International Journal of Computing and Network Technology
ISSN (2210-1519)
Int. J. Com. Net. Tech. 4, No. 3 (Sep-2016)

A Survey of Lexical Functional Grammar in the Arabic Context

Said A. Salloum [1, 2], **Mostafa Al-Emran** [1] **and Khaled Shaalan** [2]

[1] *Al Buraimi University College, Al-Buraimi, Oman*
[2]*The British University in Dubai, Dubai, UAE*

Received: 20 Jun. 2016, Revised: 15 Aug. 2016, Accepted 20 Aug. 2016, Published: 1 Sep. 2016

Abstract: Lexical Functional Grammar (LFG) plays a vital role in the area of Natural Language Processing (NLP). LFG is considered as the constraint-based philosophy of grammar. C-structure and F-structure are the two basic forms of LFG. We have perceived from the existing literature that LFG has not studied in details; the reason that encouraged us to work on this study. This study highlights the brief history of LFG along with its architecture. Arabic language along with its parsing techniques is demonstrated. Moreover, this study addresses the efforts that LFG played in resolving various NLP issues. New trends have been triggered while conducting this survey and have been demonstrated for pursuing further research.

Keywords: LFG, NLP, Arabic, parsing.

1. Introduction

Natural Language Processing (NLP) is one of the attractive research fields that focus on how computer machines process, analyze and intrepret human-being langauges for developing effective applications [1]. Lexical Functional Grammar (LFG) is one of the hotest areas in the field of NLP. LFG includes two basic forms: c-structure and f-structure [2], [3]. Differences in languages may occur in its structural representation, while it may keep using identical syntactic functions. Arabic is rich in its morphology and sophisticated in its syntactic structure [3]. Arabic sentences are characterized by a group of distinct features that cause parsing Arabic sentences to be a very difficult and challenging task [4]. Developing an Arabic parsing system is not an easy process due to the language complexity and morphological richness. Different techniques have been developed by several scholars like [3], [5], [6], [7], [8], [9], [10] for resolving Arabic parsing issues. Moreover, terminologies like Context-free Grammar (CFG) and Definite Clause Grammar (DCG) are also discussed.

The paper first gives an overview of Lexical Functional Grammar (LFG), LFG Architecture, parsing and Arabic language in section 2. Section 3 addresses the efforts of LFG in resolving several NLP issues. Conclusion and further research implications are discussed in section 4.

2. Background

LFG is considered as one of the well-known terminology in the writing of grammars for any language. This section gives a comprehensive background about LFG and its architecture. An overview about parsing and Arabic langauge is also demonstrated.

A. *Lexical Functional Grammar (LFG)*

Lexical Functional Grammar (LFG) is a linguistic hypothesis of grammar which concerns the nature of the statement structure and generate realistic framework for natural language processing [11], [12]. LFG has been invented by Joan Bresnan in 1970 and it has given the sentence structure formalism intended for typologically in various natural languages such as: Europe, Australia, Africa, South and East Asia processing [13].

B. *LFG Architecture*

LFG distinguishes two levels of representation to each sentence of the language, this approach presents these two completely different formalisms: trees form or Constituent

E-mail address: 2015328028@student.buid.ac.ae, malemran@buc.edu.om, Khaled.shaalan@buid.ac.ae

structure (c-structure) and functional structure represent grammatical functions like subject and object and the relation between them as attribute-value matrices (f-structure). Figure 1 shows both the *c* and *f* structures for the sentence “***Ayham eats”.***

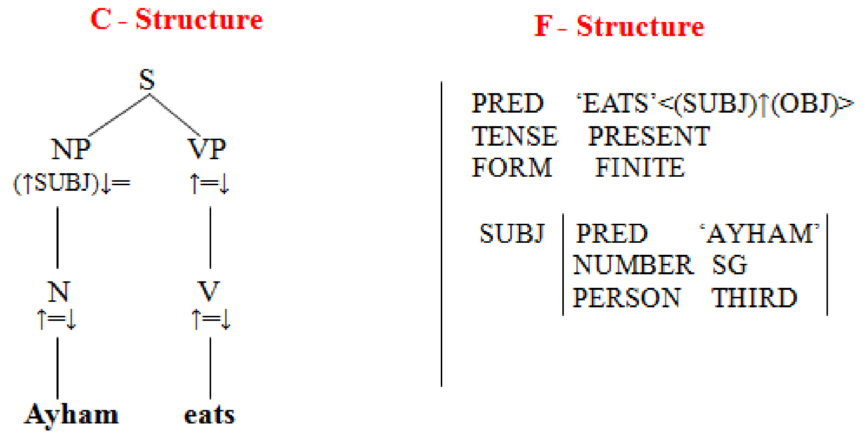

Figure 1: C-Structure and F-structure for the statement "*Ayham eats*".

