International Journal of Computer Applications (0975 - 8887)
Volume 53 - No. 3, September 2012

Context-Free Grammar Analysis for Arabic Sentences

Shihadeh Alqrainy
Software Engineering Dept.
Albalqa Applied University
Salt - Jordan

Hasan Muaidi
Computer Science Dept.
Albalqa Applied University
Salt - Jordan

Mahmud S. Alkoffash
Software Engineering Dept.
Albalqa Applied University
Salt - Jordan

ABSTRACT

This paper presents a simple parser to parse Arabic sentences. The aim of this parser is to check whether the syntax of an Arabic sentence is grammatically correct or not by constructing new efficient Context-Free Grammar that makes Top-Down technique much valuable. A set of experiments were ran on a dataset contains 150 Arabic sentence. The system achieved an average accuracy of 95%.

KEYWORDS

Natural Languages Processing, Arabic Language Processing, Parser, Context-Free Grammar, Parse Tree, Top-Down Parser

1. INTRODUCTION

In computer science, Natural Language Processing (NLP) deals with analyzing, understanding and generating the human languages in order to interface with computers in both written and spoken contexts using natural human languages.

Building a generic parser system (as one of NLP tasks) for Arabic language is a daunting; not an easy task due to the difficulty and complexity in both rich morphological and syntactical system the Arabic language own.

Parsing (syntactic analysis) is the process of mapping the sentence (string of words) to its parse tree. To do that, an efficient Context-Free Grammar (CFG), which defines the language, is extremely necessary step. Furthermore, a robust syntactical analysis system to check whether the parser input sentence may generate by a given CFG is also very important step, which requires an efficient Part-Of-Speech (POS) tagging system to assign the syntactic category (noun, verb, and particle) to each word in the input sentence [6][17].

This paper presents a simple parser to parse Arabic sentences. The aim of this parser is to check whether the syntax of an Arabic sentence is grammatically correct by constructing a new efficient Context-Free Grammar that makes Top-Down technique much valuable.

The paper starts with a brief summary of Arabic parsing related work. Arabic Context Free Grammar (CFG) is highlighted. Parsing with (CFG) including parse tree in more details also discussed. Finally, we present the experimental results and conclusion.

2. RELATED WORK

Many different approaches and a variety of ways related to Arabic Parsing have been done. Some of these works are listed below:

McCord et al. [9] presented their work on building an efficient parsing system that works on Arabic Language using a bottom up chart parser. Tounsi et al. [18] have used Treebank-based parsers and automatic LFG f-structure annotation methodologies to create a parser over Arabic language. Bataineh and Bataineh [4] developed new parser that uses the mechanism of recursive transition networks. Chiang et al. [5] highlighted this problem of parsing dialect language vs the Modern Standard Arabic (MSA).

Other attempts were made to develop the parser for Arabic such as the work done by [1][16][11][12]. However, the current literature in the field of Arabic NLP (particularly Parsing Systems) shows very few attempts have been done in developing an efficient parsing system for Arabic. Many reasons lie behind the lack of research on the Arabic language. A richly inflected and a complex morphological system that Arabic exhibits on one hand, and the lack of resources such as the availability of large manually tagged Arabic corpus on the other hand may constitutes the main reason behind the lack of research on the Arabic language [2].

3. ARABIC CONTEXT FREE GRAMMAR

Arabic language is the prominent member of Semitic languages family. It is written from right to left. The Arabic alphabet consists of 28 letters that change shape depending on their position within a word and the letters by which they are surrounded. Some Arabic letters must be connected to other letters; others may stand alone. Additionally, there are no special forms, such as the use of capital letters in English [19].

Arabic has a rich and complex morphological system and syntactical as well. It is highly inflectional and derivational language. This make parsing Arabic sentences is a daunting, not an easy task, due to fact that some of Arabic sentences are too long in terms of sentences words. The average length may exceed 60 words. Additionally, the free word order nature of Arabic sentences from one hand and the presence of an elliptic personal pronoun from other hand increase the difficulty not only for parsing system, but also for building an efficient context free grammar (CFG) [1].

