Automatic Extraction and Evaluation of Arabic LFG Resources

Mohammed Attia†‡, Khaled Shaalan‡, Lamia Tounsi†, and Josef van Genabith†
†School of Computing, Dublin City University, Ireland
‡The British University in Dubai, UAE
E-mail: {mattia,ltounsi,josef}@computing.dcu.ie, khaled.shaalan@buid.ac.ae

Abstract

This paper presents the results of an approach to automatically acquire large-scale, probabilistic Lexical-Functional Grammar (LFG) resources for Arabic from the Penn Arabic Treebank (ATB). Our starting point is the earlier, work of (Tounsi et al., 2009) on automatic LFG f(eature)-structure annotation for Arabic using the ATB. They exploit tree configuration, POS categories, functional tags, local heads and trace information to annotate nodes with LFG feature-structure equations. We utilize this annotation to automatically acquire grammatical function (dependency) based subcategorization frames and paths linking long-distance dependencies (LDDs). Many state-of-the-art treebank-based probabilistic parsing approaches are scalable and robust but often also shallow: they do not capture LDDs and represent only local information. Subcategorization frames and LDD paths can be used to recover LDDs from such parser output to capture deep linguistic information. Automatic acquisition of language resources from existing treebanks saves time and effort involved in creating such resources by hand. Moreover, data-driven automatic acquisition naturally associates probabilistic information with subcategorization frames and LDD paths. Finally, based on the statistical distribution of LDD path types, we propose empirical bounds on traditional regular expression based functional uncertainty equations used to handle LDDs in LFG.

Keywords: Arabic subcategorization frames, Arabic long-distance dependencies, Arabic LFG annotation

1. Introduction

The automatic extraction of LFG language resources from treebanks has been described for many languages including English (Cahill et al., 2004), German (Rehbein and van Genabith, 2009), French (Schluter and van Genabith, 2008) and Chinese (Guo et al, 2007). Here we present our research on extracting similar LFG language resources for Arabic from the Penn Arabic Treebank (ATB) (Maamouri and Bies, 2004), which contains 22,524 sentences, 787,235 tokens, and 587,665 words. These language resources consist mainly of two distinct and complementary parts: subcategorization frames and long-distance dependency (LDD) paths. Subcategorization frames describe the argument structure requirement of predicates, or semantic forms, while LDD paths describe the grammatical functions that exist in the path between two co-indexed syntactic elements. These two language resources can be used to augment the output of a probabilistic treebank-based parser with deeper syntactic information including unbounded dependencies (Cahill et al. 2004, 2008), not captured by many current statistical parsing approaches (Bikel, 2004; Petrov et al., 2006).

Although this method has been implemented for a number of languages, Arabic (with its rich morphology and relatively free word order) presents particular challenges addressed in this paper. For instance, the extraction of subcategorization frames requires handling the intricate issue of lemmatizing Arabic surface forms which is particularly challenging. Regarding the extraction of LDD paths we discuss particularly interesting grammatical phenomena in Arabic such as resumptive pronouns which mark the lower end in an LDD relationship, fronted subjects, and estimating the maximum length of the path. Moreover, relying on the probability distributions over LDD paths, we are able to propose empirical upper bounds on the lengths of paths.

Our annotation, subcategorization frames and LDD resources are based on the formalism of Lexical Functional Grammar (LFG) (Dalrymple, 2001). LFG is a constraint-based non-derivational syntactic framework which essentially distinguishes between two distinct but related levels of representation: c(onstituent)-structure and f(functional)-structure. C-structure takes the form of phrase structure trees, while F-structure is represented in terms of attribute-value structures (or matrices AVMs). F-structure is not directly derived from the c-structure, but the two levels are related through f-equations annotated to CFG tree nodes (Austin, 2001).

2. Arabic Subcategorization Frames

The subcategorization requirements of lexical entries are an important type of lexical information, as they indicate the argument(s) a predicate needs in order to form a well-formed syntactic structure. Producing such resources by hand is costly and time consuming. In the current research we create a lexicon of subcategorization frames through automatic induction from the ATB, which we call ArabicSubcats.

