Bel-Arabi: Advanced Arabic Grammar Analyzer

Michael Nawar Ibrahim, Mahmoud N. Mahmoud, and Dina A. El-Reedy

Abstract—**This paper proposes a framework to automate the grammar analysis of Arabic language sentences (إعراب الجمل). The grammar analysis is considered one of the complex tasks in the Natural Language Processing (NLP) field; since it determines the relation between noun and verb on the level of sentence, or noun with the letter before it or after it or noun and a character on the last level of the preposition. The construction of a rule-based high-accuracy grammar analyzer is a complex, high resource consuming task. Then, we proposed a hybrid system between learning-based approaches and rule-based approaches, which provides an acceptable accuracy and could be simply implemented. However the results of the proposed framework are really promising and it has the potential to be further improved.**

Index Terms—**Arabic natural language processing, case ending diacritization, grammar analyzer.**

I. Introduction

Arabic grammar analysis is the process of determining the grammatical role, and case ending diacratization of each word in an Arabic sentence. Grammar analysis is distinct from parsing, since it assign additional information like case ending diacratization of each word. Grammatical role of a word is determined by the relation between a word and its dependents. Grammar analyses are flatter than regular parsing tree structures because they lack a finite verb phrase forms. Once the Arabic grammar analysis of a sentence is completed many problems can be simply solved such as automatic diacritics, Arabic sentences correction and accurate translation.

As example for the task of grammar analysis, let's consider the sentence " الأولاد يلعبون في حديقة المدرسة مع بعضهم" to grammatically analyze it. The output of the framework for such sentence is shown in Arabic in Table I.

The proposed framework is divided into five main components. Three of them: Stemmer, Part of Speech Tagger (POS tagger), and Base Phrase chunker are learning-based. The learning-based components use a "Conditional Random Field" classifier [1]. The remaining two components: Morphological Analyzer and Arabic Grammar Database are rule-based.

The proposed framework covers the basic grammar rules for verbal and nominal sentence. However, it has the following limitations:

First, the system is assuming that sentence has been written correctly, whether morphologically or grammatically, and grammar correction is not included right now.

TABLE I: Example of Grammar Analysis

Word in Arabic	Transliterated Word	Grammatical Role	Sign
الأولاد	Alawlad	Subject	Nominative with damah
يلعبون	ylEbwn	Present verb	Nominative with existing noon
في	fy	Uninflected particle	-
الحديقة	AlHadyqp	Genitive noun	Genitive with kasrah
مع	mE	Uninflected circumstance	-
بعض	bED	Possessive	Genitive with kasrah
هم	hm	Uninflected pronoun	-

Second, as a nature of Arabic verbs, the verb could be in passive or active voice e.g., (ضرب, "drb") could be read as ضُرِبَ (doreb, "beaten") or ضَرَبَ (darab, "beat"), the system assumes the verb as it is in the active voice.

Third, the grammar analyzer does not prevent errors that are related to incorrect use of semantic meaning, means that the semantic analysis is not verified.

It is not a simple matter to evaluate the Bel-Arabi framework, due to the absence of standard data for the Arabic grammar analysis task. So, we have generated 600 sentences for the evaluation of the system.

This paper is organized as the following: in Section II, an overview of Arabic natural language processing is presented. In Section III, previous work in the field of Arabic grammar analysis is discussed. In Section IV, the proposed framework is explained. The data collected for the evaluation, and the evaluation process are presented in Section V. Finally, concluding remarks are presented in Section VI.

II. Arabic NLP and Data

There are three main categories of Arabic language; classical – the language of Qur'an, modern standard (MSA) – which is a simplified form of classical that is extracted from news and written documents, and dialectical Arabic which differs from one country to another. One variation of it is the colloquial language which is the daily used language by Egyptians.

In general Arabic has a very rich morphological language where each word can include number, gender, aspect, case, mood, voice, mood, person, and state. The Arabic basic word form can be attached to a set of clitics representing object

Manuscript received February 5, 2014; revised January 21, 2015.