C. *Parsing*

Parsing is the analysis of an input sentence into its constituents, resulting in a parse tree showing their syntactic relation to each other, which may also contain semantic and other information. Traditional sentence parsing is often performed as a method of understanding the exact meaning of a sentence. There are different Arabic parser forms for using Arabic Treebank resources such as Bikel parser, Maltparser, Stanford parser, and Attia’s rule-based parser. Morphological processing is considered as a challenging task when parsing morphologically rich languages such as Arabic [9]. Arabic sentences are characterized by a group of distinct features that cause parsing Arabic sentences to be a very difficult and challenging task, the difficulty comes from various reasons such as: ambiguous or inaccurate parsing can occur if one fails to address these complicated features such as complex sentence structure, free word order, length of sentences, existence of elliptic personal pronoun, diacritics (vowels), and punctuation omission [4].

D. *Arabic Language*

Arabic is the native spoken and written language for more than 330 million people who are distributed among several parts on the globe. Furthermore, more than 1.4 billion Muslims around the world use Arabic language to perform their prayers every day. It contains 28 letters and it has been written from right to left with various formats. Arabic language is considered as both *interesting* (in terms of culture, religion and history) and *challenging* (in terms of its complexity) [14]. The complexity of Arabic language is recognized by considering both *conservative*, *templatic* as well as many of its random forms in a few morphological features [15]. Both complexity and morphological features provide an opportunity for the Arabic language to play a considerable and challenging role in NLP. Arabic is Semitic language in which morphology is considered to be one of the main elements that causes the language as to be a very derivational and well-structured language [3].

E. *Modern Standard Arabic*

The original roots of Arabic language came from the Holy Qura'an, so-called Qura’anic or Classical Arabic, which have been developed across various countries and came to be known as Literary Arabic or Modern Standard Arabic (MSA) [14]. Nowadays, MSA has become the standardized linguistic Arabic that is used in an official spoken occasions, (such as conferences and lectures) and in official documents (such as books, magazines, newspapers). In MSA, there is no orthographic representation such that Arabic NLP tasks require a higher disambiguation degree [16]. Moreover, a linguistic resource in MSA would be rich in both Arabic NLP and Penn Arabic Treebank annotations.

3. LFG EFFORTS IN RESOLVING VARIOUS NLP ISSUES

LFG plays a crucial role in Natural Language Processing (NLP) as it gives formalism for representing syntactic knowledge for any language. Moreover, LFG serves as a device for explaining and expressing the syntax of various natural languages [17]. This section contains the efforts that LFG plays in various NLP issues.

*A. **Resolving Ambiguity in Arabic Sentence***

A study by [18] stated that the Arabic language does not use diacritics when writing vowels and thus makes the language unclear and slows down its development of Arabic Natural Language Processing (ANLP). In case the ambiguity is resolved, the range of possible interpretations will be reduced and the language would become clearer. The way to resolve ambiguity will be influenced by certain linguistic constraints while parsing an Arabic sentence. These hindrances are heuristics based on grammar to ensure correct formation. Syntactic analysis system has been developed for Arabic language including three NLP elements: a lexicon, a morphological analyzer and a syntactic parser. As a result of applying disambiguation approach that based on the parser and analyzer, the morphology analyzer gives all the probable readings of the given Arabic word. This would become clearer by adhering to the grammar rules which would ensure correct parse and resolve ambiguity [18].

