J Inf Process Syst, Vol.12, No.3, pp.358~380, September 2016
http://dx.doi.org/10.3745/JIPS.02.0041

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

A Hybrid Approach for the Morpho-Lexical Disambiguation of Arabic

Kheira Zineb Bousmaha*, Mustapha Kamel Rahmouni*,
Belkacem Kouninef**, and Lamia Belguith Hadrich***

Abstract

In order to considerably reduce the ambiguity rate, we propose in this article a disambiguation approach that is based on the selection of the right diacritics at different analysis levels. This hybrid approach combines a linguistic approach with a multi-criteria decision one and could be considered as an alternative choice to solve the morpho-lexical ambiguity problem regardless of the diacritics rate of the processed text. As to its evaluation, we tried the disambiguation on the online Alkhalil morphological analyzer (the proposed approach can be used on any morphological analyzer of the Arabic language) and obtained encouraging results with an F-measure of more than 80%.

Keywords

Alkhalil Morphological Analyzer, Approach to Multi-Criteria Decision (MCA), Arabic Language Processing (ALP), Augmented Transition Networks (ATNs), Contextual Exploration, Tagging, Diacritization, Disambiguation Method, Segmentation

1. Introduction

Ambiguity is the characteristic of a word that can be subject to various interpretations or several grammatical labels, which constitutes a key obstacle in the understanding of texts. Ambiguity is a central and current issue in the morphosyntactic analysis of Arabic. The analyzers are frequently confronted with situations of ambiguity at all levels of the analysis: 1) at the lexical level, where ambiguity is mainly associated with segmentations into lexical units, especially with homography, although the pronunciation is different and poly-categorical, plus the lack of diacritization and inflectional and agglutinative morphology; 2) at the syntactic level, where ambiguity is a result of the wealth of syntactic constructions and their multiple interpretations; and 3) at the semantic level, where ambiguity is mainly due to the multiple possible meanings.

Properly determining the meaning of an ambiguous word is not as simple as it appears. What should be done to disambiguate a word? We have to find a way to define the possible meanings of the word since we have to assign an appropriate meaning to each occurrence of the ambiguous word. The need

Manuscript received April 3, 2015; first revision August 7, 2015; accepted October 15, 2015.
Corresponding Author: Kheira-Zineb Bousmaha (kzbousmaha@yahoo.fr)
* Dept. of Computer Science, LRIIR, University of Oran1, Ahmed Ben Bella, Oran, Algeria (kzbousmaha@yahoo.fr, kamelrahmouni1946@gmail.com)
** National Institute of Telecommunication and Information and Communication Technology of Oran (INTTIC), Algeria (bkouninef@ito.dz)
*** Dept. of Computer Science at Faculty of Economics and Management of Sfax (FSEGS), University of Sfax,Tunisia (l.belguith@fsegs.rnu.tn)

for a disambiguation method helps create an efficient, robust, fast, and less ambiguous system for proper grammatical analysis.

Different methods of disambiguation have been studied in [1-4]. However, these approaches are limited when disambiguation is conducted on a text in Arabic. In fact, the main feature of the Arabic language is that it is an agglutinated language and 'vowelized', where without diacritics, it is difficult to distinguish the meaning and function of words and so the result is high ambiguity.

We will focus on the morpho-lexical ambiguity related to the automatic restitution of diacritics. We propose in this paper a new hybrid approach that combines the linguistic disambiguation method based on contextual exploration at the segmentation level and the augmented transition network (ATN) techniques at the syntactic level in order to a yield formal method of aggregating the multi-criteria decision aid analysis for the restitution of diacritizations. Our approach for disambiguation is based on choosing the right diacritics in various analyses to remove any ambiguity of the word and thus, recognize its grammatical and lexical features.

The rest of this paper is organized as follows: Section 2 describes some characteristics of the Arabic language, especially those related to diacritization. Section 3 deals with work related to the automatic analysis of Arabic, where we present some works that deal with lexical and morphological disambiguation and mainly the disambiguation approaches that have been adapted for diacritization. In Section 4, we present our morpho-lexical disambiguation approach and the analyzer Alkahlil+ proposed. We then depict the multi-criteria decision process as applied to disambiguation. In Section 5, we present the experiments we conducted to evaluate our system along with the results obtained and compare these results with other leading work. Finally, we present the conclusion and some assessments of this work in Section 6.

2. Several Characteristics of the Arabic Language

The Arabic language is the 5th most-used language in the world. It is both challenging and interesting. It is especially challenging because of its complex linguistic structure. The Arabic language is composed of nouns, verbs and particles. Nouns and verbs are morphemes and derived from around 10,000 roots [5]. Particles are used to complete the meaning of verbs and nouns. In Arabic, many stem words can be derived from the finite Arabic root (usually a three-letter word) consonant combinations into known patterns or schemes [6]. Different words, with different meanings and pronunciations, often differ only by their diacritics. Arabic is written from right to left, its letters changing shape according to their position in the word. The Arabic language has rich and complex morphological, grammatical, and semantic aspects since it is a highly inflectional and derivational language that includes root, prefixes, suffixes, and clitics—all of which makes morphological analysis a very complex task. Like Italian, Spanish, Chinese, and Japanese, Arabic is a pro-drop language, that is, it allows subject pronouns to drop the subject or delete it, and like Chinese, Japanese, and Korean there are no capital letters in Arabic. Besides the classical phenomena like coordination, anaphora, and ellipsis [7] that exist in Latin languages, there are complexities specific to the Arabic language. These complexities generate problems in its different process tasks, such as complex morphology [8]; the absence of short vowels; the agglutination[1], which increases the syntactic difficulties since it leads to exceptional

[1] In Arabic, articles, prepositions, pronouns, etc., can be affixed to adjectives, nouns, verbs, and particles to which they are related. The agglutinative form can constitute a whole sentence, as for instance "واستقبلهم" (Then he welcomed them), and it requires some specific treatments to find their correct syntactic structure.

structures; the free order[2] language, which causes artificial syntactic ambiguities and complicates the grammar construction; and other issues.

One of the complexities of the Arabic language is the lack of short vowels in the text. Table 1 shows the basic set of Arabic diacritics of which there are eight basic ones representing short vowels, nunation (for doubled case endings), and syllabification marks [10].

Table 1. The basic Arabic diacritics

Type of diacritic	Diacritic	Example on a letter	Pronunciation
Short vowel	Fatha	بَ	/b//a/
	Kasra	بِ	/b//i/
	Damma	بُ	/b//u/
Doubled case ending (Tanween)	Tanween Fatha	بً	/b//an/
	Tanween Kasra	بٍ	/b//in/
	Tanween Damma	بٌ	/b//un/
Syllabification marks	Sukuun	بْ	No vowel: /b/
	Shadda	بّ	Consonant doubling: /b//b/

Nunation (Tanween) can only be placed at the end of the word. The gemination (Shaddah) is used to duplicate the letter it is placed on phonetically.

They usually are absent in written Arabic, which generates several cases of lexical and morphological ambiguities concerning the meaning of the word and difficulty in identifying its function in the sentence (distinguishing between the subject and the complement, classifying the POS, etc.).

Example 1 presents an example of an Arabic sentence transcribed without diacritics, in (a) the word "كتب"(ktb) possesses a lot of possible forms that have valid interpretations when adding diacritics. It can be interpreted as the verb “to write/kataba/” and can also be interpreted in the noun form as “books/kutubun.” In Table 2 the Arabic word "مدرسة" (mdrs) is shown. It has three meanings as a noun. One word, without diacritics, could have the same spelling and POS tag but a different lexical sense.

Example 1: كتب كثيرة في المدرسة / many books in the school/

(a) The word with no diacritics كتب (ktb) has 21 potentialdiacritizations, representing 9 different grammatical categories [11]: “Kataba” (he wrote); “Kutiba” (It was written); “Kutub” (books); “Katb” (written); “Kattaba” (He was writing); “Kuttiba” (to write-factitive form); “Kattib” (am writing), etc.

This example illustrates that it is not possible to automatically diacritize a word based only on its past context. The successor words often must be considered in order to achieve the correct analysis and diacritization. Moreover, it is often necessary to do a long-range context analysis to accomplish this goal.

(b) For a category and/or grammatical feature of a given word in a given sentence, several diacritisations are possible, which means that there can be many meanings for the word and for the sentence.

[2] The grammar rules should provide all possible combinations to describe the word order in the sentence [9]. The basic order of Arabic words in a sentence is Verb–Subject-Object (VSO). However, other orders are possible: SVO and VOS. An example of a very simple sentence with the words in a different order, all the form being correctly is shown below.
Sentence order 1: في الحديقة يلعب الطفل / in the garden play the boy.
Sentence order 2: يلعب الطفل في الحديقة/ play the boy in the garden.
Sentence order 3: الطفل يلعب في الحديقة/ the boy play in the garden.