M. N. Nawar, M. N. Mahmoud, and D. A. El-Reedy are with the Computer Engineering Department, University of Cairo, Giza, 12613 Egypt (e-mail: michael.nawar@eng.cu.edu.eg, mah.nabil@ieee.org, dina.elreedy@gmail.com).

DOI: 10.7763/IJSSH.2016.V6.669

pronouns, possessive pronouns, particles and single letter conjunctions. Obviously the previous features of Arabic word increase its ambiguity. Generally Arabic stems can be attached three types of clitics ordered in their closeness to the stem according to the following formula:

{[proclitic1] {[proclitic2] {Stem [Affix] [Enclitic]}

where proclitic1 is the highest level clitics that represent conjunctions and is attached at the beginning such as the conjunction [(و, w, ‘and’),(ف , f, ‘then’)]. Proclitic2 represent particles [(ب, b, ‘with/in’) ,(ل , l, ‘to/for’) (ك, k, ‘as/such’)]. Enclitics represent pronominal clitics and are attached to the stem directly or to the affix such as pronoun [(ه , h , ’his’), (هم , hm , ‘their/them’)].

The following is an example of the different morphological segments in the word وبقدراته that has the stem (قدر ,qdr ,power), the proclitic conjunction (و, w, ‘and’) , the proclitic particle (ب , b ,‘with/in’) , the affix (ات, At ,for plural) ,and the citicized pronoun (ه , h , ’his’).

The set of proclitics considered in this work are the particles prepositions {b, l, k}, meaning {by/with, to, as} respectively, and the conjunctions {w, f}, meaning {and, then} respectively. Arabic words may have a conjunction and a preposition and a determiner cliticizing to the beginning of a word. The set of possible enclitics comprises the pronouns and (possessive pronouns) {y, nA, k, kmA, km, knA, kn, h, hA,hmA, hnA, hm, hn}, respectively, my (mine), our (ours), your (yours), your (yours) [masc. dual], your (yours) [masc. pl.], your (yours) [fem. dual], your (yours) [fem.pl.], him (his), her (hers), their (theirs) [masc. dual], their (theirs) [fem. dual], their (theirs) [masc. pl], their (theirs) [fem. pl.]. An Arabic word may only have a single enclitic at the end. We define a token as a (stem + affixes), proclitics, enclitics, or punctuation.

III. Arabic NLP Systems

For the last two decades concentration on Arabic language processing has focused on morphological analysis. In this field, many working systems have been achieved [2]-[4]. Few systems for more complicated NLP tasks are developed.

One of the developed NLP systems is MADA and TOKAN [5], [6], which is a suite of tools for morphological disambiguation, POS tagging, diacritization, lexicalization, lemmatization stemming and other tasks. MADA and TOKAN have been done on addressing different specific natural language processing tasks for Arabic. MADA is a system for Morphological Analysis and Disambiguation for Arabic. TOKAN is a general tokenizer for MADA-disambiguated text. In simple words, the MADA system along with TOKAN provide one solution to different Arabic NLP problems.

Other developed system for different Arabic NLP problems is the AMIRA system [7]. AMIRA is a toolkit for Arabic tokenization, POS tagging, Base Phrase Chunking, and Named Entities Recognition. AMIRA is a successor suite to the ASVMTools. The AMIRA toolkit includes a clitic tokenizer (TOK), part of speech tagger (POS) and base phrase chunker (BPC) - shallow syntactic parser. The technology of AMIRA is based on supervised learning with no explicit dependence on knowledge of deep morphology; hence, in contrast to systems such as MADA, it relies on surface data to learn generalizations. In general the tools are based on using a unified framework casting each of the component problems as a classification problem.

Also, one of the large groups interested in Arabic NLP is RDI Egypt. RDI has been one of the regional and international leading key players in the R&D of Arabic Human Language Technologies for the last 10 years. RDI provides automatic Arabic diacritizer [8], Arabic morphological analyzer [9], Arabic part-of-speech tagger [10], Arabic Lexical Semantic Analyzer [11], Text to Speech System, Arabic Text Search Engine, and Arabic Lexical Dictionaries.

