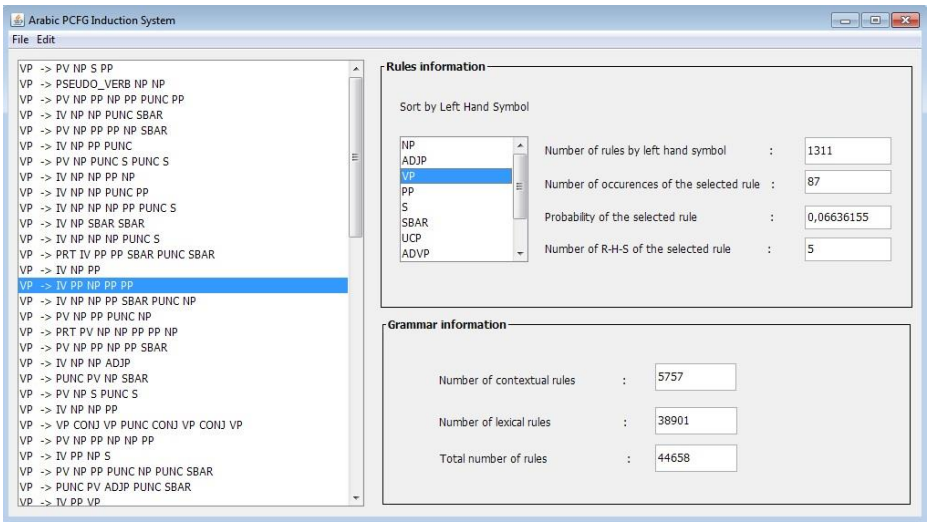


Fig. 4. Arabic PCFG browsing interface

6 Conclusion and Future Work

In this paper, we presented our work in inducing an Arabic PCFG from the PATB corpus. As a result, we obtained a new resource which provides a wide coverage and inherits PATB qualities such as its reliability, adherence to consensus and rich annotation. The proposed method consists of two steps: in the first one we induce CFG rules from the PATB parse trees using induction rules. Tag set of the PATB was factorized and filtered to reduce the complexity of the grammar and keeping only useful information. Second we calculated the probability of each rule using the frequency of its occurrence in the corpus. The obtained grammar is composed of 5757 contextual rules and 38 901 lexical rules.

Our main perspective is to test the performance of this grammar in the parsing task. This is a work in progress as we intend to test the obtained grammar using several PCFG parsing algorithms such as Viterbi or CYK. Such comparison between parsing algorithms results on our PCFG is very interesting. This work is part of a hybrid method to parse the Arabic language. The aim of this hybrid method (symbolic / statistical) is the collaboration of two parsers, the first based on a statistical model obtained using supervised learning techniques [10;11] and the second based on the induced grammar described in this paper.

Acknowledgments

We are grateful to Dr. Hela Azaiez, assistant researcher scientist at university of Iowa for proofreading the manuscript.

Reference

1. Habash, N., Rambow, O.: Extracting a tree adjoining grammar from the Penn Arabic Treebank. In: Proceedings of Traitement Automatique du Langage Naturel (TALN-04), pp. 277-284 (2004)
2. Kaplan,R. M., BresnanJ.:Lexical functional grammar: A formal system for grammatical representation. In: J. BRESNAN, Ed., The Mental Representation of Grammatical Relations. Cambridge, Mass, MIT Press, pp. 173-281 (1982)
3. Attia, M. A.: Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation. Doctoral thesis, Faculty of humanities, University of Manchester(2008)
4. Pollard, C., Sag I.: Head-driven Phrase Structure Grammar. CLSI series, University of Chicago(1994)
5. Haddar, K., Boukedi, S., Zalila, I.: Construction of an HPSG grammar for the Arabic language and its specification in TDL. International Journal on Information and Communication Technologies, vol. 3, n° 3, pp. 52-64 (2010)
6. Habash, N. Y.: Introduction to Arabic natural language processing. Synthesis Lectures on Human Language Technologies, pp. 1-187(2010)
7. Hajic, J., Vidová-Hladká, B., Pajas, P.: The Prague dependency Tree-bank: Annotation structure and support. In: Proceedings of the IRCS Workshop on Linguistic Databases, pp. 105-114 (2001)
8. Habash, N., Roth, R. M.: Catib: The Columbia Arabic Treebank. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pp. 221-224, Association for Computational Linguistics (2009)
9. Maamouri, M., Bies, A., Buckwalter, T., Mekki, W.: The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In: Proceedings of NEMLAR conference on Arabic language resources and tools, pp. 102-109 (2004)
10. Khoufi N., Louati S., Aloulou C., Hadrich Belguith L.: Supervised learning model for parsing Arabic language. In: Proceedings of the 10th International Workshop on Natural Language Processing and Cognitive Science (NLPCS 2013), Marseille, France, pp129-136. (2013)
11. Khoufi N., Aloulou C., Hadrich Belguith L.: Chunking Arabic Texts Using Conditional Random Fields. In: Proceedings of the 11th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2014), November, 2014, Doha, Qatar, pp 428-432 (2014)