Modern Standard Arabic Grammar Automatic Extraction from Penn Arabic Treebank Using Natural Language Toolkit

Amira Abdelhalim[1], Sameh Alansary[2]
Phonetics and Linguistics Department, Faculty of Arts, Alexandria University, Alexandria, Egypt
[1]Amira.Abdelhalim@yahoo.com
[2]s.alansary@alexu.edu.eg

Abstract: ***This paper presents a methodology for rule based bottom up parsing technique forModern Standard Arabic (MSA) in Context Free Grammar (CFG) formalism in Phrase Structure Grammar (PSG) representation, where the grammar is automatically extracted from a syntactically annotated corpus.The extracted grammar is used to build an automatic lexicon and grammar rules module. Furthermore, the extracted CFG is further transformed into Probabilistic Context Free Grammar (PCFG) that could be used in a hybrid approach, which is also calculated automatically. The used corpus is the Penn Arabic Treebank(PATB)and algorithm implementation is performed with Natural Language Processing Toolkit (NLTK).The parser showed that automatic extraction of grammar improved the grammar building phase in both coverage of structures and time needed, but still needs further manual constrains addition. Automatic extraction of grammar is able to enhance rule based grammar parsers and it will enable a new paradigm of statistically directed symbolic parsing.***

Keywords*: Observational Based Grammar - Automatic Grammar Extraction- Rule Based Grammar – EnhancingArabic Grammar Parsing - Statistically Directed Symbolic Parsing.*

1 INTRODUCTION:

Parsing is responsible of determining the syntactic structure of an expression. Syntactic parsing is a vital step in any Natural Language Processing (NLP) application. Many attempts have been proposed to the study of syntactic structure analysis and generation, but only some of them have been proposed to Arabic. Syntax is concerned with describing the logical sequence of sentence units. Syntactic analysis process have been defined as ―the process of analyzing a sequence of tokens to determine its grammatical structure with respect to a given formal grammar. Parsing is used to refer to the process of building automatically syntactic analysis of sentences according to a given grammar [8].The parsing transforms input text into a data structure, usually a tree, which is suitable for later processing and which captures the implied hierarchy of the input, where different grammatical frameworks have been proposed [2]. Symbolic parser, rule based parsing, suffers from low structures coverage and long time needed for building. This paper presents an automatic extraction technique for automatic building of lexicon and grammatical rules to be used in a symbolic rule based parser.

Usually, such automatic extraction technique is used only for statistical parsing, i.e. for training parsers. This paper is intended for a new paradigm of parsing that adopts automatic extraction of grammar from a syntactically annotated Treebank for a statistically directed symbolic parsing. The symbolic parser is supposed to overcome usual rule based problems of low coverage and long-time required for grammar building due to extraction automation. In addition, the symbolic parser is supposed to be able to take decisions of grammar rules quantification for construction based on real statistical findings. For example, a simple question regarding the sequence of the grammatical rules, ex: what type of phrases should be parsed (identified) first prepositional phrase or noun phrase, is usually answered logically, prepositional phrase as they have less structures diversity. This type of decision and many others should be judged with statistical guidance through grammar extraction technique of grammar. The most important facility that this technique presents is syntactic relations quantification. Symbolic parsing suffers due to lack of quantification of syntactic relations (categorical and functional) both constituency and dependency relations needs to be captured. The automatic extraction technique enables symbolic parsing to quantify syntactic relations through a huge amount of real data and study frequencies and distribution of structures statistically. It is claimed that this approach will enable symbolic parsing to start a competitive challenge in front of statistical parsing. Figure one shows parsing for grammar extraction that should be stored for further quantification later on.

