Building an Arabic linguistic resource from a treebank: the Case of Property Grammar

Raja Bensalem Bahloul, Marwa Elkarwi, Kais Haddar, Philippe Blache

Building an Arabic Linguistic Resource from a Treebank: the Case of Property Grammar

Raja Bensalem Bahloul[1], Marwa Elkarwi[1], Kais Haddar[1], and Philippe Blache[2]

[1] Multimedia Information Systems and Advanced Computing Laboratory, Higher Institute of Computer Science and Multimedia, Sfax, Tunisia
raja_ben_salem@yahoo.com, marwaelkarwi89@gmail.com, kais.haddar@fss.rnu.tn
[2] Laboratoire Parole et Langage, CNRS, Université de Provence, France
pb@lpl.univ-aix.fr

Abstract. This paper presents a survey of Arabic treebanks to facilitate their reuse for the building of new linguistic resources. In our case, we created from a treebank an automatically induced Property Grammar (GP). So, we discussed characteristics of these treebanks to choose the appropriate one. To build our resource, we adopted an automatic technique, acquiring first a context-free grammar (CFG) from the chosen treebank, and second, inducing a GP by generating relations between grammatical units described in the CFG.

Keywords: treebanks, Arabic language, reuse, property grammar

1 Introduction

Treebanks, as rich corpora with annotations, are used to build other linguistic resources, such as extensional and intentional lexicons, context-free grammars (CFG), property grammars (GP), bilingual dictionaries, etc. This promotes their reuse as well as makes explicit their implicit information. Also, 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, inheriting the original treebank qualities, while gaining on construction time. For Arabic, treebanks are scarce while the most important are: the Penn Arabic Treebank (ATB) [6], the Prague Arabic Dependency Treebank (PADT) [5], the Columbia Arabic Treebank (CATiB) [4] and the Quranic Arabic Dependency Treebank (QADT) [3]. But, what resource can we choose to build, particularly a GP in our case? This depends on several factors to consider such as the size of the corpus, its richness with different types of annotations, the annotation granularity level reached, the representation format of annotation suitable and easy to manipulate, the syntactic representation structure, the used grammar, etc. And even if we find the appropriate treebank, understanding of its categories describing linguistic units is not an easy task. Another difficulty may be encountered when we build a GP namely, the induction of properties connecting categories, which can be easily deduced, or require heuristics.

In this paper, the building process of our GP consists of two tasks: The first one induces a CFG from the treebank. The second one specifies relations between categories

of each syntactic unit from the CFG rules. In view of the defined types of properties in this paper, the technique of GP induction we adopted is purely automatic and independent of any language and of the source treebank formalism. This promotes its reuse. In addition, this technique produces properties by providing changes of different granularity levels of grammatical categories. To the best of our knowledge, the obtained GP induced from a treebank, represents the first test product for Arabic. This product can be used by several other resources to extract their implicit information.

This paper is organized as follows: Section 2 is devoted to comparing the different Arabic treebanks under various criteria. Section 3 gives the reasons for the selection of appropriate treebank. Section 4 explains our approach. Experimental results will be presented and discussed in section 5. Section 6 gives a conclusion and perspectives.

2 Comparison among Arabic Treebanks

Many criteria distinguish the ATB, PADT, CATiB, and QADT treebanks, namely:

- **Source corpus:** The ATB source corpus is composed of newswires [6]. It was divided into more than 12 sets of texts (called divisions); comprising about 750K tokens [2]. It proved its effectiveness in a large number of works in various fields, and its divisions were converted by other treebanks to their syntactic representations, like PADT and CATiB [8,4]. As against, QADT treats the Holy Quran, from which it annotates 11K words and represents them on syntactic dependency graphs [3].
- **Used grammar:** PADT and ATB annotations follow the MSA theories suitable to their source texts [5,6]. CATiB facilitates annotation, based on traditional Arabic genre [4]. QADT follows, also, the same grammar suitable to its source text [3].
- **Syntactic representation:** The phrase structure, used by ATB, is a tree representation in which the words of a sentence appear as leaves and the categories as non-terminal nodes. Dependency structure, used by the 3 other treebanks, has also a tree representation except that the words of the sentence are the nodes of the tree.
- **POS tags:** ATB uses more than 400 tags and specifying different morphological features of Arabic words and includes empty pronouns [6]. PADT has a more complex morphology than ATB, including more detail in the features [5]. CATiB uses only 6 tags [4]. QADT uses 44 tags based on the 3 the main lexical categories (noun, verb and particle) of traditional Arabic grammar [3].
- **Syntactic and semantic relations:** ATB uses about 20 dashtags to represent syntactic and semantic features [6]. But, CATiB marks only syntactic functions [4]. PADT uses about 20 detailed tags deeper than CATiB, and QADT uses 43 syntactic and semantic relations based on dependency links of traditional Arabic grammar [3].

