ISSN 2277-3061

An Improved General Purpose Arabic Morphological Analyzer and Generator Model (GPAM)

Abdelmawgoud Mohamed Maabid, Tarek Elghazaly, and Mervat Ghaith

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

abdelmogod@hotmail.com
tarek.elghazaly@cu.edu.eg
mervat_gheith@yahoo.com

ABSTRACT

Although, morphological analysis is a vital part of natural language processing applications, there are no definitive standards for evaluating and benchmarking Arabic morphological systems. This paper proposes assessment criteria for evaluating Arabic morphological systems by scrutinizing the input, output and architectural design to enables researchers to evaluate and fairly compare Arabic morphology systems. By scoring some state of the art Arabic morphological analyzers based on the proposed criteria; the accuracy scores showed that the best algorithm failed to achieve a reliable rate. Hence, this paper introduced an enhanced algorithm for resolving the inflected Arabic word, identifies its root, finds its pattern and POS tagging that will reduce the search time considerably and to free up the deficiencies identified by this assessment criteria. The proposed model uses semantic rules of the Arabic language on top of a hybrid sub-model based on two existing algorithms (Al-Khalil & IAMA rules). Based on applying the proposed assessment criteria the efficiency and speed have been enhanced where the system achieved up to 1500 words per second in small text up to 3000 words per second in larger documents

Indexing terms/Keywords

Morphology; Arabic Morphology; NLP;Morphology Assessment Criteria;Analyzer , Arabic Analyzer.

Academic Discipline And Sub-Disciplines

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

SUBJECT CLASSIFICATION

Computational linguistics--Arab countries

TYPE (METHOD/APPROACH)

Proposed Model

ISSN 2277-3061

1 INTRODUCTION

Morphology in linguistics concerns with the study of the structure of words[1]. In other words, morphology is simply a term for that branch of linguistics concerned with the forms words take in their different uses and constructions[2].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[3]. The root is a semantic abstraction consisting of two, three, or (less commonly) four consonants from which words are derived through the superimposition of template based patterns[4]. Unfortunately if understanding is considered, non-diacritic words may make problems of meaning; where many words when they appears in non-diacritic text can have more than one meaning; these different meanings rises problems of ambiguity[5].

In Arabic, like other Semitic languages, word surface forms may include affixes, concatenated to inflected stems. In nouns, prefixes include conjunctions (“و” “and”, ف “and, so”), prepositions (“بـ” “by, with”, “كـ” “like, such as”, “لـ” “for, to”) and a determiner, and suffixes include possessive pronouns. Verbal affixes include conjunction prefixes and negation, and suffixes include object pronouns. Either object or possessive pronouns can be captured by an indicator function for its presence or absence, as well as by the features that indicate their person, number and gender[6]. A large number of surface inflected forms can be generated by the combination of these features, making the morphological generation of these languages a non-trivial task[7].

Natural Languages processing and analysis improved substantially in recent years due to applying data intensive computational techniques[8]. However, state of the art approaches are essentially language specific stemmer (Morphology), considering every surface word in the language[9]. A shortcoming of this word-based analysis of the Arabic language is that it is sensitive to lack of data and information about Arabic words and it morphemes. This is an issue of importance as aligned corpora are an expensive resource, which is not abundantly available for many language analysis levels. This is particularly problematic for morphologically rich languages, where word stems are realized in many different surface forms, which exacerbates the hindering higher level of language analysis.

Morphological analysis can be performed by applying language specific rules. These may include a full-scale morphological analysis, or, when such resources are not available, simple heuristic rules, such as regarding the last few characters of a word as its morphological suffix. In this work, we will adapt some major assessment criteria for measuring advantage or drawback of any Arabic morphological system[10].

2 BACKGROUND AND PREVIOUS WORK

We believe that this is the first proposed work to sum up assessment criteria for Arabic morphological analyzers and Generators. Several researches talked about building powerful stemmers for the Arabic language with accuracies normally exceeding 90% but none of these stemmers offer the source code and/or the datasets used. It is therefore difficult to verify such claims or make a comparison between different stemmers without having the full description of the proposed method or the source code for the implementation of the algorithm[11]. In this section we review some efforts in this direction.

Mohammed N. Al-Kabi and Qasem A. Al-Radaideh[11] proposed analysis of the accuracy and strength of four stemmers for the Arabic language using one metric for accuracy and four other metrics for strength as following:

The first metric called empirical evaluation (EE), which represents a percentage of the correct roots produced by the stemmer under consideration.

The mean number of words per conflation class (MWC) depends on the number of words processed.

Index compression factor (ICF) represents the extent to which a collection of unique words is reduced (compressed) by stemming.

Word change factor (WCF) represents the proportion of the words in a sample that have been changed in any way by the stemming process.

The mean number of characters removed in forming stems (Average CR): Usually strong stemmers remove more characters from words to form stems.

Azze Al-din Al-Mazroui, et.al[12] proposed a specification of morphological analysis system in the Arabic language. In this study the researcher outlined the general characteristic that has to consider during process and building Arabic morphological system in terms of input, analysis and output. The study doesn’t provide any criteria or automation to compare different systems.

Dassouki[13] proposed a tabulate items as mechanism for assessing morphological analyzer in terms of development of the system speed, input, output, integrating with other applications and capabilities of analyzing new and non-Arabic words. The study doesn’t provide any criteria for these selected terms.

William B. Frakes and Christopher J. Fox[14] evaluated the strength and similarity among, four affix removal stemming algorithms. Strength and similarity were evaluated in different ways, including new metrics based on the Hamming distance measure. Data was collected on stemmer outputs for a list of 49,656 English words derived from the UNIX spelling dictionary and the Moby corpus. The study doesn’t provide any criteria for these selected measures and it is specific to English stemmers.

Al-Khalil Arabic Morphological System [15]; is java compiled application published on April 2010. The system can analyze a word or sentences typed in the text area. The system can analyzes up to 10 words per second in small text and up to 35 words per second in larger documents.

Arabic Morphological Analyzer[16]; Is an Arabic analyzer system published over the internet based on Quatrab system that can analyze, generate and categorize Arabic words in phrases.

Shereen Kahoka's stemmer[17] is available in form of java open source application. Kahoka's stemmer removes the longest suffix and the longest prefix. It then matches the remaining word with verbal and noun patterns, to extract the root. The stemmer makes use of several linguistic data files such as a list of all diacritic characters, punctuation characters, definite articles, and 168 stop words[18].

An Improved Arabic morphology analyzer (IAMA)[19] is an algorithm has been developed by using some new semantic rules of the Arabic language to reduce the searching time in the ATN. Also, this research introduces an algorithm for root identification that will reduce the search time considerably.

3 PROPOSED ASSESSMENT CRITERIA OF ARABIC MORPHOLOGICAL SYSTEMS

Assessing and evaluating Arabic morphological systems depends on the input words and resulted output[12] according to a predefined criteria to measure and analyze given system in order to study its weakness and strength, trying to find an Arabic morphological analyzer free from all mistakes. then we will apply these criteria on some of existing available systems; these criticism will not detract from its value and effectiveness[20].

3.1 Input

In computer science, input is something put into a system or expended in its operation to achieve output or a result. Within the context of systems theory, the inputs are what are put into a system. A very fundamental problem with software testing is that testing under all combinations of inputs and preconditions (initial state) is not feasible, even with a simple product. This means that the number of defects in a stemmer can be very large and defects that occur infrequently are difficult to find in benchmarking. More significantly, non-functional dimensions of quality (how it is supposed to be versus what it is supposed to do)—usability, scalability, performance, compatibility, reliability—can be highly subjective; something that constitutes sufficient value to one person may be intolerable to another.

In case of stemmer algorithms the input can be considered as bulk of text passed to the system in form of word or phrase fully or partially diacritized.