*B. **Definite Clause Grammar***

A study by [19] declared that Arabic language is different from other languages; the use of the grammar in Arabic is presented only in descriptive form. Various efforts are attempted to formalize the Arabic sentences [20], [21], [22] such as LFG model [23], dependency grammar and functional grammar. Nevertheless, this issue is still a big debate. A formal description of Arabic syntax has been developed by [19] in Definite Clause Grammar. It has been developed in prolog and implemented in syntactic analyzer. The argument in this grammar of non-terminals are hold for a better understanding of many structures which in turn will increase the ability of the Definite Clause Grammar for understanding the context.

C. Parsing Arabic Sentences

Making up an analytic system or Generic Parser System for Arabic is not an easy task because of the complexity and difficulty of the language known for its rich morphological and syntactical system. Several efforts have been resulted in different models and different methodologies for the development of Arabic parsing.

A study by [24] developed an efficient statistical parser that uses 40,000 sentences as a training set and 2416 sentences as a testing set via the Penn Treebank features. The developed parser input tagged sentences and creates an output of phrase-structure tree form. This parser attempted to create a parse tree between the set of words relying on dependencies' probabilities.

[6] built an efficient parsing system for Arabic language. The system uses a bottom-up chart parser. Another study by [3] has developed an Arabic parser that is based on both (Treebank along with the automatic LFG f-structure annotation methodologies). [7] developed a parser based on the usage of recursive transition networks. In spite of all of these attempts, we have to admit that only few researchers attempted to develop an efficient parsing system for Arabic. Such a lack of research can be understood if one takes into consideration the rich and complex morphological system and the lack of resources.

A study by [8] has developed a simple parser to parse Arabic sentences which aims to check if the syntax of the given Arabic sentence is grammatically correct or not through building new efficient Context-Free Grammar which makes the Top-Down techniques much more valuable. Many experiments were conducted using a dataset of 150 Arabic sentences. The system scored 95% on accuracy level. A series of experiments were conducted for examining the NLTK parser performance. Excellent results were reported in all the conducted experiments scenarios for different sentence sizes. Results revealed efficient outcomes while analyzing the nominal and verbal sentences via the development of both (CFG and NLTK parser).

As argued before, developing parsers for Arabic was not done on a large scale. Many scholars in Arabic NLP systems focused on morphological analysis [25], [26]. [27] discussed the problem of implementing a morphological analyzer for inflected Arabic vocabulary. [28] integrated both the developed morphological analyzer and the Arabic parser and presented their work on developing an efficient chart parser system that creates a parse tree representing the syntactic structure of the Arabic sentence. Therefore, the parser is able to satisfy the syntactic constrains reducing parsing inexactness using features related to the words or vocabulary of Arabic.

[5] has developed an Arabic parser for modern scientific text. This paper focused on the design and implementation of Arabic parser issue. This parser was built to be a part of a Machine Translation System and was written in DCG (Definite Clause Grammar). It was developed in two phases. Phase One includes acquiring the rules that form the grammar as a whole for Arabic and which will give an acute account of a grammatically-correct sentence. Sentences were driven from the field of agricultural documents. Phase Two comprises the actual implementation of the parser along with the parser assigning grammatical structure to input sentences. The

parser encodes the rules of Arabic grammar of "***I'rab***" "***الإعراب***" and the effects of applying such rules on the components of sentences. The parser was built as a module, which means that it can be used for any other related systems or applications. When designing this parser, problems of ambiguities were avoided as much as possible. Ambiguity can be semantic and hence the resolving process can't be comprehensive, and this is another research problem. Experiments were conducted on real extension document and the results were satisfactory.

[29] offered a new mechanism that uses mainly three parsers depending on two major methodologies: *parser switching and parser hybridization* in order to achieve new results with high precision of parsing to reduce the individual parsers' bugs as comparing with other techniques; the three compound parsers have been tested and trained on Penn Treebank.

[28] highlighted the development of a chart-parser that uses MSA sentences. The developed parser uses syntactic constraints to minimize the ambiguity of parsing through using some features in lexical semantic that is mainly utilized to solve the structure of ambiguous sentences. Prolog language has been used to implement the developed parser and has the capability to assure syntactic constraints.

[3] has used two methodologies to state the parsing of Arabic by using Treebank-based parsers and automatic Lexical Functional Grammar (LFG) that applies f-structure annotation method. Using feature of Arabic Annotation Algorithm (A^3) has accepted and used the PATB functional annotations in order to embed f-structure with parse tree. An effective change was performed on the Bikel's parser in order to merge the phrasal group and to find out the PARB functional tags by selecting the training data that covers functional tags. After applying this technique, result has shown 77% as a dependency f-score.