3 Choice of Treebank

The choice of source treebank depends on the goal we want to achieve. Indeed, we aim to induce a GP from an Arabic treebank. This grammar must describe, for each syntactic category, all grammatical categories contributing in its construction and the

relations existing between them. Syntactic categories are intermediate representations of the constituents of a sentence from the source treebank. This representation corresponds to a syntactic phrase structure but not to a dependency structure. Only ATB, described in Section 2, represents its annotations according to this structure. Several other issues led us to use ATB, namely: its rich POS tags and syntactic and semantic relations, its grammar adapted to MSA, the syntactic relevance of its source documents (converted to several other treebank representations), the variable category granularity offered, and the availability of a parentheses simple format to manipulate.

4 Proposed Approach

Our goal is to automatically induce a GP from ATB. This mechanism consists of two automatic tasks: The first one is to induce a CFG from ATB. The second task is to infer for each syntactic unit, the various relations (called properties) between its constituents from the rules described in the CFG. A constituent can be a syntactic or a lexical unit. The application of these two tasks allows us to obtain a GP.

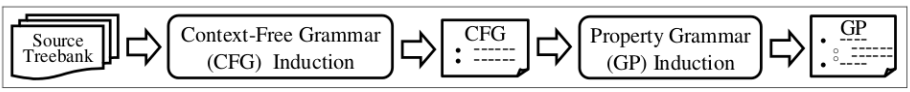

Fig. 1. Property grammar induction approach

As shown in Figure 1, the two outputs generated by the process tasks are not similar. Indeed, the first output represents each syntactic unit by a set of rules combining its various constituents (categories). But, the second task represents each syntactic unit in terms of its constituents (const), and of its properties. These properties have these different types described in the following figure:

Table 1. Functions of properties in the GP

Properties	Symbols	Functions
Uniqueness (unic)	Uniq	Set of constituents that cannot be repeated in a syntactic unit
Obligation (oblig)	Oblig	Set of possible heads of a syntactic unit
Linearity (lin)	$\prec$	Linear precedence relations between constituents of a syntactic unit
Requirement (req)	$\Rightarrow$	Mandatory co-occurrence between constituents
Exclusion (excl)	$\otimes$	Cooccurrence restriction between constituents

The contribution of the obtained GP focuses on the representation form of linguistic information in its formalism. Unlike the CFG, information is represented independently of its type or its position [1]. Thus, the GP also describes incomplete, partial and non-canonical information, which enhances its flexibility and robustness.

5 Experiments and Results

In a first step, we used an annotated corpus extracted from ATB to induce a CFG. From this, we induced a GP by deducing the set of properties describing the categories.

5.1 The used annotated corpus

We adopted the Part 2 of ATB (ATB2 v 3.1) consisting of 501 news, and including POS tags, morpho-syntactic structures at many levels and gloss [7]. 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. Each terminal is on the form of ("POS tag" "word"). And the "integrated" format brings together information about the source tokens, about the tree tokens, and the mapping between them and the tree structure.

After the presentation of these formats, we should choose the format to use as input in the CFG induction step. Since the CFG describes only the category level, our choice depends on 3 criteria: the simplicity of representation, the presence of a tree structure and the annotation at the syntactic level. The "penntree" format was the only one selected because it meets all these criteria. More specifically, we used the vowelized version of this format to avoid ambiguities related to the absence of vowels in Arabic.

5.2 Obtained grammar

With this approach, we obtained successively two different grammars: The CFG and the GP. Their size depends on the granularity level of categories it describes. A category specifies many features like mood, gender, number, etc. The higher this level, the more these grammars are complex, but the more it respects the language and vice versa.