- The possibility of analyzing the modern standard texts: Most western scholars distinguish two standard varieties of the Arabic language: the Classical Arabic (CA) of the Qur'an and early Islamic (7th to 9th centuries) literature, and Modern Standard Arabic (MSA), the standard language in use today[21]. The modern standard language is based on the Classical language. Most Arabs consider the two varieties to be two registers of one language, although the two registers can be described in Arabic as (MSA) and (CA)[22].
- The possibility of analyzing the common error words: Common typing errors "common error words" are those words mistyped but are traditionally considered correct; typically a feminine ending character "ـة T" written without dots "ـه h", the character "ى E" instead of "ي I" and the letter "ا a" without instead of "أA" [23].
- The possibility of analyzing new words (Neologisms): Neologisms are often created by combining existing words or by giving words new and unique suffixes or prefixes. Portmanteaux are combined words that are sometimes used commonly. Neologisms also can be created through abbreviation or acronym, by intentionally rhyming with existing words or simply through playing with sounds. Neologisms analysis in morphological system measures the capability of processing the new Arabic words which can be added later to morphological systems' predefined knowledge base.
- Processing of Arabized and transliterated words: Transliteration is a subset of hermeneutics. It is a form of translation, and is the practice of converting a text from one script into another. From an information-theoretical point of view, systematic transliteration is a mapping from one system of writing into another, word by word, or ideally letter by letter. Transliteration attempts to use a one-to-one correspondence and be exact, so that an informed reader should be able to reconstruct the original spelling of unknown transliterated words. Ideally, reverse transliteration is possible. In Arabic transliteration is writing e non-Arabic words written by Arabic alphabet characters as 'فاكس' "Fax" in English and "انترنت" "Internet" In English.
- Processing of non- tripartite verbs: Arabic words are divided into three types: noun, verb, and particle. Nouns and verbs are derived from a closed set of around 10,000 roots. The roots are commonly three or four letters and are rarely five letters. Arabic nouns and verbs are derived from roots by applying templates to the roots to generate stems and then introducing prefixes and suffixes. [6]. Assessing and evaluating Arabic considering the system capability of analyze quadrilateral and quinqueliteral verbs like "طمأن" "Reassure" and all possible cases of their forms of transitivity and weakness[12].

3.2 Output

Input is something put into a system or expended in its operation to achieve output or a result. Output is the information produced by a system or process from a specific input. Within the context of systems theory, the inputs are what are put

into a system and the outputs are the results obtained after running an entire process or just a small part of a process. Because the outputs can be the results of an individual unit of a larger process, outputs of one part of a process can be the inputs to another part of the process. Morphology output is all possible combination of affixes that produced a valid Arabic word, roots and patterns.

- Covering analysis of all input words:
- The system should cover all cases of analysis.
- Determine word types (pattern, root, stem and attached affixes)[12].
- Analyzing the words all domains of the language (Geographic, Historical, Religion, and Math).
- Considering syntactic case of input word (within phrase)
- Meet all possible cases for analysis: The system has to assume that the input word is a verb, name and character so,

1. Verb: has to cover non- tripartite, quadrilateral, quinqueliteral with their forms of transitivity, augmentation, hollow…etc.[4].
2. Name: has to cover names, infinitives, adjectives and adverbs.
3. Particle: has to cover prepositions, conjunctions, vowel, and vocative particles.

- Express grammatical function of the affixes
- Identify and express the word prefixes and suffixes with names: Prefixes: determining if the prefix is part of the word or a prefix to a name: example "بطاقات" (Cards), or "b بـ" and "طاقات" (Capacities). And Suffix, determining if the suffix is part of the word or a concatenated pronoun to the name: example "نكره" which come from the root " ن ك ر"; it can acts as a verb (deny) or original word character (adjective : unknown)[12].
- Identifying and express the affixes functions attached to verbs: Verb tense: The prefix "س" with verbs determining that the verb in future (present) tense; while with the word "سعي" "Saa'h" is not as the word "عي" cannot be in imperative.
- Ambiguity and Overlapping of syntactic cases:

Many words in Arabic are homographic [5]: they have the same orthographic form, though the pronunciation is different. There are many recurrent factors that contributed to this problem. Among these factors are:

1. Orthographic alternation operations (such as deletion and assimilation) frequently produce inflected forms that can belong to two or more different lemmas.
2. Some lemmas are different only in that one of them has a doubled sound which is not explicit in writing. Arabic Form I and Form II are different only in that Form II has the middle sound doubled.
3. Many inflectional operations underlie a slight change in pronunciation without any explicit orthographical effect due to lack of short vowels (diacritics).
4. Some prefixes and suffixes can be homographic with each other. The prefix t can indicate 3rd person feminine or 2nd person masculine.
5. Prefixes and suffixes can accidentally produce a form that is homographic with another full form word. This is termed "coincidental identity"
6. Similarly, clitics can accidentally produce a form that is homographic with another full word.
7. There are also the usual homographs of uninflected words with/without the same pronunciation, which have different meanings and usually different POS's.
8. That means determining the lack of morphological knowledge of the word analyst; in case of partially diacritized or non-diacritic words, the ambiguity problem may appear, so, the better is to determine all possible cases of the input word; as an example the work "رب" many be either "رَبٌّ" (God) or "رُبَّ" (maybe).

- Identifying the root of the word and determining all possible roots for the analyzed word; Right root identification of the input word, and with all generated words the system has to be capable to determine their roots and patterns.
- Grammatical errors and misspellings in the context of the expression of results of the analysis: The output representation of the system has to be error free in terms of expression and representation of output result.
- Cover all possible cases of syntactic word analyst: The system also should be represent and explain the analysis result of each of analyzed word and there generated words.
- Consistency between analyzed word and its patterns: The system should produce correct and consistent patterns for the analyzed and generated words.

- The result has to be coming from Arabic dictionary: The system should combine the Arabic morphological rules while processing the word with its knowledgebase to reflect a better analysis and generation which measures the trust of morphological analysis result.

3.3 System Architecture and design.

- Complete Functionality of the system
- The ability of word/phrase analysis
- The ability of word- generation
- What is the information represented for analysis output
- How many level of analysis
- Determining if the system is Analyzer and/or Generator.
- Percentage of non-reliance on predefined knowledgebase of affixes, roots and patterns.
    1. An affix is a morpheme that is attached to a word stem to form a new word. Affixes may be derivational, like English -ness and pre-, or inflectional, like English plural -s and past tense -ed. They are bound morphemes by definition; prefixes and suffixes may be separable affixes.
    2. Affixation is, thus, the linguistic process which speakers use to form different words by adding morphemes (affixes) at the beginning (prefix), the middle (infix) or the end (suffix) of words. As an example, the word for "I wrote" is constructed by combining the root k-t-b "write" with the pattern -a-a-tu "I wrote" to form katabtu "I wrote".
    3. Other verbs meaning will typically have the same pattern but with different consonants, "قرأت" "I read", "أكلت" "I ate","ذهبت" "I went", although other patterns are possible (e.g. "شربت" "I drank", "قلت" "I said", "تكلمت" "I spoke", where the sub pattern used to signal the past tense may change but the suffix –"تtu" is always used).
- Percentage of non-reliance on common words (Stop List): Common words (stop word) are the words that are frequently used in Arabic text with the same meaning such as day names, month names, numbers names, adverbs... etc.
- Processing Speed: Performance testing is a subset of performance engineering, an emerging computer science practice which strives to build performance into the implementation, design and architecture of a system. The processing speed can be measured by how many words processed per second.
- Ease of use and integration with larger applications. In engineering, system integration is the bringing together of the component subsystems into one system and ensuring that the subsystems function together as a system. In information technology, systems integration is the process of linking together different computing systems and software applications physically or functionally, to act as a coordinated whole.
- How much the system is capable for use and what are the prerequisites for the system to run.
- The ability to integrate the system within larger applications.
- The ability of modifying some of the system behavior of output or even input procedures and functions. (Customization).
- The ability to add inputs to the system knowledgebase.
- Availability and documentation: Software documentation or source code documentation is written text that accompanies computer software. It either explains how it operates or how to use it, or may mean different things to people in different roles. In terms of Arabic morphological system, it measures the availability of the system and its algorithms for newcomer and researchers considering the cost of commercial systems.
- User Interface (English - Arabic): The goal of human-machine interaction engineering is to produce a user interface which makes it easy, efficient, and enjoyable to operate a machine in the way which produces the desired result. This generally means that the operator needs to provide minimal input to achieve the desired output, and also that the machine minimizes undesired outputs to the human. There are two major factors for judging morphological system interface as follows:
- The Interface language of system itself.