Table 2. Different meaning of word مدرسة

Lexical unit	1st interpretation		2nd interpretation		3rd interpretation	
مدرسة	مَدرَسَة	School	مُدَرَسَة	Taught	مُدَرسَة	Teacher

Debili's statistics [12] show that there is an average of 11.6 possible diacritizations for every undiacritized word when analyzing a text of 23,000 script forms. In regards to morphological ambiguity, only 19% of the corpus words are unambiguous. The grammatical ambiguity rate reaches 5.6 on average for the vocalized words and 8.7 on average for the unvocalized ones. The ambiguity for the lexical forms with vowel marks is equal to 2.8 on average. This rate increases to 5.6 due to the absence of vowels. The absence of diacritics adds layers of confusion for novice readers and for automatic computation. For instance, the absence of diacritics becomes a serious obstacle to many applications including text to speech (TTS), intent detection, and automatic understanding in general. The process of adding vowels and other diacritic marks to Arabic text is called diacritization or vowelization. Therefore, automatic diacritization is an essential component for the automatic processing of a highly ambiguous Arabic text.

3. Related Works

The automatic analysis of texts written in Arabic faces the crucial problem of the lack of diacritics in those texts. This issue causes many cases of lexical ambiguity given the polysemous nature of un-vowelized words. The same is true for the syntactic analysis that takes the result of the lexical analysis (possibly the morphosyntactic labeling) as input and outputs a hierarchical structure of structural groups and functional relations uniting these groups. It should be noted that vowel and grammatical ambiguities concerning non-diacritization words create challenges for both this level and the semantic level where, in the absence of diacritization, a sentence can have several interpretations that are syntactically correct. Diacritics distinguish the grammatical category of the word, its function, the anaphoric relations, time indications, the aspect, mode, gender, and number.

The problem of diacritization has been the focus of several studies. However, it is still an open area for researchers to improve the accuracy and coverage of diacritization [13]. The techniques used in diacritization can be classified into three main categories: rule-based, statistical approaches, and hybrid approaches. The earliest approaches were rule-based ones, which often exploited human knowledge to solve the problem intelligently and heuristically. More recently, there have been several statistical, machine-learning approaches, such as the hidden Markov model (HMM), n-gram, statistical machine translation (SMT), and finite state transducers (FSTs). And the most current work in the area relies on hybrid approaches that combine rule-based and statistical modules [14]. Also, several systems[3] and tools have been developed for the resolution of the ambiguity for different levels of the analysis related to automatic diacritization for works such as [15-22]. Gal [23] used a HMM based on learning done on totally diacritized texts in his work, which achieved 85% good diacritizationwith some texts belonging to the training corpus. Nelken and Shieber [24] used a weighted finite state transducer

[3] A complete study of the different systems of automatic Arabic diacritization was done in [14].

based on training carried out on the LDC[4] database ATB3, which achieved an almost 90% correct vocalization rate. Shaalan [25] introduced a morphological and a syntactic analyzer for Arabic that was developed using a rule-based approach. The morphological analyzer utilizes the ATN to represent context-sensitive relation between stems and affixes. These two analyzers are the main components of diacritization systems, and thus could be used to develop a diacritics restoration system that follows a rule-based approach. Furthermore, the calculation of the maximum entropy has been the subject of a study by Zitouni and Sarikaya [26]. The authors presented an approach for restoring a comprehensive list of diacritics. They used a statistical model based on the framework of maximum entropy. Their model combines various sources of information, including lexical characteristics, based on segmentation. They used the morphological statistical analysis of Arabic to segment each Arabic word into a prefix, radical, and suffix. Each morpheme is called a segment and these characteristics are generated in an analytical model that allows maximum entropy. All of these features are then integrated into the framework of the maximum entropy to infer the complete diacritization of the sequence of input words. They have led to an error rate of 5.1% at the diacritical level as compared to 17.3% at the word level.

The Morphological Analysis and Disambiguation for Arabic (MADA) system [28] is specifically dedicated to the analysis and disambiguation of Arabic. It is based on three resources that must be downloaded and installed separately. The first resource is the Buckwalter Arabic morphological analyzer (BAMA) [29], the second is the SRILM[5] toolbox for its disambiguation utility [30]. MADA use this utility to construct n-gram[6] lexemes. Finally, MADA uses SVM tools to perform machine learning. MADA uses up to 19 orthogonal characteristics to select, for each word, a proper analysis from a list of potential analyses provided by the BAMA. The BAMA analysis most closely matching the collection of presaid and weighted features is elected. The 19 characteristics include 14 morphological characteristics that MADA predicted using 14 separate SVMs. Each analysis considered by MADA constitutes the diacritized form of the word, its lexeme, its morphological characteristics, and the input of an English glossary. Roth et al. [31] enriched this analysis with the use of lemmas.

Rashwan et al. [32] developed an Arab stochastic diacritizer based on a hybrid approach of factored and not factored textual features. They introduced the stochastic system with dual methods to automatically diacritize plain Arabic text. The first of these modules determines the most likely diacritics by choosing the sequence of the full form diacritizations of a word in Arabic. The choice is made with a maximum marginal probability using a search latticed A* and a multilevel diacritizer of Arabic for the estimation of n-gram probability with long horizon texts. When the full forms of the words are not in the vocabulary, the system uses the second module, which factorizes each word in Arabic in all of its possible morphological constituents, while also using the same techniques of the first module in order to obtain the most likely sequence of morphemes, and from there the most likely diacritization. Said and al. [33] developed a hybrid system that relies on automatic correction, morphological analysis, POS tagging, and out of vocabulary diacritization. This system is similar to Rashwan et al.'s system, but uses HMMs for morphological analyses disambiguation and resolving the syntactic ambiguity to restore the syntactic diacritic. It achieves the best results prior on LDC ATB3.

[4] LDC: Linguistic Data Consortium Arabic Treebank 3-v3.2 [27], http://catalog.ldc.upenn.edu/LDC99T42.
[5] SRILM toolkit: an extensible language modeling toolkit.
[6] According to the authors, their best results are the ones obtained with the lexemes form with trigram SLM (Statistical Language Model).

KAD [34] is statistical-based diacritizer that relies on quad-gram letters. The input to the system is a sequence of undiacritized letters. It begins by creating a database with the most frequently diacritized quad-gram patterns along with their probabilities. Then, the input letters are diacritized by retrieving from the database all of the quad-grams constituting these letters. For each letter, all the occurrences, which are at most four quad-grams, are considered and the probabilities of the sequences that have the same diacritic for that letter are totaled. The diacritic with the highest aggregated probability is chosen or that letter.

Hifny [35] used statistical n-gram language. The possible diacritized word sequences of an undiacritized input sentence are assigned probability scores using n-gram models. Then using a dynamic programming algorithm, the most likely sequence is found. Smoothing techniques were used to handle unseen n-grams in the training data. The accuracy of n-gram models depends on the order n. A larger order provides higher accuracy, as it incorporates longer linguistic dependencies.

We can see that the problem of partial presence or the absence of diacritical forms in a text poses more difficulties in the development of automatic morphological analyzers. Indeed, some machine translation systems, such as Systran and Google Translate, and even morphosyntactic analyzers, such as Arabic Buckwalter, Xerox, and the MADA analyzer, translate and analyze unvowelized texts of Arabic resources in that form. Therefore, if the input is vowelized or partially vowelized, these systems operate by eliminating all the diacritics, and they continue as if the treatment was not vowelized entrance with all the risks that this may create.

Notwithstanding the number of automated analyzers used to treat the Arabic language, and despite the number of works devoted to the study of the diacritization/vowelization of texts written in Arabic, this intractable problem remains an issue. We are critical of these approaches for using a large amount of annotated data (for a supervised labeling) or the operation of a lexicon that includes all possible labels for each word (for unsupervised labeling). We also disagree with the fact that these methods donot take into account the diacritics in their analyses, whereas the meaning and grammatical function of a word are strongly linked to good diacritisation of the latter.

Adapting existing disambiguation methods for Arabic texts is another crucial problem that must be overcome. Even though this task is far from resolved, many current works focus on this area and there are numerous options that have been explored. Given the importance of the role this problem plays vis-à-vis the results of the analysis of the text, the robustness of our proposed solution is directly related to its ability to handle vowelized, unvowelized, and partially vowelized texts. In the latter two cases, the return of a missing diacritical must follow the morphological analysis phase. Given the significant proportion of ambiguous words that have resulted from the developed tools and methods, we can conclude that, for Arabic, this phenomenon is very common and requires further study.