Table 2. Frequency of the rule "PREP NP" describing the PP subcategories at the highest granularity level in the CFG

Phrases	PP	PP-CLR	PP-PRP	PP-TMP	PP-LOC	PP-PRD	PP-MNR	PP-DIR	Others
Σ# Rules	50	44	15	15	13	13	12	9	–
#Occ of "PREP NP"	12,834	3,025	445	754	1511	762	246	154	–
Σ# Occ of rules	13,814	3,781	684	805	1537	805	286	165	222

The obtained CFG contains sets of rules describing each non-terminal unit XP. The rule form is an ordered list of grammatical categories describing a syntactic category. Table 1 shows information about the obtained CFG at the highest granularity level, describing the category PP (Prepositional Phrase) and its subcategories (e.g. PP-MNR and PP-TMP), which include more details than PP [7]. In this level, there are 263 rules of different types. According to what we observed in Table 1, we note that the highest

granularity does not make a big difference for some subcategories of PP. “PREP NP” remains the most frequent rule whatever the PP subcategory it describes. Other rules are not frequent, sharing the rest of occurrences. For example, the subcategory PP-LOC is described, in addition, by a set of rules, needless to represent, each one not exceeding 10 occurrences and bringing together only 19 occurrences. Futhermore, the sign “#” assigned to some parameters denotes their cardinality. But, if we observe the CFG, illustrated in part in Table 2, we can note that regardless of the granularity level, the occurrences of “PREP NP” represents 90% of all rules in the treebank, often making unnecessary the increase of the granularity level. By reducing this granularity to 0, we obtain a CFG for PP more compact comprising only 59 rules incorporating mostly the rule “PREP NP”. Generally, for all syntactic categories, we noticed that the granularity of categories affects also the size of all the grammar. Indeed, the number of rules in the CFG is divided by 6 at the lowest level compared to the highest one (2998/14452).

Table 3. Excerpt from the CFG at the lowest granularity level describing the category PP

Rules	#Occ	Rules	#Occ	Rules	#Occ
PREP NP	19886	PREP ADVP	32	PP PREP NP	10
PREP SBAR	1346	NP PREP NP	28	PREP NP PUNC	10
PREP S	237	PREP PUNC NP	22	PREP UCP	8
PP CONJ PP	126	PP PP	20	PUNC PREP SBAR PUNC	7
-NONE-	87	PUNC PREP NP	20	PREP NP PP	6
PRT PREP NP	63	ADVP PREP NP	19	14 rules	≤ 5
PP PUNC CONJ PP	48	PREP PUNC NP PUNC	18	25 Other rules	1
PUNC PREP NP PUNC	42	Σ# Occurrences			22099

The GP generated at a given granularity level describes, for each syntactic category, all of its constituents and the properties which connect these constituents. Fig. 3 illustrates an excerpt of the obtained GP at the highest granularity level for the category PP. Thanks to the GP formalism, implicit information in the treebank are made explicit. It induces different types of properties connecting the various constituents. For example, in Fig. 3, we have “PREP $\prec$ S-NOM” as linearity property describing the subcategory PP-DIR and indicating that if the category PREP (Preposition) appears with the category S-NOM (Nominative clause) in the same construction, it will always directly or indirectly proceed S-NOM. Such information is not explicit in the treebank. But, with the highest granularity level of categories, a lot of implicit information may be repeated for several subcategories, increasing the GP size and making its run more difficult. In Fig. 3, this is so for the properties connecting the categories PREP and NP, which are repeated at least 6 times in the grammar for the indicated subcategories. The GP at the lowest granularity level is very different. It becomes much more compact, the categories are simpler and the properties are not repeated. This is because these categories were generalized and factored. But, this lack of precision may lead to a loss of information. The linearity property “PRON_3MS $\prec$ DET+ADJ+ CASE_DEF_NOM” describing

the subcategory NP-ADV-1 is an example, among others, proving this idea. After its generalization, normally, its precision should be reduced, and should be converted to the following rule "PRON $\prec$ ADJ" to describe the basic category NP. But this did not happen. This is explained by the fact that the validity of this property has not been guaranteed for all NP subcategories. The absence of several properties due to generalization can increase the error rate.