To our knowledge, the only resource that currently exists for Arabic subcategorization frames is the lexicon manually developed for the Arabic LFG Parser (Attia, 2008). It is published as an open-source resource under the GPLv3 licence[1]. It contains 64 frame types, 2,709 lemmas types, and 2,901 lemma-frame types, averaging 1.07 frames per lemma. The resource incorporates control information and details of specific prepositions with obliques. From the f-structure annotated treebank we extract 240 frame types for 3,295 lemmas types, with 7,746 lemma-frame types (for verbs, nouns and adjectives), averaging 2.35 frames per lemma. We use the handcrafted resources (Attia, 2008) in the evaluation of the treebank based ArabicSubcats.

2.1 LFG subcategorization frames

LFG syntactic theory (Dalrymple, 2001) distinguishes between governable (subcategorizable) and non-governable (non-subcategorizable) grammatical functions (GFs). The governable GFs are the arguments required by predicates in order to produce a well-formed syntactic structure, and they include SUBJ(ect), OBJ(ect), OBJ_{Θ}, OBL(ique) $_{\Theta}$, COMP(lement) and XCOMP. Non-governable GFs are optional, and they include ADJ(junct) and XADJ. The subcategorization requirements in LFG are expressed in this format (O'Donovan et al., 2005):

$$\pi\langle gf_1, gf_2, \ldots gf_n\rangle$$

where π is the lemma (predicate or semantic form) and *gf* is a governable grammatical function. The value of the argument list of the semantic form ensures the well-formedness of the sentence. For example, in the sentence {iEotamada Al-Tifolu EalaY wAlidati-hi "The child relied on his mother", the verb {iEotamada "to rely" has the following argument structure {iEotamada<(↑SUBJ)(↑$\text{OBL}_{\text{>alaY}}$)>. By including a subject and an oblique with the preposition >alaY, we ensure that the verb's subcategorization requirements are met and that the sentence is well-formed, or syntactically valid.

2.2 Extracting subcategorization frames

In developing ArabicSubcats, we follow the successful model of LFG-based language resource extraction of O'Donovan et al. (2005) taking into consideration the specifics of the Arabic language and the resources available for evaluation. We automatically extract Arabic subcateogorization frames by utilizing the automatic Lexical-Functional Grammar (LFG) f-structure annotation algorithm for the ATB developed in (Tounsi et al., 2009). The syntactic annotations in the ATB provide information on deep representations in the phrase structure trees, such as traces and co-indexation for a number of missing argument types, which help the automatic extraction of subcategorization frames to be complete. After we extract surface forms we lemmatize all forms by re-analysing the words using the Buckwalter morphology and then choosing the analysis where the word diacrization and the POS gold tag set in the ATB match those in the Buckwalter analysis (Buckwalter, 2004).

We provide information on prepositions for obliques, distinguish between active and passive frames, and provide information on the probability score for each frame based on the frequency count for each lemma-frame pair attested in the data. We extract 240 frame types for 3,295 lemmas types, with 7,746 lemma-frame types (for verbs, nouns and adjectives), averaging 2.35 frames per lemma. We compare and evaluate the complete set of subcateorization frames extracted against the manually developed subcategorization frames in the Arabic LFG Parser (Attia, 2008).

Lemma with argument list	Conditional Probability
$Ahad_1([subj,obj,comp-s])	0.025
$Ahad_1([subj,obj,comp-sbar])	0.050
$Ahad_1([subj,passive])	0.100
$Ahad_1([subj,obj])	0.800
$Ahad_1([subj])	0.025

Table 1: Subcategorization frames with probabilities.

2.3 Estimating subcategorization probabilities

In order to estimate the likelihood of the occurrence of a certain argument list with a predicate (or lemma), we compute the conditional probability of subcategorization frames based on the number of token occurrences in the ATB, according to the following formula (O'Donovan et al., 2005);

$$P(ArgList \mid \Pi) = \frac{count(\Pi\langle ArgList\rangle)}{\sum_{i=1}^{n} count(\Pi\langle ArgList_i\rangle)}$$

where ArgList_1 … ArgList_n are all the possible argument lists that co-occur with Π. Because of the variations in verbal subcategorization, probabilities are useful for discriminating prominent frames from less frequent ones. An example is shown in Table 1

[1] http://arasubcats-lfg.sourceforge.net

for the verb $Ahada “watch” which has a total of 40 occurrences in the ATB.