A grammar in human language represents understandable specification of language syntax. It is not concern with semantic. On other word, the grammar is collection of rules that describes well-informed sentences in a language. Furthermore, context free grammar in natural languages represents a formal system which describes a language by specifying how any legal text can be derived from a distinguished symbol called the sentence symbol [15].

In this paper, we introduced a context free grammar (CFG) which for the sake of simplicity developed to cover most valid sentences over Arabic Language. Figure 1 shows the CFG which

is extremely necessary and pre-request step for any parsing system.

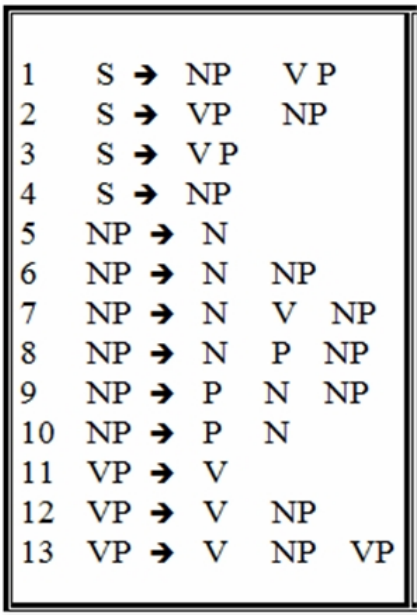

Fig. 1. Context-Free Grammar (CFG)

The main component of any CFG is a set of production rules. For example VP ⇒ V NP represents one of the above CFG production rules. Furthermore, it is clear that the recursive nesting of phrases can be easily done in all formal languages that can be generated by a CFG [15].

Arabic language as many other natural languages has nominal (NP) and verbal sentences (VP). It well known that nominal sentences begin with noun while verbal begin with verb. The following section describes the parse tree and the parsing system in more details.

4. PARSING WITH CFG

4.1 PARSE TREE

The main goal of a parse tree is to show the hierarchical structure of the language. On other word, the parse tree is a graphical representation of a derivation which represents the hierarchical structure of the language [15][17].

In this paper, the parse tree for some Arabic sentences which have been used to test the parsing system is generated by using Natural Language Toolkit (NLTK) recursive descent (Top-Down) parser [14] with an update in its source code to deal with Arabic sentences in a correct way. Figure 2 shows the parse tree for the following Arabic sentence as an example:

EXAMPLE 1. اركب السيارة البيضاء *[Take the white car]*

The grammatical production (which extracted from CFG shown in Figure 1) for the above sentence is as follows:

- S ⇒ VP
- VP ⇒ V NP
- NP ⇒ N
- NP ⇒ N NP

The interpretation of the CFG described in Figure 2 requires syntactic analysis or parsing. Table 1 presents the syntactic analysis (category) of Arabic sentence of Example 1.

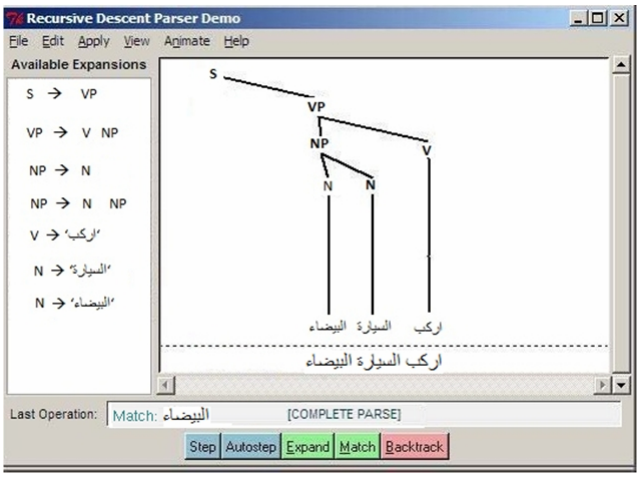

Fig. 2. Parse Tree for the Arabic Sentence of Example 1

Table 1. The Syntactic Category of Arabic Sentence of Example 1

Symbol	Meaning	Example
S	Sentence	اركب السيارة البيضاء
VP	Verb Phrase	اركب
NP	Noun Phrase	السيارة البيضاء
V	Verb	اركب
N	Noun	السيارة
N	Adjective	البيضاء