4. Proposed Disambiguation Approach Adapted to the Arabic Language

Our approach to disambiguation is based on the choice of the right diacritization of the word of a text since a word in Arabic accepts many. Although diacritization is equivalent to contextually choosing the proper diacritical of one word to determine the meaning and the function, the problem is twofold: (i) How to restore the diacritic potential of each word of a text analyzed morphologically, while several grammatical categories can be assigned to a word? (ii) How to choose from among a proposed set of

diacritical schemes, the right one for the same grammatical category attributed to this word?

As such, we adopted a two-step approach: a linguistic analysis followed by the application of a method for a multi-criteria decision (MCA). The approach was implemented and Fig. 1 shows the architecture of our Alkhalil+ analyzer.

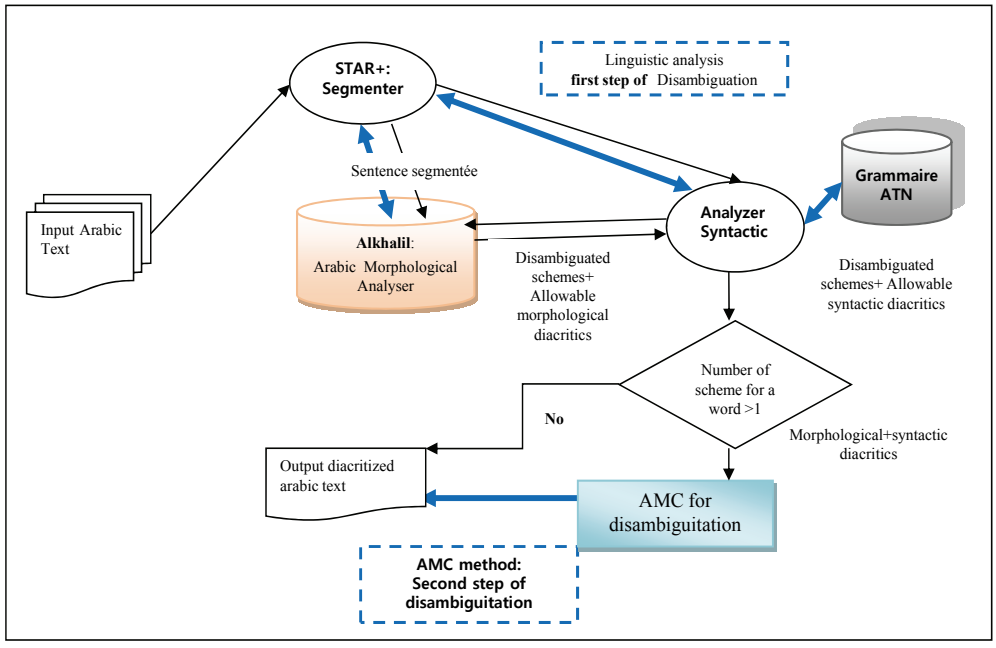


Fig. 1. The architecture of our analyzer Alkhalil+.

Linguistic Analysis: After being segmented into sentences, a morphosyntactic analysis was performed with the aim to associate each lexical unit with its grammatical category (noun, verb, adjective, etc.). The main interest of this labeling is that it allows to perform a first disambiguation processing of words. The tagger used can be associated with each lexical unit (especially for an ambiguous word) several labels (grammatical categories), and therefore, several schemes (several possibilities of diacritization). Next, a set of grammar rules implemented as a set of ATN was applied to the pre-labeled text. It must choose from among all the categories previously associated with a lexical unit a result that corresponds to a word in the considered sentence based on the corresponding ATN. Once the associated grammatical label is determined, a second disambiguation is made. The most obvious diacriticals of this word are found and all candidate schemes associated with the word are reduced.

MCA Application: In order to filter all successful applicant patterns and determine the final diacritics of this word (sense) an MCA must be applied. Our proposal is based on methods based on the decision theory. The interest to adopt a multi-criteria decision support approach to lift the morphological ambiguity in the ALP is twofold [36,37], as explained below.

1. To reduce the set of candidates labels (scenarios) MCA immediately reduces the number of correction labels by eliminating the dominant ones and generating the set of efficient labels;

2. In decreasing order, the effective labels are arranged as an overall score obtained after treatment. But the scenarios associated with a given word in isolation may not be disambiguated and the number of probable scenarios soft his word can be reduced regardless of the selected decision criteria.

To implement our approach, the choice of an automatic processing tool of the Arabic language is focused on the Alkhalil Morpho Sys [38], which is a morphological online analyzer for Standard Arabic text. Alkhalil can process texts that are not partially or totally diacritic. It is based on one part of the modeling of a large set of Arabic morphological rules and, for the other part, on the integration of linguistics resources[7] that are useful to other analyses. Despite the performance of this analyzer and of most analyzers of the Arabic language, the number of outputs assigned to each word of the text to be analyzed is considerable due to the absence of a disambiguation module.

We propose to extend and integrate in Alkhalil a morpho-lexical disambiguation module. We called this the analyzer and hence developed the Alkhalil+.

4.1 Linguistics Analysis: First Step of Disambiguation

In this section, we explain the different stages to follow linguistics analysis, which is considered to be the first step of disambiguation.

1. Alkhalil performs segmentation into words, not sentences. To make up for that, we integrated the STAR (Arabic Texts Slicer) tool [39] as a module for the segmentation of the text into sentences. The STAR tool is based on the contextual exploration method that takes into account the right and left context of each marker/trigger (punctuation, coordinating conjunctions, and some tools' words) liable to segment the sentence. The tool has 183 STAR proposals and phrases segmentation rules. For example, if we sometimes could not decide on the end the proposal for the meeting of a comma, we then called upon the Alkhalil morphological information on the right and/or the left context of comma was provided and stored for further processing. Even the parser can be applied in the segmentation of an ambiguous word. The evaluation of the STAr+[8] system (changes STAR) showed a return rate of 94.78% and an accuracy rate of 93.14%;
2. The Alkhalil Morpho Sys could be considered as the best Arabic morphological system. Actually, Alkhalil won the first position, among 13 Arabic morphological Alkhalil an analysis on the received data (morpho-lexical analysis, parsing, etc.). The morphosyntactic analysis is performed in five steps and the Alkhalil Morpho Sys allows for the identification of all possible solutions to a given word. The outputs of the analysis of Arabic words are presented in a comprehensive table showing possible roots, the corresponding schemes fully diacritic, grammatical categories, and its proclitics and enclitics.

[7] Alkhalil contains about 7,000 roots obtained from Sarf (Sarf 2007, an open source system http://sourceforge.net/projects/sarf/Arabic morphology) and NEMLAR corpus [40], NEMLAR: Euro-Mediterranean Network for Language Resources. This corpus was produced within the NEMLAR project and consists of 500,000 lexical units grouped into 13 different categories.

[8] We reprogrammed the analyzer "STAR" with Java developed in our laboratory, Miracl.

In Fig. 2 an example for the analysis of the ambiguous word "ذهب "(it can be the noun "gold" or the verb "to go"), Alkhalil attributed it 20 different schemes (patterns) with different diacritics.

Morphological analysis must recognize all bad segmentations and associates all of the lexical units that are from their various potential diacritics. The contribution of the morphological analysis diacritization process is that it eliminates some diacritics although the analysis of the clustered forms. In some cases, resolution can be achieved, but sometimes the proposed grammatical categories for a given word can be very numerous; hence, the need for the disambiguation phase exists.

3. In regards to the parser a disambiguation phase parsing is instrumental in automatic natural language processing. It continues to pose many problems in grammatical formalisms or in algorithmic complexity, especially in the managing of ambiguities [41-43]. Our analyzer is essentially based on the technology of ATN [44]. An ATN is usually represented by a graph and has the ability to make notes during the operation's notation stage and refers to these notes to make further decisions describing a state. It arches, allowing to move from one node to another while recording the states already borrowed from registers with flags. This is one of the highlights of an ATN in that it does not impose formal restrictions and has a practical and computational interest.