2.4 Evaluating the Subcategorization Frames extraction output

We compare our subcategorization frames, ArabicSubcats, against a manually created subcategorization frame lexicon used in a rule-based LFG Parser (Attia, 2008). The Arabic LFG Parser has detailed subcategorisation information for lexical entries that includes the preposition of obliques, control relationships (or XCOMPs), and the type of complementizer in verbs that have complements. For nouns and adjectives, the number of lemma-frame type pairs collected in ArabicSubcats is comparable to the manually constructed frames in the Arabic LFG parser, but it is almost four times larger for verbs, as shown in Table 2.

	Verbs	Nouns	Adjectives
lemma-subcat pair types in ArabicSubcats	6596	855	295
lemma-subcat pair types in the LFG Parser	1621	991	289
Common lemmas	1447	268	70

Table 2: Number of lemma-frame types in ArabicSubcats and the Arabic LFG Parser

Fair and equitable evaluation of data-driven against hand-crafted resources is difficult: while hand-crafted resources tend to be high quality their coverage is often limited; data-driven resources may include some noise but often display coverage well beyond that achieved by hand crafted resources. This situation is manifest in the evaluation below. For “common lemmas” in Table 2, or lemmas that are common between ArabicSubcats and the hand-crafted resources, we compare the subcategorization frames in terms of recall, defined here as:

$$recall = \frac{tp}{tp + fn}$$

where *tp* is the number of true positives (where the semantic forms are the same) and *fn* is the number of false negatives (where the frames appeared in the hand-crafted lexicon but not in the automatically generated on). Table 3 shows results of matching on all GFs and on selected GFs. We conduct the evaluation experiment at four levels: (1) we match the full argument list between the two data sets, (2) we remove the value of the preposition in obliques, (3) we also remove COMPs and XCOMPs, and (4) we only leave SUBJs, OBJs and OBJ2s, denoting transitivity, or the most important type of argument.

		Recall		
		Verbs	Nouns	Adjectives
1	Full argument list	0.78	0.51	0.58
2	Without preps	0.82	0.55	0.62
3	Without preps, comps and xcomps	0.84	0.55	0.63
4	Without obls, comps and xcomps	0.96	0.77	0.86

Table 3: Evaluating the ArabicSubcats against the resource in the Arabic LFG Parser.

We notice that the recall is high for verbs which constitute the largest portion of the data and the most important type of predicates when dealing with subcategorization frames. Yet, we also notice that some of the scores in Table 3 are low, particularly for nouns and adjectives. In a sense, this is not too surprising, as, compared to the level of annotation detail for verbal projections (VP, S, etc.), NP constituents in Penn-style treebanks are rather flat (and to date patches with more detailed NP annotations are available only for the Penn-II treebank of English (Vadas and Curran, 2007)). We conduct a manual error analysis for 20 mismatches with nouns and 20 mismatches with adjectives in order to obtain some insight into where the problem is coming from. We found that in 80% of the cases, nouns with the targeted subcategorization frame are not encoded in the ATB as verbal nouns (gerunds), but as common nouns. There are also instances of errors in the hand-crafted lexicon and in the ATB tagging. In the case of adjectives, 50% of the non-matching frames in the hand-crafted lexicon do not have examples in the ATB, 25% have tagging errors in the ATB, 20% are treated as ordinary adjectives, not verbal adjectives, and there was also an instance of error in the hand-crafted lexicon.

3. Extraction of Long-Distance Dependencies

LFG distinguishes between Grammatical Functions (GFs) such as SUBJ, OBJ, OBL, etc., and Discourse Functions (DFs), such as TOPIC, TOPIC-REL and FOCUS. LDDs always involve a DF. A DF represents an extraposed element, or the upper end of an LDD relation, and it is linked to a gap in the domain of extraction. Dalrymple (2001) defined LDDs as “constructions in which a displaced constituent bears a syntactic function usually associated with some other position in the sentence.” The relation between the two positions must be controlled according to the Extended Coherence Condition which states that a DF must be linked to a GF either by functionally or by anaphorically binding an argument.

These LDD constructions are also called “unbounded dependencies” because the distance between the initial position, the filler, and the grammatical function from which it has been extracted, the gap, can be potentially unlimited (Austin, 2001). LDDs are accounted for in the LFG literature in terms of “functional uncertainty” (Kaplan and Zaenen, 1989), where a functional equation of the form $(\uparrow DF) = (\uparrow COMP * GF)$ (Austin, 2001) involves a regular expression identifying the initial element bearing a DF liked to a GF via a path containing any number of COMPlement clauses.