The main task of syntactical analysis is to check whether an input sentence can be generated by a given grammar or not. When the answer is no, the program does not generate any parse tree and it displays "the sentence is syntactically incorrect" as an output [13].

The syntactical analysis requires lexical (syntactical) information for each word in the sentence that needs to parse, such this information usually obtained from the output of a Part-Of-Speech (POS) tagger. POS tagging system is an important first step and an integral part for any parsing system. The aim of POS tagging system is to assign the lexical category (N: noun, V: verb: P: particle) to each word in the parsing sentences.

Since a parser must be tested using annotated data sets (annotated corpus), a POS tagging system called AMT, developed by the first author [2] has been used to assign the correct POS general tag to each word in the parsing sentences, especially those words belonging to noun (N) or verb (V) category. In addition, a complete list of Arabic stopwords including the Arabic particles has been compiled by the second author [10] and also used to tag those words belonging to particle (P) class. Figure 3 shows how AMT perform tagging while a screenshot of AMT tagger is shown in Figure 4.

The AMT tagger used pattern-based approach which depends on the pattern of the word to assign the correct tag either noun (N) or verb (V). Lexical and Contextual rules also have been used to tag those words when the pattern module fails to assign the correct tag to a given word or token. In fact, AMT tagger produced detailed tag to each word in the raw text including the inflectional feature of the word such as person, gender, number, etc. In this paper, we have used only the three main general tags (N: (for all types of noun), V: verb, P: (for all types of particle)) to each word in testing corpus. Further information about AMT tagger is available at [2].

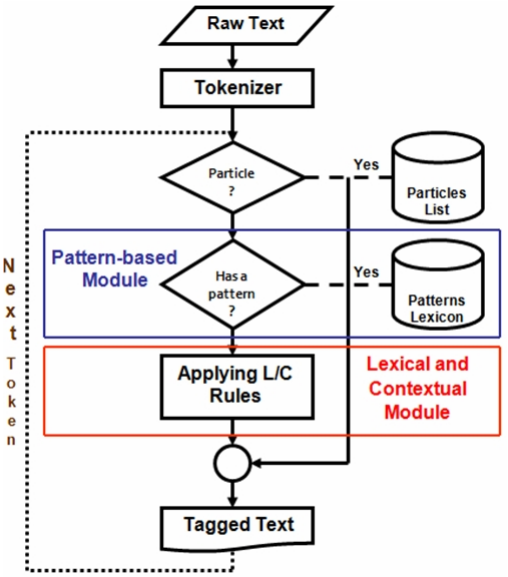

Fig. 3. How AMT Performs Tagging

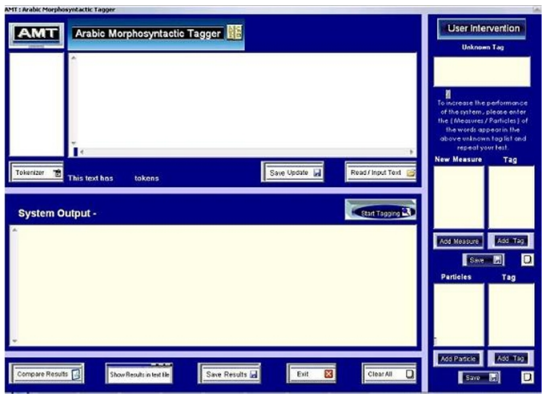

Fig. 4. A Screenshot of AMT Tagger

4.2 PARSER

Building a CFG is not sufficient to determine whether a given sentence belong to the language that the CFG defines, this is not the main goal of a CFG. It's only defines a language [7]. The parser task is to map a sentence (string of words) to its parse tree. Once the mapping is correctly done, the sentence is definitely belonging to the language that the CFG defines, and the task of the parser is completely successful.

