Automatic Treebank-Based Acquisition of Arabic LFG Dependency Structures

Lamia Tounsi **Mohammed Attia** **Josef van Genabith**

NCLT, School of Computing, Dublin City University, Ireland
{lamia.tounsi, mattia, josef}@computing.dcu.ie

Abstract

A number of papers have reported on methods for the automatic acquisition of large-scale, probabilistic LFG-based grammatical resources from treebanks for English (Cahill and al., 2002), (Cahill and al., 2004), German (Cahill and al., 2003), Chinese (Burke, 2004), (Guo and al., 2007), Spanish (O'Donovan, 2004), (Chrupala and van Genabith, 2006) and French (Schluter and van Genabith, 2008). Here, we extend the LFG grammar acquisition approach to Arabic and the Penn Arabic Treebank (ATB) (Maamouri and Bies, 2004), adapting and extending the methodology of (Cahill and al., 2004) originally developed for English. Arabic is challenging because of its morphological richness and syntactic complexity. Currently 98% of ATB trees (without FRAG and X) produce a covering and connected f-structure. We conduct a qualitative evaluation of our annotation against a gold standard and achieve an f-score of 95%.

1 Introduction

Treebank-based statistical parsers tend to achieve greater coverage and robustness compared to approaches using handcrafted grammars. However, they are criticised for being too shallow to mark important syntactic and semantic dependencies needed for meaning-sensitive applications (Kaplan, 2004). To treat this deficiency, a number of researchers have concentrated on enriching shallow parsers with deep dependency information. (Cahill and al., 2002), (Cahill and al., 2004) outlined an approach which exploits information encoded in the Penn-II Treebank (PTB) trees to automatically annotate each node in each tree with LFG f-structure equations representing deep predicate-argument structure relations. From this LFG annotated treebank, large-scale unification grammar resources were automatically extracted and used in parsing (Cahill and al., 2008) and generation (Cahill and van Genabith, 2006). This approach was subsequently extended to other languages including German (Cahill and al., 2003), Chinese (Burke, 2004), (Guo and al., 2007), Spanish (O'Donovan, 2004), (Chrupala and van Genabith, 2006) and French (Schluter and van Genabith, 2008).

Arabic is a semitic language and is well-known for its morphological richness and syntactic complexity. In this paper we describe the porting of the LFG annotation methodology to Arabic in order to induce LFG f-structures from the Penn Arabic Treebank (ATB) (Bies, 2003), (Maamouri and Bies, 2004). We evaluate both the coverage and quality of the automatic f-structure annotation of the ATB. Ultimately, our goal is to use the f-structure annotated ATB to derive wide-coverage resources for parsing and generating unrestricted Arabic text. In this paper we concentrate on the annotation algorithm.

The paper first provides a brief overview of Lexical Functional Grammar, and the Penn Arabic Treebank (ATB). The next section presents the architecture of the f-structure annotation algorithm for the acquisition of f-structures from the Arabic treebank. The last section provides an evaluation of the quality and coverage of the annotation algorithm.

1.1 Lexical Functional Grammar

Lexical-Functional Grammar (LFG) (Kaplan and Bresnan, 1982); (Bresnan, 2001), (Falk, 2001) 2001, (Sells, 1985) is a constraint-based theory of grammar. LFG rejects concepts of configurationality and movement familiar from generative grammar, and provides a non-derivational alternative of parallel structures of which phrase structure trees are only one component.

LFG involves two basic, parallel forms of

Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages, pages 45–52,
Athens, Greece, 31 March, 2009.

knowledge representation: c(onstituent)-structure, which is represented by (f-structure annotated) phrase structure trees; and f(unctional)-structure, represented by a matrix of attribute-value pairs. While c-structure accounts for language-specific lexical idiosyncrasies, syntactic surface configurations and word order variations, f-structure provides a more abstract level of representation (grammatical functions/ labeled dependencies), abstracting from some cross-linguistic syntactic differences. Languages may differ typologically as regards surface structural representations, but may still encode similar syntactic functions (such as, subject, object, adjunct, etc.). For a recent overview on LFG-based analyses of Arabic see (Attia, 2008) who presents a hand-crafted Arabic LFG parser using the XLE (Xerox Linguistics Environment).

