



Theoretical and Computational Perspectives of Arabic Morphological Analyzers and Generators : Theoretical Survey

Abdelmawgoud M. Maabid and Tarek Elghazaly

Department of Computer and Information Sciences, Institute of Statistical studies and Research, Cairo University, Egypt

amawgoud@pg.cu.edu.eg

tarek.elghazaly@cu.edu.eg

ABSTRACT

Morphology analysis is an essential part of most applications of natural language processing (NLP) which included different applications like Machine Translation (MT) and language rule based Information Retrieval (IR). Many Arabic morphological systems had built for different purposes with different algorithms and approaches; this paper is considered a survey of Arabic Morphological system from researchers' perspectives and approaches used to build them. Based on this survey; in the first part of this paper; the perspective of Arabic morphological systems had been classified into two major issues; one of them is the theoretical perspective and the second is the computational perspective of Arabic morphology. While the second part of this paper deals with approaches used to build the Arabic Morphology systems itself which are Table Lookup Approach, Combinatorial Approach, Linguistic Approach, Traditional Approaches, Finite-state Automata and Two-Level Morphology Approach and Pattern-Based Approach.

Indexing terms/Keywords

Morphology; Arabic Morphology; NLP; Computational Morphology; Morphology Approaches.

Academic Discipline And Sub-Disciplines

Natural Language Processing, Text Processing; Morphology, Natural language generation, Natural language interfaces;

SUBJECT CLASSIFICATION

Computational linguistics--Arab countries

TYPE (METHOD/APPROACH)

Literary Analysis; Survey

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATION JOURNAL OF COMPUTERS AND TECHNOLOGY

Vol. 13, No. 11

editorijctonline@gmail.com

www.ijctonline.com, www.cirworld.com



1. INTRODUCTION

Arabic is one of the languages having the characteristics that from one root the derivational and inflectional systems are able to produce a large number of words (lexical forms) each having specific patterns and semantics. Morphology in linguistics concerns with the study of the structure of words. In other words, morphology is simply a term for that branch of linguistics concerned with the forms words take in their different uses and construction. Morphology analysis is an essential part of most applications of natural language processing, NLP. Applications of NLP include Machine Translation, MT., Information Retrieval, IR., etc... While processing natural language applications are based on the Morphology part, it has to meet some criteria in order to satisfy the required functionality.

In Arabic, there are about 10,000 roots and 900 patterns. Not every root is compatible with every pattern. Linguistic experts must be consulted when creating comprehensive compatibility vectors. If compatibility vectors for all roots are available, then the general linguistic algorithm can be used. Once found, roots should be checked in the vectors to verify the compatibility between the roots and the suggested patterns.

For noun analysis, a large number of lists need to be investigated, such as noun derivatives (e.g., nouns of place, of time, of instrument), nonstandard or solid nouns and different forms (e.g., plural, dual, and feminine). The authors state that the system covers the Arabic dictionary and can analyze vowelized and non-vowelized Arabic words. They also state that their implementation of pattern matching requires almost no time and lists do not require vast memory space.

Whereas many Arabic morphological systems are built with different strategies and methodologies this paper is considered a theoretical survey of Arabic morphological systems in two parts; the first part deals with the perspectives of Arabic Morphologies. The second part deals with the different approaches used in some Arabic Morphologies.

2. PERSPECTIVES OF ARABIC MORPHOLOGY

2.1. Theoretical Perspective of Arabic Morphology

The notion of the morpheme as a meaningful string of segments delimited by the morpheme boundary mainly considered as blank spaces and sentences end marks, and containing no internal morpheme boundary, is challenged by the facts of Arabic Morphology, which exhibits properties that can be recalled [1], in very short words, as follows:

- Roots: built of three or four consonants. Each root dominates a clustering of Arabic lexical morphemes around a semantic field, which can be single, subdivided or multiple.
- Certain changes in nouns, verbs or adjectives based on these consonantal roots yield derivatives. Some vowel and syllabic patterns seem, subsequently, to be associated with a constant set of meanings.
- Traditional treatment of Arabic morphology – especially in computational morphology – sometimes remains taxonomic, abstracting away from the particular root and citing or generating all possible patterns [2].