[9] developed a methodology that produces Arabic sentences parse trees and specifically the Qur'anic sentences were proposed through the usage of NLTK. The defined process consists of building a lexicon, a context-free grammar and using the NLTK recursive-descent parser. The produced parse trees are considered as Treebank components. This approach can be further used in many other ways because the integration of a morphological analyzer and a parser simply automate the process. Top-Down (Recursive Descent) parsing algorithm was applied to parse Arabic sentences through the usage of NLTK.

A recent study by [10] developed a system that parses Modern Standard Arabic (MSA) sentences through the use of treebank resources. The developed system takes an Arabic sentence as input and produces the parsed tree for that sentence based on the built model that has been created within the training stage. Experiments results pointed that the system achieved a score of 82.4 % for precision, 86.6 % for Recall and 84.4 % for F-measure.

D. Arabic LFG Dependency Structures

[3] described that statistical parsers on Treebanks researches try to get more support than those that utilize handcrafted grammar. The observed weakness points are being unable to mark grammatical and pragmatic properties that require certain meaning applications. It has been noticed that the importance of supplying weak parsers with deep subordinated data is considered as weak points. Information encoded in the Penn-II Treebank (PTB) trees could be utilized automatically in order to annotate on each node in the tree along with its LFG f-structure from which grammar resources were removed and used in both generation and parsing. This approach was applied to various languages such as: German, Chinese, Spanish, and French. Moreover, LFG acquisition has been applied to Arabic and Penn Arabic Treebank. Results revealed that most of ATB trees generate full connected f-structure. Qualitative evaluation has been conducted using the gold standard set and achieved an f-score of 95%.

E. The Penn Arabic Treebank (ATB)

The study of the Penn Arabic Treebank (ATB) began in 2001 to explain Modern Standard Arabic news. The Treebank contains 23611 sentences [30], [31]. The ATB annotation makes use of empty nodes concept, and traces to mark long dependencies such as relative clauses and questions. Empty nodes after the verb marked with a -SBJ functional tag, appear in the ATB annotation, the matter that adapts Arabic which is a subject pro-drop language that allow a null category (pro) in the subject position of a finite clause [32].

F. Automatic annotation of the Penn treebank with LFG

[2] stated that LFG f-structures are considered as abstract syntactic forms bordering basic subject argument structure. F-structure information had been annotated in Treebank, required as training resources for random versions of unity, chain-based grammar and self-removal of such resources. Various studies like [33], [34] have expanded methods for self-annotating Treebank resources with f-structure data. Yet, all of these methods were only relevant to Treebank fractions of a few hundred Trees. Another study by [35] applies an annotation method which measures a complete Treebank

with more than 1,000,000 words in approximately 50,000 sentences and with 19,000 CFG rules. The algorithm is applied as a recursive process in Java. It addresses the Penn-II Treebank nodes that have f-structure data along with its annotations. The annotation requires less than 30 minutes.

4. CONCLUSION

Lexical Functional Grammar (LFG) has a significant impact on Natural Language Processing (NLP) issues. In this study, a review for several research studies has been intensively analyzed critically. We have focused on Arabic language in surveying these studies as less attention has been paid in this area. LFG has been discussed from different viewpoints like: resolving ambiguity in Arabic, Definite Clause Grammar, parsing Arabic sentences, Arabic Dependency structure, Penn Arabic Treebank and its annotation.

[5] has developed an Arabic parser system for parsing modern scientific text with satisfactory results. However, huge amount of parse trees are generated due to the ambiguity issues. [18] has worked on resolving ambiguity for Arabic, however, this study has focused on limited categories of ambiguity and examined the system performance on small datasets. [8] has developed a parser system for parsing Arabic sentences, however, some sentences have not been parsed properly due to the reason that some of these sentences couldn’t match the proper production rule. [10] developed a system that parses Modern Standard Arabic (MSA) sentences through the use of treebank resources with relatively high score results. Nevertheless, the system accepts the Arabic sentence in a transliterated form only and generates the corresponding parsed tree in a text format.