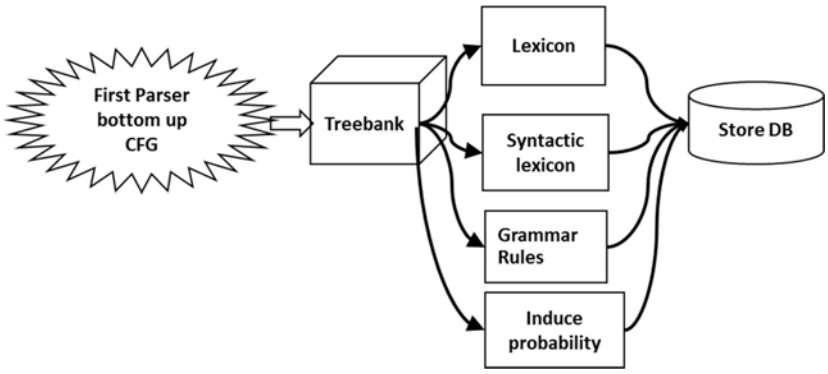

Figure 1: Grammar Extraction from Treebank

2 RELATED WORKS:

Some trials concentrated on rule based parser such as [14] used Affixs Grammars Over Finite Lattices formalism to build Arabic morpho-syntactic analyzer. [13]Used Unification Based Grammar formalism. Some trials also concentrated on statistical parser such as [15] developed a parser that learns from Penn Tree Bank (PTB) the functional labels to use it in Lexical Functional Grammar formalism. [5]Used PTB as a learning data in order to extract most common trees for syntactic interpretation of new sentences with accuracy 89.85%. [7] used machine learning methods for tokenization and part of speech (POS) tagging and base phrase chunking, it used 10% of the PAT corpus with F-score of 96.33%.

As for Arabic and CFG,[4] used CFG for designing a top down parser for simple Arabic sentences with specific domain. They developed a precise description of Arabic grammatical sentences to feed their parser with. The parser starts with word classification, rule identification then parsing. They mentioned that it showed effective results for MSA sentences. They used simple sentences both verbal and nominal from real documents, but for a specific domain with accuracy 70%. [3]Implemented a parser that checks Arabic sentence grammatical structure well-formedness. Their top-down parser scored average accuracy rate of 95%.

It is obvious that each trail whether statistical or rule based has its own formalism, parser and even evaluation metric; which causes comparison difficulty to researchers.

3 PARSING APPROACHES:

Three main approaches are recognized for parsing: the linguistic rule based approach, statistical approach and hybrid mixture of the two.The first linguistic approach uses lexical knowledge and language rules in order to parse a sentence. It is a very promising approach but requires huge amount of work and time. On the other hand, statistical approaches are based on statistics and probabilistic models. It is based on the frequencies of occurrences that are automatically derived from corpora. It is known for fast development that saves time and effort but still has many challenges due to the complexity of language infinite identity type, reflecting human mind. The third hybrid approach integrates both of them, taking advantage of grammar rules robustness and statistical models fastness. This paper extracts grammar automatically, for both rule based parsing in CFG and PCFG and uses the set of grammar rules from them on further data.

4 FORMAL LANGUAGE CFG AND REWRITE RULES:

In both mathematics and linguistics a formal language is a set of strings of symbols that may be constrained by rules that are specific to it. It is used as an agreed language to describe some knowledge of certain kind of data or to define the relationship between elements (linguistic data) and their representation formally. For linguistics, one of the commonly used formal languages is called CFG. CFG consists of a set of rewrite rules with certain categories of terminal and non-terminal symbols defined by the linguist of the form A --> B, where A belongs to the set of non-terminals and B belongs to the set of terminal or non-terminal symbols. CFG defines a formal relationship between a set of possible texts and their representations. Using this language with any linguistic representation (dependency, phrase structure or feature based) is able to supply a representation of sentences using these rewrite rules.

It is used to describe or define the sentences, whereas the representation combined with a certain linguistic theory used as a procedure or instructions to be followed. This bundle of rules as well as the chosen approach of sentence representation is called Generative Grammar.

(Sarkar, 2011) illustrated parsing issues, CFG as one, and stressed the important point that using CFG for the syntactic analysis of natural language is very problematic. The grammar of natural languages is far too complicated than just listing a set of rules; he described it as being similar to an acquisition problem. He also highlighted the second problem of resolving ambiguity such as recursive rules.

However, this limitation has been reinforced by the addition of augmentation and features to rules. The sub - categorization features of the categories may also be added between brackets V [transitive] and sequence is represented by order and sentence position by dash [- NP]. Sub-categorization features is added to CFG as appropriate restriction formal representation added to represent context.