2.2. Computational Perspective of Arabic Morphology

2.2.1 Knowledge-based Approaches

The computational treatments of Arabic morphology have been classified into the following major linguistic frameworks:

A. Syllable-based Morphology (SBM)

European languages especially Germanic language dealt with their morphology in terms of their syllable structure of word, thus this concept has been intended to apply to all languages [3]. Semitic languages especially Arabic language appears on the surface to offer biggest challenge to this theory [4]. Syllable strings therefore have the final syllable as syl1, the penultimate syllable as syl2 and so on. This ignores higher level structures such as feet or tone groups, as well as lower level structures such as mora. In Arabic it must therefore define a default syllable structure for trilateral stems, with default positions for the stem consonants; assuming that the simplest structure as exemplified by the stem "KATAB". The standard phonotactics of Arabic require syllables of CV (preferred) or CVC structure. The root must be divided into two syllables, because there are three consonants, and syllables in Arabic may have a maximum of two consonants

B. Root-and-Pattern Morphology

McCarthy proposed a claim that the stems are formed by a derivational combination of a root morpheme and a vowel melody. Roots are said to overlap with patterns to form stems. McCarthy's analysis differs from Harris' [5] in abstracting out or auto segmentation of the vowels from the pattern and placing them on a separate tier of the analysis [6, 7].

McCarthy's auto segmental approach is reflected in most of the computational attempts to model Arabic morphology, especially in the systems written within finite state morphology [8, 9]. It would be useful to briefly review one of the largest systems ever built for Arabic morphology on the basis of finite-state technology, namely the Arabic morphology system implemented using Xerox finite state technology. Verbs in Arabic are either trilateral (having three radical letters) as "ktb" (breaking) or quadrilateral (having four) as "dHrj" (rolling), for the root "ktb" if the pattern "_a_a_" applied then a new stem "katab" can be produced.



C. Lexeme-based Morphology

Lexeme-based morphology supports the claim that the stem is the only morphologically relevant form of a lexeme [10]. Lexeme-based morphology is (usually) an Item-and-Process approach. Instead of analyzing a word form as a set of morphemes arranged in sequence, a word form is said to be the result of applying rules that alter a word form or stem in order to produce a new one. An inflectional rule takes a stem, changes it as is required by the rule, and outputs a word form; a derivational rule takes a stem, changes it as per its own requirements, and outputs a derived stem; a compounding rule takes word forms, and similarly outputs a compound stem. In lexeme-based Morphology; given the forms 'cat' and 'cats', we would say that there is a lexeme [cat] that has two word forms 'cat' and 'cats'

D. Morpheme-based morphology

In morpheme-based morphology, word forms are analyzed as arrangements of morphemes. A morpheme is defined as the minimal meaningful unit of a language. In a word like independently, the morphemes are in-, depend, -ent, and ly; depend is the root and the other morphemes are, in this case, derivational affixes. In a word like dogs, we say that dog is the root, and that -s is an inflectional morpheme. In its simplest (and most naïve) form, this way of analyzing word forms treats words as if they were made of morphemes put after each other like beads on a string, is called Item-and-Arrangement. More modern and sophisticated approaches seek to maintain the idea of the morpheme while accommodating non-concatenative, analogical, and other processes that have proven problematic for Item-and-Arrangement theories and similar approaches[11]. Morpheme-based morphology presumes three basic:

- Baudouin's SINGLE MORPHEME HYPOTHESIS: Roots and affixes have the same status in the theory, they are MORPHEMES.
- Bloomfield's SIGN BASE MORPHEME HYPOTHESIS: As morphemes, they are dualistic signs, since they have both (phonological) form and meaning.
- Bloomfield's LEXICAL MORPHEME HYPOTHESIS: The morphemes, affixes and roots alike, are stored in the lexicon.

Morpheme-based morphology comes in two flavors, one Bloom fieldian and one Hockettian. For Bloomfield, the morpheme was the minimal form with meaning, but it was not meaning itself. For Hockett, morphemes are meaning elements, not form elements. For him, there is a morpheme plural, with the allomorphs -s, -en, -ren etc. Within much morpheme-based morphological theory, these two views are mixed in unsystematic ways, so that a writer may talk about "the morpheme plural" and "the morpheme -s" in the same sentence, although these are different things [12].