Finally, Stanford natural language processing group, which is a group for natural language processing research scientists, postdocs, programmers and students, is developing Arabic NLP tools. The developed Arabic NLP products are a word segmenter [12], state-of-the-art part-of-speech tagger [13] and a high performance probabilistic parser [14] the data set used is the Penn Arabic Treebank [15].

IV. Arabic Grammar Analysis Current Research

Although the importance or Arabic grammar analysis, few researchers tried to solve the issue of grammar analysis. There are two main techniques used to deal with grammar analysis for Arabic language: rule-based technique, and parsing technique.

Al Daoud *et al.* [16] propose a framework to automate the grammar analysis of Arabic language sentences in general, although it focuses on the simple verbal sentences but it can be extended to any Arabic language sentence. This system assumes that the entered sentences are correct lexically and grammatically. This system assumes that verb as it is in the active voice.

Attia [2], [3] investigates different methodologies to manage the problem of morphological and syntactic ambiguities in Arabic. He built an Arabic parser using Xerox linguistics environment which allows writing grammar rules and notations that follow the LFG formalisms. Attia tested his approach on short sentences randomly selected from a corpus of news articles; he claimed a performance of 92%.

Habash *et al.* [17] construct The Columbia Arabic Treebank (CATiB). Columbia Treebank is a database of syntactic analyses of Arabic sentences. CATiB contrasts with previous approaches to Arabic Treebanking in its emphasis on speed with some constraints on linguistic richness. Two basic ideas inspire the CATiB approach: no annotation of redundant information and using representations and terminology inspired by traditional Arabic syntax. So the task of grammar analysis can be done by applying a simple parsing approach.

Duke *et al.* [18] constructed the Quranic Arabic Dependency Treebank (QADT), which is an annotated linguistic resource consisting of 77,430 words of Quranic Arabic. This project differs from other Arabic tree banks by providing a deep computational linguistic model based on historical traditional Arabic grammar.

Most of the related work reported in this study concentrated on short sentences and used hand-crafted grammars, which are time-consuming to produce and difficult to scale to unrestricted data. Also, these approaches used traditional parsing techniques like top-down and bottom-up parsers demonstrated on simple verbal sentences or nominal sentences with short lengths.

V. The Proposed Framework

The proposed framework takes an input of sentence, and it assigns each token an appropriate tag, case, and a sign as follow:

Arabic tags :{present verb (فعل مضارع) , imperative verb (فعل أمر) , past verb (فعل ماضي) , doer (فاعل) , direct object (مفعول به) , cognate accusative (مفعول مطلق) , cognate accusative delegate (نائب للمفعول المطلق) , subject (مبتدأ) , predicate (خبر) , delayed subject (مبتدأ مؤخر) , ena subject (أسم إن) , ena predicate(خبر إن) , kan subject (أسم كان) , kan predicate (خبر كان), kad subject (أسم كاد) , kad predicate (خبر كاد), apposition (بدل) ,adjective (نعت) , incorporeal emphasis (توكيد معنوي) , verbal emphasis (توكيد لفظي) , conjunction (معطوف), possessive (مضاف اليه) , genitive noun (أسم مجرور) , specifier (تمييز) , exception (مستثني) , vocative (منادي) , circumstance (ظرف) , pronoun (ضمير) , particle ena(حرف ناسخ) , accusative particle (حرف نصب) , jussive particle (حرف جزم) , preposition (حرف جر) , exception particle (حرف استثناء) , coordinating conjunction (حرف عطف) , vocative particle (أداة نداء) , realization particle (حرف تحقيق) , diminishing particle (حرف تقليل) , punctuation (علامة ترميز) , particle (حرف) }.

Arabic cases: {nominative (مرفوع), accusative (المنصوبات), genitive (مجرور), jussive (مجزوم), and uninflected (مبني)}.
Arabic signs :{fatha (الفتحة) ,removing noun(حذف النون) , removing weak ending letter (حذف حرف العلة), kasra(الكسرة), damah (الضمة), sukun (السكون), waw and noun (الواو والنون), ya' and noun (الياء والنون), alef and noun (الألف والنون)}.

