Arabic Syntactic Trees: from Constituency to Dependency

Zdeněk Žabokrtský and **Otakar Smrž**
Center for Computational Linguistics
Faculty of Mathematics and Physics
Charles University in Prague
{zabokrtsky,smrz}@ckl.mff.cuni.cz

Abstract

This research note reports on the work in progress which regards automatic transformation of phrase-structure syntactic trees of Arabic into dependency-driven analytical ones. Guidelines for these descriptions have been developed at the Linguistic Data Consortium, University of Pennsylvania, and at the Faculty of Mathematics and Physics and the Faculty of Arts, Charles University in Prague, respectively.

The transformation consists of (i) a recursive function translating the topology of a phrase tree into a corresponding dependency tree, and (ii) a procedure assigning analytical functions to the nodes of the dependency tree.

Apart from an outline of the annotation schemes and a deeper insight into these procedures, model application of the transformation is given herein.

1 Introduction

Exploring the relationship between constituency and dependency sentence representations is not a new issue—the first studies go back to the 60's (Gaifman (1965); for more references, see e.g. Schneider (1998)). Still, some theoretical findings had not been applicable until the first dependency treebanks with well-defined annotation schemes came into existence just in the very last years (Hajič et al., 2001).

The need to convert Arabic treebank data of different descriptions arises from a co-operation between the Linguistic Data Consortium (LDC), University of Pennsylvania, and three concerned institutions of Charles University in Prague, namely the Center for Computational Linguistics, the Institute of Formal and Applied Linguistics, and the Institute of Comparative Linguistics.

The two parties intend to share the resources they create. Prior to this exchange, 10,000 words from the LDC Arabic Newswire A Corpus were manually annotated in both syntactic styles as a step to ensure that the annotations are re-usable and their concepts mutually compatible. Here we attempt the constituency–dependency direction of the transfer.

1.1 Phrase-structure trees

The input data come from the LDC team (Maamouri et al., 2003). The annotation scheme is based on constituent-syntax bracketing style used at the University of Pennsylvania (Maamouri and Cieri, 2002). The trees include nodes for surface text tokens as well as non-terminal nodes following from the descriptive grammar. Not only syntactic elements, but also several kinds of structural reconstructions (traces) are captured here.

1.2 Analytical trees

Under the analytical tree structure we understand a representation of the surface sentence in form of a dependency tree. The node set consists of all the tokens determined after morphological analysis of the text, and the sentence root node. The description recovers the relation between a governor and a node dependent on it. The nature of the government is expressed by the analytical functions of the nodes being linked.

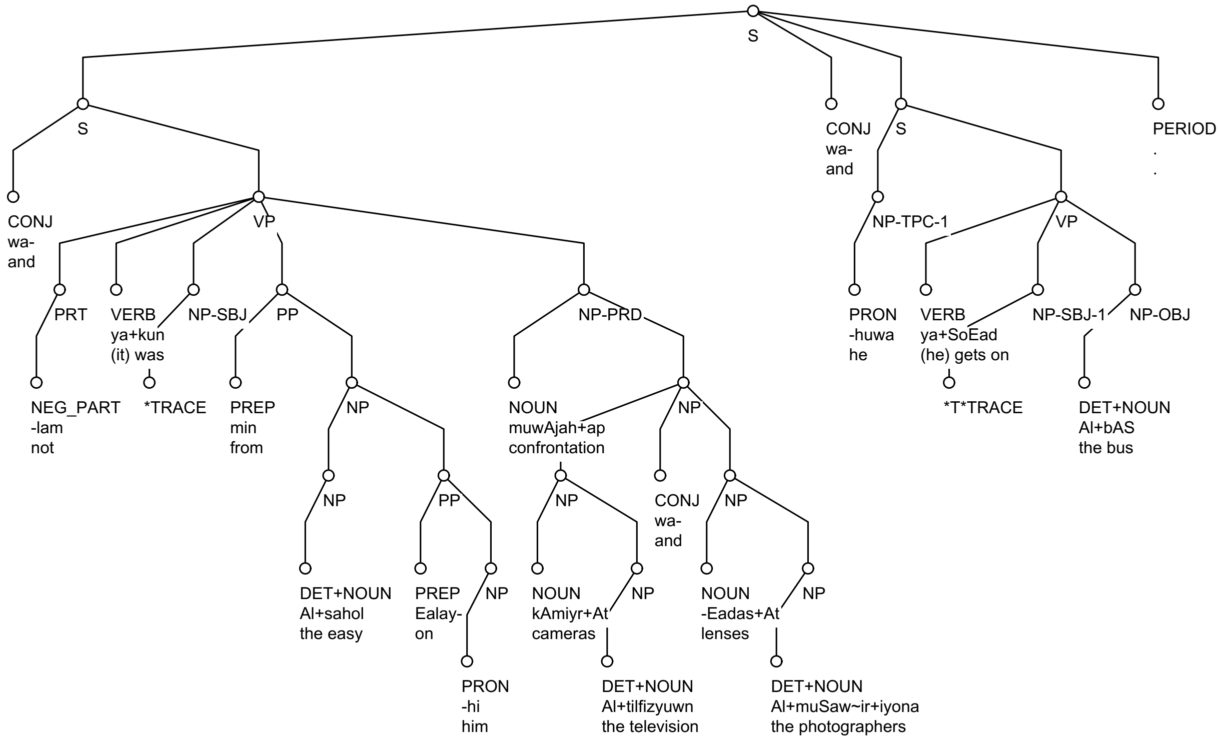

Figure 1: The model sentence in the phrase-structure syntactic description. The nodes are labeled either with part-of-speech (POS) tags, or with the names of non-terminals.

