Journal of King Saud University – Computer and Information Sciences 29 (2017) 156–163

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

# Morphological, syntactic and diacritics rules for automatic diacritization of Arabic sentences

Amine Chennoufi *, Azzeddine Mazroui

*Department of Mathematics and Computer Science, Faculty of Sciences, University Mohamed First, B-P 717, 60000 Oujda, Morocco*

ARTICLE INFO

*Article history:*
Received 13 January 2016
Revised 25 May 2016
Accepted 23 June 2016
Available online 9 July 2016

*Keywords:*
Arabic language
Automatic diacritization
Arabic diacritical marks
Morphological analysis
Smoothing techniques
Hidden Markov model

ABSTRACT

The diacritical marks of Arabic language are characters other than letters and are in the majority of cases absent from Arab writings. This paper presents a hybrid system for automatic diacritization of Arabic sentences combining linguistic rules and statistical treatments. The used approach is based on four stages. The first phase consists of a morphological analysis using the second version of the morphological analyzer Alkhalil Morpho Sys. Morphosyntactic outputs from this step are used in the second phase to eliminate invalid word transitions according to the syntactic rules. Then, the system used in the third stage is a discrete hidden Markov model and Viterbi algorithm to determine the most probable diacritized sentence. The unseen transitions in the training corpus are processed using smoothing techniques. Finally, the last step deals with words not analyzed by Alkhalil analyzer, for which we use statistical treatments based on the letters. The word error rate of our system is around 2.58% if we ignore the diacritic of the last letter of the word and around 6.28% when this diacritic is taken into account.


## 1. Introduction

The diacritical mark is a sign accompanying a letter to modify the corresponding sound or to distinguish the word from another homonym word. Diacritical marks are widely used in Semitic languages including Arabic, Hebrew and other languages like Urdu. The purpose of these signs is to clarify the morphological structure, the grammatical function, the semantic meaning of words and other linguistic and voice features (Debili and Achour, 1998). Diacritical marks in the Arabic texts are often absent (Farghaly and Shaalan, 2009), unlike Latin languages like French, where the presence of vowels in the texts is mandatory (the vowels in Latin languages play in most cases the same function as diacritical marks in Arabic language). Indeed, according to Habash (2010), diacritical marks are absent in 98% of Arabic texts, and an undiacritized word can have several potential diacritizations in over 77% of cases (Boudchiche and Mazroui, 2015).

Arabic diacritical marks are classified into three groups (Zitouni et al., 2006):

1) The first group consisting of three single short diacritics: “ ◌َ ” fatha, “ ◌ُ ” damma and “ ◌ِ ” kasra. Thus, by adding any of these signs with the letter “ م ” /m[1]/, we obtain the following respective sounds: “ مَ ” /ma/, “ مُ ”/mu/ and “ مِ ”/mi/.
2) The second group represents the doubled case ending diacritics (called tanween): “ ◌ً ” tanween fatha, “ ◌ٌ ” tanween damma and “ ◌ٍ ” tanween kasra. These diacritical marks are reserved only for the last letter of nominal words (nouns, adjectives and adverbs). This phenomenon, called nunation”, has the phonetic effect of adding an N” sound after the corresponding short vowel at the word ending. Thus, the letter “ م ” /m/ with these three signs gives the following sounds: “ مًا ” /mF/ (man), “ مٌ ” /mN/ (mon) et “ مٍ ” /mK/ (min).
3) The third group is called syllabification marks and composed of “ ◌ّ ” shadda (geminate: consonant is doubled in duration) and “ ◌ْ ” sukun. This last group indicates the absence of a short vowel, and reflects a glottal stop while shadda reflects the doubling of a consonant and is always followed by a single diacritic or by a tanween. With the letter “ م ” /m/ and the diacritical mark fatha, we get “ مَّ ” /m∼a/.

* Corresponding author.
*E-mail addresses:* chennoufi.amin@gmail.com (A. Chennoufi), azze.mazroui@gmail.com (A. Mazroui).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

[1] Buckwalter transliteration.

http://dx.doi.org/10.1016/j.jksuci.2016.06.004
1319-1578/

The diacritization operation of Arabic words occurs at two levels: morphological and syntactic levels (Diab et al., 2007). The morphological (lexical diacritics) consists of the internal diacritization of the word (the stem of the word without the last letter) and clarifies the meaning of the word. The syntactic level (casual diacritics) is interested in diacritization of the last letter of the stem and it is used to identify the syntactic role of words in the sentence. Lexical diacritics do not change with the position of the word in the sentence while the casual diacritic depends on the position of the word in the sentence. Thus, the Arabic-speaking reader should understand the Arabic text before reading it properly (Elshafei et al., 2006). This is a difficult for readers who do not have extensive knowledge of the Arabic language. Indeed, Hermena et al. (2015) studied the reaction of the readers facing the diacritized and undiacritized Arabic texts in eye-tracking experience. The results show that readers have benefited from the lifting of the ambiguity of words when diacritical marks are present.

The absence of diacritical marks is a source of complexity for automatic processing systems of the Arabic language that cannot easily determine the meaning of the sentence (Said et al., 2013). Therefore, the need for an automatic diacritization tool of Arabic is more than necessary to remove ambiguity and improve the performances of automatic processing of Arabic applications such as machine translation (Vergyri and Kirchhoff, 2004) and speech recognition (Messaoudi et al., 2004). The introduction of diacritical marks in Arabic dialect speech corpus Levantine[2] (BBN/AUB Babylon DARPA) has helped to increase its reliability and efficiency (Alotaibi et al., 2013).

In addition, the lack of diacritical marks in Arabic sentences represents the main cause of the confusion encountered during its analysis (Boudchiche and Mazroui, 2015) and (Debili and Achour, 1998). The study of Bouamor et al. (2015) showed that the automatic text diacritization increases quality manual tagging of the corpus.

