Parsing Arabic Using Treebank-Based LFG Resources

Lamia Tounsi Mohammed Attia Josef van Genabith
NCLT, School of Computing, Dublin City University, Ireland
{ltounsi, mattia, josef}@computing.dcu.ie

Introduction 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 [13] exploits the rich functional annotations in the Penn Arabic Treebank (ATB) [3], [10] 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 [1]. Currently we achieve a dependency f-score of 77%.

Related Work Arabic parsing systems have been reported in ([7], [14], [11], [12], [2]). 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 [4]. Bikel reports parse quality "far below" the required standard [9]. 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. [6] and [8] 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-structure). 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-structure), using the Arabic A^3 annotation algorithm [13], extending the approach of Cahill et al. [5], originally developed for English.

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 cliticised language. The ATB consists of 23,611 parse-annotated sentences [3], [10] from Arabic newswire text in Modern Standard Arabic (MSA). The ATB annotation scheme involves 24 basic POS-tags (497 different tags with morphological information), 22 phrasal tags, and 20 functional tags (and 52 combined functional tags, as functional tags can stack).

The Arabic Annotation Algorithm (A^3) The A^3 algorithm [13] is constructed adapting and revising the methodology of Cahill et al. [5] for English: (i) automatic extraction of the most frequent rule types from the treebank[1]. (ii) head lexicalisation of ATB trees to identify local heads. (iii) default f-structure equations are assigned to ATB functional tags. (iv) left/right annotation principles for COMPs, XCOMPs, ADJUNCTs, etc[2]. (v) Coordination and finally, (vi) Traces to handle non-local dependencies. Lexical macros exploit the rich morphological information provided by the ATB. Tounsi et al. [13] report an f-score of 95% on automatically annotated gold ATB trees against the DCU250 Arabic Dependency Bank.

Adapting the Parser We use Bikel's implementation of Collins' Model 1 as our c-structure engine [4]. As the A^3 of [13] 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 NP-OBJ $\Rightarrow$ NP_OBJ and adjust the head-finding rules in Bikel's Arabic language pack accordingly. After parsing, we unmask ATB function tags and make them available to A^3.

[1] With 85% token coverage.

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

Experiments and Evaluation 250 of the 23,611 parse-annotated sentences in ATB were randomly selected as test set [6]. The DCU 250 gold standard dependency bank for Arabic [1] 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.

Discussion and Further Work Compared to similar results for English, 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 our current experiments we 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.

References

[1] 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.

[2] 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.

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

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

[5] 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*. ACL 04, pp. 319-326.

[6] M. Diab, K. Hacioglu, D. Jurafsky. 2004. *Automatic tagging of arabic text: From raw text to base phrase chunks*. HLT-NAACL04.

[7] 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.

[8] N. Habash, O. Rambow. 2005. *Arabic Tokenization, Morphological Analysis, and Part-of-Speech Tagging in One Fell Swoop*. ACL'05.

[9] S. Kulick S, G. Ryan, M. Mitchell. 2006. *Parsing the Arabic Treebank: Analysis and improvements*. TLT 06.

[10] 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.

[11] 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.

[12] 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.

[13] L. Tounsi, M. Attia, J. van Genabith. 2009. *Automatic Treebank-Based Acquisition of Arabic LFG Dependency Structures* EACL 2009, Workshop Computational Approaches to Semitic Languages.

[14] Z. Zabokrtsky, O. Smrz. 2003. *Arabic syntactic trees: from constituency to dependency* EACL 2003, Hungary, pp. 183-186.