5 BASIC SEARCH AND MATCHING STRATEGIES FOR PARSING:

Two basic approaches of Top-down and Bottom-up parsing, as other approaches are based on them.The start point of handling the data is the first basic decision that needs to be taken in the parsing process. In top-down parsing, the process starts from the most abstract point, in our case study of syntactic structure of PSG, it is the S and directs towards the lowest level building the structure reaching words. On the other hand, in the bottom-up approach the parsing starts at the lower level, which is words, and attempts to build upwards. In most real applications, the top-down approach is commonly used with statistical parser whereas bottom-up is used with rule-based applications. The recursive rules of rule-based applications with a large grammar and many potentially ambiguous sentences predicts along with top down approach an infinite variety of possible structures. On the other side, using bottom-up approach makes it possible to parse the hypothesis list faster as it goes upwards testing through a defined set of restricted categories. Suppose the proposed grammar contains the following set of rules that are written in terms of categories, taking into consideration that the lexicon also contains the words with their features attached,

a. S -> NP - VP
b. S -> NP – VP – PP
c. NP ->Det – N
d. NP ->Det – Adj – N
e. NP ->Pron
f. NP ->Det – Adj – N – NP
g. NP ->Det – N – PP
h. PP -> Prep – NP

Looking at the number of possibilities using top-down technique along with these possibilities embedded in the recursive rules the number of predicted structures is enormous even before consulting any word in the lexicon. Bottom-up process less possibilities but does not consider a backtrack solution.

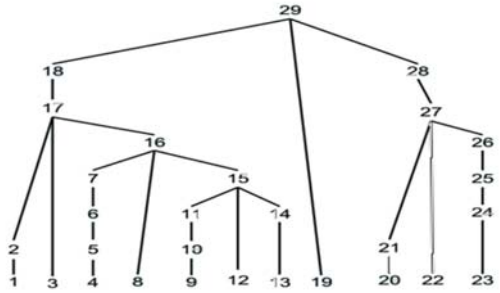

Figure 2: Bottom-Up Parsing

6 INTUITION BASED Vs. OBSERVATION BASED GRAMMERS:

In order to formulate rules two main approaches have to be discussed, intuition-based grammars and observational grammars [1]. The intuition based grammar was adopted by Chomsky, it is based on constructing sentences and introspection. The second is based on actual texts taken as evidence to draw conclusions as corpus linguists do.

Corpus provides empirical data and in case conjoined with a computational tool, it addresses issues that were previously intractable, as not only it allows for quantitative analysis, but also investigation of structures embedded in real discourse [6]. Corpus have had opened new areas of research in grammar. It facilitates the study of a single grammatical

construction and obtains information about the usage of different grammatical constructions and uses this information as the basis for writing a reference grammar [12].

7 GRAMMER DEVELOPMENT STRATEGIES:

The rule-based grammar is usually built either with Manual Grammar Development, toy grammar, that needs a skilled human team with a solid experience and knowledge in both theoretical linguistics and grammar formal representation. The major problem is time and consistency of each rule represented. That‘s why different Grammar Development Environment of software systems offer grammar writers incremental input, grammar editing, browsing, searching and tracing or debugging. The other approach is Automatic Grammar Induction which isbased on Treebank‘s, as the linguistic intuition is externalized into the annotation of the Treebank and the grammar. It is a fast and cheap method [10].

8 THE PATB CORPUS:

Tree banks are a collection of syntactically annotated sentences of a large amount of corpora. PATB is considered the most usable Treebank that uses PSG and also available for Arabic. It is a syntactically annotated Treebank‘s that is vital for training parsers as well as finding constructions for any syntactic study, specifically development of grammar based parsers.

The PTB project started in 2001 at the Linguistic Data Consortium and University of Pennsylvania. It offers two types of conventions, the original constituency and a converted dependency representation in the Columbia Arabic Treebank (CATiB), for many languages including Arabic. It consists of 23,611 parse annotated sentences from Arabic newswire text in MSA. It is one of the most significant transitions of Arabic NLP as many researches and tools for morphology and syntax, data-driven or rule based depended on it as a standardized source of annotated data [11]. Many of the significant Arabic NLP is based on it, the morphological analysis, disambiguation, POS tagging and tokenization [9].

