PARSING ARABIC USING TREEBANK-BASED LFG RESOURCES

Lamia Tounsi Mohammed Attia Josef van Genabith
NCLT, School of Computing
Dublin City University

Proceedings of the LFG09 Conference

Miriam Butt and Tracy Holloway King (Editors)

2009

CSLI Publications

http://csli-publications.stanford.edu/

Abstract

In this paper we present initial results on parsing Arabic using treebank-based parsers and automatic LFG f-structure annotation methodologies. The Arabic Annotation Algorithm (A^3) (Tounsi et al., 2009) exploits the rich functional annotations in the Penn Arabic Treebank (ATB) (Bies and Maamouri, 2003; Maamouri and Bies, 2004) to assign LFG f-structure equations to trees. For parsing, we modify Bikel's (2004) parser to learn ATB functional tags and merge phrasal categories with functional tags in the training data. Functional tags in parser output trees are then "unmasked" and available to A^3 to assign f-structure equations. We evaluate the resulting f-structures against the DCU250 Arabic gold standard dependency bank (Al-Raheb et al., 2006). Currently we achieve a dependency f-score of 77%.

1 Related Work

Arabic parsing systems have been reported in (Ditters, 2001; Zabokrtsky and Smrz, 2003; Othman et al., 2003; Ramsay et Mansour, 2007). (Attia, 2008) gives an overview of an LFG rule-based analysis of Arabic using XLE (Xerox Linguistics Environment). He concentrated on short sentences and used robustness techniques to increase the coverage. All of these use hand-crafted grammars, which are time-consuming to produce and difficult to scale to unrestricted data. More recently, the Penn Arabic Treebank (ATB) has been employed to acquire wide-coverage parsing resources. The best-known Arabic statistical parser was developed by Bikel (Bikel, 2004). Bikel reports parse quality "far below" English and Chinese (Kulick et al., 2006). The main reasons cited were a significant number of POS-tag inconsistencies (in the version of the ATB available at the time) and the considerable differences between Arabic and English sentence structure. (Dieb et al., 2004) and (Habash and Rambow, 2005) present knowledge- and machine-learning-based methods for tokenisation, basic POS tagging with a reduced tagset and base phrase chunking. Bikel's parser produces phrase-structure trees (c-structures). The main objective of our research is to automatically enrich the output of Bikel's parser with more abstract and "deep" dependency information (in the form of LFG f-structures), using the Arabic A^3 annotation algorithm (Tounsi et al., 2009), extending the approach of (Cahill et al., 2004), originally developed for English.

2 The Penn Arabic Treebank (ATB)

Arabic is a subject pro-drop language. It has relatively free word order: mainly S(ubject) V(erb) and O(bject), with VSO and VOS also possible. Arabic is a highly inflectional and cliticizing language. The ATB consists of 23,611 parse-annotated sentences (Bies and Maamouri, 2003; Maamouri and Bies, 2004) from Arabic newswire text in Modern Standard Arabic (MSA). The ATB annotation scheme involves 497 different POS-tags with morphological information (reduced to 24 basic POS-tags by Bikel e.g. NN, NNS, JJ), 22 phrasal tags e.g. NP, VP, PP and 20 functional tags e.g. SBJ, OBJ, TPC (52 combined functional tags, as functional tags can stack).

3 The Arabic Annotation Algorithm (A^3)

The Arabic Annotation Algorithm (Tounsi et al., 2009) is constructed adapting and revising the methodology of (Cahill et al., 2004) for English as follows:

1. Automatic extraction of the most frequent rule types from the treebank[1].

2. Head lexicalisation of ATB trees to identify local heads.

3. Default f-structure equations are assigned to ATB functional tags. In addition, lexical macros exploits the rich morphological information provided by the ATB.

[1] With 85% token coverage.

4. Left/right annotation principles for COMPs, XCOMPs, ADJUNCTs, etc[2].

5. Coordination

6. Traces to handle non-local dependencies.

(Tounsi et al., 2009) report an f-score of 95% on automatically annotated gold ATB trees against the DCU250 Arabic Dependency Bank.

4 Adapting the Parser

We use Bikel's implementation of Collins' Model 1 as our c-structure engine (Bikel, 2004). As the A^3 of (Tounsi et al., 2009) heavily relies on ATB function tags, we modify the Bikel parser to learn ATB tags. We "mask" ATB function tags in the training data by merging phrasal with function tags and adjust the head-finding rules in Bikel's Arabic language pack accordingly. For example, the functional and phrasal tag NP-OBJ are stuck together as NP_OBJ which makes the shallow parser interpret it as one phrasal tag during training and parsing (NP-OBJ ⇒ NP_OBJ). After parsing, we unmask ATB function tags and make them available to A^3.