3.1 LDD Co-indexation in the ATB

In the ATB, LDDs are marked by traces represented as empty categories (ECs) co-indexed with antecedents as shown by example (3), where WHNP-1 and NP-1 have the same index number which indicate the connectedness of the two constituents. This makes the task of detection and extraction easy for our LFG f-structure annotation algorithm. 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 ‘arrived’.

(3) Al+riyAH+a (WHNP-1 Al~atiy) waSal+at (NP-SBJ-1 *T*)
“the-wind which arrived”

Tounsi et al. (2009), describe a methodology for annotating the ATB with LFG functional equations to generate f-structures for the ATB sentences. But as Bikel’s (Bikel, 2004) parser (and most treebank-based probabilistic parsers) does not capture LDDs, the automatically generated f-structures produced from parser output trees are proto-f-structures, as they only represent purely local dependencies. In order to produce proper f-structures, LDDs found in topicalisation and relativization and wh-questions must be captured. We develop a post-processing step to help recover LDDs on the automatically generated proto-f-structures by exploiting trace information in the ATB treebank and translate LDDs into corresponding reentrancies at the f-structure level using co-indexation. We then compute the probabilities of different paths with each LDD type.

3.2 LDD Extraction Methodology

Following the methodology devised by (Cahill et al., 2004), we recover LDDs for Arabic at the level of f-structures using probabilistic estimation of the finite approximations of functional uncertainty paths between the LDD re-entrant elements. We extract LDD resolution paths by collecting co-indexed elements in the f-structures that have been automatically generated from the ATB trees. The probability a path given a DF is estimated according to this formula (Cahill et al., 2004):

$$P(p \mid t) := \frac{count(t, p)}{\sum_{i=1}^{n} count(t, p_i)}$$

where *p* is the path and *t* is the LDD type (either TOPIC, TOPIC-REL or FOCUS). In our work, we make a more fine-grained classification of TOPIC-REL according to the type of the relative pronoun heading the clause (relative pronoun, relative adverb, subordinating conjunction, and null relative pronoun).

Type	Treebank Tags (Regular Expression)	Count in Treebank	Recall %
TOPIC	\-TPC	12,708	99.52
TOPIC-REL rel_ pron	REL_PRON	9,927	94.33
TOPIC-REL wh-less	\(WHNP\-[^-]+ \(\-NONE\- *0*\)	5,360	99.87
TOPIC-REL rel_adv	\-. \(REL_ADV	526	89.73
TOPIC-REL sub_conj	\(WHADVP\-. \(SUB_CONJ	506	92.69
FOCUS	\-.) INTERROG_	178	31.46
Total		29,205	97.11

Table 4: Recall results of path approximations

LDD Type	Unique path types
TOPIC	55
TOPIC-REL (rel_ pron)	98
TOPIC-REL (wh-less)	71
TOPIC-REL (rel_adv)	13
TOPIC-REL (sub_conj)	41
FOCUS	9

Table 5: Number of unique paths with LDD types

3.3 LDD Extraction Results and Examples

The total number of nodes with possible co-indexation in the ATB is 29,205, and the automatically extracted finite approximations of paths cover 97.11% of all possible LDD paths, as shown in Table 4. From the ATB, we extract 55 unique TOPIC, 123 collapsed TOPIC-REL and 9 FOCUS path types (with a total of 15,858 token occurrences). The distribution of unique paths with different LDD types is shown in Table 5.

Table 6 gives the most frequent paths extracted for each LDD type in the ATB furnished with probability information. The table shows that there is a marked preference for a single path with a high probability score, while the other path choices fall far behind.

Type	Frequency	Probability
TOPIC		
subj	10,207	0.81
subj:np_adjunct	934	0.07
FOCUS		
adjunct	39	0.71
subj	4	0.07
TOPIC-REL (rel_ pron)		
subj	4,560	0.49
obj:np_adjunct	1,783	0.19
TOPIC-REL (wh-less)		
subj	2,578	0.48
obj:np_adjunct	781	0.15
TOPIC-REL (rel_adv)		
adjunct	471	0.92
xcomp:adjunct	19	0.04
TOPIC-REL (sub_conj)		
adjunct	451	0.69
subj	105	0.16

Table 6: Most frequent LDD paths in the ATB

Below we provide some demonstrative examples of the most frequent LDD types and paths in Arabic according to our statistics.

1) AlwaDoEu tagay~ar+a "The situation changed."