1.2 The Penn Arabic Treebank (ATB)

The Penn Arabic Treebank project started in 2001 with the aim of describing written Modern Standard Arabic newswire. The Treebank consists of 23611 sentences (Bies, 2003), (Maamouri and Bies, 2004) .
Arabic is a subject pro-drop language: a null category (pro) is allowed in the subject position of a finite clause if the agreement features on the verb are rich enough to enable content to be recovered (Baptista, 1995), (Chomsky, 1981). This is represented in the ATB annotation by an empty node after the verb marked with a -SBJ functional tag. The ATB annotation, following the Penn-II Treebank, utilises the concept of empty nodes and traces to mark long distance dependencies, as in relative clauses and questions. The default word order in Arabic is VSO. When the subject precedes the verb (SVO), the construction is considered as topicalized. Modern Standard Arabic also allows VOS word order under certain conditions, e.g. when the object is a pronoun. The ATB annotation scheme involves 24 basic POS-tags (497 different tags with morphological information), 22 phrasal tags, and 20 individual functional tags (52 different combined tags).
The relatively free word order of Arabic means that phrase structural position is not an indicator of grammatical function, a feature of English which was heavily exploited in the automatic LFG annotation of the Penn-II Treebank (Cahill and al., 2002). Instead, in the ATB functional tags are used to mark the subject as well as the object.
The syntactic annotation style of the ATB follows, as much as possible, the methodologies and bracketing guidelines already used for the English Penn-II Treebank. For example, in the Penn English Treebank (PTB) (Marcus, 1994), small clauses are considered sentences composed of a subject and a predicate, without traces for an omitted verb or any sort of control relationship, as in example (1) for the sentence "I consider Kris a fool".

(1) (S (NP-SBJ I)
(VP consider
(S (NP-SBJ Kris)
(NP-PRD a fool))))

The team working on the ATB found this approach very convenient for copula constructions in Arabic, which are mainly verbless (Maamouri and Bies, 2004). Therefore they used a similar analysis without assuming a deleted copula verb or control relationship, as in (2).

(2) (S (NP-SBJ `Al-mas>alatu` المسألة)
(ADJ-PRD `basiyTatuN` بسيطة))

المسألة بسيطة
`Al-mas>alatu basiyTatuN`
the-question simple
'The question is simple.'

2 Architecture of the Arabic Automatic Annotation Algorithm

The annotation algorithm for Arabic is based on and substantially revises the methodology used for English.
For English, f-structure annotation is very much driven by configurational information: e.g. the leftmost NP sister of a VP is likely to be a direct object and hence annotated $\uparrow$ OBJ $=\downarrow$. This information is captured in the format of left-right annotation matrices, which specify annotations for left or right sisters relative to a local head.
By contrast, Arabic is a lot less configurational and has much richer morphology. In addition, compared to the Penn-II treebank, the ATB features a larger functional tag set. This is reflected in the design of the Arabic f-structure annotation algorithm

(Figure 1), where left-right annotation matrices play a much smaller role than for English. The annotation algorithm recursively traverses trees in the ATB. It exploits ATB morpho-syntactic features, ATB functional tags, and (some) configurational information in the local subtrees.

We first mask (conflate) some of the complex morphological information available in the preterminal nodes to be able to state generalisations for some of the annotation components. We then head-lexicalise ATB trees identifying local heads. Lexical macros exploit the full morphological annotations available in the ATB and map them to corresponding f-structure equations. We then exploit ATB functional tags mapping them to SUBJ, OBJ, OBL, OBJ2, TOPIC and ADJUNCT etc. grammatical functions. The remaining functions (COMP, XCOMP, SPEC etc.) as well as some cases of SUBJ, OBJ, OBL, OBJ2, TOPIC and ADJUNCT, which could not be identified by ATB tags, are treated in terms of left-right context annotation matrices. Coordination is treated in a separate component to keep the other components simple. Catch-all & Clean-Up corrects overgeneralisations in the previous modules and uses defaults for remaining unannotated nodes. Finally, non-local dependencies are handled by a Traces component.

The next sub-sections describe the main modules of the annotation algorithm.

2.1 Conflation

ATB preterminals are very fine-grained, encoding extensive morpho-syntactic details in addition to POS information. For example, the word سنقف `sanaqifu` '[we will] stand' is tagged as (FUT+IV1P+IV+IVSUFF_MOOD:I) denoting an imperfective (I) verb (V) in the future tense (FUT), and is first person (1) plural (P) with indicative mood (IVSUFF_MOOD:I). In total there are over 460 preterminal types in the treebank. This level of fine-grainedness is an important issue for the annotation as we cannot state grammatical function (dependency) generalizations about heads and left and right contexts for such a large tag set. To deal with this problem, for some of the annotation algorithm components we masked the morpho-syntactic details in preterminals, thereby conflating them into more generic POS tags. For example, the above-mentioned tag will be conflated as VERB.