E. Word-based morphology

Word-based morphology is (usually) a Word-and-paradigm approach. This theory takes paradigms as a central notion. Instead of stating rules to combine morphemes into word forms, or to generate word forms from stems, word-based morphology states generalizations that hold between the forms of inflectional paradigms. The major point behind this approach is that many such generalizations are hard to state with either of the other approaches. The examples are usually drawn from fissional languages, where a given "piece" of a word, which a morpheme-based theory would call an inflectional morpheme, corresponds to a combination of grammatical categories, for example, "third person plural." Morpheme-based theories usually have no problems with this situation, since one just says that a given morpheme has two categories. Item-and-Process theories, on the other hand, often break down in cases like these, because they all too often assume that there will be two separate rules here, one for third person, and the other for plural, but the distinction between them turns out to be artificial. Word-and-Paradigm approaches treat these as whole words that are related to each other by analogical rules. Words can be categorized based on the pattern they fit into. This applies both to existing words and to new ones. Application of a pattern different from the one that has been used historically can give rise to a new word, such as older replacing elder (where older follows the normal pattern of adjectival superlatives) and cows replacing kine (where cows fits the regular pattern of plural formation).

F. Stem-based Lexicon

The central claim of this approach is that stem dependent on lexical databases, with entries associated with grammar and lexis specifications, is the most appropriate organization for the storage of pertinent information for Arabic.

Also known as "Stem-based Lexical Resources, Including Root-Pattern and Grammar-lexis Information"

Table 1. Knowledge-based Approaches

Approach	Claim	Use of word	Use of Stem	Use of Morpheme
Syllable-based Morphology	Deal with morphology in terms of their syllable structure of word	Yes	No	Yes
Root-and-Pattern Morphology	stems are formed by a derivational combination of a root morpheme and a vowel	No	Yes	Yes
Lexeme-based Morphology	The stem is the only morphologically relevant form of a lexeme	No	Yes	No
Morpheme-based morphology	Word forms are analyzed as arrangements of morphemes	Yes	No	Yes
Word-based morphology	states generalizations that hold between the forms of inflectional paradigms	Yes	No	Yes
Stem-based Lexicon	Stem dependent on lexical databases, with entries associated with grammar and lexis specifications	Yes	Yes	Yes

2.2.2 Empirical Approaches

A major benefit of the knowledge-based methods is that the rules and constraints for recognizing and classifying the internal structure of words are defined on a precise linguistic basis. Thus, under the assumption that the set of morphological rules and constraints define a linguistically consistent system, then, if a word can be morphologically analyzed, we can be sure that the resulting structure is correct [13]. Of course, this requires the computational basis of the analysis to be sound and complete; however, since we also assume this for the computational basis of the empirical methods, we do not consider this as a unique feature of knowledge-based methods. Furthermore, it is also often assumed that the modeled linguistic system is domain independent that is valid and applicable in any domain. As a consequence, the ultimate goal is to implement a linguistic knowledge base -in our case, a morphological system- that covers all possible allowable structures and only these. However, this requires not only that all possible allowable structures are known by the linguist, but that they can be formalized and implemented consistently preferable as non-redundant as possible [14].

In large scale implementation of Arabic NLP components the following drawbacks many raised:

- **Ambiguity:** The aim of formalizing all possible allowable structures means that an NLP component is to be expected to return all possible analyses (or readings) for a given input for further processing (by a human or another NLP component), as long as the system does not dispose of any decision criteria on how to rank or select between the alternative readings relative to a given domain or application.
- **Coverage:** Even if a linguistic domain is completely understood, it is extremely challenging to provide a complete implementation of all phenomena from scratch, because it might still be unclear how to represent a certain linguistic entity using the implementation formalism at hand or because not all possible ways and constraints (e.g., about the nature of the input data) are known in order to properly embed the NLP component into a larger application context.