PP-DIR	Const	{PP, PREP, NP, ADVP, S-NOM, SBAR, PRT}
	Uniq	{PREP, NP, ADVP, S-NOM, SBAR, PRT}
	Lin	PP $\prec$ {PREP, NP}; PREP $\prec$ {NP, ADVP, S-NOM, SBAR}; PRT $\prec$ {PREP, NP}
	Req	{NP, ADVP, S-NOM, SBAR}$\Rightarrow$ PREP; PRT $\Rightarrow$ {PREP, NP}
	Excl	PP $\otimes$ {ADVP, S-NOM, SBAR, PRT}; NP $\otimes$ {ADVP, S-NOM, SBAR} ADVP $\otimes$ {S-NOM, SBAR, PRT}; S-NOM $\otimes$ {SBAR, PRT};SBAR $\otimes$ {PRT}

PP-DTV	Const	{PREP, NP}
PP-TPC	Uniq	{PREP, NP}
PP-LOC	Oblig	{PREP, NP}
PP-MNR	Lin	PREP $\prec$ NP
PP-PRD	Req	NP $\Rightarrow$ PREP; PREP $\Rightarrow$ NP

Fig. 2. Excerpts from the GP at the highest granularity level describing the category PP

According to the results, we note that the granularity level has a major impact on the complexity of the GP inducing problem. Indeed, with a high granularity level, we have representative and detailed properties, because of the high number of categories in the CFG. But, this produces particularly an over-generation for exclusion properties, which increases the complexity of the problem. On the opposite, with a low granularity level, we obtain a more reduced number of categories in the CFG, which makes the properties safe but very general, losing thus information. We should control the granularity level to make a compromise between the quantity and the quality of these properties.

6 Conclusion and Perspectives

We proposed in this paper an approach of building a GP from ATB. In this new resource, we made explicit different types of implicit information by inducing properties connecting various grammatical categories of ATB. The result is a resource on a wide coverage provided at different granularity levels, and inheriting ATB qualities among which its reliability, its submission to consensus and its rich annotation. This resource was built in an internship at the laboratory LPL (Aix-en-provence). The technique that we adopted to build this resource is generic. Indeed, it is independent of any language as well as of the source formalism, since properties are directly generated from the CFG. In addition, by applying only the types of properties defined so far, this technique is purely automatic, which promotes its reusability.

As perspectives, we will show a control of the granularity level of categories in future works. Besides, in order to offer a very precise representation of syntactic information, we can enrich or modify the relations presented in the GP. In future works, this grammar can also be used to enrich the Arabic treebank with a property-based representation to improve its quality. To optimize this enrichment, several control mechanisms can be integrated into determining syntactic categories and evaluability of their linguistic properties. The ATB size can also be enriched by converting the source documents annotated by other Arabic Treebanks (like PADT and CATiB) to the ATB representation as they have already done. So, having a corpus format unifying the different representations is useful, offering thus interoperability of annotated data and extending the applicability of divergent NLP tools in different research contexts.

References

1. Blache, P.: Les Grammaires de Propriétés: Des contraintes pour le traitement automatique des langues naturelles. Hermès Sciences Publications (2001)
2. Diab, M. T., Habash, N., Rambow, O., Roth, R.: LDC Arabic Treebanks and Associated Corpora: Data Divisions Manual. Columbia University, Technical Report, Center for Computational Learning Systems (2013)
3. Dukes, K., Buckwalter, T.: A Dependency Treebank of the Quran using traditional Arabic grammar. Institute of Electrical and Electronics Engineers (2010)
4. Habash N., Faraj R., Roth, R.: Syntactic Annotation in the Columbia Arabic Treebank. In Conference on Arabic Language Resources and Tools, Cairo, Egypt (2009)
5. Hajič, J., Smrž, O., Zemánek, P., Snaidauf, J., Beska, E.: Prague Arabic Dependency Treebank: Development in Data and Tools. In Proceedings of the NEMLAR international conference on Arabic Language Resources and Tools (2004)
6. Maamouri, M., Bies, A., Buckwalter, T.: The Penn Arabic Treebank: Building a Large-scale Annotated Arabic Corpus. In Proceedings of the Network for Euro-Mediterranean Language Resources Conference on Arabic Language Resources, Cairo, Egypt (2004)
7. Maamouri, M., Bies, A., Krouna, S., Gaddeche, F., Bouziri, B.: Penn Arabic Treebank guidelines v4.8. Technical report, LDC, University of Pennsylvania (2009)
8. Smrž O., Bielický, V., Kouřilová, I., Kráčmar, J., Hajič, J., Zemánek, P.: Prague Arabic Dependency Treebank: A Word on the Million Words. In Proceedings of the Workshop on Arabic and Local Languages (LREC 2008), pp. 16–23, Marrakech, Morocco (2008)