For future perspectives, researchers may work on enhancing the work of [5] by resolving the ambiguity issues and getting a specific parse tree instead of several possible readings. Further research may focus on resolving ambiguity for a large dataset of Arabic sentences in correspondence with [18]. Future research may also work on the expansion of the developed CFG by [8] in order to work with multiple sentences in Arabic. In accordance with [10], researchers may also work on developing an Arabic parser that takes a free Arabic sentence as input and produce the corresponding parsed tree in a GUI format.

REFERENCES

[1] Al Emran, M., & Shaalan, K. (2014, September). A Survey of Intelligent Language Tutoring Systems. In *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on* (pp. 393-399). IEEE.

[2] Cahill, A., McCarthy, M., Van Genabith, J., & Way, A. (2002). Automatic annotation of the Penn-Treebank with LFG F-structure information.

[3] Tounsi, L., Attia, M., & van Genabith, J. (2009). Parsing Arabic using treebank-based LFG resources. *Proceedings of the LFG09 Conference.*

[4] Al-Taani, Ahmad T., Mohammed M. Msallam, and Sana A. Wedian. "A top-down chart parser for analyzing arabic sentences." *Int. Arab J. Inf. Technol.*9.2 (2012): 109-116.

[5] Shaalan, K., Farouk, A., & Rafea, A. (1999, April). Towards an Arabic parser for modern scientific text. In *Proceeding of the 2nd Conference on Language Engineering* (pp. 103-114).

[6] McCord, Michael C., and Violetta Cavalli-Sforza. "An arabic slot grammar parser." *Proceedings of the 2007 Workshop on computational approaches to semitic languages: Common issues and resources*. Association for Computational Linguistics, 2007.

[7] Bataineh, B. M., & Bataineh, E. A. (2009, July). An efficient recursive transition network parser for Arabic language. In *Proceedings of the World Congress on Engineering* (Vol. 2, pp. 1-3).

[8] Alqrainy, S., Muaidi, H., & Alkoffash, M. S. (2012). Context-free grammar analysis for Arabic sentences. *International Journal of Computer Applications*, *53*(3).

[9] Shatnawi, M., & Belkhouche, B. (2012). Parse Trees of Arabic Sentences Using the Natural Language Toolkit. *College of IT, UAE University, Al Ain*.

[10] Al-Emran, M., Zaza, S., & Shaalan, K. (2015, May). Parsing modern standard Arabic using Treebank resources. In *Information and Communication Technology Research (ICTRC), 2015 International Conference on* (pp. 80-83). IEEE.

[11] Kaplan, R. M., & Bresnan, J. (1982). Lexical-functional grammar: A formal system for grammatical representation. *Formal Issues in Lexical-Functional Grammar*, 29-130.

[12] Bresnan, J. (2001). Lexical-Functional Syntax Blackwell.

[13] Bresnan, J., Asudeh, A., Toivonen, I., & Wechsler, S. (2015). *Lexical-functional syntax* (Vol. 16). John Wiley & Sons.

[14] Shaalan, K. (2010). Rule-based approach in Arabic natural language processing. *The International Journal on Information and Communication Technologies (IJICT)*, *3*(3), 11-19.

[15] Habash, N., Dorr, B., &Monz, C. (2006). Challenges in building an Arabic-English GHMT system with SMT components. In *Proceedings of the 11th Annual Conference of the European Association for Machine Translation (EAMT-2006)*, pp. 56 - 65.

[16] Farghaly, A., & Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, *8*(4), 14.

[17] Shaalan, K., Abo Bakr, H. M., & Ziedan, I. (2009, March). A hybrid approach for building Arabic diacritizer. In *Proceedings of the EACL 2009 workshop on computational approaches to semitic languages* (pp. 27-35). Association for Computational Linguistics.

[18] 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).

[19] El-Shishiny, H. (1990, August). A formal description of Arabic syntax in definite clause grammar. In *Proceedings of the 13th conference on Computational linguistics-Volume 3* (pp. 345-347). Association for Computational Linguistics.

[20] Bakir, M. J. (1980). *Aspects of clause structure in Arabic: a study in word order variation in literary Arabic*. Indiana University Linguistics Club.

[21] Al-Khuli, M. A. (1979). *A contrastive transformational grammar: Arabic and English* (Vol. 10). Brill Archive.