As a solution direction for these kinds of problems, empirical-based methods are explored and developed in computational linguistics since the 1990s [15]. Empirical methods employ machine learning techniques to automatically extract linguistic knowledge from natural language data directly rather than require the system developer to manually encode the requisite knowledge. Since these methods are by definition data-driven, they actually also learn how to weight between alternative solutions and how to predict useful information (e.g., a missing class label) for unknown entities through a rigorous statistically analysis of the data. In the beginning of the development of the new field of empirical methods for NLP, the proposed approaches have often been considered as alternative or even competing approaches to the corresponding knowledge-based methods [16]. However, in recent years the trend has become to consider both approaches more as complementary to each other, and new ways of integrating knowledge-based and empirical-methods are envisaged and actively explored.

3. ARABIC MORPHOLOGY APPROACHES

Many Arabic stemming algorithms have been developed, where each one has been implemented according specific methodology, in the subsequent sections we will represent an overview of such stemming algorithms used for Arabic systems.



3.1 Table Lookup Approach

The table lookup approach depends mainly on very large tables listed in alphabetical order storing Arabic words found in natural texts with their corresponding morphological parts. Such parts include stem, root, and affixation. Words may include functional words, foreign words, and proper names where each word uses a unique entry in the table. Multiple entries may exist for words spelled the same to reflect the possibility of multiple analyses [17].

3.2. Combinatorial Approach

Combinatorial approaches compare tested words against prepared lists for roots, patterns, particles, and affixes. The comparison is based on a combinatorial algorithm that tests all combinations of three or four letters of a given word in order to extract the root. In general, such approaches are simple but take a long time to process and require very large lists.

Al-Fedaghi and Al-Anzi [18] enhanced an algorithm proposed previously by Al-Fedaghi and Al-Sadoun. The enhancement is achieved by solving some important phonological and orthographical problems, such as assimilation, mutation, vocalization, and germination. The modified algorithm uses lists of trilateral roots and patterns with all combinations of affixes. It starts by comparing the input word against the pattern list to extract the root. The algorithm then runs in four modes to cover all possible orthographical and phonological cases.

3.3. Linguistic Approach

Linguistic-based approaches require a deep analysis of the Arabic morphological system. They simulate the behavior of a linguist during the analysis of a given Arabic word. In this approach, tested words are compared against lists of affixes or function words to derive stems and then compared against lists of patterns and roots in order to derive roots.

In general, such linguistic approaches are more accurate but require too many lists that need to be prepared and checked linguistically. Such lists incur a time overhead for accessing them. In addition, the mechanism for removing affixes is almost a process of trial and error, in which results are not guaranteed to be accurate.

Linguistic approaches can be subdivided into **traditional** or **cut-and-paste** approaches, and **finite-state automata and two-level morphology** approaches, and **Pattern-Based** Approach. In the subsequent section, traditional linguistic approaches will be discussed, followed by finite-state and two-level morphology models.

3.3.1 Traditional Approaches

In this section, a survey of Arabic morphological analysis techniques is conducted that includes a description of the algorithm, the method of its implementation.

Al-Khalil Arabic Morphological System is java compiled trial version applications published on April 2010. The system can analyze word or sentences typed in the text area [19]. The system can analyzes up to 10 words per second in small text up to 35 words per second in larger documents.

Al-Bawab [20, 21] proposed a morphosyntactic system for generating and analyzing Arabic words. The analyzer guesses the type of input word as verb, noun, or article. A little information can be used as a hint as to word type, such as a notation mark. Verb analysis starts by removing affixes and, in the case of false partitioning; it will be reattached completely or partially. After partial or full removal of affixes, the remainder will be compared with patterns that are equal in length. Entries in the pattern list are sorted according to pattern lengths and vowelization and consist of patterns, roots, and morphological properties. Many test processes are carried out to assure correct analysis including compatibility between prefix and patterns, the availability of the root in the list, and checking pronouns and syntax case of verbs.

Khoja and Garside [22] presented a linguistic algorithm that analyzes vowelized and non-vowelized Arabic words. In this algorithm, a list of function words is used to detect and filter them. The algorithm also uses backtracking as a remedy for the erroneous removal of affixes. Straightforward pattern matching is used to extract roots, where a prepared list of roots is used to check the validity of extracted roots. The algorithm tries to solve special cases, such as words containing weak letters or germination mark and hamza or glottal stop. The output of the system gives some useful statistics. The work concludes with a statement of the advantages and disadvantages of the system.