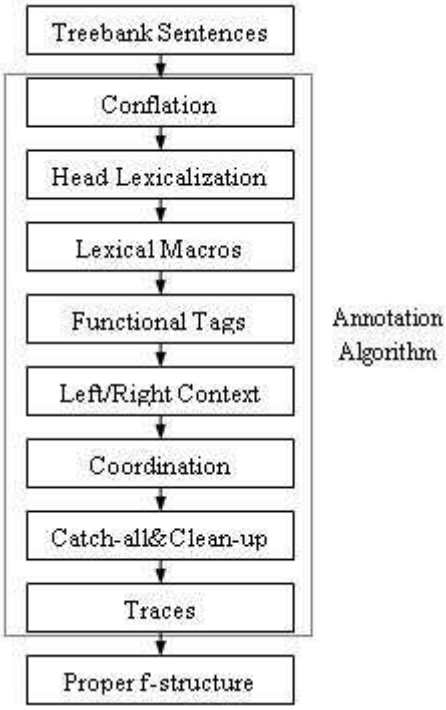

Figure 1: Architecture of the Arabic annotation algorithm

2.2 Lexical Macros

Lexical macros, by contrast, utilise the detailed morpho-syntactic information encoded in the preterminal nodes of the Penn Arabic Treebank trees and provide the required functional annotations accordingly. These tags usually include information related to person, number, gender, definiteness, case, tense, aspect, mood, etc.

Table 1 lists common tags for nouns and verbs and shows the LFG functional annotation assigned to each tag.

2.3 Functional Tags

In addition to monadic POS categories, the ATB treebank contains a set of labels (called functional tags or functional labels) associated with functional information, such as -SBJ for 'subject' and -OBJ for 'object'. The functional tags module translates these functional labels into LFG functional equations, e.g. -OBJ is assigned the annotation ↑OBJ=↓. An f-structure equation look-up table assigns default f-structure equations to each functional label in the ATB (Table 2).

A particular treatment is applied for the tag -PRD (predicate). This functional tag is used with copula complements, as in (3) and the corresponding c-structure in Figure 2. Copula complements

Tag	Annotation
Nouns	
MASC	↑ GEND = masc (masculine)
FEM	↑ GEND = fem (feminine)
SG	↑ NUM = sg (singular)
DU	↑ NUM = dual
PL	↑ NUM = pl (plural)
ACC	↑ CASE = acc (accusative)
NOM	↑ CASE = nom (nominative)
GEN	↑ CASE = gen (genitive)
Verbs	
1	↑ PERS = 1
2	↑ PERS = 2
3	↑ PERS = 3
S	↑ NUM = sg
D	↑ NUM = dual
P	↑ NUM = pl
F	↑ GEND = masc
M	↑ GEND = fem

Table 1: Morpho-syntactic tags and their functional annotations

Functional Label	Annotation
-SBJ (subject)	↑ SUBJ = ↓
-OBJ (object)	↑ OBJ = ↓
-DTV (dative), -BNF (Benefactive)	↑ OBJ2 =↓
-TPC (topicalized)	↑ TOPIC=↓
-CLR (clearly related)	↑ OBL =↓
-LOC (locative), -MNR (manner), -DIR (direction), -TMP (temporal), -ADV (adverbial) -PRP (purpose),	↓∈↑ ADJUNCT

Table 2: Functional tags used in the ATP Treebank and their default annotations

correspond to the open complement grammatical function XCOMP in LFG and the ATB tag -PRD is associated with the annotation in (4) in order to produce the f-structure in Figure 3. The resulting analysis includes a main predicator 'null_be' and specifies the control relationship through a functional equation stating that the main subject is co-indexed with the subject of the XCOMP.

(3) الهدنة ضروريّة
Al-hudonapu Daruwriy˜apN
the-truce necessary
'The truce is necessary.'