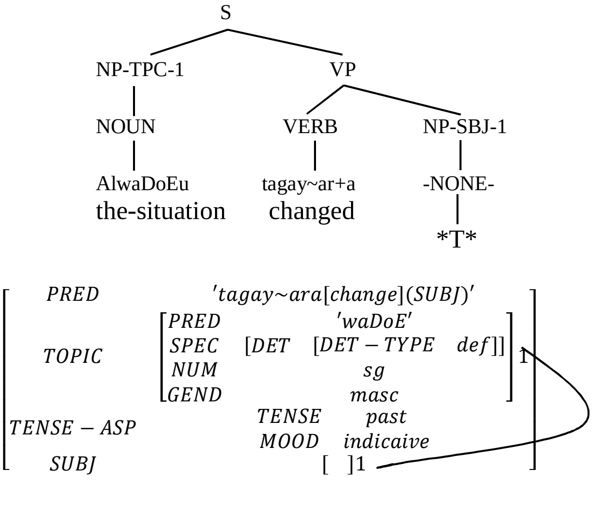

Figure 1: c- and f-sturcture of topicalization

Topicalized constructions. Subject-verb-object sentences are treated as topicalized constructions, where the subject is treated as a fronted TOPIC co-indexed with a gap in the original subject position, i.e. following the verb. This analysis is based on the theoretical assumption that Arabic is verb initial be default. Example (1) shows the FU equation (↑ TOPIC) = (↑ SUBJ), with a probability of 81%.

2) Alsay~idapu Al~atiy taEomalu fiy $arikapK
"the lady who works in a company"

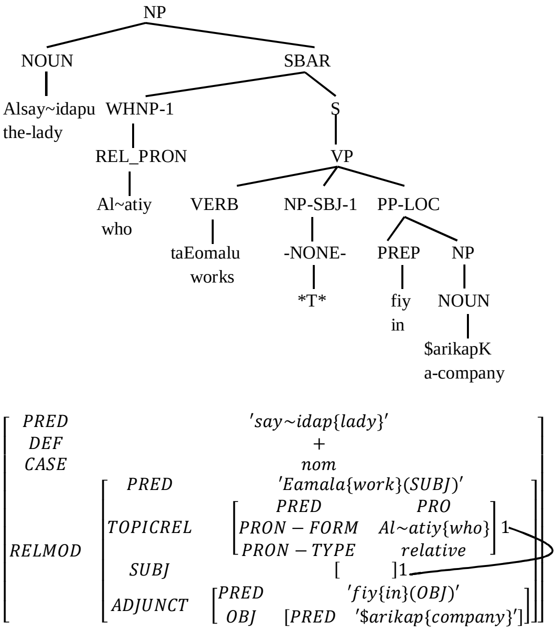

Figure 2: c- and f-structure of a relative clause.

3) >a$oxASN yuwAjihuwna ma$Akila
"person [who] face troubles"

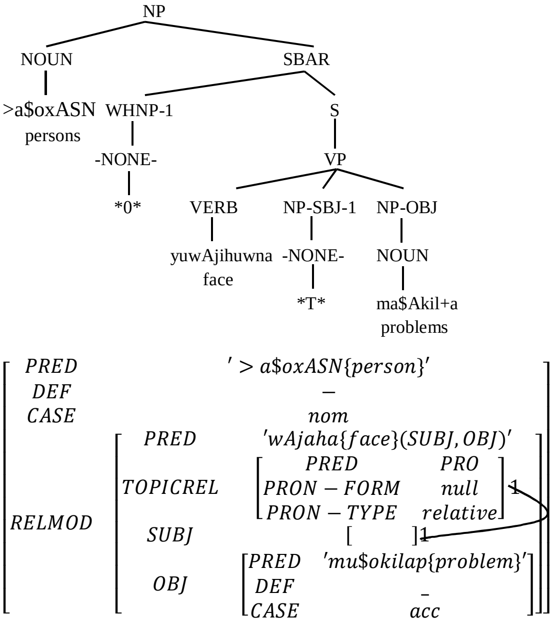

Figure 3: c- and f-structure of a wh-less clause.

Relative Clauses. Most frequently, relative clause in Arabic are headed by a relative pronoun. The relative pronoun has the DF of TOPIC-REL and is linked to a GF in the sentence. Example (2) has the FU equation (↑ TOPIC-REL) = (↑ SUBJ), with a probability of 49%.