Darwish [23] designed a light Arabic stemmer that removes common prefixes and suffixes. The stemmer is very compact in size and has been significantly improved by a modification done by Leah Larkey. The stemmer accepts input in either CP-1256 or UTF-8 encoding. Internally, it transliterates input string into Roman characters, removes diacritics, and makes letter normalization. This stemmer is available publicly for research purposes.

Abuleil et al. [24] described and implemented a learning system that can analyze some types of Arabic nouns to produce their morphological information using a rule base that uses suffix analysis as well as pattern analysis. The system utilizes user feedback to classify the noun and identify the group that it belongs to. The authors classified the nouns into 84 groups according to their patterns for singular, plural, masculine, and feminine. For each group, a function was implemented to find the morphological information for a given noun. Very few groups have a unique pattern for plural and singular, and most of them share the same pattern with other groups. The system consist of the following modules: interface module, type-finder module, database module, noun morphology analyzer module, suffix-analyzer module, pattern generator module, database-checker module, and user-feedback module. It reads a noun from the text, isolates



suffixes, and generates its pattern. It uses the classified noun table, the suffix/pattern analysis, or the user-feedback module to find the group to which the noun belongs and updates the database.

3.3.2 Finite-state Automata and Two-Level Morphology

Kay [25] explains that finite-state and two-level morphology has been built over the concepts of auto segmental phonology proposed by Goldsmith and auto segmental morphology proposed by McCarthy. McCarthy, as cited by Kay [25], proposed in 1979 the auto segmental theory for Arabic morphology. This work has been supported and defended by Haile and Mtenje, but Hudson raised serious theoretical challenges [26]. Beesley [27] believes that it is not clear how McCarthy's Arabic morphology can be implemented to do practical morphological analysis or generation. Ali [28] commented that the theory developed by McCarthy might be suitable for phonology, but that applying it to morphology might be very complicated. McCarthy updated his proposal so that mosaic analysis can be applied to Arabic nouns and affixational analysis can be applied to Arabic verbs [27, 29, 30].

ALPNET and Xerox projects: In non-Semitic languages, morphemes are concatenated with roots and stems, while in Semitic language inflectional patterns are not completely concatenative but interdigitate or intercalate [31]. Stems are generated through interdigitation of roots and patterns. The standard system of Arabic orthography perhaps represents the extreme of complexity because of the great irregularities between the lexical and surface strings caused by phenomena including weak roots, hamza orthography, and the zero realization of short vowels and germination mark [27]. Even though Arabic morphology is an extreme test for any theory, a two-level model will be very suitable for the analysis of Arabic [32]. Kiraz [30] believes that nobody has dealt computationally with the challenging problem of the Arabic broken plural. Broken plurals are not handled by most of current Arabic stemmers [33].

In the ALPNET project, Beesley [27] implemented the two-level theory whereby lexicons drive the search process, while rules act as filters. In traditional two-level systems, lexicons contain morphemes, and the target is to find the correct lexicon entries that comprise the correct word. This mechanism, without modification, is not suitable for Arabic.

3.3.3 Pattern-Based Approach

Al-Kharashi and Al-Sughaiyer [34, 35] mentioned that researchers proposing different morphological analysis techniques were seeking a high degree of accuracy. This caused proposed systems to be based on heavy computational processes or the use of large amounts of associated information. Consequently, they proposed a pattern based algorithm that did not require either complex computations or associated tables. The proposed technique was based on fast-surface morphological analysis of Arabic words. Al-Kharashi and Al-Sughaiyer concluded their work as follows: Reducing the number of rules and increasing language coverage while maintaining the same level of performance and functionality can achieve enhancement. Merging rules is one method that can be used for enhancing the pattern-based stemmer. While keeping time efficiency almost the same, merging rules reduces the rule list dramatically.

4. CONCLUSION

Based on this survey we can classify the perspective of Arabic morphological systems into two major issues; one of them is the theoretical perspective and the second is the computational perspective of Arabic morphology. The computational perspective considered the Knowledge-based Approaches which consider how the algorithm deals with the Arabic word; some of these systems consider the syllable structure of the word, derivational combination of a root morpheme and a vowel melody, stem is the only morphologically relevant form of a lexeme, word forms are analyzed as arrangements of morphemes, Word-and-paradigm approach. This theory takes paradigms as a central notion, stem dependent on lexical databases, with entries associated with grammar and lexis specifications. And Empirical Approaches which deals with the rules and constraints for recognizing and classifying the internal structure of words are defined on a precise linguistic basis.