The version used is part three, version one that consists of, basically 600 stories from Al Nahar News Agency, referred to as ANNAHAR. The stories are specified with a DOC ID along with date. The average number of words per story is 567 and total word token is 340,281.
The corpus is first annotated with Tim Buckwalter’s lexicon and morphological analyzer to generate a list of candidate POS tags for each word. The second step is manual choice from candidate tag (lexical category) along with inflectional features and gloss and then automatic clitic separation and then parsing annotation of constituent structure along with functional function categories for each non-terminal node. The main files that are vital are the “.sgm” file that contains the raw corpus, the “.tree” file that has the parsed annotated corpus.

- **Features and Their Annotation:**

Main Features of Penn Treebank Constituent tags:

S	sentence
NP	noun phrase
VP	verb phrase
PP	prepositional phrase
SBAR	S-bar (subordinate clause, complementizer or WH- and sentence)
SBARQ	S-bar that is a question
SQ	S that is a question
NX	noun head in certain complex coordination contexts
PRN	parenthetical
PRT	particle
QP	quantity phrase (multi-word numbers)
ADJP	adjective phrase
ADVP	adverb phrase
FRAG	fragment
WHNP	WH- noun phrase
WHPP	WH- prepositional phrase
WHADJP	WH- adjective phrase
WHADVP	WH- adverb phrase
CONJP	conjunction phrase (multi-word conjunction)
INTJ	interjection
NAC	Not-A-Constituent (mostly rightward moved conjuncts with conjunction)

UCP	Unlike-Coordinated-Phrase (dominates coordination of NP and PP, e.g.)
X	unknown, technical problem, etc.

9 GRAMMAR EXTRACTION AND PARSING:

A. NLTK Framework:

NLTK is a python platform for building and testing NLP applications. It provides easy to use libraries based on Object Oriented Model of programming. Its libraries are organized into packages of modules, classes and functions that are easily used for different purposes such as classification, stemming, tagging, parsing and semantic reasoning. It also offers a powerful API documentation.

It is used in this paper as the platform that is responsible for reading the parsed corpus, extracting CFG and CFG augmented with Features grammar, calculated probability for PCFG augmented with features productions generation, drawing parsed trees representation, generating files of written extracted grammar both rule based and probabilistic grammars and testing these extracted files on further data.

B. Algorithm:

The CFG class first identifies the non-terminal symbol as an object and then expands it to the right hand side. It accepts a feature structure object, a grammatical category along with its features description in CFG representation, which is used in a feature based grammar and equivalent to CFG but all non-terminals are feature-struct non-terminal of feature based grammar in CFG augmented with features. This feature structure is important to represent annotated data grammar of a parsed corpus.The grammar production maps a single symbol on the left to sequences on the right.

It can construct a probabilistic production by creating another new object from the given start state and a set of probabilistic productions. It takes the featured CFG productions and return featured PCFG production. A featured PCFG consists of a start state and a set of productions with probabilities. The set of terminals and non-terminals is implicitly specified by the production. Any given left hand must have a probability that sum to 1.

TABLE 1: ALGORITHM FOR EXTRACTING GRAMMAR WITH NLTK

Read Arabic Penn Treebank	Preprocessing.
	Create a Parsed Corpus Reader.
	Iterate over Parsed Sentences.
	Draw Sentences Trees (to check reader and sentences).
	Extract Grammar Rules and Lexical Rules.
	Write Them in a File.
	Print Number of Sentences and Tokens.
PCFG	Split Data To Training and Testing
	Extract CFG Grammar and Lexicon
	Add Probability on CFG
	Write PCFG To a File
Testing Grammar:	Open and Read Extracted PCFG File Open and Read CFG File
	Create a Probability Parser to use PCFG Extracted Grammar For Parsing Test Sentences. Create a CFG Parser and Read Extracted FCFG Grammar For Parsing Test Sentences.