Wh-less relative clauses. Wh-less clauses are abundant in Arabic (5,360 instances in the ATB), and they have their own specific morpho-syntactic constraints. The modified noun must be indefinite, and the relative clause must start with a verb, as shown by the following example. Example (3) shows the FU equation (↑ TOPIC-REL) = (↑ SUBJ), with a probability of 48%.

4) Al>aroDu Al~atiy yamolikuhA AlmuzAriEu
"Lit. the land which owns it the farmer"

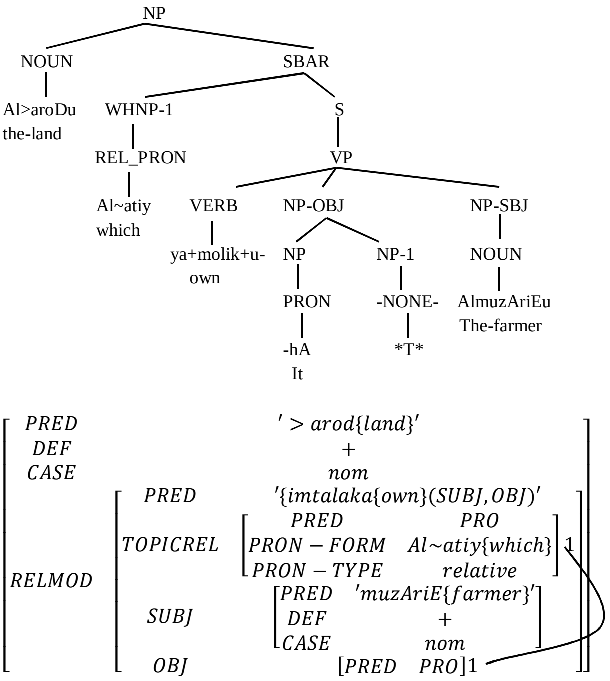

Figure 4: c- and f-structure for a clause with a resumptive pronoun.

Resumptive Pronouns. Resumptive pronouns are an interesting phenomenon in Arabic LDDs. They are defined as pronouns that are used in some languages to mark the lower end of an LDD (Falk, 2002). Resumptive pronouns fill the gaps in the domain of extraction, and like gaps, resumptive pronouns are linked to a discourse function. The Extended Coherence Condition allows an anaphoric link. Dalrymple (2001) pointed out that some languages signal the domain of extraction in LDD constructions by means of special morphological or phonological forms. The distribution of the resumptive pronouns in Arabic (Attia, 2008) shows they are optional when extracting from the object position. Example (4) shows the FU equation (↑ TOPIC-REL) = (↑ OBJ), with a probability of 19%.

3.1 Determining empirical upper-bounds for FU Equations

Theoretically, LDDs can span unbounded amounts of intervening syntactic elements (Kaplan and Zaenen, 1989). FUs are regular expressions denoting the set of possible paths in an f-structure between a source and a target f-structure. In the example below:

$$(\uparrow DF)\ =\ (\uparrow COMP * GF)$$

the Kleene closure operator indicates a potentially infinite number of grammatical functions intervening in the LDD path.

This however, is neither practical nor realistically descriptive of human language which is infinite due to its creative and dynamic nature, but at the same time limited by human memory temporal and physical and space. Table 7 shows the results of our statistics and probability estimation of the number of possible paths in the ATB. As the number of GFs in the LDD path increases the probability decreases significantly. We did not find any instances of an LDD path containing more than 5 GFs.

# of GFs in the LDD path	# of instances	% in the LFG-Annotated ATB
5	10	0.03
4	108	0.38
3	1,717	5.98
2	7,967	27.73
1	18,927	65.88

Table 7: Statistics and probability estimation on the number of GFs in the Arabic LDD paths

Therefore, we propose a revised version of the FU equation as follows:

$$(\uparrow DF) = (\uparrow GF\{0,l\}\ GF)$$

where l is the limit specified for a language (in Arabic the limit attested in the ATB is 5). The suggestion here is to replace the Kleene star with lower and upper bounds. Therefore the notion of

5) Al>arSidapu Al~atiy kAna AlmaSirifu yaDoTar~u <ilaA Allujuw'i <ilayohA
Lit. The assets which the bank was obliged to resort to it.