Top-Down is one of the parsing strategies. This approach builds a parse from the starting symbol (S). The goal of Top-Down parsing system is to parse the input sentence according to the given grammar productions. It picks a production rule and tries to match the input sentence words [15].

To build a parse, the following steps should be repeated until the fringe of the parse tree matches the input sentence (string).

(1) At the Start node S, Select a production with S on its left hand side and for each symbol on its right hand side, construct the appropriate child.

(2) When a terminal is added to the fringe that does not match the input string, then backtrack.

(3) Find the next node to be expanded.

If the parse tree did not match the input sentence (string) then it means that input sentence has a syntax error with respect to grammatical production that is whitened [3].

In this paper, we have used NLTK recursive descent (Top-Down) parser [14] to test our compiled data set. The NLTK recursive descent (Top-Down) parser (nltk.parser) is one of the natural language toolkit task modules. This module encompasses to the task of parsing, or deriving the syntactic structure of a sentence [14][8]. The NLTK was developed in conjunction with a computational linguistics course at the University of Pennsylvania in 2001. It has many modules such as parser, stemmer (porter), tokenizer, corpus, chart, sense, etc. NLTK is an open source project. We have used only parser module with an update in its source code to test our compiled testing corpus over Arabic language. A screenshot of (NLTK) recursive descent (Top-Down) parser described how to parse the following Arabic sentence "جامعة البلقاء التطبيقية" is shown in Figure 5.

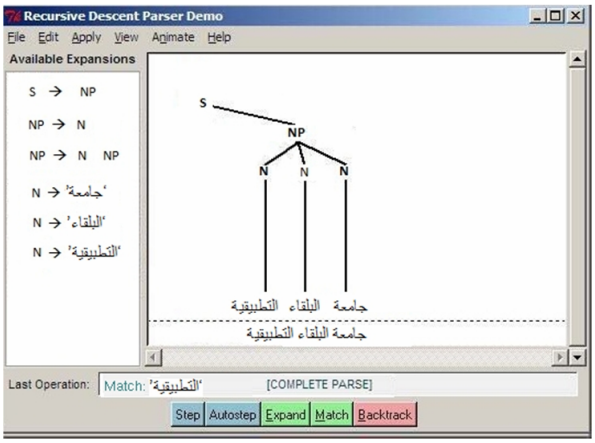

Fig. 5. A Screenshot of NLTK Parser

5. EXPERIMENTAL RESULTS

An Arabic corpus to test NLTK recursive descent (Top-Down) parser has been compiled. It contains 150 Arabic sentences. The corpus is chosen and extracted from Arabic documents. Some of these sentences are shown below.

