JEP-TALN 2004, Session Traitement Automatique de l'Arabe, Fès, 20 avril 2004

Extracting a Tree Adjoining Grammar from the Penn Arabic Treebank

Nizar Habash
Institute for Advanced Computer Studies
University of Maryland
College Park, Maryland, USA
habash@umiacs.umd.edu

Owen Rambow
Department of Computer Science
Columbia University
New York, New York, USA
rambow@cs.columbia.edu

Mots-clefs – Keywords

syntaxe de l'arabe, banque d'arbres, grammaire d'arbres adjoints
Arabic syntax, treebanks, Tree Adjoining Grammar

Résumé - Abstract

Nous décrivons l'extraction d'une grammaire d'arbres adjoints (TAG) à partir d'une banque d'arbres de l'arabe écrit. Nous montrons quelques exemples d'arbres élémentaires ainsi obtenus, et les structures de dérivation (donc, de dépendence syntaxique) qui y correspondent.

We describe the extraction of a Tree Adjoining Grammar from the Penn Arabic Treebank.[1] We show some examples of extracted trees for different constructions, and the corresponding derivation structures (which represent syntactic dependency).

1 Introduction

Much progress in natural language processing (NLP) over the last decade has come from the combination of using corpora of annotated naturally occurring text along with machine learning algorithms. Following this trend, corpora have been created for other languages, such as the Penn Arabic Treebank (PATB) (Maamouri et al.2003). However, the corpora almost invariably need to reinterpreted for the machine learning algorithm; for example, the English parser of Collins (1997) uses a notion of head percolation on top of the phrase structure of the Penn Treebank (a syntactically annotated corpus of English). In this paper, we reinterpret the PATB as a dependency corpus and we also extract a Tree Adjoining Grammar from the corpus. We use part 1, version 2.0 of the PATB, which comprises around 160,000 words of annotated Arabic text from newswire sources.

[1]This work was funded in part by the National Science Foundation under grant 0329163, "Arabic Dialect Modeling for Speech and Natural Language Processing".

There are several reasons why we may want to convert the PATB to a dependency representation and extract a TAG grammar. A dependency representation is often seen as being closer to semantics and is therefore useful for many applications. A TAG grammar, whether hand-crafted or extracted, can be used for developing standard parsers (Chiang2000), but they can also be used for "supertagging" in which we automatically assign tags representing rich syntactic structure to words in an input string (Bangalore and Joshi1999). Supertagged text can be used for chunking and quick parsing. Furthermore, extracted grammars can be used in generation as well (Bangalore and Rambow2000). Finally, the explicit linguistic information that the decomposition into TAG elementary trees gives us is useful when we want to analyze the corpus to see how frequent certain linguistic phenomena are.

2 Related Work

There has been extensive work on extracting TAGs from English corpora and on creating a dependency corpus for Arabic. To our knowledge, this is the first attempt to extract a TAG from an Arabic corpus.

There have been several efforts aimed at extracting TAGs from the PTB (Xia et al.2000; Chen2001; Chiang2000). Our work follows that of Chen (2001) most closely, as its goal is less to replicate the phrase structure of the original corpus, and more to produce a dependency structure which is well motivated.

A dependency treebank of Arabic in the style of the Prague Dependency formalism is being created (Smrž and Zemánek2002; Žabokrtský and Smrž2003). The work includes automatic conversion of the PATB. While our level of representation is somewhat "deeper" than the Praguian analytical level, but entirely syntactic, unlike the Praguian tectogrammatical level (which includes semantic labels), our dependency representations of course in many ways are similar to those proposed by these authors.

3 Tree Adjoining Grammar and Dependency Syntax

We give a very shory introduction to Tree Adjoining Grammar (TAG). For more information, see (Abeillé and Rambow2000). In TAG, the elementary structures are phrase-structure trees. We combine elementary structures in a TAG by using two operations, *substitution* and *adjunction*. We can substitute tree β into tree α if there is a nonterminal symbol on the frontier of α which has the same label as the root node of β. We can then simply append β to α at that node. In adjunction, a tree α contains a non-terminal node labeled A; the root node of tree β (an "auxiliary tree") is also labeled A, as is exactly one non-terminal node on its frontier (the "foot node"). All other frontier nodes are terminal nodes or substitution nodes. Adjunction has the effect of inserting one tree into the center of another. TAG elementary structures have an extended "domain of locality". This increased domain of locality allows the linguist to associate each tree with one lexical head, and to state linguistic relationships of the lexical head locally in the elementary tree (such as subcategorization, semantic roles of arguments, case assignment, agreement, and word order). We call a formalism which has one lexical head per tree a "lexicalized grammar".