The language used to represent the output of the system in case of analysis or generation.

- Encoding and word representation: Identifying the character encoding used in the system itself for processing and representing the data. As Arabic letters need to be represented in Unicode set; some systems needs to transliterate the input as a preparation for processing step and then revert the transliterated results into Arabic to match user input and user interface.

3.4 Assessment behavior

Assessments are carried out by executing some of the available Arabic morphological systems on a randomly selected Arabic political news article, an Arabic Sport News article “from Al-Ahram newsletter” and chapter number 36 of the Holy Qur’an “Surah Yassin”. We then manually extracted the roots of the test documents’ words to compare results from different stemming systems, thus creating our baseline. Roots extracted were then checked manually in an Arabic dictionary. Voting weights are assigned to each assessment item in order to accurately make comparisons between these algorithms. Each assessment item has to be applied and calculated as per the result of applying the analysis to the sample input words. Table 1, shows assessment items where the voting mark of each individual item is assumed to be 100.Here is the step by step procedure of executing the assessment criteria:

1. Manually extract the roots of the test documents’ words.
2. Assign voting mark for each assessment item.
3. Manually check the extracted roots against Arabic dictionary.
4. Apply each assessment item separately on each of Arabic Morphological system.
5. For the output results, check them manually against Arabic dictionary.

Finally, we applied the assessment items separately on each of Arabic Morphological system and all items have been assigned a maximum value of 100 marks. Each assessment item has been applied and calculated as per system result of applying the analysis of the sample document words. The following table shows a tabulated assessment items.

Table 1. Assessment Criteria Items

	Factor No.	Assessment Criteria	Wight
Input	1	The possibility of analyzing the modern standard texts	100
	2	The possibility of analyzing the common error words	100
	3	The possibility of analyzing new words	100
	4	Processing of Arabized and transliterated words	100
	5	Processing of non- tripartite verbs.	100
Output	6	Covering analysis of all input words	100
	7	Meet all possible cases for analysis	100
	8	Express grammatical function of the affixes	100
	9	Ambiguity and Overlapping of syntactic cases	100
	10	Identifying the root of the word and determining all possible roots	100
	11	Grammatical errors and misspellings in the context of the results of the analysis	100
	12	Cover all possible cases of syntactic word analyst	100
	13	Consistency between analyzed word and its patterns	100
	14	The result has to be coming from Arabic dictionary	100
System Architecture	15	Percentage of non-reliance on predefined knowledgebase of affixes	100
	16	Percentage of non-reliance on common words	100
	17	Processing Speed	100
	18	Ease of use and integration with larger applications	100
	19	Availability, documentation and customization	100
	20	User Interface (English - Arabic)	100
	21	Encoding and word representation	100
Sum			2200

4 PROPOSED ARABIC MORPHOLOGICAL ANALYZER SYSTEM

The proposed system will be a redesign, reuses and enhancement of Al-Khalil Arabic Morphological System[15] based on some of the Arabic morphology analyzer rules designed by Saad(IAMA)[19]. Al-Khalil Arabic morphological system is an open source application written using Java language and published on April 2010. The proposed system uses the modified version of Al-Khalil's knowledgebase. This modification proposed here uses linguistic algorithms and approach used for stem and root identification based on predefined linguistic rules and knowledgebase. The proposed system consists of five components, an Arabic Tokenizer, a Common word& non-derivational nouns extractor, a Stem identifier, a Stem refiner and a Root identifier. The first component, Arabic Tokenizer, receives as input the Arabic text and then converts it into a stream of tokens. The Arabic word may be a verb, noun or common word[24]. Each type may be inflected or not. The first step in the morphology analysis is to identify the common words and the non-derivational nouns. The remaining tokens are assumed to be inflected words.

The method presented is based on Root-And-Pattern techniques which is responsible for both analysis and generation tasks. A word is accepted by the rules applied if it belongs to a correct word in Arabic. Consequently, implementing such rules needs to use the predefined roots and patterns. We have to extract all the morphological rules from the lexicon and implement functionality for each rule. So to realize that implementing, we have to use some operations such as concatenation and union.

This section illustrates the proposed morphological analyzer and the new algorithms for word analysis and generation. Figure1 shows the main components of this system. The proposed system consists following components:

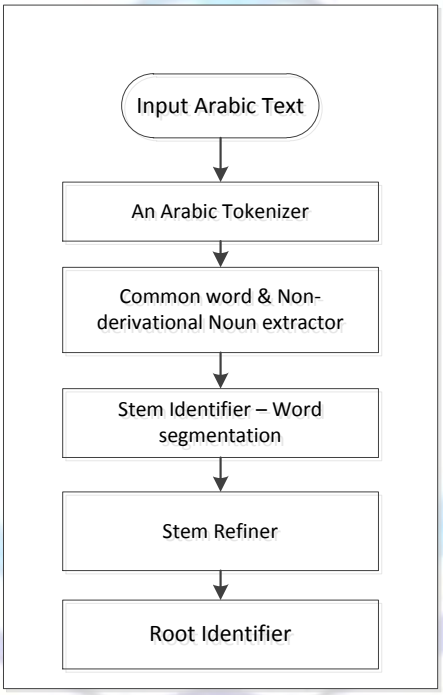

Fig 1: WordAnalyzercomponent.

4.1 Word Tokenizer

Word Tokenizer is first step in the proposed system which takes the input text and splits it into basic components where each component represents one Arabic word with no spaces or punctuation marks. The Arabic word may be a verb, noun or common word. Algorithm 1 illustrates the idea.

- Find the input text and i
- Split i into basic words ws
- **For each** word w in ws find the type of w.
- return a token of w and its non-diacritic form

Figure 2 shows the implementations step by step used in building the Arabic tokenizer part the proposed system.

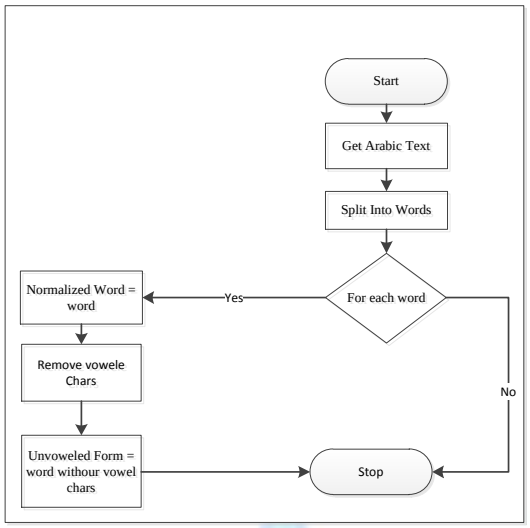

Fig 2: Word Tokenizer

4.2 Word Analyzer

The analyzer receives as input a stream of tokens one by one and applies some linguistic rules to determine the inflection result of given token. Each token type may be inflected or not. The first step in the morphology analysis is to identify the exceptional words, common words (الخ ... أدوات النداء, حروف النصب, حروف الجر) and the non-derivational nouns (علم, اسم جنس الخ...). The remaining tokens are assumed to be inflected words. The inflection means that, either there are affixes attached to the word or there are some inflectional modifications[25].

4.3 Exceptional Words

Exceptional words are the stems assumed to be non-derivational Arabic words stored in knowledgebase like ‘الله’ "Allah" with type stem prefix and suffix tagged as follow.

4.4 Word Segmentation

If the input word is not exceptional word then the analyzer will process the input word as segments. The segmentor is stem identification based component responsible for extract all possible valid Arabic components of the input word where the differences of these components are in affixes. For example the word ‘بسم’ "Besm" can be dealt as two components, ‘بسم’ "In the name of" or ‘سم’ "poison" with ‘بـ’ "with" as a prefix attached with it.

4.5 Stem Identification

To find the stem, the word inflection must be resolved. The Arabic word has to satisfies the following formula[19, 26]:

[Prefix1 | Prefix 2] + Stem + Suffix1 + Suffix 2 + Suffix 3

Where affixes are list of prefixes and suffixes attached at beginning and the end of any Arabic word respectively, the affixes serve for a special linguistic purpose. Based on ordering and linking of the affixes to each other and to the stem, the inflection was resolved.