The second classification in this paper dealt with the different approaches used to build existing Arabic morphological systems. In Table lookup Approach the system depends mainly on very large tables listed in alphabetical order storing Arabic words found in natural texts with their corresponding morphological parts. While the Combinatorial Approach compares tested words against prepared lists for roots, patterns, particles, and affixes. The Linguistic Approach requires a deep analysis of the Arabic morphological system. They simulate the behavior of a linguist during the analysis of a given Arabic word. Furthermore, Traditional Approaches consider a survey of Arabic morphological analysis techniques is conducted that includes a description of the algorithm, the method of its implementation, classified by language coverage. Then the Finite-state Automata and Two-Level Morphology Approach which built over the concepts of auto segmental phonology proposed by Goldsmith and auto segmental morphology proposed by McCarthy. Finally, The Pattern-Based Approach which considers the proposed systems to be based on heavy computational processes or the use of large amounts of associated information.

References

- [1] Bauer, L., The Linguistics Student's Handbook 2007 Edinburgh University Press. 2007
- [2] Bohas, G., J.-P. Guillaume, and D. Kouloughli, The Arabic Linguistic Tradition (Georgetown Classics in Arabic Languages and Linguistics series) 2006: Georgetown University Press
- [3] Kiraz, G.A., Computational Nonlinear Morphology: With Emphasis on Semitic Languages. Studies in Natural Language Processing 2001: Cambridge University Press 194