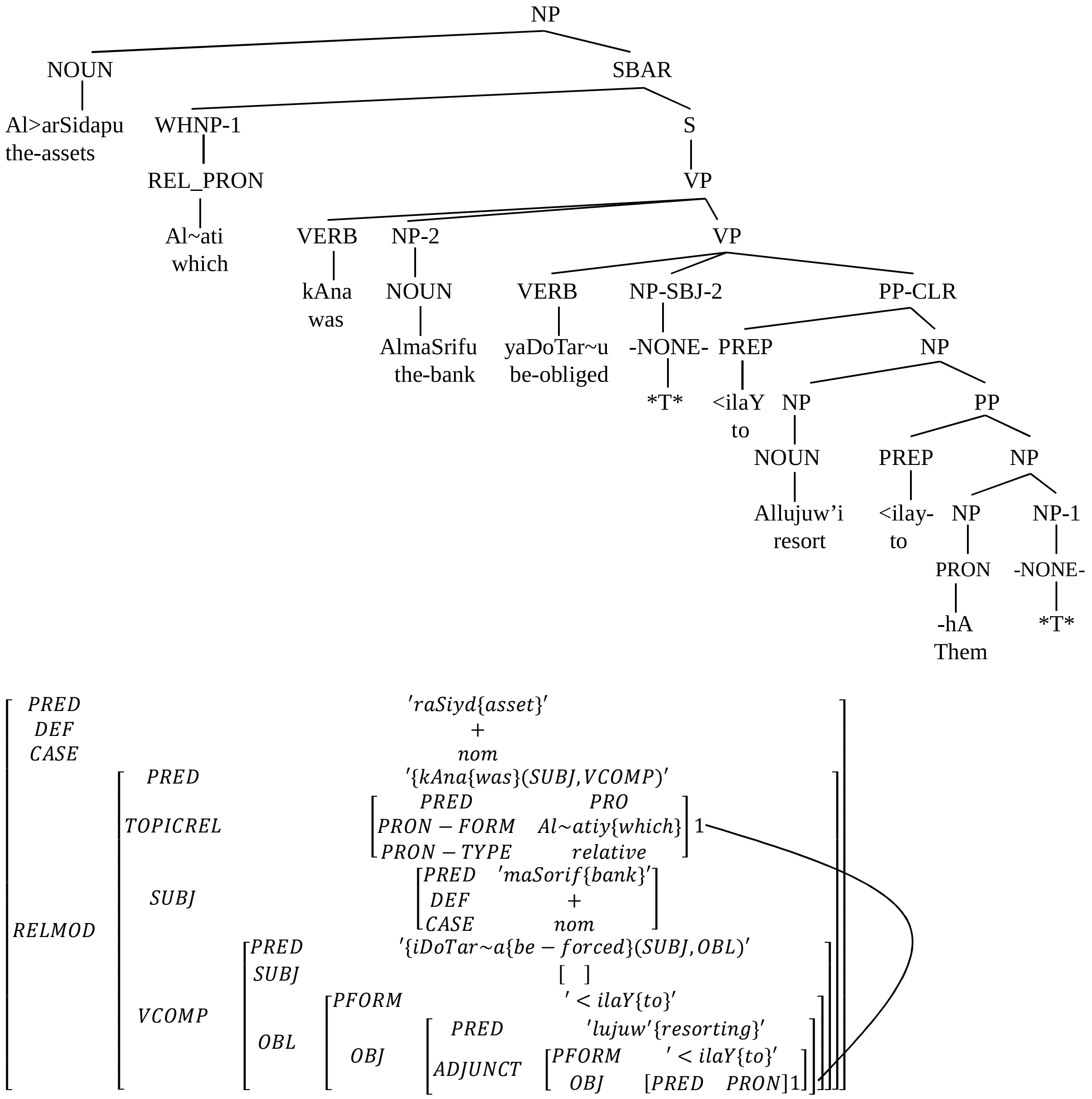

Figure 5: c- and f-structure of an LDD path containing 5 GFs.

unbounded dependency is refined by empirical facts which show that most LDD dependencies are clearly captured by an upper bound of 5. Example (5) represents an example of the longest LDD path we found in the ATB, with the LDD path described by the following equation.

$$(\uparrow TOPICREL) = (\uparrow VCOMP\ OBL\ OBJ\ ADJUNCT\ OBJ)$$

4. Future work

We plan to use the linguistic information extracted in this research in providing LDD information for Arabic free text parsed with a probabilistic CFG parser such as Bikel. According to Cahill et al. (2004), an LDD solution is ranked using the formula $P(s|l) \times P(p|t)$ which states that the probability of a solution is the product of the probability of a subcategorization frame given a lemma and the probability of a path given an LDD type.