الخرج OUTPUT								الدخل INPUT
الكلمة المشكولة Voweled Word	السابق Prefix	الجذع Stem	نوع الكلمة Type	الوزن Pattern	الجذر Root	الحالة الإعرابية POS Tags	اللاحق Suffix	
ذَهَبُ	ذهب	#	اسم جامد	فَعَلُ	ذهب	مفرد مذكر مرفوع في حالة الاضافة	#	ذهب
ذَهَبٌ	ذهب	#	اسم جامد	فَعَلٌ	ذهب	مفرد مذكر مرفوع نكرة	#	
ذَهَبَ	ذهب	#	اسم جامد	فَعَلَ	ذهب	مفرد مذكر منصوب في حالة الاضافة	#	
ذَهَبِ	ذهب	#	اسم جامد	فَعَلِ	ذهب	مفرد مذكر مجرور في حالة الاضافة	#	
ذَهَبٍ	ذهب	#	اسم جامد	فَعَلٍ	ذهب	مفرد مذكر مجرور نكرة	#	
ذَهَبْ	ذهب	#	اسم علم	#	#	#	#	
ذَهَبُ	ذهب	#	مصدر أصلي	فَعَلُ	ذهب	مفرد مذكر مرفوع في حالة الاضافة	#	
ذَهَبُ	ذهب	#	مصدر أصلي	فَعَلُ	ذهب	مفرد مذكر مرفوع في حالة الاضافة	#	
ذَهَبٌ	ذهب	#	مصدر أصلي	فَعَلٌ	ذهب	مفرد مذكر مرفوع نكرة	#	
ذَهَبٌ	ذهب	#	مصدر أصلي	فَعَلٌ	ذهب	مفرد مذكر مرفوع نكرة	#	
ذَهَبَ	ذهب	#	مصدر أصلي	فَعَلَ	ذهب	مفرد مذكر منصوب في حالة الاضافة	#	
ذَهَبَ	ذهب	#	مصدر أصلي	فَعَلَ	ذهب	مفرد مذكر منصوب في حالة الاضافة	#	
ذَهَبِ	ذهب	#	مصدر أصلي	فَعَلِ	ذهب	مفرد مذكر مجرور في حالة الاضافة	#	
ذَهَبِ	ذهب	#	مصدر أصلي	فَعَلِ	ذهب	مفرد مذكر مجرور في حالة الاضافة	#	
ذَهَبِ	ذهب	#	مصدر أصلي	فَعَلِ	ذهب	مفرد مذكر مجرور نكرة	#	
ذَهَبٍ	ذهب	#	مصدر أصلي	فَعَلٍ	ذهب	مفرد مذكر مجرور نكرة	#	
ذَهِبَ	ذهب	#	فعل ماض مبني للمعلوم	فَعِلَ	ذهب	ثلاثي مجرد مسند إلى الغائب(هو) لازم	#	
ذَهَبَ	ذهب	#	فعل ماض مبني للمعلوم	فَعَّلَ	ذهب	ثلاثي مزيد مسند إلى الغائب(هو) متعد	#	
ذَهَبَ	ذهب	#	فعل ماض مبني للمعلوم	فَعَلَ	ذهب	ثلاثي مجرد مسند إلى الغائب(هو) متعد ولازم	#	
ذُهِبَ	ذهب	#	فعل ماض مبني للمجهول	فُعِّلَ	ذهب	ثلاثي مزيد مسند إلى الغائب(هو) متعد	#	
ذُهِبَ	ذهب	#	فعل ماض مبني للمجهول	فُعِلَ	ذهب	ثلاثي مجرد مسند إلى الغائب(هو) متعد ولازم	#	
ذَهَّبْ	ذهب	#	فعل أمر	فَعِّلْ	ذهب	ثلاثي مزيد مسند إلى المخاطب أنت متعد	#	

Fig. 2. Analysis results of word "ذهب " by Alkhalil.

The formality of an ATN can be used to describe complex and deep syntactic dependencies, especially for are cursive system, in a manner that is relatively intuitive and easy to implement. The ATNs are readable, understandable, fast, efficient, and modular. The use of technology's finite state has been studied in several research projects such the Nooj platform (http://www.nooj4nlp.net) and was favored by different research teams, such as Xerox [45] and Bell Labs research center (http://www.bell-labs.com).

The ATNs implemented in our analyzer used a rule base that included two classes, which are as described below.

a) The first class of rules contained 30 ATNs that intervene directly on the schemas associated with the potential grammatical category of a word (specialized class: fine label[9]). It is based on contextual rules and heuristics. By surface parsing the sentence, the corresponding ATN decides the outset of the most obvious diacritic potential patterns of the word. Fig. 3 shows an implemented contextual rule.

Examples of contextual rules:

Rule 1: After a subordination particle (or genitive) "حرف جر" always follows a genitive name case "اسم مجرور".

Rule 2: After a particle subjunctive (or accusative) "حرف نصب" for a noun always follows subjunctive noun "اسم منصوب".

Rule 3: After a particle subjunctive (or accusative) "حرف نصب" for a verb unaccomplished always follows a verb in the unfinished "فعل مضارع".

```
Input: a particle followed by a verb
Output: success or failure of parsing. A feedback
message in case of failure
Grammar checking of a particle followed by a verb
Rule:
feedback_message=''
if  verb.tense = past
then append feedback_message with ' لا يأتي الفعل
الماضي بعد أداة النصب أو الجزم',
if  verb.end_case = nominative
then append feedback_message with ' لا يكون الفعل
مرفوع بعد أداة النصب أو الجزم',
if particle.category = preposition
then append feedback_message with ' لا يسبق الفعل حرف
جر',
if feedback_message=''
then accept particle followed by a verb
else issue feedback_message and fail
End Rule
```

Fig. 3. Example of a contextual rule [42].

This first step allowed us to effectively reduce amount of unambiguous patterns associated with different grammatical categories assigned to the ambiguous word.

b) The second class of rules is connected to the scheduling of words in the sentence and it assigned the category that each word of a text belonged to in the context in which the word appeared. It is associated with the grammatical category of the word (the generic class, which is known as the macro-category). If the sentence was not been recognized by any of the 57 ATNs of the base, then it was rejected (Note: All the details of this implementation will be the subject of another article).

In earlier stages, we found that in over 72% of the cases, the grammatical category was found. This

[9] We include the tags of Alkhalil and the tags of the LDC.

means that grammatical labeling leads to improved out comes in regards to diacritization. The number of potential schemes was reduced, the resolution was obtained and, if none existed,only one scheme inthe grammatical category was found or another phase of disambiguation was required. This step is required (in most cases) for the diacritisation word.

4.2 The Purpose of a Multi-Criteria Decision (MCA) for Disambiguation: Another Step of Disambiguation

Morphosyntactic analysis raises the problem of the multitude of scenarios assigned to a word. It generates at least two competing solutions. The major problem of morphosyntactic analysis lies in choosing a good and proper grammatical label scheme.

The principle of our proposed disambiguation method is to reduce the outset, the number of scenarios dominated by separating the scenarios (i.e., scenario shaving no better evaluation according to all criteria) and classifying effective scenarios (i.e., those that are not dominated) to bring about the best scenario, which is the one that generally has the most efficient scores according to different criteria. So, we reduce the number of interpretations from the first analyzes through a MCA, we take the Help method to the multi-criteria decision TOPSIS (Technique for Order by Similarity to Ideal Solution) [47], which is known for its robustness and mathematical foundation.

In this section, we explain the different stages to follow the MCA analysis, which is to be considered at the formal robustness step of disambiguation.

4.2.1 Problem formalization of MCA

We defined X = {x_1, x_2, x_3, ...,x_n} as a set of scenario candidates for disambiguation. These scenarios, which are an infinite number and different one from another, constitute all the possible solutions. To choose the best scenario "X" we used the set F = {f_1, f_2, f_3, ...,f_n}, which is a coherent family of criteria.

A series of steps was performed starting with a list of potential actions. Then the criteria were listed, the evaluation function (performance based) defined, a table of performance was established, the performances were weighed and aggregated, and finally, classified in the scenarios.

In order to evaluate each scenario according to each criterion, we defined the evaluation function as follows:

$$f_j: \mathrm{X} \rightarrow \mathrm{R}$$

$$x \rightarrow f_j(x),$$

where, $f_j(x)$ represents the evaluation of scenario x according to criterion f_j. Each of these functions should be maximized (or minimized) according to the type of criteria that was used.

(a) Ideal Scenario: A scenario is said to be ideal if it corresponds to the best solution for all criteria. It is represented by a point in $\mathbf{R}^q$ coordinates, which are: (y^1+ ... + y^q) where, y+ = Max ($f_j(x)$); j = 1, .., q with $x \in$ X are.The ideal scenario in this case is to find a single scheme for a given word (i.e., a good grammatical label and correct diacritical); in other words, a perfect morpho-lexical disambiguation (100%).

(b) Dominance relation between scenarios: It can only be said that scenario x_1 dominates scenario x_2, if and only if $f_j(x_1) \geq f_j(x_2)$; $j = 1, ..., q$ wherein at least one of the inequalities is strict ($f(x_1) \neq f(x_2)$). In other words, x_2 has no evaluation (according to all used criteria) that is better than that of x_1.

Note, that for this dominance relation "≥" means "better than". So if f_j is a criterion to be minimized so that $f_j(x_1)$ is "better than" $f_j(x_2)$, it must verify: $f_j(x_1) \leq f_j(x_2)$.