C. The Training Phase:

The training phase involves the usage of the parsed corpus to extract the grammar rules along with their features. The corpus is divided into a training set and testing set. For training, a preprocessing phase is performed where each annotated sentence is copied manually to a file, each sentence in a separate line.The combination of the features and categories allows the training corpus to learn allocation of each word in the sentence as grouping of sequence of labels, both features and categories, in the most probable syntactic group. The extracted grammar is saved to a file.

Rule: S -- > (C1) (C2)
C1-- > (W1,W2)
C2 -- > (W3,W4)
W -- > terminal word as written in raw text

where S represents the sentence, C represents the constituent category non- terminal label and W represents the word category along with their features. The first three rules are called grammar rules, whereas the last is a lexical rule.

Examples of Extracted Rules:

Grammar productions (start state = S)
NP-OBJ-1 -> -NONE-
DET+NOUN+CASE_DEF_GEN -> "Al$y'"
PP-TMP -> PREP NP
DET+NOUN+CASE_DEF_GEN -> 'AlfEl'
S -> S CONJ S PUNC
NP -> NP SBAR
NOUN+CASE_DEF_NOM -> 'qA}d'
VP -> PRT IV3MS+IV+IVSUFF_MOOD:J NP-SBJ PP NP-PRD
PV_PASS+PVSUFF_SUBJ:3MS -> 'qtl'
NP -> NP PP

```
PREP -> 'mn'
PP -> PREP NP
ADJP-PRD -> ADJ+CASE_INDEF_ACC
NP-SBJ -> PRON_3FS
ADJ+CASE_DEF_NOM -> '|xr'
NP -> DET+NOUN+CASE_DEF_GEN
VP -> IV3MS+IV+IVSUFF_MOOD:I NP-SBJ-3 NP-OBJ NP-ADV
-NONE- -> '*ICH*'
IV3MS+IV+IVSUFF_MOOD:I -> 'ykAd'
IV3MS+IV+IVSUFF_MOOD:J -> 'ykn'
NP -> PRON_3MS
NP-SBJ -> -NONE-
DET+NOUN+NSUFF_MASC_PL_GEN -> 'AlmSwryn'
DET+NOUN+CASE_DEF_ACC -> 'AlbAS'
ADJ+CASE_INDEF_GEN -> 'xAft'
NP-SBJ-1 -> NP NAC-2
NP -> NOUN+CASE_DEF_NOM NP
PUNC -> '-LRB-'
VP -> PV_PASS+PVSUFF_SUBJ:3MS NP-SBJ-1 NP-OBJ-1 PP-TMP NAC-2
NP-SBJ-2 -> -NONE-
SBAR -> PART S
```

D. Calculate PCFG:

NLTK could be used as well for constructing probabilistic models. Internally the library generates a descriptive extraction vector for each word by its morphological features, both its category along with features. The vector is completed by the appropriate syntactic class of the non-terminal constituent label. Each vector represent the corpus in a tabular way which consists of the words sequence, represented in terms of features, and the return at the end of it (vector 1: Det N ?, NP).

E. The Testing Phase:

The testing corpus also contains a set of annotated sentences and same set raw un-annotated each in a line. Both extracted PCFG and CFG grammar are used to analyze it. The parsing is generated in a file along with the tracing of the steps.

10 CONCLUSIONS:

This paper presented a bottom-up CFG parser using NLTK for PATB parse trees. The technique enabled the automatic extraction of grammar that is intended to improve rule based symbolic parsing in terms of coverage and time needed for building. Grammatical structures will be further refined with the addition of manual constrains. This approach will support the statistically directed symbolic parsing that enhances the architecture of symbolic parsing. In addition, the quantification of syntactic relations is now available due to automatic grammar extraction from the Treebank.