- **Prefix list**: all possible prefixes that can be attached at the beginning of word. Prefixes are listed in XML file with some properties to identify each one as:
  - *Class*: a flag to determine type of word that this prefix can attach with.
  - *Desc*: is a description of the prefix.
  - *Non-diacritic form*: is the prefix characters without any *diacritization* attached with it.
  - *Diacriticized form*: is a prefix with all possible *diacritization* characters attached with it.
- **Suffix list**: all possible suffixes that can be attached at the end of word. Suffixes are listed in XML file with some properties to identify each one as:
  - *Class*: a flag to determine type of word that this prefix can attach with.
  - *Desc*: is a description of the prefix.
  - *Non-diacritized form*: is the prefix characters without any *diacritization* attached with it.
  - *Diacritized form*: is a prefix with all possible *diacritization* characters attached with it.

The system in hand applies new semantic and linguistic rules to reduce the number of search possibilities. Studying the Arabic affixes carefully will show that there is a semantic contradiction between these affixes determined with affix classes stated in the affix list. For example the prefixes for the present verbs cannot be joined in the same word with the suffixes which can be attached to past verbs only, (e.g. "ت" "t" cannot be joined, as a suffix, to present verbs, "فس" "fas" cannot be joined, as a prefix, to the past verbs or the imperative).The same idea applies to the affixes that can be attached to nouns only and not to verbs. The division of these affixes into contradicted groups, then building separate methods for the non-contradicted groups only, will reduce the searching time considerably.

Table 2. Affix classes

Class	Available	
	Prefix	**Suffix**
C1	Yes	Yes
C2	Yes	Yes
C3	Yes	Yes
N1	Yes	Yes
N2	Yes	No
N3	Yes	No
N4	Yes	No
N5	Yes	No
V1	Yes	Yes
V2	Yes	Yes
V3	Yes	Yes
V4	No	Yes

Another point to note is that the Arabic verbs cannot be more than six characters long[27]. Also they cannot be less than one character long. This is due to the fact that each Arabic verb must correspond to a certain pattern (الوزن), these patterns cannot be more than six characters or less than one character long. On the other hand, verbs which are one character long are rarely used in the modern Arabic texts (e.g 'ع' A'ie" 'الأمر من وعى"). Then we can deduce that the Arabic verbs cannot be less than two characters long. The same idea can be applied to the Arabic derivational nouns(active participle "اسم فاعل"; passive participle "اسم مفعول"; noun of time "اسم زمان" .. etc.)[28].

Now, the following constraints can be formulated which can be applied during the process of the stem identification to determine a validity of stem; where the valid stem has to satisfy the following rules.

- Stem length between 2 and 9 characters
- Stem with prefix of class N cannot be suffixed by V class
- Stem with prefix of class V cannot be suffixed by N class
- Stem with prefix of class N1 , N2, N3 or N5 has to suffixed with any valid suffix

Figure 4 presents the implantation of stem validator for valid segment.

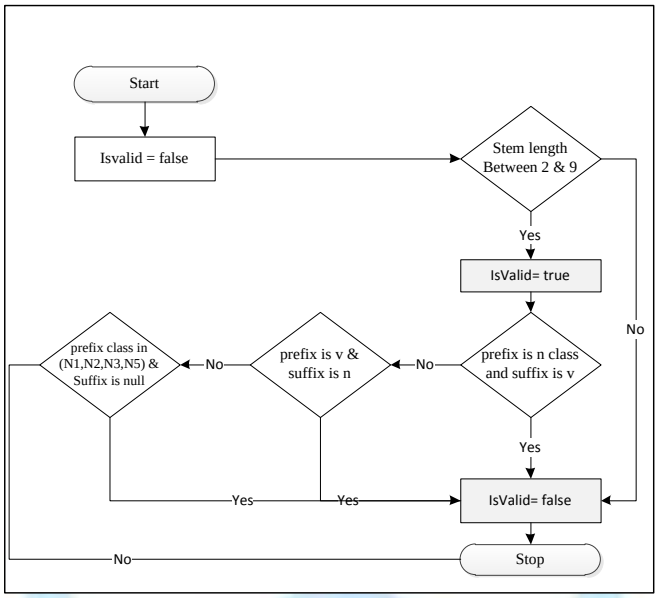

Fig 3: Stem validator

The figure 5 shows the relation between the word length and the total number of transitions before and after applying the constraints. Also the achieved reduction is drawn against the word length.

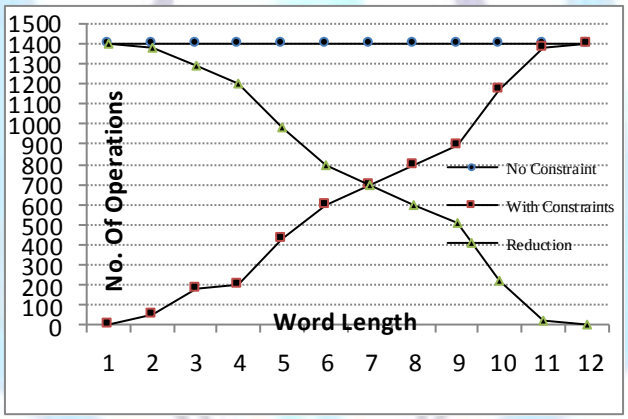

Fig 4: Word length and total number of transitions

The number of transitions in[24]follows the diamond shaped points shown in figure 6. Also if the ATN developed in [25] were used for the Arabic verbs only it will follow the same points. The circles shaped points shows the number of transitions needed by the system in hand after applying the previously mentioned constraints. The triangular shaped points are the reductions[19].

Two contemporary Arabic political news article, Arabic Sport and sample of Holy Qur'an text were used in a statistical experiment. The statistics shows that 45% of the Arabic verbs were 4 characters long. The statistics for the percentage of word length repetitions and the transitions percentage reduction done by the system in hand are shown in figure 6.From the diagram it is apparent that about 95% of the Arabic verbs were between 3 and 6 characters long. Hence, we can deduce that the average transitions percentage reduction for the system in hand is 80% which is the average percentage repetitions for lengths 3, 4, 5 and 6.[19]

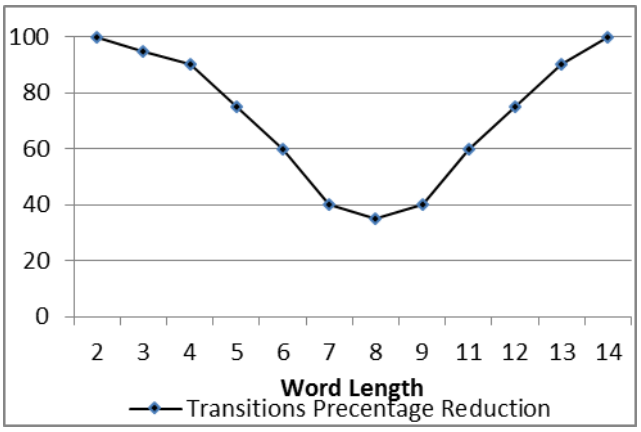

Fig 5: Statistics for Transitions percentage reduction

4.6 Tool Word Analysis

Tool word can be extracted from a predefined list of tool words where each are defined with type of suffixes and prefixes can be attached with and diacritic and non-diacritic form of the word.

The tool word extractor method uses the exact matches to identify all possible tool words form given segment where the segment has to satisfy the following conditions as illustrated in algorithm 6:

The diacritic form of stem equals non- diacritic form property in tool words.

The prefix class and suffix class of the stem in the allowed prefixes and suffixes of the tool word respectively.