(c) Effective scenario: Scenario x is said to be efficient, if and only if it is not dominated by any scenario. The set of all effective scenarios is considered to be a set of more interesting solutions.

(d) Ranking of scenarios: In order to determine the best scenario (the best scheme), we proceeded to classifying all of the effective scenarios. The goal was to determine the scenario that satisfied the overall best ratings. Thus, we calculated for each scenario x_i, an overall evaluation score $S(x_i)$, which is the weighted sum of the different evaluations of x_i, according to all criteria:

$$\alpha_j > 0, \sum_{j=1}^{q} \alpha_j = 1 \; with \; S(x_i) = \sum_{j=1}^{q} \alpha_j f_j(x_i) \tag{1}$$

where, α_j is the weight associated with the criteria f_j. The problem of determining the best scenario, denoted $P(x)$, is defined by:

$$P(x) = \max_{x \in X} \sum_{j} \alpha_j f_j \; with \; j = 1, ..., q \tag{2}$$

4.2.2 Basic criteria for the evaluation of scenarios

The problem with determining coherent criteria that are comprehensive, exhaustive, non-redundant, and forming a cohesion between themselves is a complex task. The criteria used to discriminate between the scenarios can change from one language to another. Note that some criteria used for French, do not apply to Arabic, mainly when it comes to diacritizations. However, the overlapping between the criteria is quite possible. This is the case of the general criteria that does not take into account the specific characteristics of languages, such as the frequency of occurrence of a word. Thus, we proposed two basic criteria for discriminating between scenarios (schemes).

1) The juxtaposition of diacritics: This criterion uses the position of diacritics to disambiguate. To match each between the input diacritic word and diacritic candidate scheme, it is assigned 1.
2) Computing the frequency of occurrence of each scenario candidate in the text.

4.2.3 Weighting of criteria

It is necessary, however, to weight the criteria. There are local and global weightings, such as Normal, Gfldf, Idf, and Entropy. Our choice was focused on an overall weighting (entropy or the average uncertainty). On one hand, it is a method that takes into account the distribution of lexical units in the text and, on the other, we recognized the need to locally weight the importance of the scheme in its grammatical category and measure the overall representativeness of the lexical unit in the text. This method is an objective technique for weighting the criteria. This principle is that a criterion "*j*" is more important than the dispersion of the valuation of actions. Thus, the most important criteria are those that discriminate the most between actions (x_i).

The entropy of a criterion *j* is calculated using the following formula:

$$E_j = -K \sum x_{ij} \log(x_{ij}) \tag{3}$$

where, K is a constant chosen such that for any *j*, we have0 ≤ E_j ≤ 1, for example, K = 1/log(*n*) (*n*being the number of patterns candidates for disambiguation). The entropy E_j gets larger as the values x_j are getting. Therefore, the weights are calculated according to the dispersion measurement (as opposed to the entropy):

$$D_j = 1 - E_j \tag{4}$$

The weight is normalized by:

$$W_j = D_j / \sum D_j \quad (j = 1, \ldots, n) \tag{5}$$

4.2.4 Aggregation of criteria

In order to aggregate the different assessments of a scenario calculated according to the two criteria used, we propose the use of the TOPSIS method. Hwang and Yoon [48] developed the TOPSIS method. It chooses a solution that is closest to the ideal solution, and is based on the dominance relation, which results from the distance to the solution (the best of all criteria) and as far away as possible from the worst solution (which degrades all criteria). Theaim is to reduce the number of disambiguation scenarios (the scenarios are the schemes assigned at a word) and classify effective ones according to their calculated total scores. In case of similar scenarios, we calculated their measures of separation in accordance to the ideal scenario (step inspired by the TOPSIS method). The best scenario is the closest to the ideal scenario.

The main steps of the proposed method for ranking scenarios for the morpho-lexical disambiguation are as listed below.

Step 0. Construct the evaluation matrix/decision.

$$E = (e_{ij})\ i = 1, \ldots, m;\ j = 1, \ldots, n$$

This matrix represents the respective scores of the different scenarios according to all of the selection criteria, where, e_{ij} is the scenario x_i score according to criterion f_j.

Step 1. Standardize the decision matrix.

The resulting decision matrix is normalized to allow a homogeneous comparison through the various criteria that have different units of measure. The elements of the decision matrix are normalized by:

$$e'_{ij} = \frac{f_j(a_i)}{\sqrt{\sum_{i=1}^{m} \left[f_j(a_i) \right]^2}};\ i = 1, \ldots, m;\ j = 1, \ldots, n \tag{6}$$

$f_j(a_i)$ corresponds to the deterministic values of "I" actions for the "j" criterion.

Step 2. Weight the normalized decision matrix.

$e''_{ij} = \pi_j \, . \, e'_{ij}, i = 1, \ldots, m$; $j = 1, \ldots, n$; π_j is the weight of the j^{th} criterion

$$\sum_{j=1}^{n} \pi_j = 1 \tag{7}$$

This is obtained by multiplying each column of the normalized matrix by the relative weight of the criteria for that column.

Step 3. Determine the ideal scenario (a^+) and the worst-case scenario (a_-)

$$a^+ = \left\{ \max e''_{ij}, i = 1, \ldots, m \; and \; j = 1, \ldots, n \right\}; \; e_j^+ = \max_i \left\{ e''_{ij} \right\} \tag{8}$$

$$a_- = \left\{ \min e''_{ij}, i = 1, \ldots, m \; and \; j = 1, \ldots, n \right\}; \; e_{j-} = \max_i \left\{ e''_{ij} \right\} \tag{9}$$

The ranking of the scenarios is performed in descending order of their assessments.

Step 4. Calculate the remoteness measures: Calculate the Euclidean distance between each candidate scenario and the ideal scenario (compared with a^+ and a_-):

$$D_i^+ = \sqrt{\sum_{j=1}^{n} \left(e''_{ij} - e_j^+ \right)^2} \; with \, i = 1, \ldots, m \tag{10}$$

$$D_{i-} = \sqrt{\sum_{j=1}^{n} \left(e''_{ij} - e_{j-} \right)^2} \; with \, i = 1, \ldots, m \tag{11}$$

The best scenario is the lowest measure of separation (the same for the worst solution).

Step 5. Calculate the approximation coefficients with respect to the ideal.

$$C_i^+ = \frac{D_{i-}}{D_i^+ + D_{i-}} \; with \; i = 1, \ldots, m; \; 0 \le C_i^+ \le 1 \tag{12}$$

Step 6. Classify the actions according to their preference orders.

("i" is better than "j" if $C^*i > C^*i$).

The following example illustrates the principle of themethod.

4.3 An Example

We used the sentence Ph of a text Tgivenas the input for the Alkhalil+ tool. After the firstlanguage-processing step already explained, the sentences go through a disambiguation phase according to the CMA steps: Ph = "ذهب الطفل إلى البستان" / the child went to the garden /

Analyze the ambiguous word ""ذهب" / verb: to go / or / Name: gold /

Determination of scenarios

After conducting linguistic analysis, we obtained a reduced set of patterns for that word (the likely scenario "name" as a grammatical category was eliminated by the previous steps mentioned above). In this case, the set E considered as the likely scenarios for that word will be:

E = {Verbs: فَـعَـلَ,فَـعِـلَ, فَـعُـلَ, فَـعِـلٌ, فُـعِـلَ }

Application of the criteria on the scenarios

Criterion 1. Juxtaposition diacritical: Each diacritic was assigned the value 0 or 1 depending on whether its position was good in the scheme.

Scenarios: { فَـعَـلَ 1 1 1, فَـعِـلَ 1 0 1, فَـعُـلَ 1 0 1, فَـعِـلٌ 1 0 0, فُـعِـلَ 0 0 1, فَـعَـلْ 1 1 0}

Criterion 2. Frequency of occurrence (PA, PB, ..., Pn; Paragraphs of text): For each likely scenario, we recorded the frequency of occurrence in each paragraph of the text (without diacritical, the last letterbecause it is a casual mark that depends on the grammatical terms [position] of the word).

For example, for scenario فَـعَـلَ, we have 2 times in PA, PB 1 time in ... and 3 times in Pj.

Step 0: Application of the evaluation function and generation of the evaluation matrix (Table 3)

Table 3. Evaluation matrix from text

Criterion	Scenarios					
	فَـعَـلَ	فَـعِـلَ	فَـعُـلَ	فَـعِـلٌ	فُـعِـلَ	فَـعَـلْ
Diacritical juxtaposition	1+1+1=3	2	2	1	1	2
Frequency of occurrence	16	5	3	5	5	16

For example, for the scenario 1 **فَـعَـلَ**, we allocated the number 3, which is : 1 + 1 + 1 (see step 2, criterion 1), to it in the assessment matrix.