A TAG derives a phrase-structure tree, called the "derived tree". In addition to the derived tree, a second structure is built up, the "derivation tree". The derivation tree records how the derived tree was assembled from elementary trees. In this structure, each of the elementary trees is represented by a single node. Since the grammar is lexicalized, we can identify this node with the (base form of the) lexeme of the corresponding tree. If a tree t_1 is substituted or adjoined into a tree t_2, then the node representing t_1 becomes a dependent of the node representing t_2 in the derivation tree. Furthermore, the arcs between nodes are annotated with the position in the "target tree" at which substitution or adjunction takes place. We can see that the derivation structure is a dependency tree. For more discussion of the relation between the derivation structures and dependency structures, see (Rambow and Joshi1997; Candito and Kahane1998).

4 Imposing a Dependency Analysis on the PATB and Extracting a TAG

In this section, we give analyses for several constructions, illustrating how we go from phrase structure to dependency and TAG elementary trees. The dependency structure is the derivation structure one obtains when using the given TAG elemntary trees to derive the Arabic sentence fragment.

We would like to emphasize that both the conversion of a phrase-structure corpus to dependency, and the extraction of a TAG, involve many possible choices: there is not single correct answer for either. Rather, the coders of the PATB have made choices about how to encode certain constructions and have documented these decisions in the annotation manual. When extracting a TAG, or converting to a different representation such as dependency, the researcher has the option of devising new representations for these constructions – the PATB does not dictate which representation we choose. To extract or convert the new representations, we must write *ad-hoc* procedures to analyze the original treebank. We cannot expect that a small set of rules along with a generic rule engine will be sufficient for the mapping (though they may be sufficient for a first rough approximation).

4.1 Basic Clause Structure

Arabic has both VSO (verb-subject-object) word order and SVO (subject-verb-object) order, with VSO more common. The two orders are illustrated with the following highly simplified sentences from the PATB:

(1) a. إستضافت هانوي مؤتمراً
AstDAft hAnwY m&tmrA
hosted Hanoi conference
Hanoi hosted a conference

b. هذا يكشف الترابط
h*A yk$f AltrAbT
this shows the+connection
this shows the connection

The PATB represents the VSO order as underlying, with the subject in the VP, while the SVO and OVS orders are represented in a similar manner, namely with the subject or object preposed as sister to the VP and coindexed with a *T* trace in its original position. The preposed element has dashtag TPC, while the NPs in the VP have dashtags indicating their grammatical function (SUBJ and OBJ).

The extracted TAG elementary trees are shown in Figure 1. We see that the trees, as usual in TAG elementary trees, comprise the head, positions for the arguments, traces, and all antecedents for traces. The tree for VSO is tree 54, with 448 occurrences in the training corpus, while the SVO tree is tree 285, with only 55 occurrences in the training corpus. (There were no cases of OVS order in the training corpus.)

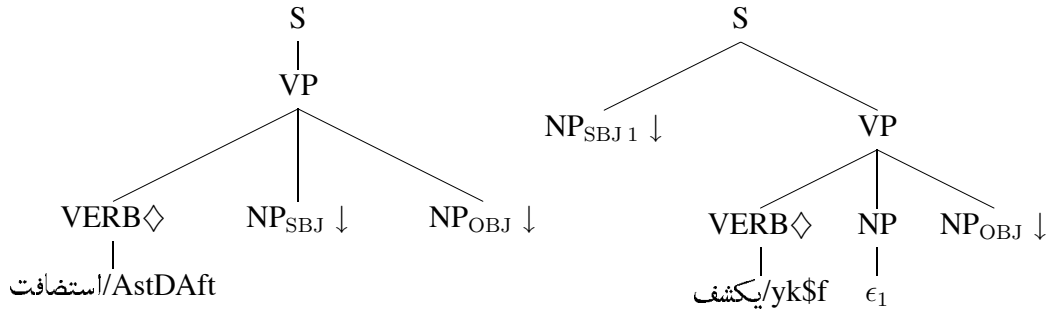

Figure 1: Tree for main-clause SOV construction (left) and tree for main-clause SVO construction right)

In the dependency representation, the different word orders are represented with the same trees, with the only difference (apart from the word order, if it is encoded) being the feature real:topic on the constituent head which is pre-verbal Figure 2.