1.3 Model sentence

Let us give a model sentence which in its phonetic transcript and translation reads

> *Wa lam yakun mina ’s-sahli ʿalay hi muwāǧahatu kāmīrāti ’t-tilfizyūni wa ʿadasāti ’l-muṣawwirīna wa huwa yaṣʿadu ’l-bāṣa.*
>
> It was not easy for him to face the television cameras and the lenses of photographers as he was getting on the bus.

Its respective representations in Figures 1 and 2 use glossed tokens which are further split into morphemes and transliterated in Tim Buckwalter’s notation of graphemes of the Arabic script.

There are three phenomena to focus on in the trees. Firstly, occurrence of the empty trace `(*TRACE) NP-SBJ` or the `(*T*TRACE) NP-SBJ-1` one with its contents moved to `NP-TPC-1`. Secondly, subtree interpretation may be sensitive to other than the top-level nodes, like when the coordination `S CONJ S` produces the subordinate complement clause `Pred (Atv)` due to the idiomatic context of the pronoun. Finally to note are complex rearrangements of special constructs, as is the case of `NP-SBJ PP NP-PRD` versus `AuxP AuxP Sb` nodes and their subtrees. More discussion follows.

1.4 Outline of the transformation

The two tree types in question differ in the topology as well as in the attributes of the nodes. Thus, the problem is decomposed into two parts:

i) creation of the dependency tree topology, i.e. contraction of the phrase-structure tree based mostly on the concept of phrase heads and on resolution of traces,

ii) assignment of labels describing the analytical function of the node within the target tree.

2 Structural Transformation

2.1 The core algorithm

The principle of the conversion of phrase structures into dependency structures is described clearly in Xia and Palmer (2001) as (a) mark the

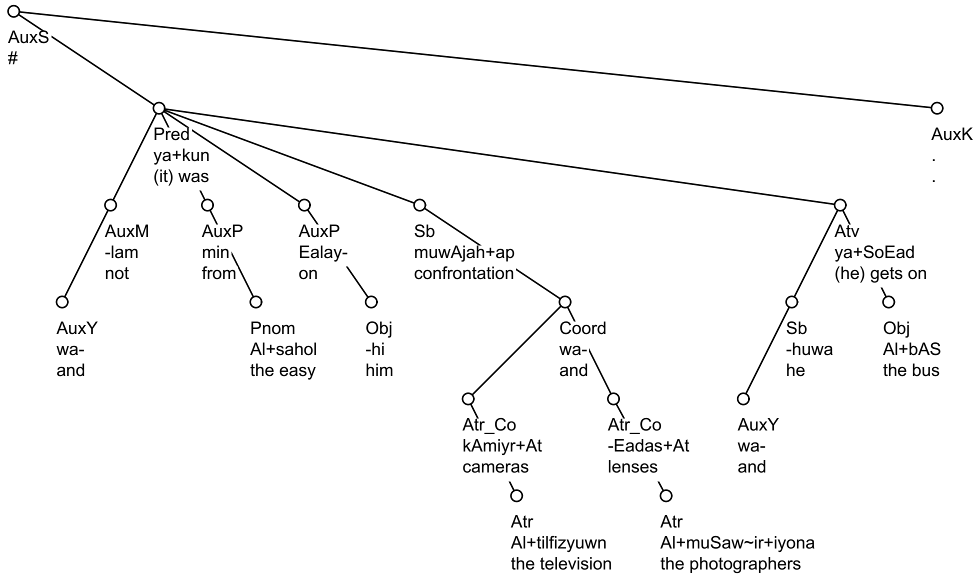

Figure 2: The model sentence in the dependency analytical description, showing the nodes and their functions in the hierarchy.

head child of each node in a phrase structure, using the head percolation table, and (b) in the dependency structure, make the head of each non-head child depend on the head of the head-child.