The objective of this paper is to present an automatic Arabic diacritization system combining linguistic rules and statistical treatments. This article is structured as follows: the second paragraph presents the previous works on this area. The third paragraph is devoted to the presentation of the different steps of our system. Indeed, we describe the morphological analysis adopted in the first part of the system. Then, we explain the syntactic control used in the second part and some diacritical rules. We conclude this section by presenting the statistical model adopted in the third and fourth steps of the system. The fourth paragraph deals with the experimentation and evaluation system. We end this paper by a conclusion and some perspectives.

## 2. Related work

Automatic diacritization approaches can be classified into four categories. The first one includes approaches based only on statistical processing. The second category includes hybrid approaches using a morphological analysis followed by a statistical processing. The third category consists of hybrid approaches using morphological analysis, syntactic rules and statistical processing. The last one contains the automatic diacritization systems developed by commercial companies. Approaches based solely on the rules are rarely used because of their complexities due to the high level of ambiguity and the large number of morphosyntactic rules (Debili and Achour, 1998).

### 2.1. Statistics-based models

Gal (2002) was one of the first to use an approach based on hidden Markov models (HMM) for the vocalization of Semitic texts. He has tested his method on the Quran as Arabic texts and the Old Testament for the Hebrew language. The developed application does not extend to all Arabic diacritical marks. Emam and Fischer (2005) extended the statistical processing of diacritization based on examples for Statistical Machine Translation (SMT). Alghamdi et al. (2010) introduced a method based on the quad-gram at the letters. Recently, the researcher (Hifny, 2013) presented a statistical method based on n-gram and compared some smoothing techniques to treat the case of unseen transitions. More recently, Abandah et al. (2015) used a training phase based on recurrent neural networks (RNN) for automatically adding diacritical marks to Arabic text without relying on any prior morphological or contextual analysis. The diacritization is solved as a sequence of transcription problem. Their approach uses a deep bidirectional long short-term memory network that builds high-level linguistic abstractions of text and exploits long-range context in both input directions.

### 2.2. Morphological hybrid approaches

These approaches use both morphological analysis and statistical processing. The works of Vergyri and Kirchhoff (2004) are among the first to use these approaches. Thus, diacritical marks in the Arab conversations are restored by combining morphological and contextual information with a statistical model labeling (acoustic signal). However, they did not model the Shadda diacritic. Similarly, Nelken and Shieber (2005) presented a system that uses an automatic finite state probability, and incorporated a trigram model based on words, a quad-gram language model based on letters and an extremely simple morphological model to identify the prefix and the suffix of word. Zitouni et al. (2006) combined a statistical model based on maximum entropy with the classification of words. The input parameters of this model are the simple letter of the word and the morphological segments and the syntactic state. Habash and Rambow (2007) use the outputs of the morphological analyzer BAMA (Buckwalter, 2004) and individual taggers to choose among these outputs the most selected by these taggers. Diab et al. (2007) were inspired by the machine translation system (SMT), and they introduced six different diacritization schemes developed from observations of the naturally relevant diacritical marks. For these schemes, the morphological analyzer used was MADA (Habash et al., 2013). Recently, Bebah et al. (2014) exploited the morphological analyzer Alkhalil Morpho Sys (Bebah et al., 2011) in a process based on hidden Markov models.

### 2.3. Morphosyntactical hybrid approaches

These methods use both morphological and syntactic rules, and statistical processing. The architecture of the automatic diacritization system proposed by Shaalan et al. (2009) combines three approaches: automatic segmentation, part-of-speech (POS) tagging and the chunk parsing. This method is based on the lexicon of extraction, the bi-gram model and the support vector machines (SVM). The syntactic information is used to treat for each word the diacritical mark of its last letter in a separate final process. The solution, proposed by Rashwan et al. (2011) uses in the first step morphological and syntactic information from ArabMorp[3] and ArabTagger[4] tools, and then an n-gram model and the A* algo-

[2] https://catalog.ldc.upenn.edu/LDC2005S08.

[3] http://www.rdi-eg.com/technologies/Morpho.aspx.

[4] http://www.rdi-eg.com/technologies/POS.aspx.

rithm to select the most likely solution. Said et al. (2013) developed a system based on auto-correction, morphological analysis, part-of-speech tagging and a diacritization process of unseen words in the training corpus. Pasha et al. (2014) presented MADAMIRA (v1.0) which is a disambiguation morphological analysis system of Arabic words in context. This system combines some aspects of both systems MADA (Habash et al., 2013) and AMIRA (Diab et al., 2007). MADAMIRA provides several morphosyntactical outputs including word diacritization. This system uses in disambiguation step the SVM model or the N-gram model. More recently, Shahrour et al. (2015) presented an automatic Arabic diacritization approach that provides the type of the word and the POS tag in the context using additional morphological and syntactic information to re-label the nominal output of the morphological analyzer MADAMIRA.

### 2.4. Applications developed by commercial companies

As for most applications of natural language processing, commercial companies have developed independent automatic diacritization systems or as part of other applications such as a speech synthesizer or a word processor. Among the most interesting projects, we cite the diacritizer ArabDiac[5] developed by RDI society[6], the mobile application Harakat developed by the company multillect[7] and those developed by IBM[8] society, INFO ARAB–ISIS[9], AppTek[10], Sakhr[11] and Aljazeera[12] companies. Recently, Microsoft Research subsidiary of Microsoft Corporation launched an automatic diacritization application of the Arabic language called Arabic Authoring Services[13] (version 1.0) in the version 2013 of Microsoft Word.