Steps 1 and 2: Aggregation of the performance and the weighting of criteria

(a) Normalization of the evaluation table: Table 4 shows the normalization of the evaluation matrix obtained ($e'_{i,j}$).

Table 4. Normalization of the evaluation matrix

Criterion	Scenarios					
	فَـعَـلَ	فَـعِـلَ	فَـعُـلَ	فَـعِـلٌ	فُـعِـلَ	فَـعَـلْ
Diacritical juxtaposition	0.13	0.09	0.09	0.04	0.04	0.09
Frequency of occurrence	0.03	0.008	0.008	0.005	0.008	0.03

(b) Weighting of the evaluation matrix (normalized) (Table 5).

Table 5. The weighting of the normalized evaluation matrix obtained ($e''_{i,j}$)

Criterion	Scenarios					
	فَـعَـلَ	فَـعِـلَ	فَـعُـلَ	فَـعِـلٌ	فُـعِـلَ	فَـعَـلْ
Diacritical juxtaposition	0.13	0.09	0.09	0.04	0.04	0.09
Frequency of occurrence	0.02	0.08	0.08	0.05	0.08	0.02

Step 3: Determination of the ideal e^+ and the worst e^- solutions

The ideal and non-ideal solutions obtained following each criterion are:

- Criterion 1: "diacritical juxtaposition"
  Ideal " e_1^+" = 0.33, worst solution "e_1^-" =0.33
- Criterion 2: "frequency of occurrence"
  Ideal " e_2^+ "= 0.50, worst solution "e_2^-" = 0.03

Step 4: Calculation of remoteness for each candidate scenario (Table 6)

Table 6. The separation distances for each candidate scenario

	Scenarios					
	فَـعَـلَ	فَـعِـلَ	فَـعُـلَ	فَـعِـلٌ	فُـعِـلَ	فَـعَـلْ
D^+	0.00	0.34	0.40	0.46	0.46	0.46
D^-	0.50	0.15	0.12	0.22	0.22	0.45

Step 5: Computation of the coefficient of the measurement of remoteness (CR) with the ideal profile (Table 7)

Table 7. The coefficients obtained for remoteness with ideal profile

	Scenarios					
	فَـعَـلَ	فَـعِـلَ	فَـعُـلَ	فَـعِـلٌ	فُـعِـلَ	فَـعَـلْ
CR	1	0.31	0.22	0.28	0.28	0.50

Step 6: Selection of scenario (i.e., the appropriate schema and final diacritical)

As seen in the table above, we established a ranking of these coefficients in descending order ("*i*" is better than "*j*" if $C^*_i > C^*_i$). The scenario that obtained the highest score (the highest coefficient) was chosen. In our case, this was scenario1 " فَـعَـلَ ", generating the following information (Fig. 4); (remember 20 schemes for this word before disambiguation in Section 4.1, Fig. 2).

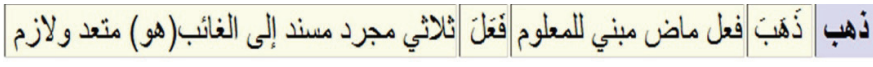


Fig. 4. Result after disambiguation analysis Alkhalil+ for the word "ذهب".

If the remoteness ratio (CR) is equal to 1, then one scenario is possible and the ambiguity is resolved. This is the case in this particular example, where the rate of disambiguation reached 100% because the verbs were better disambiguated by their arguments (importance of local information), while a larger

context was necessary for names, as Alkhalil+ only displays scenarios including CR>= 0.7. We chose a threshold of 70%, because after several tests, we found that below this value, the correct scenarios could not be displayed. In the case where the threshold was not reached, we posted both scenarios that had the maximum values out of all of the proposed scenarios for the same word.

5. Results and Discussions

For a sentence containing ambiguous words, we tested it with both analyzers Alkhalil and Alkhalil+: كتب كثيرة في المكتبة (many books in the library) (Table 8).

Table 8. Disambiguation rate for this sentence by Alkhalil+

Words of the sentence to be analyzed	Number of schemes assigned to this word by Alkhalil	Number of schemes selected to this word by Alkhalil+	Disambiguation percentage (%) Alkhalil+ for each word in the sentence
كتب	17	1	100%
كثيرة	8	1	100%
المكتبة	**21**	**2**	**90.47%**

Disambiguation rate for this sentence: 96.823%.

We can see from this example that the number of schemes assigned by Alkhalil was 21 for the undiacritized word "المكتبة". When using the Alkhalil+ analyzer, the first disambiguation (linguistic analysis) only leaves the genitive for it, that is to say, 7 elected scenarios (schemes) for 21 probable, the second disambiguation regroups the identical schemes and apply the steps of MCA then ending with two elected scenarios: المَكْتَبَة (el maktabati / the library) and المُكَتِّبَة (el moukattibati/librarian) where a semantic disambiguation is required (We could use Arabic WordNet ontology with only two synsets in this case).

In order to evaluate Alkhalil+, experiments were performed on a corpus consisting of a collection of texts in Arabic by maintaining existing potential diacritical. The corpus contains about 51,404 words including 81 Arabic texts (8th year school book of basic education 899 paragraphs, 3,871 sentences and 29,188 words) [49] (Table 9).

Table 9. The number of words of the corpus and the number of words that were not recognized by our system Alkhalil+ (the rate of analyzed words went from 93% in Alkhalil to 96,0003%[10] in Alkhalil+)

Number of words of the corpus	51,404
Number of words that were not recognized by Alkhalil+	2,056

As shown in Table 10, we obtained a disambiguation rate of 96.555% for the grammatical labeling of words recognized by Alkhalil+. This was due to the different disambiguation steps implemented in this work.

Table 10. An evaluation of the labelling Alkhalil+ (after grammatical labeling)

	Grammatical labeling
% of words correctly diacritized taking into account non recognised words	91.733
% of properly diacritized words without considering the unrecognized words	96.555

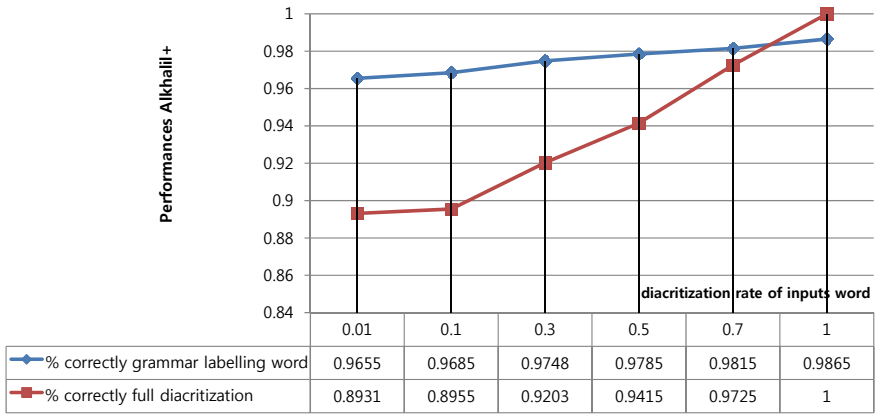
[10] We improved the lexical base of Alkhalil (i.e., Alkhalil did not make the processing of the numbers and letter Latin).

This precision is slightly greater that those obtained by other systems, such as MADA, which reported 96.09% as grammatical labeling rate. Table 11 shows a comparison of performances by some well-known Arabic language processing systems.

Table 11. The rate of grammatical labeling of some known analyzers Arabic language processing

Analyzers	Grammatical labeling
Stanford POS tagger [50]	96.42% on Arabic (without diacritics)
Khoja APT [51]	Around 90% (without diacritics)
ASVMTools [52]	95.49% (without diacritics)
AMIRA [53]	Over 96% (without diacritics)
MADA+TOKAN [54]	96.09%

MADA+TOKAN takes the word without diacritics when Alkhalil+ takes into consideration the diacritics that are present in the input corpus. Alkhalil+'s performance improves when the input word contains diacriticals[11], as they are reflected in the different treatments of the analysis. Fig. 5 shows that theinput diacritics rate greatly influenced the morpho-lexical disambiguation of the word on the grammatical labeling and on diacritization.



	0.01	0.1	0.3	0.5	0.7	1
% correctly grammar labelling word	0.9655	0.9685	0.9748	0.9785	0.9815	0.9865
% correctly full diacritization	0.8931	0.8955	0.9203	0.9415	0.9725	1

Fig. 5. The impact of input diacritics on the performance of Alkhalil+.