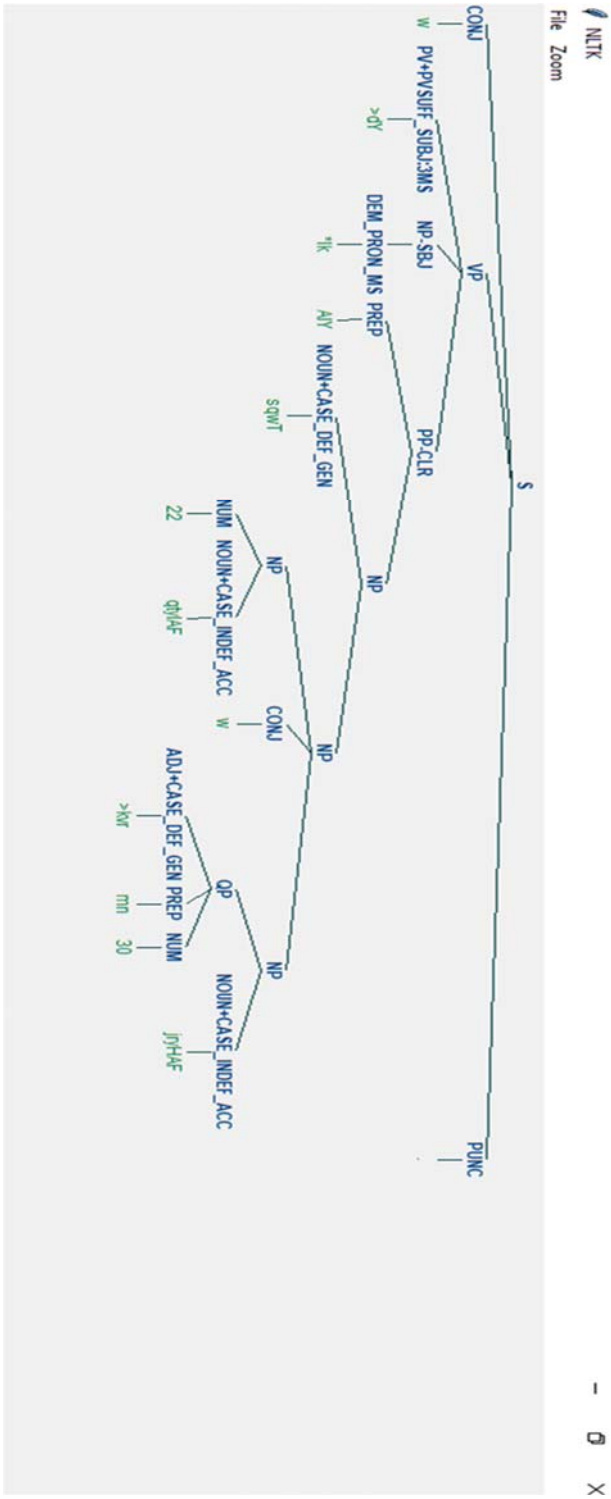

Figure 3: Tree for an Extracted Grammar Rule and Lexical Items for a Sentence in PATB 30 "وأدى ذلك الى سقوط 22 قتيلاً وأكثر من 30 جريحاً"

REFERENCES:

[1] Aarts, J. (1991). Intuition Based and Observation-Based Grammar. In Aijmer

[2] Al-Daoud, E., &Basata, A. (2009). A framework to automate the parsing of Arabic language sentences. Int. Arab J. Inf. Technol., 6(2), 191-195.

[3] Alqrainy, S., Muaidi, H., &Alkoffash, M. S. (2012). Context-Free Grammar Analysis for Arabic Sentences. International Journal of Computer Applications, 53(3), 7-11.

[4] Al-Taani, A. T., Msallam, M. M., &Wedian, S. A. (2012). A top-down chart parser for analyzing arabic sentences. Int. Arab J. Inf. Technol., 9(2), 109-116.

[5] Ben Fraj, F., Ben Othmane-Zribi, and Ben Ahmed, M., 2010, ―Parsing Arabic Texts Using Real Patterns of Syntactic Trees‖ The Arabian Journal for Science and Engineering, Volume 35, Number 2C.

[6] Biber, D., Conrad, S., &Reppen, R. (1998). Corpus linguistics: Investigating language structure and use. Cambridge University Press.

[7] Diab, M. (2009). Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and base phrase chunking. In 2nd International Conference on Arabic Language Resources and Tools (Vol. 110).

[8] Grune, D., & Jacobs, C. Parsing Techniques–A Practical Guide. 1990. VU University. Amsterdam.