## 3. Description of our automatic diacritization system

Given the morphological and syntactic richness of Arabic language, the proposed solution for automatic diacritization will reflect this richness and will be performed in four stages (see Fig. 1). The first stage (module M2) includes morphological analysis out of context and it provides for each word all its possible diacritization forms. In the second step (M3 module), the system uses the syntactic rules to eliminate invalid transitions. The third phase is devoted to statistical processing to choose among the solutions of the second phase those most likely. This is done through the use of an HMM modeling (M4 module), smoothing techniques (module M5) and the Viterbi algorithm (module M6). The last step (M7 and M8 modules) treats the not analyzed words in the morphological stage. It consists of a statistical treatment similar to that of the third step with a model based on letters rather than words.

### 3.1. Morphological analysis

After pre-treatment of the undiacritized text (tokenization and normalization of words), and segmentation into sentences and then into words, the latter are treated with the second version of Alkhalil Morpho Sys analyzer (Boudchiche et al., 2014). Thus, we get all possible diacritization forms of each word taken out of context accompanied by their morphosyntactic information. Indeed, for each diacritization form, the system provides the stem, the clitics attached to the stem, the POS tags and the lemma. In the case of a noun or a verb, the system also provided the root, the syntactic form and the patterns of the stem and the lemma. We opted for the use of this analyzer because their performances are much better than those of the first version of BAMA (Buckwalter, 2002) or the first version of Alkhalil analyzer (Chennoufi and Mazroui, 2016). In particular, the analyzed rate of words is very high since it reached 98.49%.

It should be noted that when the Alkhalil system analyzes a word partially or totally vowelized, it only keeps the outputs whose diacritization is compatible with that of the input word.

### 3.2. Syntactic control

Most research on automatic diacritization has shown that the rate of syntactic errors (error on the last letter of the word) is at least as important as the rate of morphological errors (error related to the word without its last letter). These papers have recommended the use of syntactic rules for improving the performance of the automatic diacritization (Chennoufi and Mazroui, 2016; Schlippe et al., 2008; Shaalan et al., 2009).

We have exploited morphosyntactic information obtained from the morphological analysis to keep only the transitions of words that respect the linguistic rules of Arabic language. We have therefore sought to use the majority of outputs provided by Alkhalil analyzer. Thus, information such as POS tags (noun, verb or particle), syntactic form (genitive name, jussive form of verbs...) and enclitics of words will be very useful in this stage. For example, a preposition without suffix is always followed by a genitive noun. It means that only the transitions between prepositions and genitive nouns are kept. We have implemented 36 syntactic rules and we present in Table 1 some examples of them.

At the end of this step, if no transition between two successive words of a sentence is enabled by the 36 rules, we do not reject any transition for these two words.

### 3.3. Diacritic rules

After preliminary testing of our system, we noticed a significant portion of diacritization errors come from the non-application of the rule relating to the succession of two sukun diacritics ("قاعدة التقاء الساكنين"). In this case, the second sukun is always the Alif letter "ا" /A/. To address this problem and improve the performance of our system, we have adopted in this case the following diacritic rules:

1) If the stem of the predecessor word is the preposition particle "مِنْ" /mino/, then the sukun of its last letter will be replaced by the diacritical fatha ("مِنْ الْكِتَابِ" /mino AlokitaAbi/ (from the book) becomes "مِنَ الْكِتَابِ" /mina AlokitaAbi/).
2) If the predecessor word ends with the letter "م" /m/ "ميم الجمع" (/m/ plural), so the sukun of the word's last letter "ميم الجمع" /m/ will be replaced by the diacritical damma ("قَرَأْتُمْ الْكِتَابَ" /qaraOotumo AlokitaAba/ (you've read the book) becomes "قَرَأْتُمُ الْكِتَابَ" /qaraOotumu AlokitaAba/).
3) If the above cases do not attend (the most common case), then the sukun at the last letter of the word will be replaced by the diacritical kasra ("خُذْ الْكِتَابَ" /xu*o AlokitaAba/ (takes the book) becomes "خُذِ الْكِتَابَ" /xu*i AlokitaAba/).

### 3.4. Statistical analysis at word level

After morphological analysis step that gives for each word all its possible diacritizations, and following the validation step of transitions between pairs of diacritized words and the application of diacritic rules, we present the third stage of diacritization

[5] http://www.rdi-eg.com/technologies/Diac.aspx.
[6] http://www.rdi-eg.com/.
[7] https://multillect.com/apidoc/harakat.
[8] www.ibm.com.
[9] http://www.isisintl.com/.
[10] http://www.apptek.com/.
[11] http://www.sakhr.com/index.php/en/.
[12] http://learning.aljazeera.net/TextEditor.
[13] https://store.office.com/arabic-authoring-services-WA104030856.aspx?assetid=WA104030856.