For each token in the sentence, knowing its POS tag, BP chunk and its morphological features like: token definiteness, we use a rule based system to determine the tag, case, and sign of each word in the sentence.

The grammar analyzer input and features could be characterized as follow:

Input: A complete sentence of Arabic words.

Context: The whole sentence.

Features: To extract the grammatical role of the words of the sentence, we use stemmer, POS tagger, BP chunker, and a morphological analyzer to extract extra morphological features of the words in the sentence.

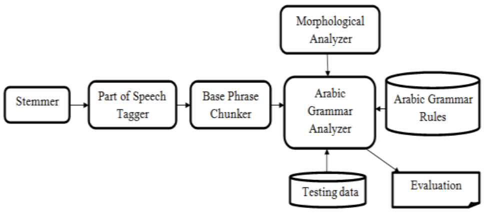

Fig. 1. Proposed framework architecture.

A. The Architecture of the Framework

The framework is presented in Fig. 1. The Arabic grammar analyzer module uses stemmer to separate proclitics and enclitics of the word. Then the POS tagger assigns an adequate POS tag to each token. Then, the base phrase chunker groups words belonging to the same phrases. Additional morphological information extracted for each word using the morphological analyzer. Finally, it applies the Arabic grammar rules to assign a tag, case and sign for each word.

B. Framework Components Description

1) Morphological analyzer

The morphological analyzer is based on BAMA-v2.0 (Buckwalter Arabic morphological analyzer version 2.0) [19], and it contains additional features like the extraction of the pattern of the word. For example, the pattern of (“كاتب”,”kAtb”) is (“فاعل”,”fAEl”) and the pattern of (“مكتب”,”mktb”) is (“مفعل”,”mfEl”). Also, it could be used to extract the root of the word. For example, the root of (“كاتب”,”kAtb”) is (“كتب”,”ktb”) and the root of (“مكتب”,”mktb”) is (“كتب”,”ktb”). Also, the morphological analyzer is developed to determine if a word is definite or not, is masculine or feminine, is plural or dual or singular.

2) Stemmer

The stream of characters in a natural language text must be broken up into distinct meaningful units (or tokens) before any language processing. The stemmer is responsible for defining word boundaries, demarcating clitics, multiword expressions, abbreviations and numbers.

In this task, the classifier takes an input of raw text, without any processing, and assigns each character the appropriate tag from the following tag set {B-PRE1, B-PRE2, B-WRD, I-WRD, B-SUFF, I-SUFF}. Where I denotes inside a segment, B denotes beginning of a segment, PRE1 and PRE2 are proclitic tags, SUFF is an enclitic, and WRD is the stem plus any affixes and/or the determiner Al. These tags are similar to the tags used by Diab *et al*. [20].

The classifier training and testing data could be characterized as follow:

Input: A sequence transliterated Arabic characters processed from left-to-right with break markers for word boundaries.

Context: A fixed-size window of -5/+5 characters centered at the character in focus.

Features: All characters and previous tag decisions within the context, and the characters corresponding to the word patterns with the context.

3) Part of speech tagger

POS tagging represents the task of marking up a word in a text as corresponding to a particular part of speech, based on both its definition, as well as its context. There are basically two difficulties in POS tagging. The first one is the ambiguity in the words, meaning that most of the words in a language have more than one part of speech. The second difficulty arises from the unknown words, the words for which the tagger has no knowledge about.

In this task, the POS tagger takes an input of tokenized text, and it assigns each token an appropriate POS tag from the Arabic Treebank collapsed POS tags, which comprises 24 tags as follows: {ABBREV, CC, CD, CONJ+NEG PART, DT, FW, IN, JJ, NN, NNP, NNPS, NNS, NO FUNC, NUMERIC_COMMA, PRP, PRP$, PUNC, RB, UH, VBD, VBN, VBP, WP, WRB}.

The classifier training and testing data could be characterized as follow:

Input: A sequence of transliterated Arabic tokens processed from left-to-right with break markers for word boundaries.

Context: A window of -2/+2 tokens centered at the focus token.