[9] Habash, N., &Rambow, O. (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 (pp. 578-580).

[10] Kakkonen, T. (2007). Framework and resources for natural language parser evaluation. University of Joensuu.

[11] Maamouri, M., Bies, A., Buckwalter, T., &Mekki, W. (2004, September). The pennarabictreebank: Building a large-scale annotated arabic corpus. In NEMLAR conference on Arabic language resources and tools (pp. 102-109).

[12] Meyer, C. F. (Ed.). (2002). English corpus linguistics: An introduction. Cambridge University Press.

[13] Othman, E., Shaalan, K., &Rafea, A. (2004, September). Towards resolving ambiguity in understanding arabic sentence. In International Conference on Arabic Language Resources and Tools, NEMLAR (pp. 118-122).

[14] Ouersighni, R. (2001, July). A major offshoot of the DIINAR-MBC project: AraParse, a morphosyntactic analyzer for unvowelled Arabic texts. In ACL 39th Annual Meeting (pp. 9-16).

[15] Tounsi, L., Attia, M., & van Genabith, J. (2009). Parsing Arabic using treebank-based LFG resources.

BIOGRAPHIES:

Amira Abdelhalim: Teacher Assistant, Faculty of Arts, Phonetics and Linguistics Department, Alexandria University. She got her MAwith excellent degree on "A Formal Approach to Modern Standard Arabic Syntax: A Corpus Based Study" in 2016.Her main areas of interest are corpus based studies, Arabic morphology, Arabic syntax, Arabic semantics, Machine Learning Techniques and Language Modeling.

Dr. Sameh Alansary: *Director of Arabic Computational Linguistic Center at Bibliotheca Alexandrina, Alexandria, Egypt.*

He is professor of computational linguistics in the Department of Phonetics and Linguistics and the head of Phonetics and Linguistics Department, Faculty of Arts, Alexandria University. He obtained his

MA in Building Arabic Lexical Databases in 1996, and his PhD from Nijmegen University, the Netherlands in building a formal grammar for parsing Arabic structures in 2002. His main areas of interest are concerned with corpus work, morphological analysis and generation, and building formal grammars.

He is also the head of Arabic Computational Linguistics Center in Bibliotheca Alexandrina. He is supervising and managing the Universal Networking Language project in Library of Alexandria since 1-6-2005 till now.

Dr. Alansary is the co-founder of the Arabic Language Technology Center (ALTEC), an NGO aims at providing Arabic Language resources and building a road map for Arabic Language Technology in Egypt and in the Middle East. He has many scientific works in Arabic Natural Language Processing published in international conferences and periodicals, and a member in many scientific organizations: (1) Egyptian Society of Language Engineering, Cairo, (2) Arabic Linguistic Society - USA, (3) Association of Computational Linguistics - USA – Europe, (4) Universal Networking Language foundation, United Nations, Geneva, Switzerland.

استخلاص قواعد النحو لتراكيب جمل اللغة العربية المعاصرة آليا بإستخدام عينة لغوية من Penn Arabic Treebank باستخدام NLTK

Amira Abdelhalim[1], Sameh Alansary[2]
Phonetics and Linguistics Department, Faculty of Arts, Alexandria University, Alexandria, Egypt
[1]Amira.Abdelhalim@yahoo.com
[2]s.alansary@alexu.edu.eg

ملخص—تقدم هذه الورقة البحثية نظاما للتحليل النحوي الآلي للغة العربية المعاصرة. تعتمد المنهجية على بناء محلل نحوي يقوم بإستخلاص القواعد النحوية للجملة آليا من خلال مدونة لغوية موسومة بتحليل تركيبي للجمل. المدونة المستخدمة هي **Penn Arabic Treebank** والأداة الحاسوبية لبناء المحلل النحوي واستخلاص القواعد لبناء معجم آلي للقواعد والكلمات هي **NLTK**. وقد أدى بناء معجم الكلمات والقواعد إلى سرعة البناء وشمولية المركبات الممثلة ولكن لازالت القواعد تحتاج لإضافة قيود لفك اللبس التركيبي للجمل المتشابهه في أجزاء منها من الوحدات التركيبية.