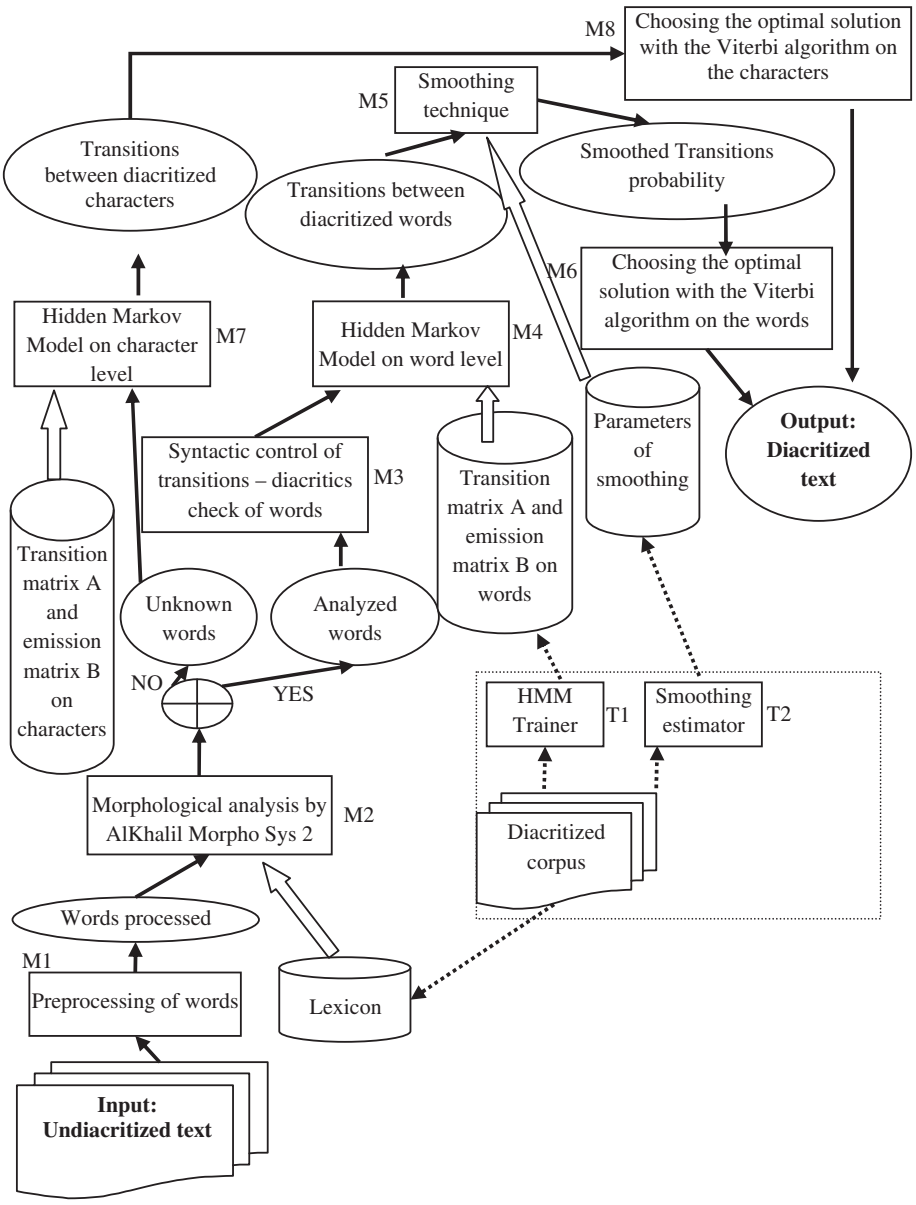


**Figure 1.** Overview of the automatic diacritization architecture.

process. It consists of a statistical treatment based on the hidden Markov models and the Viterbi algorithm (Neuhoff, 1975), which provides the most likely diacritized sentence (Fig. 2). The representation of observed states of HMM are the Arabic words without diacritics (eg "فهمتم" /fhmtm/) and the hidden states are diacritized word forms (eg "فَهِمْتُمْ" /fahimotumo/) (Elshafei et al., 2006; Bebah et al., 2014). This model states provided the best scores of automatic diacritization compared to other hidden states like lists of diacritical marks (Bebah et al., 2014). To smooth the unseen valid transitions in the training corpus, we used the Absolute Discounting Smoothing Technique (Ney and Essen, 1991), which has achieved the highest scores in previous works (Hifny, 2013;Chennoufi and Mazroui, 2014).

### 3.5. Statistical analysis at letter level

During the test phase, another constraint was encountered related to words not analyzed by Alkhalil Morpho Sys and for which the label unknown" was associated. Thereby, the fourth phase of diacritization system relates only to these cases. These words are not diacritized by the third stage of the system. Thus, for each unanalyzed word, another hidden Markov model is used and for which the Arabic letters are the observed state and the diacritized letters are the hidden states. The Viterbi algorithm is also used to choose the most probable solution.

## 4. Experimental phase

### 4.1. Methodology

To achieve statistical phase, transition and emission probabilities $a_{ij}$ and $b_i(t)$ will be estimated during the training step (for details see Bebah et al., 2014). The used estimation method is based on the calculation of maximum likelihood (Manning and Schütze, 1999). Indeed, if we note:

$C = \{Ph_1, \ldots, Ph_M\}$ a representative corpus of Arabic texts formed by $M$ phrases $Ph_k$,

**Table 1**
Examples of syntactic rules used in the automatic diacritization system.

| N° | Rules | Examples |
|---|---|---|
| 1 | The preposition "حرف جر" is always followed by a genitive noun "اسم مجرور" | The transition "مِنَ الْمَدْرَسَةِ" /mina Alomadorasati/ (from the school) is valid<br>The transition "مِنَ الْمَدْرَسَةَ" /mina Alomadorasata/ is not valid |
| 2 | The particle "لَمَّا" /lam∼aA/ is always followed by a verb in the past tense "فعل ماض" or an apocopative verb in the present tense | The transition "لَمَّا ذَهَبَ" /lam∼aA *ahaba/ (when he left) is valid<br>The transition "لَمَّا ذَهَبْ" /lam∼aA *ahaba/ is not valid<br>The transition"لَمَّا يَذْهَبُ" /lam∼aA ya*ohabo/ (when he leaves) is valid<br>The transition "لَمَّا يَذْهَبَ" /lam∼aA ya*ohaba/ is not valid |
| 3 | The relative pronoun "اسم موصول" is always followed by a nominative verb in the present tense "فعل مضارع مرفوع" or a verb in the past tense or a nominative noun "اسم مرفوع" or a particle "حرف" | The transition "الَّذِي يَكْتُبُ" /Al∼a*iy yaktubu/ (who writes) is valid<br>The transition "الَّذِي يَكْتُبَ" /Al∼a*iy yaktuba/ is not valid<br>The transition "الَّتِي أُمَّهَا" /Al∼atiy Âum∼uhaA/ (who his mother) is valid<br>The transition "الَّتِي أُمَّهَا" / Al∼atiy Âum∼ahaA / is not valid |
| 4 | An adverbe "ضرف" not attached to a pronoun is always followed by a genitive noun "اسم مجرور" or a demonstrative pronoun "اسم إشارة" or a relative pronoun "اسم موصول" or a particle "حرف" | The transition "فَوْقَ سُلْمِ" /fawoqa sul∼amĩ/ (above the stair) is valid<br>The transition "فَوْقَ سَلَمْ" /fawoqa sal∼ama/ is not valid<br>The transition"مَسَاءَ الْخَمِيسِ" /masaA'a Alxamiysi/ (on Thursday evening) is valid<br>The transition"مَسَاءَ الْخَمِيسَ" / masaA'a Alxamiysa/ is not valid |