(4) ↑ PRED = 'null_be'
↑ XCOMP = ↓
↑ SUBJ= ↓ SUBJ

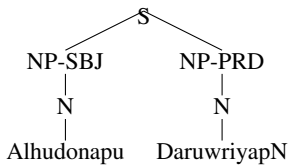

Figure 2: C-structure for example (3)

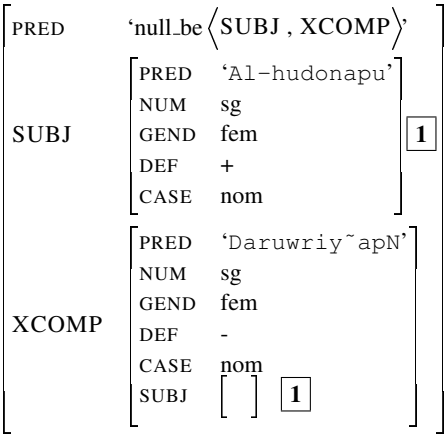

Figure 3: F-structure for example (3)

2.4 Left-Right Context Rules

The left-right context annotation module is based on a tripartite division of local subtrees into a left-hand-side context (LHS) followed by a head (H) followed by a right-hand-side context (RHS). We developed our own head finding, or head lexicalization, rules based on a variety of heuristics and manual inspection of the PS rules.

Initially, we extracted 45785 Phrase Structure (PS) rules from the treebank. The reason for the relatively large number of PS rules is the fine-grained nature of the tags encoding morphological information for pre-terminal nodes. When we conflate pre-terminals containing morphological information to basic POS tags, the set of PS rules is reduced to 9731.
Treebanks grammars follow the Zipfian law: for each category, there is a small number of highly frequent rules expanding that category, followed by a large number of rules with a very low frequency. Therefore, for each LHS category we select the most frequent rules which together give 85% coverage. This results is a reduced set of 339 (most frequent) PS rules. These rules are manually examined and used to construct left-right LFG f-structure annotation matrices for the treebank. The annotation matrices encode information about

the left and right context of a rule's head and state generalisations about the functional annotation of constituents to the left and right of the local head.

Consider sentence (5), where an NP is expanded as NP NP ADJP. The first NP is considered the head and is given the annotation ↑=↓. The second NP and the ADJP are located to the left (Arabic reading) of the head (LHS). The left-right context matrix for NP constituents analyses these phrases as adjuncts and assigns them the annotation ↓ ∈ ↑ ADJUNCT.

(5) جمعيّة الطيّارين الأنجوليّة
jamoEiy˜apu Al-Tay˜Ariyna Al->anoguwliy˜apu
society the-pilot the-Angolian
'Angolian Pilot Society'

The left-right annotation matrices also cover other non-subcategorisable functions (such as XADJUNCT, SPEC, etc.) as well as constituents with subcategorisable grammatical functions (SUBJ, OBJ, OBL, COMP, etc.) which are not identified via ATB functional tags (and hence left unannotated by the Functional Tags component)

2.5 Coordination

Treebanks tend to encode co-ordination in a rather flat manner. In the LFG framework coordinated constituents are treated as sets. The phrase structure functional annotations for creating a set function for such constituents is given in (6) where the f-structures of the two coordinated NPs on the right-hand side are members of the set valued f-structure of the NP on the left-hand side.

(6) NP → NP CONJ NP
↑∈↓ ↑∈↓

To keep the other modules simple and perspicuous coordination is treated in the annotation algorithm as a separate component. The coordination module localizes the coordinating conjunct, marks it as head and adds the coordinated elements to the f-structure set representation of the coordination ↓∈↑ COORD. Figure 2.5 shows the f-structure for the NP in sentence (7).
(7) الكرات والتسديدات
Al-kurAtu wa-Al-tasodiydAtu
the-balls and-the-scores

$$\begin{bmatrix} \text{COORD\_FORM} & \text{'wa-'} \\ \text{COORD} & \left\{ \begin{bmatrix} \text{PRED} & \text{'Al-kurAtu'} \\ \text{NUM} & \text{pl} \\ \text{GEND} & \text{fem} \\ \text{CASE} & \text{nom} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \text{'Al-tasodiydAtu'} \\ \text{NUM} & \text{pl} \\ \text{GEND} & \text{fem} \\ \text{CASE} & \text{nom} \end{bmatrix} \right\} \end{bmatrix}$$

Figure 4: An Arabic coordination example

2.6 Catch-All and Clean-Up

The previous components of the annotation algorithm give concise statements of linguistic generalisations, but sometimes they overgeneralise. Such overgeneralisations are detected and corrected by the Catch-All and Clean-Up component of the algorithm.
For example, the mutiword expression إلّا أنّ 'illaA 'anna 'but' is annotated in the treebank as two subsequent subordinating conjunctions: (SUB_CONJ 'illaA) (SUB_CONJ 'anna). In the f-structure annotation this leads to a conflict as to which lexical item should occupy the value of the SUBORD_FORM feature. The Catch-All and Clean-Up component sidelines the problem by moving the second part of the MWE to an adjunct position.