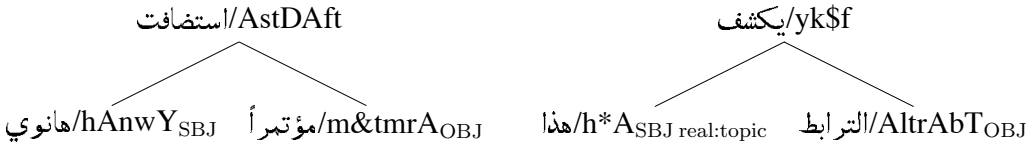

Figure 2: Dependency/derivation tree for sentence ((1)a) (left) and sentence ((1)b) (right)

4.2 Predicative Constructions

In the analysis of predicative constructions with nouns, prepositions, and adjectives, we follow the precedent set for English TAGs in the XTAG project (XTAG-Group1999) and analyze the predicate as the head, not the auxiliary. Of course, in Arabic additional support for such an analysis comes from cases in which there is no (overt) copula; for example, Smrz and Zemanek (2002) also choose this analysis. However, we are consistent in using the predicate-as-head analysis even in cases in which a copula is present. The motivation is partly to achieve consistency in our syntactic analysis: we do not want the head to vary between the predicate or the copula. Mainly, however, our motivation is semantic: we want our syntactic analysis to be as close to semantics as possible, and in the semantics, the nominal, adjectival, or prepositional head takes the subject as an argument directly (we predicate of Abdul that he is a doctor, for example). As examples we give nominal predicative constructions from the PATB (again radically simplified), one without copula, one with.

(2) a. ذلك عشية الاحتفالات
*lk E$yp AlAHtfAlAt
that eve the-celebrations
that is at the eve of the celebration

b. كان الهويدي انشط المهاجمين
kAn Alhwydy An$T AlmhAjmyn
was Al-Huwaidi most-active the-fighters
Al-Huawidi was the most active of the fighters

The PATB represents the copula-free predicative construction as two sister NPs under an S node, one with dashtag SBJ, the other with dashtag PRD. If there is a copula, a VP is added, with the copula under the VP. If the subject is fronted before the copula, the same mechanism applies as described in Section 4.1. Unfortunately, this means that for the extracted TAG grammar, while we wish to treat the presence of the copula as minor variation, we do end up with two different extracted trees, depending on whether we add the copula or not. They are shown in Figure 3 along with the auxiliary tree used to adjoin in the copula. Instead, we could, in the extraction process, always add the VP node, so that all nominal predicative constructions, whether with overt copula or not.

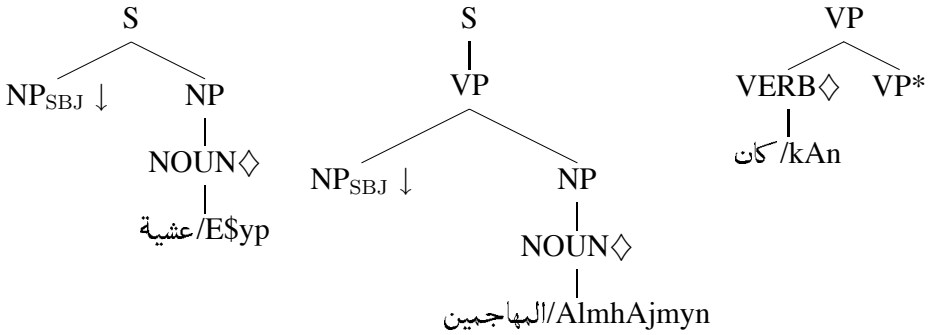

Figure 3: Tree for nominal predicative construction without copula (left), with VP for adjoining copula (center), and auxiliary tree for adjoining copula (right)

The dependency structures for the sentences in ((2)) are shown in Figure 4.