$n_i^k$ = the occurrence number of the hidden state $w_i$ (diacritized word) in the sentence $Ph_k$,
$n_{ij}^k$ = the occurrence number of the transition from the hidden state $w_i$ (diacritized word) to the hidden state $w_j$ (diacritized successor word) in the sentence $Ph_k$,
$m_{it}^k$ = the occurrence number of undiacritized word $u_t$ with the hidden state $w_i$ in the sentence $Ph_k$,
$N_{1+}^k(w_i\bullet)$ = the number of all words repeated once and more after the diacritized word $w_i$ in the sentence $Ph_k$,
$P_{MLE}(w_j) = \frac{\sum_{k=1}^{M} n_j^k}{N}$: The maximum likelihood of the word $w_j$ in the corpus $C$ of size $N$.

Then, the probabilities $a_{ij}$ and $b_i(t)$ can be estimated by the following formulas:

$$a_{ij} = \frac{\max\left\{\sum_{k=1}^{M} n_{ij}^k - D, 0\right\}}{\sum_{k=1}^{M} n_i^k} + \frac{D}{\sum_{k=1}^{M} n_i^k} P_{MLE}(w_j) \sum_{k=1}^{M} N_{1+}^k(w_i\bullet)$$
$$\text{and} \quad b_i(t) = \frac{\sum_{k=1}^{M} m_{it}^k}{\sum_{k=1}^{M} n_i^k} \qquad (1)$$

with the constant $D = 0.5$.

### 4.2. Training and test corpora

Our statistical model was trained on 90% of a large corpus of more than 72 million diacritized words. This training corpus was drawn at random. The remaining 10% (7,176,188 words) will be used to test and evaluate our model. These corpora consist of Tashkeela corpus[14] (63 million of diacritized words), Nemlar corpus (0.5 million of diacritized words) (Attia et al., 2005) and a part of RDI corpus[15] not redundant with Tashkeela corpus (8.5 million of diacritized words). They are composed of texts taken from diacritized old classic books and few modern documents. The topics covered several thematic areas including theology, grammar, history, economy and geography.

The HMM based on the letters and specific to unanalyzed words in the morphological step was trained on the same corpus as that used for the HMM related to words.

We observed that some texts contain partially diacritized words. These texts have been eliminated and are not part of the 72 million words used in the training and testing phases. Similarly, diacritical marks are not always arranged in the same way in all texts. Indeed, some diacritic writing rules differ sometimes from one Arab country to another and from one area to another. Thus, to evaluate our system we have standardized the diacritic scriptures of training and test corpora with the output of Alkhalil analyzer. Finally, some spelling mistakes often appear in some texts of the corpus. We have carried out the correction of these errors.

#### 4.2.1. Standardization of diacritic rules

By analyzing the writing rules of diacritical marks in the different texts, we found the following differences:

1) Diacritic marks on long vowels (Alif "ا"/A/, Waw "و" /W/, Yae "ي" /Y/) have three forms of writing. The first form does not put diacritical marks on long vowels ("الْمَالِيزِيونَ " /AlomAlyziywna/ (Malaisiens)), the second way brings them after long vowels ("الْمَالِيزِيوُنَ" / AlomAalyiziywuna/) and the 3rd writing puts the diacritical mark before the long vowel ("الْمَالِيزِيُونَ" /AlomaAliyziyuwna/). We adopted this last rule because it is similar to that used by Alkhalil analyzer.
2) The Tanween fatha sign with the letter Alif "ا"/A/ has two forms of writing: one before the letter ("سَلَامًا" /salaAmFA/ (peace)) and the other after the letter ("سَلَاماً" /salaAmAF/). The second form has been adopted.
3) Shadda sign also presents two forms of writing: one before the diacritical mark and the other after the diacritical mark. The rule that we have adopted is always to write the Shadda sign before the diacritical mark.

We applied these three rules to all words of the corpora.

#### 4.2.2. Correcting spelling errors

We also correct some errors that were recurrent in the corpora.

1) In some cases, there are words with Alif maksoura "ى" /Y/ instead of the letter Yae "ي" /y/. Thus, whenever the letter Alif maksoura is accompanied by a diacritical mark, we proceed to replace it by the letter Yae (e.g. the word "عَلِىّ" /EaliY∼u/ will be replaced with the proper name "عَلِيّ" / Ealiy∼u/).
2) Some words contain a succession of diacritical marks ("عَلْمَ"). In this case, we only keep the first diacritical mark and reject the others.

[14] http://sourceforge.net/projects/tashkeela/.
[15] http://www.rdi-eg.com/RDI/TrainingData/.