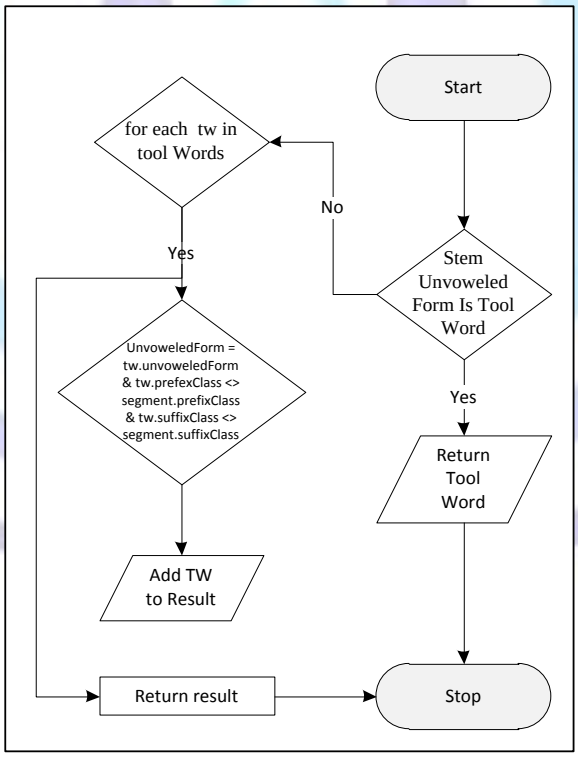

Fig 6: Tool Word Extractor

4.7 Arabic Nouns Analysis

Arabic nouns classified into two types; proper noun and nominal nouns[29],

a- Proper nouns: can be identified of any word only if non- diacritic form of input segment matches the non- diacritic form of predefined list of proper nouns and the segment's prefix is not in class of 'C'.

- **b- Nominal nouns**: using root identification algorithm to find the patterns of the input stem, nominal nouns are assumed to satisfy the following rules:

- The *word with Fathatan, Kasratan or Dammatan must has any valid suffix*
- *The word with Fathatan, Kasratan or Dammatan cannot be prefixed with any prefix in class of N1, N2,N3, or N5*
- *Prefixes of N2, C2 and C3 are not valid with noun categories NCG (13 to 18)*
- *Prefixes of N4 and N5 are not valid with noun categories NCG (13 to 18)*

If Hamza appears in the *diacritized* form of the segment then the Hamza letter must be valid against hamza rule

The *diacritized* form has to satisfy valid contaminative characters.

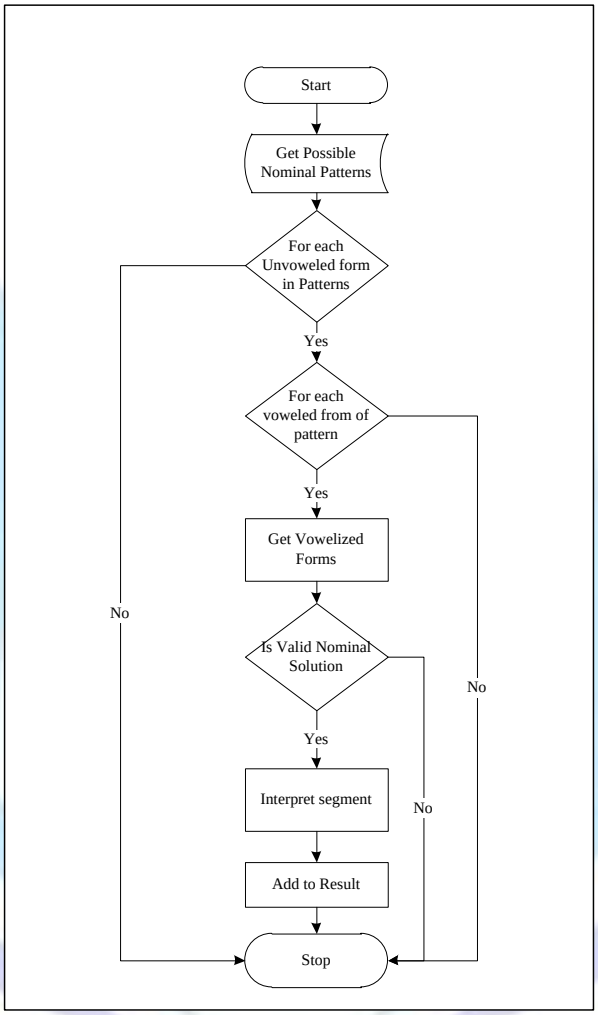

Fig 7: Get Noun Solutions

4.8 Arabic Verbs Analysis

Arabic verbs are those words in Arabic that are built according to the Arabic derivation rules to represent actions in past, current or future tense.

Arabic verbs are assumed to satisfy the following formula:

P1 + Stem + S1 + S2

Where, P1 is the set of conjugates and special articles, S1 is the subject, and S2 is the first object.

The second object is ignored, since one word that contains a verb plus a subject plus the first object plus the second object, appears rarely in the modern Arabic texts[19]. The proposed system uses the same algorithm of matching the nominal patterns to identify all possible roots of the verb.

The following table lists the verb and nouns affix types, classified according to some semantic contradiction and POS tagging criteria.

4.9 Stem Refiner

The proposed system serves for the purpose of omitting the affixes. To deal with the stem omitted or converted characters a new component has been added. An important fact will be explained which results in the need of such component. The

fact is that, Arabic verbs and derivational nouns have finite number of patterns for those roots with no weak characters (ا, و, ى) or hamza (ء). If the roots have weak characters or hamza, these basic patterns will have variants. One solution to find the root is to write down all these variants and then matching the word with them[24]. Applying this approach, 101 pattern categories to account for all pattern variants were used in; then matching the stem with these patterns to find a root. Figure 7 lists the conversion rules for the Arabic verbs. The same approach can be also applied to Arabic nouns. The omitted characters are manipulated in the next section. The proposed system stores the available conversion rules with all patterns

4.10 Root identification

This section is concerned with the root identifier component. The root identifier component is responsible for matching the stems with the stored patterns. Then the root can be easily identified. Another approach were to develop an algorithm that examines every combination of characters from the stem and check if this combination is a valid root[30], then it produces the pattern. This approach will be superior to that used in[24], but it still needs to store the patterns to check if the legality of produced pattern.

The approach in hand needs only the basic patterns set as mentioned previously since, the stem refiner component restores any converted characters in the stem due to the presence of the weak characters or the hamza in the root. The basic patterns for Arabic verbs are listed in the table 3:

Table 3. Basic Patterns for Arabic Verbs

Pattern Length	Patterns Text		Pattern Length	Patterns Text	
5	أنيت+فاعل	Anyt+fael	3	فعل	Fala
5	أنيت+نفعل	Anyt+nfaal	4	أفعل	Afaal
5	أنيت+فتعل	Anyt+ftaal	4	فاعل	Fael
5	أنيت+فعلّ	Anyt+falaa	4	فعّل	Faal
6	أنيت+تفعّل	Anyt+tfaala	4	افعل	Afaal
6	أنيت+تفاعل	Anyt+tfala	4	أنيت+فعل	Anyt+fala
6	استفعل	istfala	5	انفعل	Infala
6	أنيت+ستفعل	Anyt+stfala	5	افتعل	Ftaala
4	فعلل	falala	5	افعلّ	Falla
5	تفعلل	Tfalala	5	تفعّل	Tafaala
5	أنيت+فعلل	Anyt+falala	5	تفاعل	Tafaala
6	أنيت+تفعلل	Anyt+Tfalala	5	أنيت+فعّل	Anyt+faal

The final point is to deal with the omitted characters. The omitted characters in Arabic verbs are the weak characters or the hamza (if found in the root). This omission occurs under certain circumstances. This will cause the basic patterns to have variants. One solution to this problem is again to write these variants. Another one depends on a new matching Algorithm. The algorithm is built on the fact that the omitted characters must be elements from the set of weak characters and the hamza. The algorithm is further illustrated in the following points:

- *Find the stem length L.*
- *Get un diacritized patterns whose lengths are L with related rules.*
- *Get all the pattern excess characters (all characters except for ف ع ل)*
- *build result pattern list R*
- *Remove these characters form the corresponding locations in the stem.*
- *for each rule in pattern rules replace the corresponding numeric characters with stem character to build new root r.*
- *for new roots r replaces (أ ، ؤ ، ئ) with (ء)*
- *match r in non-diacritized patterns*
- *Repeat for all patterns in step 2.*