ضرب الولد صديقه العزيز	اشترى محمد سيارة جميلة	رأى صديقي حلما مزعجا
صنع النحات تحفة رائعه	قتل المجرم الرجل العجوز	عمل المزارع يوما طويلا
استلم الموظف معاملة مهمة	كتب التلميذ فقرة جديدة	علم المدرب المتدرب المبتدئ
غنى المطرب لحنا جميلا	رأت الفتاة حلما مزعجا	قال الشاعر قصيدا جميلا
قال الشيخ موعظة حسنة	مشيت اليوم مسافة طويلة	رسم الفنان لوحة جميلة
عمل خالد ساعات طويلة	شاهد علي فلما مثيرا	رأى الرجل مشهدا مريبا
وجد محمد حقيبة كبيرة	أكل معاذ حلوى شهية	اشترى علي بيتا كبيرا
استحسن العميل المعاملة اللطيفة	افتتح الوزير منشئة ضخمة	اكل السمين وجبة كبيرة
باع المزارع تفاحا اصفرا	نام ثائر نوما عميقا	شرب الطفل حليبا طازجا
تبعد المدينة مسافة بعيدة	تملك طارق مزرعة غربية	ذكر عاصم صحابيا جليلا
اشترى المركز طابعة كبيرة	سرق الرجل محفظة ثمينة	شكر المعلم الطالب النجيب
ركب وائل حافلة سريعة	نفذ الموظفون اعتصاما جماهيريا	يقطن السكان جبلا منيفا
نقل الرسول رسالة شفوية	اجتمع القادة اجتماعا مهما	اشترى التلميذ قلما جميلا
كرم المدير الموظف المميز	زار السائح منطقة اثرية	يعبد المسلمون الاها واحدا
أكل القرش سمكة صغيرة	زار التلاميذ محمية طبيعية	يشاهد هاشم منظرا خلابا

The testing corpus has been annotated (tagged) using AMT tagger that produced the three main general tags (N: noun, V: verb, P: particle) to each word in testing corpus. A set of experiments were ran to test the performance of the NLTK parser system. The

results were excellent in all experiment scenarios for the various sizes of sentences.

The testing corpus divided into two data sets. The first dataset contains 103 verbal sentences while the second contains 47 nominal sentences. The first experiment was performed on the first dataset. NLTK parser correctly parses 92% of set-1 as shown in Figure 6.

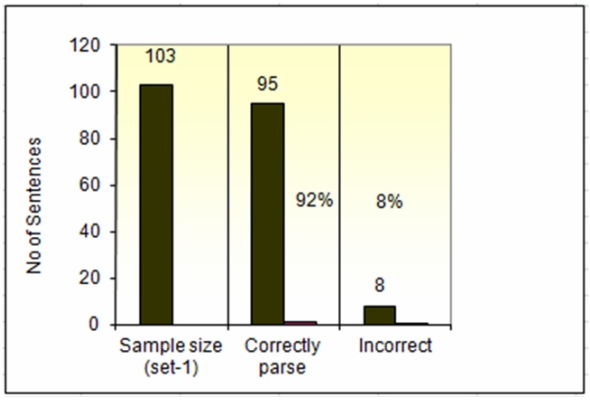

Fig. 6. Success Rate of Experiment-1

The second experiment was performed on the second data set. NLTK parser correctly parses 98% of set-2 as shown in Figure 7.

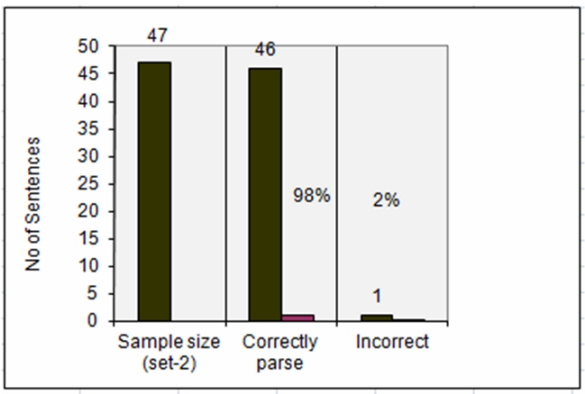

Fig. 7. Success Rate of Experiment-2

The performed experimental results showed the effectiveness employing the proposed CFG and the NLTK parser for analyzing both verbal and nominal sentences. Despite the excellent results the system achieved, some sentences were not parsing correctly due to many reasons:

- Not all the string words were correctly tagged.
- Some sentences did not match a correct production rules.
- The order of some sentences is too difficult (eg : sentence contains two prepositions).