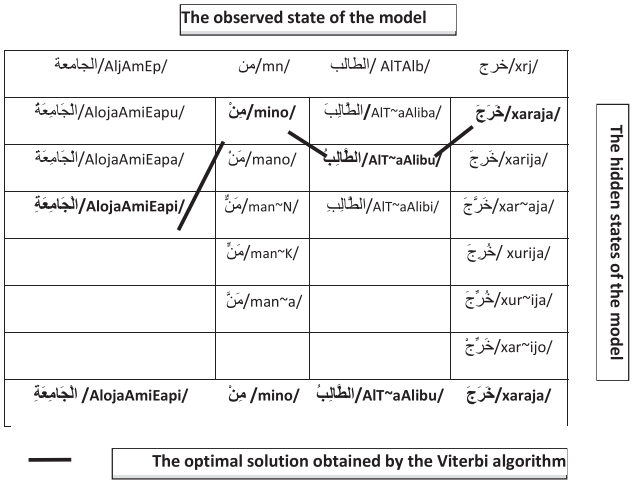


**Figure 2.** Example of using the Viterbi algorithm on an Arabic sentence to find the optimal solution.

3) Sometimes the letter Alif with hamza below"إ /I/ is not accompanied by the diacritic kasra that represents the only possibility of diacritization. In this case, we add this diacritical mark.
4) The diacritic rules mentioned in paragraph 3.3 are not always respected in the corpora. We therefore apply these rules to all words of the corpora.

### 4.3. Results

Before presenting the results, it is important to explain the evaluation methodology both at the word and at the letter level. The error rate at the word level is noted WER (WER: Word Error Rate) and the error rate at the letter level is noted DER (DER: Diacritic Error Rate). For each of these two types of errors, we introduce the rate that takes into account the diacritical mark of the last letter and the one that ignores this diacritical mark. Consequently, WER1 represents the rate of the words incorrectly diacritized by the system taking into account the diacritic of the last letter. WER2 is defined as WER1 except that it ignores the diacritical mark of the last letter. Similarly, DER1 is the rate of letters incorrectly diacritized including the last letter, while DER2 is defined as DER1 but does not consider the last letter of the word. For this metric, the numbers and the punctuations are not considered in the evaluation process.

#### 4.3.1. Contribution of syntactic and diacritic rules

To assess the impact of the integration of diacritical and syntactic rules, we evaluate three automatic diacritization systems. The first system is the one developed in a previous work (Chennoufi and Mazroui, 2016), and which is based on morphological analysis and statistical treatments without syntactic and diacritic rules. The second system is obtained by integrating the diacritic rules in the first system, and the third is one that incorporates both diacritical and syntactic rules. After completing the training steps on the same training corpus for these three systems, we tested them on the test corpus consisting of 7.17 million words. The results of the different error rates for these three systems are shown in Table 2.

We note that the integration of diacritic rules has significantly improved the accuracy of the system. Indeed, WER1 decreased from 8.29% for the system does not incorporate the diacritic rules to 6.50% for one that incorporates these rules. Similarly, WER2 decreased from 4.10% for the first system to only 2.58% for the second. Given that every word counted in calculating WER2 will be automatically counted in the calculation of WER1, we can assert that the integration of diacritic rules have benefited mainly to improve WER2. Analyzing the results of the third system which integrated syntactic rules, we find that the integration of these rules has allowed only to correct some errors made by the second system at the vowel of the last character of the word. Indeed, just WER1 decreased from 6.50% to 6.28% while WER2 remained unchanged. The other error rates related to letter (DER1 and DER2) also presented significant decreases. Thus, the integration of syntactic and diacritic rules allowed a significant improvement in the system performances.

**Table 2**
Evaluation results of the two automatic diacritization systems on the test corpus.

| Approach of automatic diacritization system | WER1 (%) | WER2 (%) | DER1 (%) | DER2 (%) |
|---|---|---|---|---|
| Morphological analysis + Statistics (Chennoufi and Mazroui, 2016) | 8.29 | 4.10 | 2.93 | 1.54 |
| Morphological analysis + Diacritic rules + Statistics | 6.50 | **2.58** | 2.05 | **0.90** |
| Morphological analysis + Diacritic rules + Syntactic rules + Statistics | **6.28** | **2.58** | **1.99** | **0.90** |

Best results are shown in boldface.

#### 4.3.2. Comparison with the results of the literature

To position our system with respect to other Arabic automatic diacritization applications, we compare the performance of our system with those of two other systems. The first one is MADAMIRA system (Version 1 – 25/08/2014) and the second is Arabic Authoring Services (كُتُبْ) integrated with Microsoft Word (version 2013). Indeed, we ran these three systems (MADAMIRA, Arabic Authoring Services and our system) on a random sample of 187,723 words from test corpus. The outputs of these three systems have undergone the same standardization treatments of paragraphs 3.3, 4.2.1 and 4.2.2 above. The results of these evaluations are presented in Table 3.

The different error rates of MADAMIRA and Arabic Authoring Services (كُتُبْ) are relatively high. Indeed, the error rate WER1 of MADAMIRA systems based on the SVM model and the language model are respectively equal to 36.07% and 27.29%. Similarly, the Arabic Authoring Services System (كُتُبْ) indicates 20.56% for WER1. However, our system shows a much lower rate of order 6.22%. Similar remarks can be raised for the other error rates WER2, DER1 and DER2.

The high error rate of the systems MADAMIRA and Arabic Authoring Services (كُتُبْ) can be explained in part by the nature of the test corpus. Indeed, this corpus is essentially made up of classical Arabic texts, while both systems are more suited to contemporary texts (MSA: Modern Standard Arabic).