Another example is provided by quantifiers. In Arabic, quantifiers have the same syntactic structure as the construct state (similar to the genitive construction in English as in *the boys' book*), so that sentences (8) and (9) are syntactically equivalent. The word 'students' is in the second part of the construct state in both phrases, but it is a modifier in the first and a head in the second. Therefore, a list of quantifiers (Table 3) is used in the Catch-All and Clean-Up module, so that they are identified and properly annotated according to certain context conditions.

The Catch-All and Clean-Up module also provides default annotations for nodes that remain unannotated by the previous components.
(8) كتب الطلّاب
kutubu Al-Tul˜abi
books the-students
'students' books'

(9) بعض الطلّاب
baEoDu Al-Tul˜abi
some the-students
‘some students’

biDoEapu	بضعة	some
kAf˜apu	كافّة	all
>ay˜u	أيّ	any
jamiyEu	جميع	all
muEoZamu	معظم	most
biDoEu	بضع	few
kul˜u	كلّ	all
baEoDu	بعد	some
baqiy˜apu	بقيّة	rest
nafosu	نفس	same
>aHadu	أحد	one-masc
<iHodaY	إحدى	one-fem

Table 3: List of Arabic quantifiers

2.7 Traces

The f-structure generated prior to the Traces module is called a proto-f-structure (i.e. a partial representation), as it is not complete with respect to long-distance dependency resolution. In order to produce proper f-structures, long-distance dependencies such as topicalisation and wh-movement must be captured. In our annotation algorithm we exploit trace information in the ATB treebank and translate long-distance dependencies into cooresponding reentrancies at the f-structure level using coindexation.

Figure 5 gives the ATB tree for the phrase in (10) containing a trace. The trace is used to capture A-movement, and the indices on the WHNP-2 and NP-SBJ-2 indicate that these constituents are related.

In the annotation algorithm we assign the equation ↑SUBJ = ↑TOPICREL to the empty node to indicate that the relative pronoun ‘which’ is interpreted as the subject of the verb ‘threaten’. This annotation produces the proper f-structure in Figure 6.

(10) العنف الّذي يهدّد السلام
Al-Eunofu Al˜a⋆iy yuhad˜idu Al-salAma
violence which threatens peace

Once every node in a tree is annotated with f-structure equations, the equations are then passed to a constraint solver. Ideally one f-structure representation is produced for each sentence. If there are conflicts in the f-structure equations, no f-structure is produced.

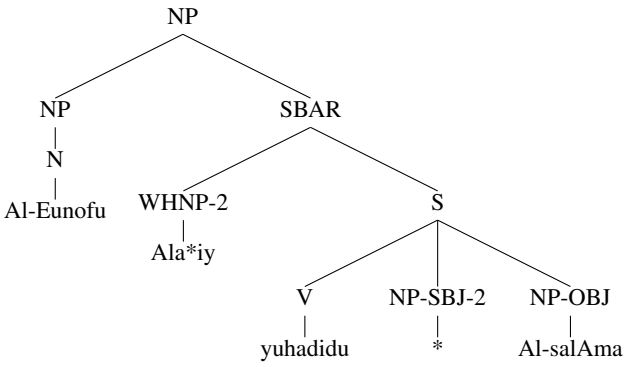

Figure 5: C-structure with a long-distance dependency

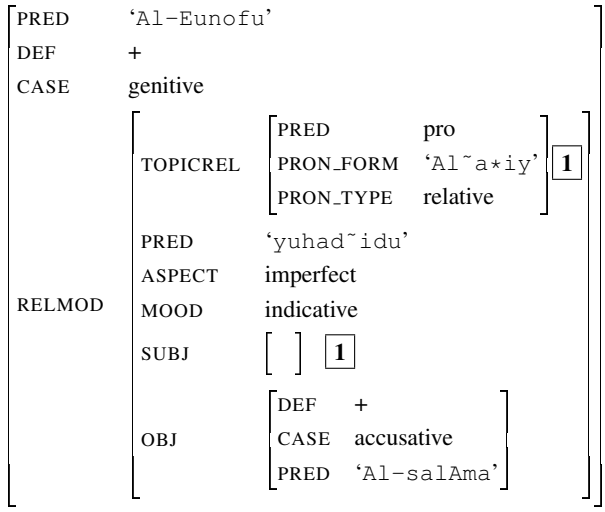

Figure 6: Proper f-structure with long-distance dependencies captured

3 Evaluation

We conduct two types of evaluation: quantitative and qualitative evaluation.