Our hybrid approach showed significant performance compared to other known existing systems of automatic diacritization. We evaluated our system's results compared to others using the same evaluation metrics[12]. Two main measures were used to evaluate system's performance, in regards to the error rates: word error rate (WER) and diacritization error rate (DER). WER is the percentage of incorrectly diacritized words in which at least one letter has a wrong diacritic. The percentage of incorrectly diacritized letters is denoted as DER. Our system shows a by WER=10, 69, and DER=3, 80. While ignoring the last diactrics letter, we obtained a WER=3, 42 and a DER=1, 15. This means an absolute reduction on the DER over the best evaluated system.

[11] We suppose that diacritics were correct in input.

[12] We calculated these two metrics as is done in [26] where: (a) all words were counted including numbers and punctuators, (b) each letter or digit in a wordwas a potential host for a set of diacritics, and (c) all diacritics in a single letter were counted as a single binary (True or False) choice.

Table 12. Acomparison between some systems surveyed (the better system is the one whose WER / DER is the smaller)

System	All diacritics		Ignore last	
	DER	WER	DER	WER
Zitouni and Sarikaya [26], 2009	5.5	18.0	2.5	7.9
Habash and Rambow [28], 2007	4.8	14.9	2.2	5.5
Rashwan et al. [32], 2001	3.8	12.5	1.2	3.1
Hifny [35], 2012	-	8.90	-	3.40
Alghamdi et al [34], 2012	5.5	-	-	-
Said et al [33], 2013	3.6	11.4	1.6	4.4
Our system Alkhalil+	**3.8**	**10.69**	**1.15**	**3.42**

As shown in Table 12, our results were satisfactory and are as good as previous works and because we used following steps 1) the treatment by sentence and not by word as was done in the Alkhalil analyzer, 2) the parser that we used reduced the rate of the WER, and 3) the MCA that we adopted reduced the rate of the DER and the WER because it regrouped the scheme was correctly diacritized it regrouped by our morphological analyzer, which is what makes this work unique. Nevertheless, the comparison would have been better if these systems had been tested on the same datasets[13] as ours. Although we compared the systems in terms of WER and DER, we have to be careful when interpreting the results.

However, the results show that the limits of the Alkhalil+ system are mainly caused by many issues such as:

1. The data that was not listed in the database of Alkhalil: so we started by extending the lexical database for unrecognized words to learn foreign words.
2. The morphological analyzer Alkhalil does not analyze words that do not belong to the lexical database of the most common words resulting in a morphological error of 11.5%
3. For 22.5% of the analyzed sentences, their failure in analysis is mainly due to the fact that their structure is not covered by our grammar (this was the case for long phrases or anaphoric sentences and/or elliptical unrecognized) or to a failure in the segmentation into sentences (a failure in recognizing the morphosyntactic characteristics of certain words, etc.) and this had a negative impact on the allocation of the POS taggingof the syntactical errors, which was around 10% error rate .
4. The CMA does not identify the right solution if there are errors are in the previous steps.

6. Conclusion

The automatic analysis of Arabic encounters the crucial problem of the lack of diacritics in the texts. Many systems and tools are developed for the automatic processing of Arabic. However, despite the performances of analyzers, the number of outputs assigned to each word of the text to be analyzed is considerable due to the absence of a disambiguation module.

We have presented in this paper, a lexical morphological disambiguation approach that is based on a multi-criteria decision method. For example, we presented the TOPSIS method, which is considered to be a formal method of disambiguation. The experiments carried out by our approach showed a

[13] Different datasets were utilized: ATB3, Tashkila, Coran, Sakhr, etc.

disambiguation rate that was above 85%. This was due to the use of correct diacritic patterns (schemes) by the Alkhalil morphological analyzer combined with the use of a MCA algorithm to formally choose good grammatical labeling and, therefore, the appropriate diacritic. The results presented in this paper open up many avenues for future research that focuses on the use of Alkhalil+. Azmi and Almajed [14] felt that hybrid-based schemes are a better choice, as they combine human knowledge with some intelligent techniques.

Alkhalil+ has made a significant contribution to MSA (Modern Standard Arabic) morphological analysis, disambiguation, POS tagging, tokenization, lemmatization, and diacritization. In order to improve the system's performance in terms of the quality of solutions and execution time, we took into account the problem of ambiguity in the analysis of language. Some potential points to consider have to be taken into account in regards to extending the basic rules of grammar for the syntactic analyzer and implementing a semantic disambiguation module using ontology that is fairly accurate in covering the meaning of words.

We completed our study by joining the opinions of Nelkel and Shieber [24] and Azmi and Almajed [14]: "the objective is to use enough of the optimal number of diacritical marks to avoid ambiguity in the written text."

References

[1] A. Tchechmedjiev, "État de l'art: mesures de similarité sémantique locales et algorithmes globaux pour la désambiguïsation lexicale à base de connaissances," in *Proceedings of Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 3: RECITAL*, Grenoble, France, 2012, pp. 295-308.

[2] L. Audibert, "Désambiguïsation lexicale automatique: sélection automatique d'indices," in *Proceedings of Traitement Automatique des Langues Naturelles (TALN-2007)*, Toulouse, France, 2007, pp. 13-22.

[3] M. Rakho, G. Pitel, and C. Mouton, "Désambiguïsation automatique à partir d'espaces vectoriels multiples clutérisés," Université Paris 7 – Diderot, Rapport Intermédiaire, 2008.

[4] R. Navigli, "Word sense disambiguation: a survey," *ACM Computing Surveys*, vol. 41, no. 2, article no. 10, 2009.

[5] A. Alsaad and M. Abbod, "Arabic text root extraction via morphological analysis and linguistic constraints," in *Proceedings of 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation (UKSim)*, Cambridge, UK, 2014, pp. 125-130.

[6] A. Al-Arfaj and A. Al-Salman, "Arabic NLP tools for ontology construction from Arabic text: an overview," in *Proceedings of 2015 International Conference on Electrical and Information Technologies (ICEIT),* Marrakech, Moroco, 2015, pp. 246-251.

[7] L. Belguith and A. Ben Hamadou, "Traitement des erreurs d'accord: Une analyse syntagmatique pour la détection et une analyse multicritère pour la correction," *Revue d'intelligence artificielle*, vol. 18, no. 5-6, pp. 679-707, 2004.

[8] M. Sawalha and E. Atwell, "Adapting language grammar rules for building morphological analyzer for Arabic language," in *Proceedings of the Workshop of Morphological Analyzer Experts for Arabic Language*, Damascus, Syria, 2009.

[9] R. Ouersighni, "La conception et la réalisation d'un système d'analyse morpho-syntaxique robuste pour l'arabe: utilisation pour la détection et le diagnostic des fautes d'accord," Ph.D. dissertation, Université Lumière Lyon 2, 2002.

[10] A. Farghaly and K. Shaalan, "Arabic natural language processing: Challenges and solutions," *ACM Transactions on Asian Language Information Processing*, vol. 8, no. 4, article no. 14, 2009.

[11] E. Souissi, "Etiquetage grammatical de l'arabe voyellé ou non," Ph.D. dissertation, Université de Paris VII, 1997.

[12] F. Debili, H. Achour, and E. Souissi, "La langue arabe et l'ordinateur de l'étiquetage gramatical à la voyellation automatique," *Correspondances: bulletin de l'IRMC*, vol. 2002, no. 71, pp. 10-26, 2002.

[13] R. Shah, P. S. Dhillon, M. Liberman, D. Foster, M. Maamouri, and L. Ungar, "A new approach to lexical disambiguation of Arabic text," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP'10)*, MIT Stata Center, MA, 2010, pp. 725-735.

[14] A. M. Azmi and R. S. Almajed, "A survey of automatic Arabic diacritization techniques," *Natural Language Engineering*, vol. 21, no. 3, pp. 477-495, 2015.

[15] A. A. Alzand and I. Rosziati, "Diacritics of Arabic natural language processing (ANLP) and its quality assessment," in *Proceedings of the 2015 International Conference on Industrial Engineering and Operations Management (IEOM2015)*, Dubai, United Arab Emirates (UAE), 2015.

[16] R. A. Haertel, P. McClanahan, and E. K. Ringger, "Automatic diacritization for low-resource languages using a hybrid word and consonant CMM," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Stroudsburg, PA: Association for Computational Linguistics, 2010, pp. 519-527.

[17] G. A. Abandah, A. Graves, B. Al-Shagoor, A. Arabiyat, F. Jamour, and M. Al-Taee, "Automatic diacritization of Arabic text using recurrent neural networks," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 18, no. 2, pp. 183-197, 2015.

[18] M. Diab, K. Hacioglu, and D. Jurafsky, "Automatic tagging of Arabic text: from raw text to base phrase chunks," in *Proceedings of HLT-NAACL 2004: Short Papers*, Boston, MA, 2004, pp. 149-152.