On the other hand, to ensure objective comparison between our system and some previous work like Abandah et al. (2015), which is the most recent work and announcing the best results, we use the same evaluation metric of diacritization introduced by Zitouni et al. (2006) and adopted by Habash and Rambow (2007), Rashwan et al. (2011), Abandah et al. (2015) and other authors. For this metric, the numbers and the punctuations are also consid-

**Table 3**
Comparison between three Arabic automatic diacritization systems.

| Automatic diacritization system | WER1 | WER2 | DER1 | DER2 |
|---|---|---|---|---|
| MADAMIRA (SVM) | 36.07 | 20.21 | 12.66 | 7.12 |
| MADAMIRA (language model) | 27.29 | 16.14 | 9.21 | 5.56 |
| Arabic Authoring services (كُتُبْ) | 20.56 | 11.18 | 7.19 | 4.16 |
| Our system | **6.22** | **2.53** | **1.98** | **0.90** |

Best results are shown in boldface.

**Table 4**
Performance comparison between Abandah diacritization system and our system.

| Automatic diacritization system | WER1 | WER2 | DER1 | DER2 |
|---|---|---|---|---|
| Abandah et al. (2015) | 5.82 | 3.54 | 2.09 | 1.28 |
| Our system | **4.45** | **1.86** | **1.52** | **0.71** |

Best results are shown in boldface.

ered in the evaluation process. We tested our system on the same corpus used as a test corpus by Abandah et al. (2015). This corpus consists on ten books of Tashkeela corpus and the Quran. Table 4 below shows the scores of Abandah et al. (2015) and our system.

Table 4 shows that the error rates WER1 and DER1 of Abandah system are respectively equal to 5.82% and 2.09%. Our system has a lower error rate WER1 and DER1 respectively equal to 4.45% and 1.52%.

It should be noted that this assessment methodology is biased and does not reflect the real performances of the system since the punctuations and numbers are never diacritized in the Arabic texts and their error rates are always equal to zero.

### 4.4. Discussion

The good performances of our system are consequences of:

1) The robustness of the second version of AlKhalil analyzer used by our system in the morphological stage;
2) The use of syntactic and diacritic rules;
3) The strong representation of the corpus used in the training phase given its large size.

The evaluation of this automatic diacritization system of Arabic sentences combining morphological analysis, syntactic and diacritic rules and statistical processing produces better performance than other systems. The integration of syntactic rules has contributed to the improvement of the error rate WER1, and they particularly allowed correcting some mistakes at the last character. In the same, the integration of diacritic rules has reduced the error rate WER2.

## 5. Conclusion

This paper presents a model of automatic Arabic diacritization based on hybrid approach that combines the linguistic rules and statistical processing. The use of morphological, syntactic and diacritic rules combined with the hidden Markov models provides the best performances. Indeed, the evaluation results are very encouraging and much better in comparison with other available systems. Spelling errors in the training and testing corpora and their enrichment by other texts will improve these scores. In addition, the integration of other syntactic rules will contribute to decrease the error rates.

## References


Abandah, G.A., Graves, A., Al-Shagoor, B., Arabiyat, A., Jamour, F., Al-Taee, M., 2015. Automatic diacritization of Arabic text using recurrent neural networks. Int. J. Doc. Anal. Recogn. 18, 183–197. http://dx.doi.org/10.1007/s10032-015-0242-2.

Alghamdi, M., Muzaffar, Z., Alhakami, H., 2010. Automatic restoration of Arabic diacritics: a simple, purely statistical approach. Arab. J. Sci. Eng. 35, 125–135.

Alotaibi, Y.A., Meftah, A.H., Selouani, S.-A., 2013. Diacritization, automatic segmentation and labeling for Levantine Arabic speech. In: 2013 IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE). IEEE, pp. 7–11. http://dx.doi.org/10.1109/DSP-SPE.2013.6642556.

Attia, M., Choukri, K., Yaseen, M., 2005. Specifications of the Arabic written corpus produced within the Nemlar project.

Bebah, M., Meziane, A., Mazroui, A., Lakhouaja, A., 2011. Alkhalil Morpho Sys. In: 7th International Computing Conference in Arabic. Riyadh, Saudi Arabia.

Bebah, M., Chennoufi, A., Mazroui, A., Lakhouaja, A., 2014. Hybrid approaches for automatic vowelization of Arabic texts. Int. J. Nat. Lang. Comput. 3, 53–71. http://dx.doi.org/10.5121/ijnlc.2014.3404.

Bouamor, H., Zaghouani, W., Diab, M., Obeid, O., Oflazer, K., Ghoneim, M., Hawwari, A., 2015. A pilot study on Arabic multi-genre corpus diacritization. In: Proceedings of the Second Workshop on Arabic Natural Language Processing. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 80–88. <http://dx.doi.org/10.18653/v1/W15-3209>.

Boudchiche, M., Mazroui, A., 2015. Evaluation of the ambiguity caused by the absence of diacritical marks in Arabic texts: statistical study. In: 2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA). IEEE, pp. 1–6. http://dx.doi.org/10.1109/ICTA.2015.7426904.

Boudchiche, M., Mazroui, A., Bebah, M., Lakhouaja, A., 2014. L'Analyseur Morphosyntaxique AlKhalil Morpho Sys 2. In: 1ère Journée Doctorale Nationale Sur L'ingénierie de La Langue Arabe, (JDILA'14). Rabat, Morocco.

Buckwalter, T., 2002. Arabic Morphological Analyzer Version 1.0. Linguist. Data Consort. n° LDC2002L49.

Buckwalter, T., 2004. Arabic morphological analyzer version 2.0. Linguist. Data Consortium. Univ. Pennsylvania.

Chennoufi, A., Mazroui, A., 2014. Méthodes de lissage d'une approche morpho-statistique pour la voyellation automatique des textes arabes. In: 21ème Traitement Automatique Des Langues Naturelles. pp. 443–448.