In our implementation, the topology of the analytical tree is derived from the topology of the phrase tree by a recursive function, which has the following input arguments: original phrase tree T_{phr}, dependency tree T_{dep} being created, one particular node s_{phr} from T_{phr} (the root of the phrase subtree to be processed), and node p_{dep} from T_{dep} (the future parent of the subtree being processed). The function returns the root of the created analytical subtree. The recursion works like this:

1. If s_{phr} is a terminal node, then create a single analytical node n_{dep} in T_{dep} and attach it below p_{dep}; return n_{dep};

2. Otherwise (s_{phr} is a nonterminal), choose the head node h_{phr} among the children of s_{phr}, recursively call the function with h_{phr} as the phrase subtree root argument, and store its return value r_{dep} (root of the recursively created dependency subtree); recursively call the function for each remaining s_{phr}'s child $n_{phr,i}$, and attach the returned subtree root $o_{dep,i}$ below r_{dep}; return r_{dep}.

2.2 Appointing heads

Rules for the selection of phrase heads follow from the analytical annotation guidelines. Predicates are considered the uppermost nodes of a clause, prepositions govern the rest of a prepositional phrase, auxiliary words are annotated as leaves etc. Non-verbal predication, so frequent in Arabic syntax, is also formalized into the terms of dependency, cf. Smrž et al. (2002).

With the algorithm taking decisions about the head child before scanning the subtrees of the level, the already mentioned clause *huwa yaṣʿadu ’l-bāṣa* qualifies improperly as a sister to the predicate *yakun* of the main clause. In fact, we are dealing with the so called state or complement clause. Therefore, corrective shuffling in this respect is inevitable.

2.3 Tree post-processing

Completion of the dependency tree also involves pruning of subtrees which are co-indexed with some trace, and attaching them in place of the referring trace node. Typically, this is the case for clauses having an explicit subject before the predicate. In the model sentence, *yaṣʿadu* retains its role as a predicate of the clause, no matter what function it receives from its governor.

3 Analytical Function Assignment

The analytical function can be deduced well from the POS of the node and the sequence of labels of all its ancestors in the phrase tree, and from the POS or the lexical attributes of its parent in the dependency tree. That is why this step succeeds the structural changes.

Problems may appear though if the declared constituents are not consistent enough, relative to the analytical concept. While `NP-SBJ`, `PP` and `NP-PRD` would normally imply `Sb`, `AuxP` and `Pnom`, these get in principal conflict in the type of nominal predicates like *mina 's-sahli* followed by an optional object and a rhematic subject. The Figures provide the best insight into the differences.

4 Evaluation and Conclusion

Preliminary evaluation gives 60 % accuracy of the generated tree topology, and roughly the same rate for analytical function assignment. The measure is the percentage of correct values of parents/functions among all values. The work is in progress, however. According to our experience with similar task for Czech, English (Žabokrtský and Kučerová, 2002) and German, we expect the performance to improve up to 90 % and 85 % as more phenomena are treated.

The experience made during this task shall be useful for the development of a rule-based dependency partial analysis, which shall pre-process data for manual analytical annotation.

Acknowledgements

Development and fine-tuning of the transformation procedures would not have been possible without the TrEd tree editor by Petr Pajas of the Charles University in Prague. The Figures were produced with it, too.

The phonetic transcription of Arabic within this paper was typeset using the ArabTEX package for TEX and LATEX by Prof. Dr. Klaus Lagally of the University of Stuttgart.

The research described herein has been supported by the Ministry of Education of the Czech Republic, projects LN00A063 and MSM113200006.

References

Haim Gaifman. 1965. Dependency Systems and Phrase-Structure Systems. *Information and Control*, pages 304–337.

Jan Hajič, Eva Hajičová, Petr Pajas, Jarmila Panevová, Petr Sgall, and Barbora Vidová-Hladká. 2001. Prague Dependency Treebank 1.0 (Final Production Label). CDROM CAT: LDC2001T10, ISBN 1-58563-212-0.

Mohamed Maamouri and Christopher Cieri. 2002. Resources for Natural Language Processing at the Linguistic Data Consortium. In *Proceedings of the International Symposium on Processing of Arabic*, pages 125–146, Tunisia, April 18th–20th. Facultè des Lettres, University of Manouba.

Mohamed Maamouri, Ann Bies, Hubert Jin, and Tim Buckwalter. 2003. Arabic Treebank: Part 1 v 2.0. LDC catalog number LDC2003T06, ISBN 1-58563-261-9.

Gerold Schneider. 1998. A Linguistic Comparison Constituency, Dependency, and Link Grammar. Master's thesis, University of Zurich.

Otakar Smrž, Jan Šnaidauf, and Petr Zemánek. 2002. Prague Dependency Treebank for Arabic: Multi-Level Annotation of Arabic Corpus. In *Proceedings of the International Symposium on Processing of Arabic*, pages 147–155, Tunisia, April 18th–20th. Facultè des Lettres, University of Manouba.

Fei Xia and Martha Palmer. 2001. Converting Dependency Structures to Phrase Structures. In *Proceedings of the Human Language Technology Conference (HLT-2001)*, San Diego, CA, March 18–21.

Zdeněk Žabokrtský and Ivona Kučerová. 2002. Transforming Penn Treebank Phrase Trees into (Praguian) Tectogrammatical Dependency Trees. *Prague Bulletin of Mathematical Linguistics*, (78):77–94.