6. CONCLUSION

In this paper, we described a Context-Free Grammar which developed to cover most valid sentences over Arabic Language. NLTK parser that uses Top-Down technique to check whether the syntax of an Arabic sentence is grammatically correct also discussed. NLTK is a broad-coverage natural language toolkit that provides a simple, extensible demonstrations and projects. An overview of AMT tagger which usually used to produces the lexical information that the parser need also highlighted. Future work will focus on expanding the developed CFG to deal with most Arabic sentences.

7. REFERENCES

[1] Ahmad T. Al-Taani, Mohammed M. Msallam, and Sana A. Wedian. A top-down chart parser for analyzing arabic sentences. *Int. Arab J. Inf. Technol.*, 9(2):109–116, 2012.

[2] Shihadeh Alqrainy. *A Morphological-Syntactical Analysis Approach For Arabic Textual Tagging*. PhD thesis, De Montfort University - UK, 2008.

[3] L. Bala, S. Ishwar, and S. Kumar. Context free grammar for natural language constructs - an implementation for venpa class of tamil poetry, 2003.

[4] Bilal M. Bataineh and Emad A. Bataineh. An efficient recursive transition network parser for arabic language. In *Proceedings of the World Congress on Engineering 2009 Vol II, WCE '09, July 1 - 3, 2009, London, U.K.*, Lecture Notes in Engineering and Computer Science, pages 1307–1311. International Association of Engineers, Newswood Limited, 2009.

[5] David Chiang, Mona Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. Parsing arabic dialects. In *Final Report, 2005 JHU Summer Workshop*, 2005.

[6] J. Earley. An efficient context-free parsing algorithm. *Communications of the ACM (CACM)*, 13(2), February 1970.

[7] Antony P J, Nandini. J. Warrier, and Dr. Soman K P. Article:penn treebank-based syntactic parsers for south dravidian languages using a machine learning approach. *International Journal of Computer Applications*, 7(8):14–21, October 2010. Published By Foundation of Computer Science.

[8] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, 2002.

[9] M. Mccord and V. Cavalli-Sforza. An arabic slot grammar parser. In *Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, 2007.

[10] Hasan Muaidi. *Extraction of Arabic word roots: An Approach Based on Computational Model and Multi-Backpropagation Neural Networks*. PhD thesis, De Montfort University - UK, 2008.

[11] E. Othman, K. Shaalan, and A. Rafea. A chart parser for analyzing modern standard arabic sentence. In *Proceedings of the MT Summit IX Workshop on Machine Translation for Semitic Languages: Issues and Approaches, USA*, 2003.

[12] R. Ouersighni. Towards developing a robust large-scale parser for arabic sentences. In *Proceedings of the International Arab Conference on Information Technology*, 2008.

[13] D. Rao, P. Bhattacharyya, and R. Mamidi. Natural language generation for english to hindi human aided machine translation. In *International Conference on Knowledge Based Computer Systems (KBCS 1998), Mumbai, December*, 1998.

[14] Recursive Descent Parser. http://nltk.googlecode.com/svn/trunk/doc/api/nltk.app.rdparserapp-module.html, 2012.

[15] B. Sagar, G. Shobha, and R. Kumar. Context-free grammar analysis for simple kannada sentences. In *International Conference [ACCTA-2010], August*, 2010.

[16] K. Shaalan, A. Farouk, and A. Rafea. Towards an arabic parser for modern scientific text. In *Proceeding of the 2nd Conference on Language Engineering, Egyptian Society of Language Engineering, Egypt*, 2010.

[17] Lee Spector, Kyle Harrington, and Thomas Helmuth. Tag-based modularity in tree-based genetic programming. In *GECCO '12: Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, pages 815–822, Philadelphia, Pennsylvania, USA, 7-11 July 2012.

[18] Lamia Tounsi, Mohammed Attia, and Josef van Genabith. Parsing arabic using treebank-based lfg resources. In *Proceedings of the LFG Conference*, 2009.

[19] Transparent Language. http://www.transparent.com/.