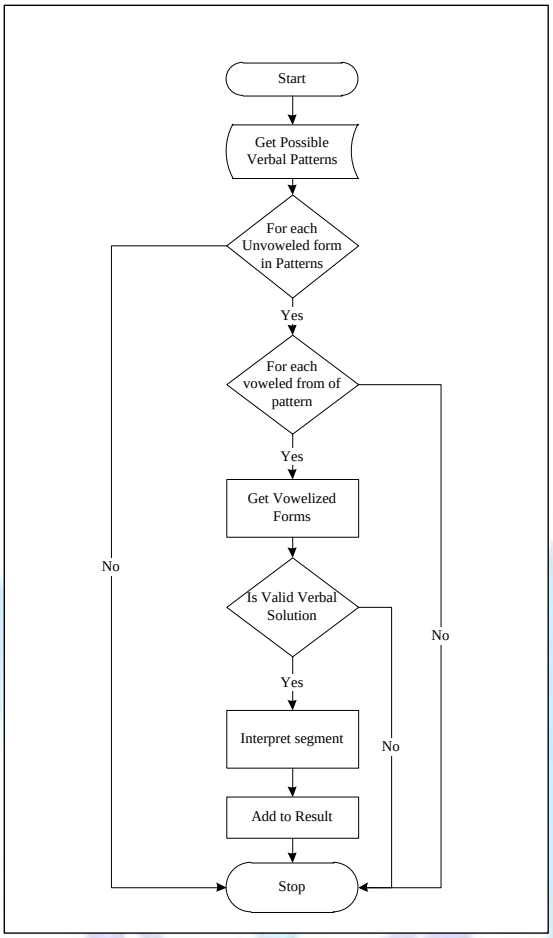

Fig 8: Get Verbal Solutions

5 User Interface

The proposed system provided as dynamic link library to be easily integrated with any natural language processing system with simple user interface used for test purposes. The system can accept one or more word; using tokenizer module it splits the input text as stream of word. The system extracts diacritized form, The Stem, Word root, Prefix, Suffix, Word type, Word pattern, and POS as result of analysis.

6 EXPERIMENTS AND DISCUSSIONS

Experiments are done by executing both the proposed system and the available Arabic morphological systems on a randomly selected contemporary Arabic political news article, Arabic Sport News article "from Al-Ahram newsletter" and chapter number 36 of the Holy Qur'an "Surah Yassin". Each test document contains domain specific words and represents contemporary and standard Arabic. The test documents contain 11000 distinct words. We manually extracted the roots of the test documents' words to compare results for each stemming algorithm. Roots extracted have been check against Arabic dictionary.

Table 4, shows a detailed analysis been done for the sample test documents, the Qur'an chapter; where the test documents are taken form Al-Ahram daily newspaper publish in Egypt.

The analysis also show that function words such as "فى" "fi", "من""min", "بين" "bian" are most frequent words in any Arabic text. In other hand, nonfunctional words with high frequency such as "الإفريقية" "al-afiriqiah", "القمة" "al-Qemah" and other words out of 30 most frequent tokens as shown in table 4 gives a general idea about the main topic of the article.

Simple tokenization is applied for the text of the baseline documents can be used to test any stemming algorithm smoothly and correctly.

Table 4. Top 10 Frequent tokens

Freq.	Is Common word	Word
11	Yes	في
9	Yes	من

6	No	علي
5	No	عن
5	No	التي
5	Yes	ان
5	Yes	على
5	No	أن
5	No	الذي
5	No	إلي

Table 4, shows a detailed analysis been done for the sample test documents, the Qur'an chapter; where the test documents are taken form Al-Ahram daily newspaper publish in Egypt.

6.1 The stemming algorithms under evaluation

Al-Khalil Morpho System

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 [15]. The system can analyzes up to 10 words per second in small text up to 35 words per second in larger documents.

RDI Arabic Morphological Analyzer

The main RDI's NLP core engine is the basis of Arabic morphological analysis, Arabic POS tagging, and Arabic Lexical Semantic Analysis [31]. "ArabMorpho" is a morpheme-based lexical analyzer/synthesizer which distinguishes it from its vocabulary-based rivals and boosts its flexibility.

Arabic Morphological Analyzer

Arabic Morphological analyzer is published system over the internet based on Quatrab system that can analyze, generate and categorize Arabic words in phrases[16].

Xerox Morphology

Is "based on solid and innovative finite-state technology"[5]. It adopts the root-and-pattern approach and includes 4,930 roots and 400 patterns, effectively generating 90,000 stems. Its main advantage is that it is rule based with wide coverage. It also reconstructs vowel marks and provides an English glossary for each word. At Xerox, the treatment of Arabic starts with a lexical grammar where prefixes and suffixes concatenate to stems in the usual way, and where stems are, similarly, represented as a concatenation of a root and a pattern[2, 32]

Shereen Khoja Stemmer

Shereen Khoja's stemmer[17]is available in form of java open source application. Kahoka's stemmer removes the longest suffix and the longest prefix. It then matches the remaining word with verbal and noun patterns, to extract the root. The stemmer makes use of several linguistic data files such as a list of all diacritic characters, punctuation characters, definite articles, and 168 stop words[18].

Sakhr Arabic Morphological Processor

It is a morphological analyzer synthesizer that provides basic analysis for a single Arabic word, covering the whole range of modern and classical Arabic. The analyzer identifies all possible stem forms of a word; i.e. extracting its basic form stripped from the affixes, the morphological data such as root, the Morphological Pattern (MP), and its part of speech.

Tri-literal Root Extraction Algorithm

Al-Shalabi, Kanaan and Al-Serhan developed a root extraction algorithm which does not use any dictionary. It depends on assigning weights for a word's letters multiplied by the letter's position [33], Consonants were assigned a weight of zero and different weights were assigned to the letters grouped in the word "سألتمونيها" where all affixes are formed by combinations of these letters. The algorithm selects the letters with the lowest weights as root letters [33].

AraFlex Arabic Morphological Analyz

Modification of Buckwalter's Arabic Morphological Analyzer for analysis of Arabic words and roots. The system published over internet at http://lexanalysis.com/araflex/araflex.html. AraFlex Arabic morphological analyzer uses only web interface and has no open source code for integration in larger applications. AraFlex cannot analyze the more complex words that have more affixes.

Tim Buckwalter Morphological analyzer

Tim Buckwalter developed a morphological analyzer for Arabic. Buckwalter compiled a single lexicon of all prefixes and a corresponding unified lexicon for suffixes instead of compiling numerous lexicons of prefixes and suffix morphemes. He included short vowels and diacritics in the lexicons [28].

TAGGAR Morphology

This module is significant in a system of voice synthesis starting from the text, because the insertion of the pauses and the generation of the prosodic markers can be made only if one has a minimum of grammatical information on each word of the sentence. The texts subjected to the entry of system are correctly diacritized. This system has as a role to identify a word given (starting from a diacritized text) and to affect a morpho-syntactic label to him (unaccomplished verb, pronoun,).

ElixirFM Functional Arabic morphology

ElixirFM can process words of Modern Written Arabic using four different modes. Here, you can learn how to use these modes for various purposes. The implementation of Functional Arabic Morphology written in Haskell and Perl [34].

Sarf

Sarf Arabic morphology system is an open source application. It is Arabic generator, as it generates verbs, nominal derivatives, and there conjugations for trilateral and quadrilateral verbs based on Arabic morphological rules and predefined knowledgebase.

6.2 The methodology of Proposed system

The proposed system is an enhancement of Al-Khalil Arabic Morphological System based on the improved Arabic morphology analyzer rules developed in "Improved Arabic morphology analyzer (IAMA)". As explored through the previous sections; the proposed system been developed by combining the two systems philosophy with enhancements of methodology using new semantic rules of the Arabic language to reduce the search steps. By using these new semantic rules of Arabic; the search time has been considerably reduced compared to the reused system. The following is an explanation of the method used for each system:

Purposed System (GPAMA): Uses a combination of Arabic rules and knowledgebase lookup where these knowledgebase is represented as XML files for any further updates.

Al-Khalil System: knowledgebase lookup

IAMA: Uses Knowledgebase of common words and non-derivational nouns, other words uses ATNs with Arabic rules.

6.3 The Algorithms used in Proposed system:

As we stated above; the proposed system is a reuse of Al-Khalil system with extending and enhancements of IAMA system. The system achieved analysis speed of 300 words per second, while Al-Khalil system achieved 35 words per second on the same processor.

Compared with Al-Khalil System:

The purposed system uses a combination of Arabic rules and knowledge base lookup while Al-Khalil system uses knowledge base for data extraction and tokens lookups.