- [4] Hayes, B., et al., *Linguistics: An Introduction to Linguistic Theory*. 1st Edition ed2001: Wiley-Blackwell. 768
- [5] Harris, Z., *The Linguistic Structure of Hebrew*. *Journal of the American Oriental Society*, 1941. 62: p. 143–67.
- [6] McCarthy, J.A., *Formal Problems in Semitic Phonology and Morphology*, 1979, MIT.
- [7] McCarthy, J.A., *Prosodic Theory of Non-Concatenative Morphology*. *Linguistic Inquiry*, 1981. 12: p. 373–418.
- [8] Kiraz, G.A. *Multi-tape two-level morphology: A case study in semitic non-linear Morphology*. in *In Proceedings of COLING'94*. 1994.
- [9] Kiraz, G.A., *A Multi-tiered Nonlinear Morphology using Multi-tape Finite State Automata: Automata: A Case Study on Syriac and Arabic*. *Computational Linguistics*, 2000. 26:1: p. 77–105.
- [10] Kiraz, G.A., *Computational Nonlinear Morphology; With Emphasis on Semitic Languages*. *Studies in Natural Language Processing*, ed. I. Branimir Boguraev, T.J. Watson Research Center and L.D.C. Steven Bird, University of Pennsylvania 2004, The Edinburgh Building, Cambridge CB2 2RU, UK: The press syndicate of the University of Cambridge, the Pitt building, Trumpington street, Cambridge, United Kingdom.
- [11] Roark, B. and R. Sproat, *Computational Approaches to Morphology and Syntax* 2007, United States: Oxford University Press, New York.
- [12] Narayanan, A. and L. Hashem *Finite-state abstractions on Arabic morphology*. *Artificial Intelligence Review*. 1994. 7, 373–399.
- [13] Holes, C., *Modern Arabic: Structures, Functions, and Varieties*. Revised Edition ed2004: Georgetown University Press. 419
- [14] Versteegh, K. and M.G. Carter, *Studies in the History of Arabic Grammar II: Proceedings of the 2nd Symposium on the History of Arabic Grammar, Nijmegen, 27 April-1 May 1987* 1990: John Benjamins Publishing Company. 319.
- [15] Cardie, C. and R.J. Mooney, *Guest editors' introduction: Machine learning and natural language*. *Machine Learning*, 1999. 11:1: p. 1–5.
- [16] Magerman, D.M. *Statistical decision-tree models for parsing*. in *ACL-95*. 1995. ACL: Ann Arbor, MI, USA.
- [17] Frakes, W.B., *Stemming algorithms*. In *Information retrieval: Data structures and algorithms* 1992: Prentice Hall.
- [18] Al-Fedaghi, S. and F. Al-Anzi. *A new algorithm to generate Arabic root pattern forms*. in *In Proceedings of the 11th national computer conference*. 1989. King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia.
- [19] Khawaja, A., A. Mazrui, and A.R. Boodlal, *AL-Khalil Arabic Morphological System*, 2010, Mohammed Al-Aoual University - Jeddah - Laboratory Research in informatics: The Arab League Educational, Cultural and Scientific Organisation (ALECSO).
- [20] Al-Bawab, M. and M.H. Al-Tayyan, *Computerized processing of Arabic morphology [in Arabic]*. *Arabian Magazine of Sciences*, Arab League Educational, Cultural and Scientific Organization, ALECSO, 1998. 32: p. 6–13.
- [21] Al-Bawab, M., et al., *A computerized morpho-syntactic system of Arabic*. *The Arabian Journal of Science and Engineering*. Published by KFUPM, Dhahran, Saudi Arabia., 1994. 19: p. 461–480.
- [22] Khoja, S. and R. Garside. *Stemming Arabic Text*. 1999 4-6-2012]; Available from: <http://www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps>.
- [23] Darwish, K. *Building a Shallow Morphological Analyzer in One Day*. in *40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*. 2002. Philadelphia, PA, USA.
- [24] Abuleil, S., S. Abuleil, and M. Evens. *Acquisition system for Arabic noun morphology*. in *Association for Computational Linguistics 40th Anniversary Meeting, In Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages*. 2002. University of Pennsylvania, Philadelphia.
- [25] Kay, M. *Nonconcatenative finite-state morphology*. in *In Proceedings of the 3rd Conference of the European Chapter of the Association for Computational Linguistics*. 1987. Copenhagen, Denmark.
- [26] Beesley, K.R., *Arabic stem morphotactics via finite-state intersection*, in *Paper presented at the 12th Symposium on Arabic Linguistics*, Arabic Linguistic Society, 6-7 March 1998 1998b: Champaign, IL.
- [27] Beesley, K.R., *Computer analysis of Arabic morphology: A two-level approach with detours* In B. Comrie & M. Eid (Eds.). *Perspectives on Arabic Linguistics III, Papers from the 3rd Annual Symposium on Arabic Linguistics*, Benjamins, Amsterdam, 1991. Volume 12: p. 155–172.
- [28] Ali, N., *Arabic language and computer [In Arabic]*. Ta'reeb 1988.
- [29] Beesley, K.R. *Arabic Morphological Analysis on the Internet*. in *6th International Conference and Exhibition on Multi-lingual Computing*. 1998a. Cambridge.



- [30] Kiraz, G.A., Computational analysis of Arabic morphology, Computer Laboratory, George.kiraz@cl.cam.ac.uk, Editor 1995: University of Cambridge (St. John's College).
- [31] Narayanan, A. and L. Hashem. On abstract, finite-state morphology. in In Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics 1993. Netherlands.
- [32] Beesley, K.R. Finite-state descriptions of Arabic morphology. in In Proceedings of the 2nd Cambridge Conference on Bilingual Computing in Arabic and English. 1990. Cambridge University, UK: Literary and Linguistic Computing Centre.
- [33] Xu, J., A. Fraser, and R. Weischedel. Empirical studies in strategies for Arabic retrieval. in In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2002. Tampere, Finland: SIGIR 2002.
- [34] Sughaiyer, I.A.A. and I.A.A. Kharashi, An efficient Arabic morphological analysis technique for information retrieval systems, in In ACIDCA'2000 International Conference 2000: Monastir, Tunisia.
- [35] sughaiyer, I.A.A. and I.A.A. Kharashi, Firing policies for an Arabic rule-based stemmer. In Proceedings of the 9th International Symposium on String Processing and Information Retrieval (SPIRE 2002) 2002, Lisbon, Portugal.

Author' biography with Photo



Abdelmawgoud Mohamed Maabid is a Master of Science in computer Science Candidate at Institute of Statistical studies and Research - Department of Computers and Information Sciences, Cairo University. He's also a software development worker who specializes in Arabic text processing systems and plug-ins