Chennoufi, A., Mazroui, A., 2016. Impact of morphological analysis and a large training corpus on the performances of Arabic diacritization. Int. J. Speech Technol. 19, 269–280. http://dx.doi.org/10.1007/s10772-015-9313-5.

Debili, F., Achour, H., 1998. Voyellation automatique de l'arabe. In: Proc. Work. Comput. Approaches to Semit. Lang. pp. 42–49.

Diab, M., Hacioglu, K., Jurafsky, D., 2007. Automatic Processing of Modern Standard Arabic Text. In: Arabic Computational Morphology. Springer, Netherlands, Dordrecht, pp. 159–179. http://dx.doi.org/10.1007/978-1-4020-6046-5_9.

Elshafei, M., Al-Muhtaseb, H., Alghamdi, M., 2006. Machine generation of arabic diacritical marks. In: The 2006 World Congress in Computer Science Computer Engineering, and Applied Computing. Las Vegas, USA, pp. 128–133.

Emam, O., Fischer, V., 2005. Hierarchical approach for the statistical vowelization of arabic text. US 8,069,045 B2.

Farghaly, A., Shaalan, K., 2009. Arabic natural language processing. ACM Trans. Asian Lang. Inf. Process. 8, 1–22. http://dx.doi.org/10.1145/1644879.1644881.

Gal, Y., 2002. An HMM approach to vowel restoration in Arabic and Hebrew. In: Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages. Association for Computational Linguistics, Morristown, NJ, USA, pp. 1–7. http://dx.doi.org/10.3115/1118637.1118641.

Habash, N., Rambow, O., 2007. Arabic diacritization through full morphological tagging. Hum. Lang. Technol. In: 2007 Conf. North Am. Chapter Assoc. Comput. Linguist. Companion, vol. Short Pap.

Habash, N., Roth, R., Rambow, O., Eskander, R., Tomeh, N., 2013. Morphological Analysis and Disambiguation for Dialectal Arabic. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). Atlanta, GA, pp. 426–432.

Habash, N.Y., 2010. Introduction to Arabic natural language processing. Synth. Lect. Hum. Lang. Technol. 3, 1–187. http://dx.doi.org/10.2200/S00277ED1V01Y201008HLT010.

Hermena, E.W., Drieghe, D., Hellmuth, S., Liversedge, S.P., 2015. Processing of Arabic diacritical marks: phonological–syntactic disambiguation of homographic verbs and visual crowding effects. J. Exp. Psychol. Hum. Percept. Perform. 41, 494–507. http://dx.doi.org/10.1037/xhp0000032.

Hifny, Y., 2013. Restoration of Arabic diacritics using dynamic programming. In: 8th International Conference on Computer Engineering & Systems (ICCES). IEEE, pp. 3–8. http://dx.doi.org/10.1109/ICCES.2013.6707161.

Manning, C.D., Schütze, H., 1999. Foundations of Statistical Natural Language Processing. MIT Press.

Messaoudi, A., Lamel, L., Gauvain, J.-L., 2004. The LIMSI RT-04 BN Arabic System. In: Darpa RT04. Palisades NY.

Nelken, R., Shieber, S.M., 2005. Arabic diacritization using weighted finite-state transducers. In: Proc. ACL Work. Comput. Approaches to Semit. Lang.

Neuhoff, D., 1975. The Viterbi algorithm as an aid in text recognition (Corresp.). IEEE Trans. Inf. Theory 21, 222–226. http://dx.doi.org/10.1109/TIT.1975.1055355.

Ney, H., Essen, U., 1991. On smoothing techniques for bigram-based natural language modelling. [Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing, vol. 2. IEEE, pp. 825–828. http://dx.doi.org/10.1109/ICASSP.1991.150464.

Pasha, A., Al-Badrashiny, M., Diab, M., Kholy, A. El, Eskander, R., Habash, N., Pooleery, M., Rambow, O., Roth, R.M., 2014. MADAMIRA: a fast, comprehensive tool for morphological analysis and disambiguation of Arabic, in: Proceedings of LREC. Reykjavik, Iceland.

Rashwan, M.A.A., Al-Badrashiny, M.A.S.A.A., Attia, M., Abdou, S.M., Rafea, A., 2011. A stochastic arabic diacritizer based on a hybrid of factorized and unfactorized textual features. IEEE Trans. Audio. Speech. Lang. Process. 19, 166–175. http://dx.doi.org/10.1109/TASL.2010.2045240.

Said, A., El-Sharqwi, M., Chalabi, A., Kamal, E., 2013. A hybrid approach for Arabic diacritization. In: Natural Language Processing and Information Systems. Springer, Berlin Heidelberg, pp. 53–64. http://dx.doi.org/10.1007/978-3-642-38824-8_5.

Schlippe, T., Nguyen, T., Vogel, S., 2008. Diacritization as a machine translation problem and as a sequence labeling problem. In: 8th AMTA Conference.

Shaalan, K., Bakr, H.M.A., Ziedan, I., 2009. A hybrid approach for building arabic diacritizer. In: Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages. pp. 27–35.

Shahrour, A., Khalifa, S., Habash, N., 2015. Improving Arabic Diacritization through Syntactic Analysis. In: The 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015. Association for Computational Linguistics, pp. 1309–1315.

Vergyri, D., Kirchhoff, K., 2004. Automatic diacritization of Arabic for acoustic modeling in speech recognition. In: Proc. Work. Comput. Approaches to Arab. Script-based Lang.

Zitouni, I., Sorensen, J.S., Sarikaya, R., 2006. Maximum entropy based restoration of Arabic diacritics. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL – ACL '06. Association for Computational Linguistics, Morristown, NJ, USA, pp. 577–584. http://dx.doi.org/10.3115/1220175.1220248.