Compared with IAMAMorphology analyzer

Applying the assessment criteria on purposed system compared with Al-Khalil System shows that there are significant advantages for both.

Table 5 the proposed system achieved 99.96% precision compared with Al-Khalil system which achieved 98.27% considering only the analyzed words where the precision has been calculated based on Arabic dictionary.

Table 5. No. of Generated words and Arabic Dictionary matched results

System	Analyzed words	Generated words	correct words	Recall	Precision
GPAMA (2014)	10567	140587	140543	87.28	99.96
Xerox Morphology	10374	137298	135513	84.16	98.7
RDI	9428	102392	94466	58.67	92.26
AMA	9035	160695	156582	97.24*	97.44
Al-Khalil	8446	118109	116062	72.08	98.27
Sakhr	8348	101884	101476	63.09	99.6
Khoja Stemmer	8250	11000	10397	6.45	94.52

System	Analyzed words	Generated words	correct words	Recall	Precision
Tri-Literal	7798	13808	13591	8.44	98.43
AraFlex	7660	37651	32737	20.33	86.95
Buckwalter	7464	87014	86518	53.73	99.43
TAGGAR	6285	91034	70642	43.87	77.6
ElixirFM	6187	94140	91918	57.09	97.64
Sarf	628	8164	8164	5.07	100.00

In table 5, the proposed system achieved 96.06% accuracy compared with Xerox system which achieved 94.31% considering only the analyzed words where the accuracy has been calculated based on Arabic dictionary.

Where the precisions are calculated based on correct words against the generated words where:

RECALL (R) is the ratio of the number of relevant generated words (A) to the total number of expected words to be generated linguistically from Arabic dictionary (B) and expressed as a percentage.

$R = (C / E) \times 100$

Where: R is Recall, C is correct words, and E is expected words

PRECISION is the ratio of the number of correct generated words to the total number of generated words and expressed as a percentage.

$P = (C / G) \times 100$

Where: P is Precision, C is correct words, and G is total generated words.

As mentioned, AMA recorded high percentage of recall as it removes all diacritics of input word, so it generates all possible words of the input ignoring the original diacritics which cause over generation problem.

6.4 Enhancments and opinion finiding of the proposed system.

As per the result of applying a set of evaluation criteria selected as a baseline for testing and comparing the result of running the proposed approach with other systems used in this study; the proposed method achieved better improvement over the state of the art systems where "Al-Khalil Morphological System" is one of the reused systems. The followings are the opinion findings:

Processing Speed:

The proposed system can analyze word or sentences typed in the text area. The system can analyzes up to 1500 words per second in small text up to 3000 words per second in larger documents

Covering analysis of all input words:

The proposed system has possibility of analyzing the common error words as "الجنوبى Western" and "إجتماع Meeting", where the "الجنوبي southern" with dotted yah and the word "اجتماع" with "ا Alef" is commonly written without dots or with hamza respectively.

Meet all possible cases for analysis:

As a result of analyzing the word "أبه abh"; GPAMA generated 41 possible words while other systems missed the following results:

"أبِهِAbihieh" : "أ: Interrogative particle + "بِ" Preposition + "ـهِ" pronoun

"أبٌ" : "أَبُهُ" : nominative + "ـه" : prepositional pronoun

"أبً" : "أَبَهُ" : accusative + "ـه" : prepositional pronoun

As a result of analyzing the word "عالم Alm"; GPAMA generated 8 possible words while other systems missed the results "عالَمAlam"

As a result of analyzing the word " جلسة Glsh" GPAMA generated 19 possible words while other systems missed the results of infinitive named entity "جِلْسَه Gilsah"

As a result of analyzing the word "مدارس madars"; GPAMA generated 19 possible words while other systems missed the results of infinitive named entity "مُدارِسُ Mudareso" and "مُدارَسُMudaraso"

Express grammatical function of the affixes:

As a result of analyzing the word "احترمتehtrmt"; GPAMA generated 8 possible words contains "احترمتْ" and expressed the suffix "ت" as feminine particle while other systems expressed this "ت" as singular subject of Third feminine person.

As a result of analyzing the word "ستذكرونناstzkronna"; GPAMA generated 3 possible words all suffixed with "ون" nominative plural: and "نا" pronoun of first person; while other system expressed the suffix "ون" as plural masculine subject of Second person.

Other enhancement

Identifying the root of the word and determining all possible roots

Grammatical errors and misspellings in the context of the results of the analysis

Cover all possible cases of syntactic word analyst

Consistency between analyzed word and its patterns

The result of analysis is fully coming from Arabic dictionary without any mistakes.

The proposed system has 150 times faster than the reused where our system can analyze 1500 words per second compared with Al-Khalil system which can only process 30 words per second on same environment.

Ease of use and integration with larger applications as it is provided as dynamic link library.

The system provides integration with any User Interface

Encoding and word representation has been represented with non-Unicode characters

Table 6. GPAMA vs. Existing systems assessment result

	Factor No.	Morphology System Score												
		Al-Khalil	Sarf	AMA	Khoja	AraFlex	ElixirFM	Buck-walter	RDI	Xerox	TAGGAR	Sakhr	Tri-Literal	GPAMA
Input	1	75	-	80	50	50	50	60	80	78	30	80	30	95
	2	85	-	90	20	15	20	10	50	40	20	50	20	98
	3	30	-	20	0	0	0	40	30	30	30	25	30	70
	4	10	-	5	0	0	0	0	0	0	0	0	0	65
	5	90	-	85	80	80	75	80	85	85	80	85	80	92
Output	6	80	-	85	78	72	58	70	89	87	60	79	74	99
	7	87	-	85	0	85	75	80	85	80	75	89	80	99
	8	92	-	80	0	0	60	75	90	78	80	90	80	99
	9	90	-	35	30	30	30	70	95	87	75	90	80	98
	10	85	-	95	30	35	25	65	95	80	80	90	75	97
	11	85	-	98	90	90	95	90	95	90	90	85	75	99
	12	45	-	40	0	20	70	70	65	85	80	75	85	97
	13	80	-	95	0	95	95	85	90	95	90	85	78	98
	14	86		97	80	80	85	85	98	90	95	95	85	97
Architecture and design	15	0	0	0	0	0	0	0	0	0	0	0	0	0
	16	0	0	0	0	0	0	0	0	0	0	0	0	0
	17	35	0	0	30	0	0	0	0	0	0	0	0	98
	18	60	60	30	60	30	30	30	30	30	30	30	30	99
	19	70	85	50	70	70	70	85	70	65	70	70	60	95
	20	50	50	50	50	80	50	80	50	50	50	50	50	90
	21	50	50	10	10	50	50	30	50	40	50	50	50	90
Sum		1285	245	1130	678	882	938	1105	1247	1190	1085	1218	1062	1775

Table 6 shows system scores in each factor in the proposed assessment criteria. In Figure 9 shows the result of applying the developed baseline items on each system where our system achieved 1775 out of 2200 points, with speed of 300

input words per second where Al-Khalil system recorded 1280 points with speed of only 35 words per second[15]. It should be noted that all test related to the speed of the systems are done over the same environment.