5 Experiments and Evaluation

250 of the 23,611 parse-annotated sentences in ATB were randomly selected as test set (Dieb et al., 2004). The DCU 250 gold standard dependency bank for Arabic (Al-Raheb et al., 2006) is semi-automatically constructed using A^3 and manual correction and extension. We use gold-POS-tagged ATB text and the lexical morphological information from ATB in the results reported below:

Precision	Recall	F-score
70.40	72.38	71.37

Table 1: C-structure evaluation (Evalb).

Precision	Recall	F-score
74.75	81.07	77.78

Table 2: F-structure evaluation.

6 Discussion and Further Work

Compared to similar results for English which have a dependency f-score of 87% against DCU 105 (Cahill et al., 2002), initial results (dependency f-score of 77%) for Arabic are somewhat disappointing. The most likely reason is the explosion in the size of the phrasal category set with 22 ATB phrasal categories as opposed to 150 (masked) categories (fusing ATB phrasal and functional tags) to be learnt by Bikel's parser, resulting in substantial data-sparseness. However, the result provides a base-line for what, to the best of our knowledge, is the first treebank-based LFG parsing approach to Arabic. In an effort to improve on the baseline presented in this paper, our current experiments use a two-stage architecture with a simple probabilistic phrase-structure parser, followed by a machine-learning-based ATB function labeller, to provide input to A^3.

[2] Left/right annotation matrices play a smaller role than for English because Arabic is a lot less configurational and has a richer morphology.

References

Y. Al-Raheb, A. Akrout, J. van Genabith, J. Dichy. 2006. *DCU 250 Arabic Dependency Bank: An LFG Gold Standard Resource for the Arabic Penn Treebank* The Challenge of Arabic for NLP/MT at the British Computer Society, UK, pp. 105-116.

M. Attia. 2008. *Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation*. Ph.D. Thesis. The University of Manchester, Manchester, UK.

A. Bies and M. Maamouri. 2003. *Penn Arabic Treebank Guidelines* URL: http://www.ircs.upenn.edu/arabic/Jan03release/guidelines-TB-1-28-03.pdf.

D. Bikel. 2004. *Intricacies of Collins' parsing model* Computational Linguistics, 30(4), 2004.

A. Cahill, M. McCarthy, J. van Genabith and A. Way. 2002. *Automatic Annotation of the Penn-Treebank with LFG F-Structure Information*. LREC 2002 workshop on Linguistic Knowledge Acquisition and Representation, pp. 8-15.

A. Cahill, M. Burke, R. ODonovan, J. van Genabith, A. Way. 2004. *Long-Distance Dependency Resolution in Automatically Acquired Wide-Coverage PCFG-Based LFG Approximations*. Meeting of the Association for Computational Linguistics ACL 2004, pp. 319-326.

M. Diab, K. Hacioglu, D. Jurafsky. 2004. *Automatic tagging of arabic text: From raw text to base phrase chunks*. North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL04).

E. Ditters. 2001. *A Formal Grammar for the Description of Sentence Structure in Modern Standard Arabic*. Workshop on Arabic Processing: Status and Prospects at ACL/EACL, Toulouse, France.

N. Habash, O. Rambow. 2005. *Arabic Tokenization, Morphological Analysis, and Part-of-Speech Tagging in One Fell Swoop*. In Proceedings of the Conference of American Association for Computational Linguistics ACL 2005.

S. Kulick S, G. Ryan, M. Mitchell. 2006. *Parsing the Arabic Treebank: Analysis and improvements*. In Proceedings of TLT 2006. Treebanks and Linguistic Theories..

M. Maamouri and A. Bies. 2004. *Developing an Arabic Treebank: Methods, Guidelines, Procedures, and Tools* Workshop on Computational Approaches to Arabic Script-based Languages, COLING 2004.

E. Othman, K. Shaalan, A. Rafea. 2003. *A Chart Parser for Analyzing Modern Standard Arabic Sentence* MT Summit IX Workshop on Machine Translation for Semitic Languages: Issues and Approaches, USA.

A. Ramsay and H. Mansour. 2007. *Towards including prosody in a text-to-speech system for modern standard Arabic* Computer Speech and Language 22:84-103.

L. Tounsi, M. Attia, J. van Genabith. 2009. *Automatic Treebank-Based Acquisition of Arabic LFG Dependency Structures* European Chapter of the Association for Computational Linguistics, EACL 2009, Workshop Computational Approaches to Semitic Languages, pp 45-52.

Z. Zabokrtsky, O. Smrz. 2003. *Arabic syntactic trees: from constituency to dependency* European Chapter of the Association for Computational Linguistics EACL 2003, pp. 183-186.