The quantitative evaluation evaluates the coverage of our annotation algorithm, while the qualitative evaluation compares the f-structures generated by the automatic annotation procedure against a gold standard of manually constructed f-structures for 250 sentences (Al-Raheb and al., 2006) selected at random from the ATB treebank. The aim of the qualitative evaluation is to ensure that the annotation quality is of a high standard, particularly as the annotation algorithm is used for extracting

wide-coverage syntactic and lexical resources.

In the quantitative evaluation experiment, the annotation algorithm achieves good coverage for 19 273 ATB sentences (remaining after removing trees with FRAG and X - labeled constituents); 98% of trees produce a complete and connected f-structure (no fragments) and 2% of trees do not produce an f-structure because of feature-value clashes.

For the qualitative evaluation, we use the evaluation methodology of (Crouch and al., 2002) and (Riezler, 2002) in order to calculate precision and recall on descriptions of f-structures. In this methodology, each f-structure is represented as a set of triples of the form: relation(argument$_1$,argument$_2$). For example the triples num(riHol+At+i, pl), case(riHol+At+i, genitive), gender(riHol+At+i, fem) encode that the number of the word riHol+At+i رحلات 'journeys' is plural; its case is genitive; and its gender is feminine. The triple subj(ta+bolug+u: *to reach*,HarAr+ap+a: *temperature*) indicates that the subject of the verb to reach is temperature. The results of the evaluation of the quality of the annotation against the DCU 250 gold standard are presented in Table 4. We achieve an f-score of 95%. In comparison, the f-scores for French, English and Chinese languages are 95%-96%. Table 5 presents the results by selected grammatical functions.

	Precision	**Recall**	**F-score**
Results	95.49	94.43	94.96

Table 4: Evaluation of the automatically produced f-structures against gold standard (all features).

	Precision	**Recall**	**F-score**
adjunct	91	91	91
coord	80	87	83
obj	81	88	85
obl	100	94	97
poss	96	89	92
subj	89	68	77
topic	93	92	92
topicrel	89	88	88

Table 5: Evaluation of the automatically produced f-structures against gold standard by selected grammatical functions.

4 Conclusion

In this paper, we have shown how the methodology for automatically annotating treebanks with LFG f-structure equations originally developed for English has been successfully adapted to Arabic. Arabic is known for its rich morphology and syntactic flexibility which allows SVO, VSO, VOS word orders. We exploit the rich morphological information in the annotation algorithm by utilising the morphological tags to add information to the f-structures. We also use ATB functional tags to specify default syntactic functions, e.g. -SBJ (subject) and -OBJ (object), provide left-right annotation matrices for the remaining constituents, treat coordination and represent non-local dependencies. The evaluation measured coverage as well as the quality of the automatic annotation algorithm. 98% of ATB trees (without FRAG and X) produce a complete and connected f-structure. When evaluated against a gold standard of 250 manually constructed f-structures, the algorithm scores an f-measure of 95%. The work presented in this paper is the first step in automatically acquiring deep resources for wide coverage parsing and generation for Arabic.

Acknowledgments

This research was supported by Science Foundation Ireland Grant 04/IN/I527.

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.

M. Baptista. 1995. *On the Nature of Pro-drop in Capeverdean Creole.* Harvard Working Papers in Linguistics, 5:3-17.

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

J. Bresnan. 2001. *Lexical-Functional Syntax*. Blackwell Publishers, Oxford, UK.

M. Burke, O. Lam, R. Chan, A. Cahill, R. ODonovan, A. Bodomo, J. van Genabith, and A. Way. 2004. *Treebank-Based Acquisition of a Chinese Lexical-Functional Grammar*. The 18th Pacific Asia Conference on Language, Information and Computation, Tokyo, Japan, pp. 161–172.