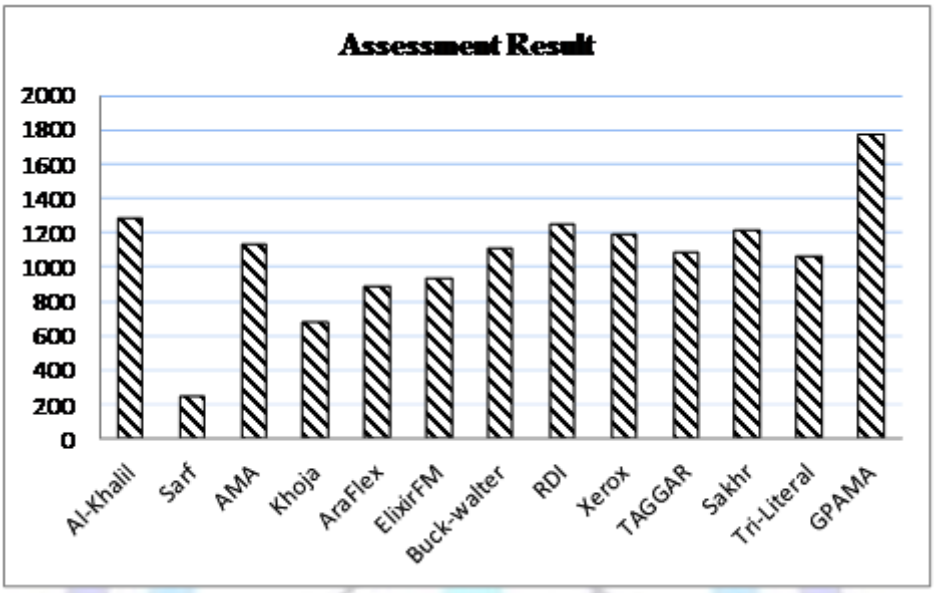

Fig 9: Assessment result of GPAMA Vs. Al-Khalil

7 Conclusion and Future Research

The proposed assessment criteria are adapted to measure Arabic Morphological Analyzers with some features intended for integration with lager applications in natural language processing. Many other criteria can be added to the proposed items and may vary in weight and phase of testing; similar to the source code related metrics used for measuring the system as a product.

The stemming algorithms involved in the experiments agreed and generate analysis for simple roots that do not require detailed analysis. So, more detailed analysis and enhancements are recommended as future work.

Most stemming algorithms are designed for information retrieval systems where accuracy of the stemmers is not important issue[33]. On the other hand, accuracy is vital for natural language processing. The accuracy rates show that the best algorithm failed to achieve accuracy rate of more than 65%. This proves that more research is required, this prove the need for developing the proposed system.

The new algorithms used in the proposed morphological analyzer are minimal in searching time as explored in the previous sections. The proposed system is directed to the standard modern Arabic that covers non- diacritized, partially diacritized and fully diacritized words. It can resolve the inflected Arabic word, identifies its root, finds its pattern and POS tagging.

By rating the proposed system with this baseline standard measurement showed that it achieved better word analysis improvement, and minimized the searching time which yielded a better performance with processing speed of up to 1500 words per second in small text up to 3000 words per second in larger documents.

Small components can be added to this system so that it will be capable of finding and extracting named entities and many other word attributes; where named entity is one or more word that seeks to locate and classify atomic elements in text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, and percentages. This can be done by using some modification of the Tokenizer part of the proposed system to identify the basic elements and adding more information in knowledgebase for names.

REFERENCES

[1] Kiraz, G.A., Computational Nonlinear Morphology; With Emphasis on SemiticLanguages. Studies in Natural Language Processing, ed. I. Branimir Boguraev, T.J. Watson Research Center and L.D.C. Steven Bird, University of Pennsylvania2004, The Edinburgh Building, Cambridge CB2 2RU, UK: The press syndicate of the University of Cambridge, the Pitt building, Trumpington street, Cambridge, United Kingdom.

[2] Beesley, K.R. Arabic Morphological Analysis on the Internet. in 6th International Conference and Exhibition on Multi-lingual Computing. 1998a. Cambridge.

[3] Buckwalter, T. Buckwalter Arabic Morphological Analyzer Version 1.0. Linguistic Data Consortium, 2002.

[4] Watson, J.C.E., The Phonology and Morphology of Arabic. The phonology of the world's languages, ed. J. Durand2007, New York, United States: Oxford University Press.

[5] Mohammed, A.A., An Ambiguity-Controlled Morphological Analyzer for Modern Standard Arabic Modelling Finite State Networks, 2006, School of Informatics, The University of Manchester.

[6] 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.

[7] Soudi, A., V. Cavalli-Sforza, and A. Jamari. A Computational Lexeme-Based Treatment of Arabic Morphology. in Arabic Natural Language Processing Workshop, Conference of the Association for Computational Linguistics (ACL 2001). 2001. Toulouse, France.

[8] Soudi, A., A.v.d. Bosch, and G.u. Neumann, Arabic Computational Morphology; Knowledge-based and Empirical Methods. Text, Speech and Language Technology, ed. N. Ide, et al. Vol. 38. 2007, P.O. Box 17, 3300 AA Dordrecht, The Netherlands: Springer. 307.

[9] Khaled, S.F. and R.A. Ahmed, Lexical Analysis of Inflected Arabic Words using Exhaustive Search of an Augmented Transition Network. Software Practice and Experience, 1993. 23(6).

[10] Roark, B. and R. Sproat, Computational Approaches to Morphology and Syntax2007, United States: Oxford University Press, New York.

[11] Al-Kabi, M.N., Q.A. Al-Radaideh, and K.W. Akkawi, Benchmarking and assessing the performance of Arabic stemmers. Journal of Information Science, 2011. 37(111).

[12] Mazrui, A., et al., Morphological analysis system specifications, in Meeting of experts in computational morphological analyzers for the Arabic language2010: Damascus.

[13] Desouki, M.S., Mechanism for assessing morphological analyzer (In Arabic), in Meeting of experts in computational morphological analyzers for the Arabic language2009, The Arab League Educational, Cultural and Scientific Organisation (ALECSO) - King Abdulaziz City for Science and Technology: Damascus.

[14] Frakes, W.B. and C.J. Fox. Strength and Similarity of Affix Removal Stemming Algorithms. in Proceedings of the Annual Conference on Research and Development in Information Retrieval. 2003. ACM SIGIR Forum

[15] 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).

[16] Zarrouki, T. Arabic Morphology. 2010 5-6-2102]; Available from: http://tahadz.wordpress.com/.

[17] 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.

[18] Larkey, L.S., Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. 2002.

[19] Saad, E.-S.M., et al. An Improved Arabic morphology analyzer. 2005.

[20] Zayed, M., Report in the Arabic Morphological Analyzers, in Meeting of experts in computational morphological analyzers for the Arabic language 2009, The Arab League Educational, Cultural and Scientific Organisation (ALECSO) - King Abdulaziz City for Science and Technology: Damascus.

[21] Mushira Eid, C.H., Perspectives on Arabic Linguistics V: Papers from the Fifth Annual Symposium on Arabic Linguistics, Volume 51993: John Benjamins Publishing Company.

[22] Elgibali, A., K. Versteegh, and M. Eid, Encyclopedia of Arabic Language and Linguistics. illustrated ed2009: Brill Academic Pub. 3250.

[23] Eid, M., V. Cantarino, and K. Walters, Perspectives on Arabic Linguistics VI: Papers from the Sixth Annual Symposium on Arabic Linguistics, Volume 4. illustrated ed. Vol. 4. 1994: John Benjamins Publishing Company. 238

[24] Ahmed, H., Developing Text Retrieval System Using NLP, in ISSR - Computer Science and Information2000, Cairo: Cairo.

[25] Farouk, A., Developing an Arabic Parser in a Multilingual Machine Translation system, in Faculty of Computers and Information1999, Cairo.

[26] Abdeldayem, A.M., 1999أبنية الأسماء والأفعال والمصادر لابن القطاع الصقلى: Egyptian National Library, Cairo.

[27] Alnahawi, J.A., 1 .الشافية فى علم الصرف، تحقيق حسن أحمد العثمانst Edition ed1995, Mecca: Royal Library.

[28] Sawalha, M. and E. Atwell, Comparative Evaluation of Arabic Language Morphological Analysers and Stemmers. 2010.

[29] Hamalawy, A.b.M., 12 .شذا العرف في فن الصرفth Edition ed1957: Dar El-Kayan.

[30] Al-Fedaghi, S.S. and F.S. Al-Anzi A new Algorithm to generate the Arabic Root-Pattern forms. 1988.

[31] RDI. RDI - Research and Development. 5-6-2012]; Available from: http://www.rdi-eg.com/rdi/technologies/arabic_nlp.htm.

[32] 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 19981998b: Champaign, IL.

[33] Sawalha, M. and E. Atwell. Comparative Evaluation of Arabic Language Morphological Analysers and Stemmers. in COLING 2008 22nd International Conference on Comptational Linguistics. 2008. Manchester: Coling 2008 Organizing Committee.

[34] Smrˇz, O., Functional Arabic Morphology, Formal System and Implementation, in Institute of formal and applied linguistics, Faculty of mathematics and physics2009, Charles University in Prague: Prague.

Author' biography with Photo

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