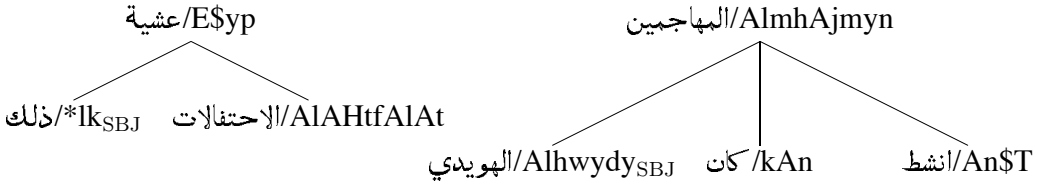

Figure 4: Dependency/derivation tree for sentence ((2)a) (left) and sentence ((2)b) (right)

5 The Extracted TAG

We first consider the growth of the grammar. We expect that the number of incrementally new tree types should decline: as the corpus from which the grammar is extracted grows, a wider variety of constructions has been seen, and fewer new constructions are encountered. As expected, the incremental number of trees quickly declines initially, but then remains constant (around 200) from 40,000 words to 110,000 words. We interpret this phase as representing the accumulation of the core grammar for Arabic. It then starts declining again. Unfortunately, the corpus is too small to see if this trend continues, but we hope to confirm this when the next,larger version of the corpus is released.

This paper has provided a very short introduction to the current state of the project. As we have stressed, the corpus does not determine a single TAG, and we will continue to experiment with extraction parameters so that we can obtain a grammar that is optimal with respect to both the quality of its linguistic description and its processing properties in, for example, parsing and generation.

References

Anne Abeillé and Owen Rambow. 2000. Tree Adjoining Grammar: An overview. In Anne Abeillé and Owen Rambow, editors, *Tree Adjoining Grammars: Formalisms, Linguistic Analyses and Processing*, pages 1–68. CSLI Publications.

Srinivas Bangalore and Aravind Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–266.

Srinivas Bangalore and Owen Rambow. 2000. Exploiting a probabilistic hierarchical model for generation. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, Saarbrücken, Germany.

Marie-Hélène Candito and Sylvain Kahane. 1998. Can the TAG derivation tree represent a semantic graph? An answer in the light of Meaning-Text Theory. In *Proceedings of the Fourth International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+4)*, IRCS Report 98–12, pages 21–24. Institute for Research in Cognitive Science, University of Pennsylvania.

John Chen. 2001. *Towards Efficient Statistical Parsing Using Lexicalized Grammatical Information*. Ph.D. thesis, University of Delaware.

David Chiang. 2000. Statistical parsing with an automatically-extracted tree adjoining grammar. In *38th Meeting of the Association for Computational Linguistics (ACL'00)*, pages 456–463, Hong Kong, China.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Madrid, Spain, July.

Mohamed Maamouri, Ann Bies, Hubert Jin, and Tim Buckwalter. 2003. Arabic treebank: Part 1 v 2.0. Distributed by the Linguistic Data Consortium. LDC Catalog No.: LDC2003T06.

Owen Rambow and Aravind Joshi. 1997. A formal look at dependency grammars and phrase-structure grammars, with special consideration of word-order phenomena. In Leo Wanner, editor, *Recent Trends in Meaning-Text Theory*, pages 167–190. John Benjamins, Amsterdam and Philadelphia.

Otakar Smrž and Petr Zemánek. 2002. Sherds from an arabic treebanking mosaic. *Prague Bulletin of Mathematical Linguistics*, (78).

Fei Xia, Martha Palmer, and Aravind Joshi. 2000. A uniform method of grammar extraction and its applications. In *Proc. of the EMNLP 2000*, Hong Kong.

The XTAG-Group. 1999. A lexicalized Tree Adjoining Grammar for English. Technical Report `http://www.cis.upenn.edu/~xtag/tech-report/ tech-report.html`, The Institute for Research in Cognitive Science, University of Pennsylvania.

Zdeněk Žabokrtský and Otakar Smrž. 2003. Arabic syntactic trees: from constituency to dependency. In *Eleventh Conference of the European Chapter of the Association for Computational Linguistics (EACL'03) – Research Notes*, Budapest, Hungary.