Features: Every character N-gram, N<=4 that occurs in the focus token, the 5 tokens themselves, POS tag decisions for previous tokens within context, and the patterns of the words within the context.

4) Base phrase chunker

Chunking represents the task of recovering only a partial amount of syntactic information to identify phrases from natural language sentences It is the process of grouping consecutive words together to form phrases, also called Shallow parsing Chunking does not provide information on how the phrases attach to each other. The structures generally specified by shallow parsers include phrasal heads and their immediate and unambiguous dependents and these structures are usually non-recursive.

In this task, the BP Chunker takes an input of tokenized text, and it assigns each token an appropriate Base Phrase Chunk tag from the Arabic Treebank collapsed BPC tags . Nine types of chunked phrases are recognized using a phrase BIO tagging scheme, Inside (I) a phrase, Outside (O) a phrase, and Beginning (B) of a phrase. The 9 chunk phrases identified for Arabic are PP, PRT, NP, SBAR, INTJ, and VP. Thus the task is a one of 12 classification task (since there are I and B tags for each chunk phrase type except PRT, and a single O tag).

The classifier training and testing data could be characterized as follow:

Input: A sequence of transliterated Arabic tokens processed from left-to-right with break markers for word boundaries.

Context: A window of -2/+2 tokens centered at the focus token.

Features: Every character N-gram, N<=4 that occurs in the focus token, the 5 tokens themselves, POS tag decisions for previous tokens within context and the previous Base phrase tag .

5) Arabic grammar rules databas

It consists of about four hundred Arabic grammar rules, when applied to the sentence after the extraction of the features like: POS tag, BP tag, and the pattern; it will assign a tag, a case and a sign to each token in the sentence. After the execution of all the rules, if some tokens remain without a tag, they will be given a default one. As Example of Arabic grammar rule: any noun after a preposition is a genitive noun. Another example of the grammar rules: any noun after a vocative particle is a vocative.

VI. Evaluation of the Framework

For the evaluation of the Bel-Arabi Advanced Arabic grammar analyzer, first the data used for the evaluation will be discussed, then the evaluation measures and results used will be discussed.

A. The Evaluation Data

For the evaluation of this framework, we have generated 600 sentences. The 600 sentences consist of 3452 tokens. The sentences lengths, tags, cases and signs are distributed as shown in Table II and Table II respectively.

TABLE III: Grammar Analysis Test Sentences Length Distribution

Sentence Length	Count
2	25
3	76
4	87
5	113
6	81
7	85
8	60
9	43
10	22
11	3
12	5

TABLE IV: Grammar Analysis Tags

Tag	Count
present verb	193
past verb	105
imperative verb	15
doer	191
direct object	227
subject	299
predicate	157
delayed subject	20
ena subject	51
ena predicate	35
kan subject	49
kan predicate	38
kan subject	26
apposition	147
adjective	155
conjuction	95
possessive	287
genitive noun	183
specifier	35
circumstance	66
pronoun	216
coordinating conjunction	101
particle	217
Other Tags	544

TABLE V: Grammar Analysis Cases

Case	Count
nominative	1081
accusative	557
jussive	58
genitive	602
uninflected	1154

TABLE VI: Grammar Analysis Signs

Sign	Count
No sign	1154
fatha	554
kasra	568
damah	1023
sukun	50
waw and noun	24
ya' and noun	31
alef and noun	9
removing noun	31
removing weak ending letter	8

B. The Evaluation Measure and Results

For the evaluation of Bel-Arabi, the following performance measures are calculated for the tag, the case and the sign.

$$\text{macro average precision} = \frac{1}{n}\sum_{i=1}^{n} precision(tag_i) \tag{1}$$

$$\text{macro average recall} = \frac{1}{n}\sum_{i=1}^{n} recall(tag_i) \tag{2}$$

$$\text{macro average}\, F_{\beta=1} = \frac{1}{n}\sum_{i=1}^{n} F_{\beta=1}(tag_i) \tag{3}$$

where *n* is the total number of tags.

$$\text{accuracy} = \frac{\text{number of true results}}{\text{number of true and false results}} \tag{4}$$