[19] M. Diab, M. Ghoneim, and N. Habash, "Arabic diacritization in the context of statistical machine translation," in *Proceedings of Machine Translation Summit XI (MT-Summit)*, Copenhagen, Denmark, 2007.

[20] M. El-Bèze, B. Mérialdo, B. Rozeron, and A. M. Derouault, "Accentuation automatique de textes par des méthodes probabilistes," *Technique et Science Informatiques*, vol. 13, no. 6, pp. 797-815, 1994.

[21] M. Maamouri, A. Bies, and S. Kulick, "Diacritization: a challenge to Arabic treebank annotation and parsing," in *Proceedings of the British Computer Society Arabic NLP/MT Conference*, London, 2006.

[22] A. O. Bahanshal and H. S. Al-Khalifa, "A first approach to the evaluation of Arabic diacritization systems," in *Proceedings of 2012 Seventh International Conference on Digital Information Management (ICDIM),* Macau, 2012, pp. 155-158.

[23] Y. A. Gal, "An HMM approach to vowel restoration in Arabic and Hebrew," in *Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages*, Philadelphia, PA, 2002, pp. 1-7.

[24] R. Nelken and S. M. Shieber, "Arabic diacritization using weighted finite-state transducers," in *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Ann Arbor, MI, 2005, pp. 79-86.

[25] K. Shaalan, "Rule-based approach in Arabic natural language processing," *International Journal on Information and Communication Technologies (IJICT)*, vol. 3, no. 3, pp. 11-19, 2010.

[26] I. Zitouni and R. Sarikaya, "Arabic diacritic restoration approach based on maximum entropy models," *Computer Speech & Language*, vol. 23, no. 3, pp. 257-276, 2009.

[27] M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki, "The Penn Arabic treebank: building a large-scale annotated Arabic corpus," in *Proceedings of NEMLAR Conference on Arabic Language Resources and Tools*, Cairo, Egypt, 2004, pp. 102-109.

[28] N. Habash and O. Rambow, "Arabic diacritization through full morphological tagging," in *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, Rochester, NY, pp. 53-56.

[29] T. Buckwalter, *Buckwalter Arabic Morphological Analyzer Version 2.0.* Philadelphia, PA: Linguistic Data Consortium, 2004.

[30] A. Stolcke, "SRILM: an extensible language modeling toolkit," in *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP)*, Denver, CO, 2002, pp. 1-4.

[31] R. Roth, O. Rambow, N. Habash, M. Diab, and C. Rudin, "Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, Columbus, OH, 2008, pp. 117-120.

[32] M. A. Rashwan, M. A. Al-Badrashiny, M. Attia, S. M. Abdou, and A. Rafea, "A stochastic Arabic diacritizer based on a hybrid of factorized and unfactorized textual features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 166-175, 2001.

[33] A. Said, M. El-Sharqwi, A. Chalabi, and E. Kamal, "A hybrid approach for Arabic diacritization," in *Natural Language Processing and Information Systems*. Heidelberg: Springer, 2013, pp. 53-64.

[34] M. Alghamdi, Z. Muzaffar, and H. Alhakami, "Automatic restoration of Arabic diacritics: a simple, purely statistical approach," *Arabian Journal for Science and Engineering*, vol. 35, no. 2, pp. 125-135, 2010.

[35] Y. Hifny, "Smoothing techniques for Arabic diacritics restoration," in *Proceedings of 12th Conference on Language Engineering (ESOLEC'12)*, Cairo, Egypt, 2012, pp. 6-12.

[36] A. Scharlig, *Décider sur plusieurs critères: panorama de l'aide à la décision multicritère*. Lausanne: Presses polytechniques et universitaires romandes, 1985.

[37] B. Roy and D. Bouyssou, *Aide multicritère à la décision: méthodes et cas*. Paris: Economica, 1993.

[38] Alkhalil Morpho Sys version 1.3, 2011; http://sourceforge.net/projects/alkhalil/.

[39] L. Belguith, L. Baccour, and G. Mourad, "Segmentation de textes arabes basée sur l'analyse contextuelle des signes de ponctuations et de certaines particules," in *Actes de la 12éme Conférence annuelle sur le Traitement Automatique des Langues Naturelles*, Dourdan, France, 2005, pp. 451-456.

[40] M. Yassen, K..Choukri, N. Paulsson., S. Haamid. and all "Building Annotated Written and Spoken Arabic LRs in NEMLAR Project," in *Proceedings of International Conference on Language Resources and Evaluation (LREC), 2*006.

[41] A. Haddad, H. B. Ghezala, and M. Ghnima, "Conception d'un catégoriseur morphologique fondé sur le principe d'Eric Brill dans un contexte multi-agents," in *Proceedings of 26th Conference on Lexis and Grammar*, Bonifacio, France, 2007, pp. 1-8.

[42] K. Belkacem and S. Abderrahmane, "Using augmented transition network for morphological processing of Arabic," *International Journal of Computer Applications*, vol. 25, no. 10, pp. 22-27, 2011.

[43] M. A. Attia, "Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machine translation," Ph.D. dissertation, University of Manchester, UK, 2008.

[44] W. A. Woods, "Transition network grammars for natural language analysis," *Communications of the ACM*, vol. 13, no. 10, pp. 591-606, 1970.

[45] K. R. Beesley, "Finite-state morphological analysis and generation of Arabic at Xerox Research: status and plans in 2001," in *Proceedings of ACL Workshop on ARABIC Language Processing: Status and Perspective*, Toulouse, France, 2001, pp. 1-8.

[46] N. Habash, *Introduction to Arabic Natural Language Processing*. San Rafael, CA: Morgan & Claypool Publishers, 2010.

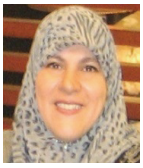
[47] P. Vincke, *L'aide multicritère a la décision*. Bruxelles: Editions de l'universite de Bruxelles, 1989.

[48] C. L. Hwang and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications: A State-of-the-Art Survey*. Berlin: Springer, 1981.

[49] L. Belguith and N. Chaaben, "Analyse et désambiguïsation morphologiques de textes arabes non voyellés," in *Actes de la 13ème confrence sur le Traitement Automatique des Langues Naturelles*, Leuven, Belgium, 2006, pp. 493-501.

[50] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Edmonton, Canada, 2003, pp. 173-180.

[51] S. Khoja, "APT: Arabic part-of-speech tagger," in *Proceedings of the Student Workshop at North American Chapter of the Association for Computational Linguistics (NAACL2001)*, Pittsburg, PA, 2001, pp. 20-25.
[52] J. Giménez and L. Màrquez, "SVMTool: a general POS tagger generator based on support vector machines," in *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, 2004.
[53] M. Diab, "Second generation AMIRA tools for Arabic processing: fast and robust tokenization, POS tagging, and base phrase chunking," in *Proceedings of 2nd International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, 2009, pp. 285-288.
[54] N. Habash, O. Rambow, and R. Roth, "MADA+ TOKAN: a toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization," in *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR),* Cairo, Egypt, 2009, pp. 102-109.



Kheira Zineb Bousmaha

She is a doctor studentand teacher at the department of computer Science, University of Oran 1, Ahmed Ben Bella, Algeria. She received her Magister in Computer Science from Oran University. Currently. Her search has focused on Specification and modeling Information Systems in Arabic language. She is member of Arabic Natural language Processing Research Group (ANLP-RG) at Sfax, Tunisia and member of laboratory RIIR (industrial computer and networks) at Oran Algeria.

Mustapha Kamel rahmouni

He is currently a researcher and Professor at the university Ben Bella, Oran 1 (Algeria), Department of Computer Science. He holds a Ph.D. in Operational Research from Southampton University (UK). His current interests include: Modeling Security in Information Systems using UML extensions, Cryptography. He is member of laboratory RIIR (industrial computer and networks) at Oran Algeria.

Belkacem Kouninef https://orcid.org/0000-0002-4665-3467

He woks as an Associate Professor at the National Institute of Telecommunication and Information and Communication Technology of Oran (INTTIC) Algeria. He is head of a research group in the LaRATIC research laboratory at the INTTIC institute. His main areas of research include: Natural Language Processing, Information Technology, Innovative web applications and e-learning techniques, Information system and database.

Lamia Belguith Hadrich

She is Professor of Computer Science at Faculty of Economics and Management of Sfax (FSEGS) - University of Sfax (Tunisia). She is teacher at the department of computer Science of FSEGS since 1992. She is author of over 80 publications. She is Head of Arabic Natural language Processing Research Group (ANLP-RG) of Multimedia, Information systems and Advanced Computing Laboratory (MIRACL).