5. Conclusion

We have successfully extracted LFG language resources for Arabic that include subcategorization frames and long-distance dependency (LDD) paths. Subcategorization frames describe the argument structure requirements of semantic forms, while LDD paths describe the grammatical functions that exist along the LDD between two co-indexed syntactic elements. With insights from the probabilistic facts on LDD paths, we propose an emprical upper bound on traditional LFG functional uncertainty equations. Subcategorization frames and LDD paths are extremely useful in enriching the output of probabilistic CFG parsers with deep syntactic knowledge. Besides its value for parsing, the capture and encoding of syntactic subcategorization frames is an essential requirement in the construction of computational and paper lexicons alike, and we believe this work will be of high interest to electronic lexicographers working with Arabic.

Acknowledgements

This research is funded by Enterprise Ireland (PC/09/037), the Irish Research Council for Science Engineering and Technology (IRCSET), and the UAE National Research Foundation (NRF) (Grant No. 0514/2011) , and the Science Foundation Ireland (Grant No. 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University.

References

Attia, Mohammed. (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.

Austin, P. K. (2001). Lexical functional grammar. In *International Encyclopedia of the Social and Behavioral Sciences,* eds. N. J. Smelser and P. Baltes, 8748-8754. Oxford, UK: Elsevier Science Ltd.

Bikel, Dan. (2004). A distributional analysis of a lexicalized statistical parsing model. In Proceedings of the 9th Conference on Empirical Methods in Natural Language Processing (EMNLP), Barcelona, Spain.

Buckwalter, Tim. 2004. Buckwalter Arabic Morphological Analyzer. Linguistic Data Consortium. (LDC2004L02).

Cahill, A., M. Burke, R. O'Donovan, J. van Genabith, and A. Way. (2004). Long-distance dependency resolution in automatically acquired wide-coverage PCFG-based LFG approximations. In: ACL 2004 - 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July 2004, Barcelona, Spain.

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

Dalrymple, Mary. (2001). *Lexical Functional Grammar*. New York: Academic Press.

Falk, Yehuda N. (2002). Resumptive Pronouns in LFG. In *The LFG 02 Conference*, Athens, pp. 154-173.

Guo, Yuqing and van Genabith, Josef and Wang, Haifeng. (2007). *Treebank-based acquisition of LFG resources for Chinese.* In: Lexical Functional Grammar 2007, 28-30, California, USA.

Kaplan, Ronald M., and Annie Zaenen. (1989). Long-distance Dependencies, Constituent Structure and Functional Uncertainty. In *Alternative Conceptions of Phrase Structure*, ed. by M. R. Baltin and A. S. Kroch. Chicago: University of Chicago Press.

Maamouri, M. and Bies, A. (2004). Developing an Arabic Treebank: Methods, guidelines, procedures, and tools. In Workshop on Computational Approaches to Arabic Script-based Languages, COLING.

O'Donovan, Ruth, Michael Burke, Aoife Cahill, Josef van Genabith and Andy Way (2005) Large-Scale Induction and Evaluation of Lexical Resources from the Penn-II and Penn-III Treebanks, Computational Linguistics, pages 329 – 366.

Petrov, Slav, Leon Barrett, Romain Thibaux, and Dan Klein. (2006). Learning Accurate, Compact, and Interpretable Tree Annotation. In Proceedings of COLING-ACL.

Rehbein, Ines and Josef van Genabith. (2009). Automatic Acquisition of LFG Resources For German - As Good As It Gets. Proceedings of the LFG09 Conference. Cambridge, UK

Schluter, Natalie and Josef van Genabith. (2008). Treebank-Based Acquisition of LFG Parsing Resources for French. In: the Sixth International Language Resources and Evaluation Conference (LREC'08), Marrakech, Morocco

Tounsi, Lamia, Mohammed Attia and Josef van Genabith. (2009). Automatic Treebank-Based Acquisition of Arabic LFG Dependency Structures. EACL-Workshop on Computational Approaches to Semitic Languages, Athens, Greece.

Vadas, D. and J. R. Curran. (2007). Adding Noun Phrase Structure to the Penn Treebank. The 45th Annual Meeting of the Association of Computational Linguistics, pp. 240–247, Prague, Czech Republic.