TABLE VI: GRAMMAR ANALYSIS TAGS RESULTS

Tag	Precision	Recall	F-measure
present verb	0.9645	0.9775	0.9710
past verb	0.9688	0.9630	0.9659
imperative verb	0.9556	0.6143	0.7478
doer	0.7248	0.8245	0.7714
direct object	0.7489	0.7816	0.7649
subject	0.9748	0.98753	0.9811
predicate	0.8834	0.9015	0.8923
delayed subject	0.9154	0.9042	0.9097
ena subject	0.9429	0.9659	0.9542
ena predicate	0.9375	0.9271	0.9322
kan subject	0.9460	0.9439	0.9449
kan predicate	0.87189	0.8918	0.8817
kan subject	0.9424	0.9294	0.9358
apposition	0.8938	0.8870	0.8903
adjective	0.9215	0.9485	0.9348
conjuction	0.8873	0.8723	0.8797
possessive	0.8762	0.8548	0.8653
genitive noun	1.0000	1.0000	1.0000
specifier	1.0000	0.7459	0.9544
circumstance	1.0000	0.9284	0.9628
pronoun	1.0000	1.0000	1.0000
coordinating conjunction	1.0000	0.9715	0.9855
particle	1.0000	0.9838	0.9918
Other Tags	1.0000	1.0000	1.0000

TABLE VIII: GRAMMAR ANALYSIS CASES RESULTS

Case	Precision	Recall	F-measure
nominative	0.9412	0.9298	0.9354
accusative	0.9415	0.9345	0.9379
jussive	0.9596	0.9667	0.9631
genitive	0.9537	0.9464	0.9500
uninflected	0.9915	0.9818	0.9866

TABLE VIII: GRAMMAR ANALYSIS SIGNS RESULTS

Sign	Precision	Recall	F-measure
No sign	1.0000	1.0000	1.0000
fatha	0.9521	0.9448	0.9484
kasra	0.9548	0.9593	0.9570
damah	0.9298	0.9334	0.9315
sukun	0.9650	0.9701	0.9675
waw and noun	1.0000	0.3237	0.4890
ya' and noun	1.0000	0.3346	0.5014
alef and noun	1.0000	0.2642	0.4179
removing noun	1.0000	0.4467	0.6175
removing weak ending letter	1.0000	0.2500	0.4000

The Detailed results of Bel-Arabi tags, parses, and signs are shown in Table VI -Table VIII respectively, the summary of the results are shown in Table IX

TABLE IX: SUMMARY OF RESULTS

	Macro-Avg. Precision	**Macro-Avg. Recall**	**Macro-Avg. F-score**	**Accuracy**	**Error**
TAGS	0.9567	0.9422	0.9504	93.33%	6.67 %
CASES	0.9575	0.9518	0.9546	94.09%	5.91 %
SIGNS	0.9801	0.6426	0.7230	94.49%	5.51 %

The overall accuracy of tokens that have correct tag, case and sign is 90.44% which is a good precision for this complex task.

VII. CONCLUSION

Arabic morphology poses special challenges to computational natural language processing systems. Its rich morphology and the highly complex word formation make computational approaches to Arabic very challenging. Also grammar analysis systems are complex and need extensive research and linguistic resources. In the proposed system we tried to highlight the most attractive property in Arabic language. The current results are promising, and the system could be further improved by adding extra grammar rules and adding a semantic analysis. The semantic analysis can be used to solve some type of ambiguity problems.

REFERENCES

[1] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. ICML*, pp. 282-289, 2001.

[2] M. Attia, "An ambiguity-controlled morphological analyzer for modern standard Arabic modeling finite state networks," in *Proc. the Challenge of Arabic for NLP/MT Conference*, London, pp. 155-159, 2006.

[3] M. Attia, "Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machine translation," PhD Thesis, University of Manchester, UK, 2008.

[4] A. Daoud, "Morphological analysis and diacritical Arabic text compression," *Computer Journal of the International Journal of ACM Jordan*, vol. 1, no. 1, pp. 41-47, 2010.

[5] N. Habash, O. Rambow, and R. Roth, "MADA + TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization," in *Proc. the 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*, Cairo, Egypt, 2009.