M. Burke. 2006. *Automatic Treebank Annotation for the Acquisition of LFG Resources*. Ph.D. thesis, School of Computing, Dublin City University, Ireland.

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

A. Cahill, M. Forst, M. McCarthy, R. ODonovan, C. Rohrer, J. van Genabith and A. Way. 2003. *Treebank-Based Multilingual Unification Grammar Development*. The 15th Workshop on Ideas and Strategies for Multilingual Grammar Development, at the 15th European Summer School in Logic, Language and Information, Vienna, Austria, pp. 17–24.

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*. 42nd Meeting of the Association for Computational Linguistics, Barcelona, Spain pp. 319-326.

A. Cahill, J. van Genabith. 2006. *Robust PCFG-Based Generation using Automatically Acquired LFG Approximations*. ACL 2006, Sydney, Australia, pages 1033-1040.

A. Cahill, M. Burke, R. O'Donovan, S. Riezler, J. van Genabith, A. Way. 2008. *Wide-Coverage Deep Statistical Parsing using Automatic Dependency Structure Annotation*. Computational Linguistics, Vol. 34, No. 1, pages 81-124.

N. Chomsky. 1981. *Lectures on Government and Binding.* Dordrecht: Foris.

G. Chrupala and J. van Genabith. 2006. *Improving Treebank-Based Automatic LFG Induction for Spanish.* In Proceedings of the LFG06 Conference.

R. Crouch, R. M. Kaplan, T. H. King, S. Riezler. 2002. *Comparison of Evaluation Metrics for a Broad Coverage Parser* LREC Workshop: Beyond PARSEVAL Towards Improved Evaluation Measures for Parsing Systems, Spain, pp. 67-74.

M. Dalrymple. 2002. *Lexical Functional Grammar*. Syntax and Semantics, Volume 34, Academic Press, San Diego, CA/London, U.K.

Y. Falk. 2001. *Lexical-Functional Grammar: An Introduction to Parallel Constraint-Based Syntax.* Stanford, Calif.: CSLI Publications.

A. Frank, L. Sadler, J. van Genabith, A. Way. 2003. *From Treebank Resources to LFG F-Structurese*. A. Abeille editor Treebanks: Building and Using Syntactically Annotated Corpora, Kluwer Academic Publishers, Dordrecht/Boston/London,The Netherlandsm pp. 367-389.

Y. Guo, J. van Genabith, H. Wang. 2007. *Acquisition of Wide-Coverage, Robust, Probabilistic Lexical-Functional Grammar Resources for Chinese*. Proceedings of the 12th International Lexical Functional Grammar Conference, USA, pp. 214–232.

R. Kaplan and J. Bresnan. 1982. *Lexical Functional Grammar: a Formal System for Grammatical Representation, in J. Bresnan (ed.).* The Mental Representation of Grammatical Relations, MIT Press, Cambridge, MA, pp. 173-281.

R. M. Kaplan, S. Riezler, T. H. King, J. T. Maxwell, A. Vasserman, and R. Crouch. 2004. *Speed and Accuracy in Shallow and Deep Stochastic Parsing.* In The Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004), Boston, MA, pp. 97–104.

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, Geneva, 2004.

M. Marcus, G. Kim, M. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz and B. Schasberger 1994. *The Penn Treebank: Annotating Predicate Argument Structure.* In Proceedings of the Human Language Technology Workshop. San Francisco, CA.

R. O'Donovan, M. Burke, A. Cahill, J. van Genabith, and A. Way. 2004. *Large-Scale Induction and Evaluation of Lexical Resources from the Penn-II Treebank.* The 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, pp. 368–375.

R. O'Donovan, A. Cahill, J. van Genabith, and A. Way. 2005. *Automatic Acquisition of Spanish LFG Resources from the CAST3LB Treebank.* The Tenth International Conference on LFG, Bergen, Norway, pp. 334–352.

S. Riezler, King, T., Kaplan, R., Crouch, R., Maxwell, J. T., and Johnson, M. 2002. *Parsing theWall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques.* The 40th Annual Conference of the Association for Computational Linguistics (ACL-02), Philadelphia, PA, pp. 271–278.

P. Sells 1985. *Lectures on Contemporary Syntactic Theories.* CSLI Lecture Notes. Stanford, CA: CSLI.

N. Schluter and J. van Genabith 2008. *Treebank-Based Acquisition of LFG Parsing Resources for French.* Proceedings of the Sixth International Language Resources and Evaluation (LREC'08).