[22] Ayoub, G. (1981). Structure de la phrase verbale en arabe standard. *Etudes Arabes Saint-Denis, 1*(2), 1-367.

[23] Fehri, A. F. (1981). *Complémentation et anaphore en arabe moderne: une approche lexicale fonctionnele* (Doctoral dissertation).

[24] Collins, M. J. (1996). A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pp. 184 - 191. Association for Computational Linguistics.

[25] Ditters, E. (2001, July). A formal grammar for the description of sentence structure in modern standard Arabic. In *EACL 2001 Workshop Proceedings on Arabic Language Processing: Status and Prospects* (pp. 31-37).

[26] Jaccarini, A. (2001). A modifiable structural editor of grammars for arabic processing. In *the proceeding of*.

[27] Rafea, A. A., & Shaalan, K. F. (1993). Lexical analysis of inflected Arabic words using exhaustive search of an augmented transition network.*Software: Practice and Experience*, *23*(6), 567-588.

[28] Othman, E., Shaalan, K., & Rafea, A. (2003, September). 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* (pp. 37-44).

[29] Henderson, J. C., & Brill, E. (1999). Exploiting diversity in natural language processing: Combining parsers. In *Proceedings of the Fourth Conference on Empirical Methods in Natural Language Processing*, pp. 187 - 194.

[30] Bies, A., & Maamouri, M. (2003). Penn Arabic treebank guidelines. *Draft: January, 28*, 2003.

[31] Maamouri, M., & Bies, A. (2004, August). Developing an Arabic treebank: Methods, guidelines, procedures, and tools. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based languages* (pp. 2-9). Association for Computational Linguistics.

[32] Baptista, M. (1995). On the nature of pro-drop in Capeverdean Creole. *Harvard Working Papers in Linguistics, 5*, 3-17.

[33] Frank, A. (2000, July). Automatic F-Structure annotation of treebank trees. In*Proceedings of the LFG00 Conference, CSLI Online Publications, Stanford, CA.*

[34] Sadler, L., van Genabith, J., & Way, A. (2000). Automatic F-structure annotation from the AP Treebank.

[35] Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., ... & Schasberger, B. (1994, March). The Penn Treebank: annotating predicate argument structure. In *Proceedings of the workshop on Human Language Technology* (pp. 114-119). Association for Computational Linguistics.

Said A. Salloum is currently a master student of Informatics (Knowledge and Data Management) at The British University in Dubai. He got his Bachelor degree in Computer Science from Yarmouk University. He is currently the Director of programming and Systems Analysis department at Al Buraimi University College. Salloum is an Oracle expert since 2013 along with various recognized international certificates that are issued by Oracle.

Mostafa Al-Emran has graduated from The British University in Dubai with a distinction level along with the top Academic Excellence Award with MSc in Informatics (Knowledge and Data Management). He is currently the Head of Technical Support & Electronic Services Sections at Al Buraimi University College. Al-Emran got his Bachelor degree from Al Buraimi University College with the first honor degree in Computer Science. Currently, he is working on different research areas in Computer Science such as: M-Learning, Knowledge Management, Educational Technology and Data Analysis.

Khaled Shaalan is a full professor of Computer and Information Sciences at the British University in Dubai (BUiD). He is also a tenure professor at Cairo University. Prof Khaled is an Honorary Fellow at the School of Informatics, University of Edinburgh (UoE). He is currently the Head of PhD in Computer Science, MSc in Informatics, and MSc in IT Management programs. His main area of interest includes computational linguistics. He is an authority in the field of Arabic Natural Language Processing, and commands a great respect among the research community in the Arab world. He is the Head of Natural Language Research Group at BUiD. Prof Khaled has several research publications in his name in highly reputed journals such as IEEE Transactions on Knowledge and Data Engineering, Computational Linguistics, Journal of Natural Language Engineering, Journal of the American Society for Information Science and Technology, Expert Systems with Applications, Software-Practice & Experience, Journal of Information Science, and Computer Assisted Language Learning to name a few. He has guided several Doctoral and Master Students in the area of Arabic Natural Language Processing and Knowledge Management. He has done extensive research in the field of Arabic Named Entity Recognition and currently working on Arabic Question Answering.