[6] N. Y. Habash, O. C. Rambow, and R. M. Roth, "MADA + TOKAN manual," 2010.

[7] Y. Benajiba and M. Diab. (2010). At the Center for Computational Learning Systems. Columbia University. [Online]. Available: http://nlp.ldeo.columbia.edu/amira/

[8] M. Attia, M. A. A. Rashwan, and G. Khallaaf, "A formalism of Arabic phonetic grammar, and application on the automatic Arabic phonetic transcription of transliterated words," 2004.

[9] M. Ahmed, "A large-scale computational processor of the Arabic morphology, and applications," A Master's Thesis, Faculty of Engineering, Cairo University, Cairo, Egypt, 2000.

[10] M. Attia and M. Rashwan, "A large-scale Arabic POS tagger based on a compact Arabic POS tags set, and application on the statistical inference of syntactic diacritics of Arabic text words," in *Proc. the Arabic Language Technologies and Resources Int'l Conference*, 2004.

[11] M. Attia *et al.*, "A compact Arabic lexical semantics language resource based on the theory of semantic fields," 2008.

[12] S. Green and J. de Nero, “A class-based agreement model for generating accurately inflected translations,” in *Proc. the 50th Annual Meeting of the Association for Computational Linguistics*, 2012.

[13] K. Toutanova, D. Klein, C. Manning, and Y. Singer, “Feature-rich part-of-speech tagging with a cyclic dependency network,” in *Proc. the HLT-NAACL 2003*, pp. 252-259, 2003.

[14] S. Green and C. D. Manning, “Better Arabic parsing: Baselines, evaluations, and analysis,” in *Proc. COLING*, 2010.

[15] M. Maamouri *et al*., “The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus,” in *Proc. NEMLAR Conference on Arabic Language Resources and Tools*, 2004.

[16] E. A. Daoud and A. Basata, “A framework to automate the parsing of arabic language sentences,” *Computer Journal of The International Arab Journal of Information Technology*, vol. 6, no. 2, pp. 191-195, 2009.

[17] N. Y Habash and R. M. Roth, “CATiB: The Columbia Arabic Treebank,” in *Proc. the ACL-IJCNLP 2009 Conference Short Papers*, Association for Computational Linguistics, 2009.

[18] K. Dukes and T. Buckwalter, “A dependency Treebank of the Quran using traditional Arabic grammar,” in *Proc. the 7th International Conference on Informatics and Systems (INFOS)*, Cairo, Egypt, 2010.

[19] T. Buckwalter, “Buckwalter Arabic morphological analyzer version 2.0. linguistic data consortium,” University of Pennsylvania, LDC Catalog, 2004.

[20] M. Diab, K. Hacioglu, and D. Jurafsky, “Automated methods for processing Arabic text: From tokenization to base phrase chunking,” in *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, A. van den Bosch and A. Soudi, Eds., Kluwer/Springer, 2007.

Michael Nawar Ibrahim was born in Egypt in 1990. He received his bachelor in computer engineering from the Faculty of Engineering, Cairo University. He is currently pursuing master’s degree from the Faculty of Engineering, Cairo University.

He is currently a teaching assistant in the Faculty of Engineering, Cairo University.

His areas of interests are natural language processing, machine learning, pattern recognition, and artificial neural networks.

Mahmoud N. Mahmoud was born in Egypt in 1990. He received his bachelor in computer engineering from the Faculty of Engineering, Cairo University. He is currently pursuing master’s degree from the Faculty of Engineering, Cairo University.

He is currently a teaching assistant in the Faculty of Engineering, Cairo University.

His areas of interests are natural language processing, machine learning, pattern recognition, and artificial neural networks.

Dina A. El-Reedy was born in Egypt in 1990. She received his bachelor in computer engineering from the Faculty of Engineering, Cairo University. She is currently pursuing master’s degree from the Faculty of Engineering, Cairo University.

She is currently a teaching assistant in the Faculty of Engineering, Cairo University.

Her areas of interests are natural language processing, machine learning, pattern recognition, and artificial neural networks.