SHAKKIL: An Automatic Diacritization System for Modern Standard Arabic Texts

Amany Fashwan
Phonetics and Linguistics Department,
Faculty of Arts, Alexandria University,
Alexandria, Egypt
a.fashwan@gmail.com

Sameh Alansary
Phonetics and Linguistics Department,
Faculty of Arts, Alexandria University,
Alexandria, Egypt
sameh.alansary@bibalex.org

Abstract

This paper sheds light on a system that would be able to diacritize Arabic texts automatically (SHAKKIL). In this system, the diacritization problem will be handled through two levels; morphological and syntactic processing levels. The adopted morphological disambiguation algorithm depends on four layers; Uni-morphological form layer, rule-based morphological disambiguation layer, statistical-based disambiguation layer and Out Of Vocabulary (OOV) layer. The adopted syntactic disambiguation algorithms is concerned with detecting the case ending diacritics depending on a rule based approach simulating the shallow parsing technique. This will be achieved using an annotated corpus for extracting the Arabic linguistic rules, building the language models and testing the system output. This system is considered as a good trial of the interaction between rule-based approach and statistical approach, where the rules can help the statistics in detecting the right diacritization and vice versa. At this point, the morphological Word Error Rate (WER) is 4.56% while the morphological Diacritic Error Rate (DER) is 1.88% and the syntactic WER is 9.36%. The best WER is 14.78% compared to the best-published results, of (Abandah et al., 2015); 11.68%, (Rashwan et al., 2015); 12.90% and (Habash et al., 2009); 13.60%.

1 Introduction

Modern Standard Arabic (MAS) is currently the sixth most widely spoken language in the world with estimated 422 million native speakers. It is usually written without diacritics which makes it difficult for performing Arabic text processing. In addition, this often leads to considerable ambiguity since several words that have different diacritic patterns may appear identical in a diacritic-less setting. In fact, a text without diacritics may bring difficulties for Arabic readers. It is also problematic for Arabic processing applications where the lack of diacritics adds another layer of ambiguity when processing the input data (Shaalan et al., 2009).

Diacritics restoration is the problem of inserting diacritics into a text where they are missing. Predicting the correct diacritization of the Arabic words elaborates the meaning of the words and leads to better understanding of the text, which in turn is much useful in several real life applications such as Information Retrieval (IR), Machine Translation (MT), Text-to-speech (TTS), Part-Of-Speech (POS) tagging and others.

For full diacritization of an Arabic words, two basic components are needed:

1) Morphology-dependent that selects the best internal diacritized form of the same spelling; e.g. the word علم "Elm" has different diacritized forms; عِلْم "Eilom" "science", عَلَم "Ealam" "flag", عَلَّمَ "Eal˜ama" "taught" and عَلِمَ "Ealima" "knew".

2) Syntax-dependent that detects the best syntactic case of the word within a given sentence; i.e. its role in the parsing tree of that sentence. For example; دَرَسْتُ عِلْمَ الرِّيَاضِيَّاتِ "darasotu Eiloma Alr˜iyADiy˜Ati" "I studied Mathematics" implies the syntactic diacritic of the target word - which is an "object" in the parsing tree - is "Fatha", while يُفِيدُ عِلْمُ الرِّيَاضِيَّاتِ جَمِيعَ الْعُلُومِ "yufiydu Eilomu Alr˜iyADiy˜Ati jamiyEa AloEuluwmi" "Mathematics benefits all sciences" implies the syntactic diacritic of the target word which is a "subject"

Proceedings of The Third Arabic Natural Language Processing Workshop (WANLP), pages 84–93,
Valencia, Spain, April 3, 2017.

in the parsing tree - is "Damma" (Rashwan et al., 2009).

2 Related Work

Diacritic restoration has been receiving increasing attention and has been the focus of several studies. Different methods such as rule-based, example-based, hierarchical, morphological and contextual-based as well as methods with Hidden Markov Models (HMM) and weighted finite state machines have been applied for the diacritization of Arabic text.

Among these trials, that are most prominent, (Habash and Rambow, 2005; 2007a), (Zitouni et al., 2006), (Diab et al., 2007), (Roth et al., 2008), (Shaalan et al., 2008; 2009), (Habash et al., 2009), (Rashwan et al., 2009;2011; 2014; 2015), (Metwally et al., 2016), (Abandah et al., 2015), (Chennoufi and Mazroui, 2016a; 2016b), and (Alansary, 2016a).

In addition, some software companies have developed commercial products for the automatic diacritization of Arabic; Sakhr Arabic Automatic Diacritizer[1], Xerox's Arabic Morphological Processor[2] and RDI's Automatic Arabic Phonetic Transcriptor (Diacritizer/Vowelizer)[3]. Moreover, there are also other free online available systems; Meshkal Arabic Diacritizer[4], Harakat Arabic Diacritizer[5], Al-Du'aly[6], Farasa[7] and Google Tashkeel which is no longer working where the tool is not available now.

It has been noticed that most of the previous systems use different statistical approaches in their diacritization systems and few trials use the rule-based approach. The difference between the current proposed system and the others is the interaction between the rule-based approach and statistical approach using different machine learning techniques, where the rule-based can help the statistical-based in detecting the right diacritization and vice versa. In addition, extracting and implementing some syntactic rules for case ending restoration where, to our knowledge, none of the previous systems make use of syntax with the exception of (Shahrour et al., 2015), (Chennoufi and Mazroui, 2016b) and (Alansary, 2016a) who have integrated syntactic rules.

[1]http://aramedia.com/nlp2.htm [Acc. 12-2-2015].

[2]http://aramedia.com/diacritizer.htm [Acc. 12-2-2015].

[3]http://www.rdi-eg.com/technologies/arabic_nlp.htm [Acc. 12-2-2015].

[4]http://tahadz.com/mishkal [Acc. 4-4-2015].

[5]http://harakat.ae/ [Acc. 4-4-2015].

[6]http://faraheedy.mukhtar.me/du2alee/tashkeel [Acc. 20-8-2016]

[7]http://qatsdemo.cloudapp.net/farasa/ [Acc. 28-12-2016]

3 System Data Sets

The used data in the current system was selected from a Morphologically Annotated Gold Standard Arabic Resource **(MASAR)** for MSA (Alansary, 2016b). The texts were selected from different sources; Newspapers, Net Articles and Books. Moreover, these selected texts covered more than one genre. Each word is tagged with features, namely, Lemma, Gloss, prefixes, Stem, Tag, suffixes, Gender, Number, Definiteness, Root, Stem Pattern, Case Ending, Name Entity and finally Vocalization. In the current proposed system, the data is subdivided into three kinds of data sets: **a) "Training Data"** about 450,000 words. **b) "Development Data"** about 55,000 words, and **c) "Testing Data"** about 55,000 words.

3.1 Extracting Arabic Linguistic Rules

It must be noted that extracting Arabic linguistic rules is not an easy task where these rules must be represented in a generalized format in a way that simulates the concerned component of the language. So these rules need to be constrained in a certain order to avoid overlapping among them. In this stage a number of rules related to morphology, definiteness and case ending are extracted from the training data set in a formalized and generalized format and implemented in the system.

As concerning to morphological rules, we are concerned with extracting some rules that help in detecting the POS of a certain words depending on the preceding or succeeding POS tags or word forms. In addition to the previous kind of rules, other rules have been extracted to detect the whole morphological analysis and hence the full internal diacritization depending on the preceding or succeeding POS tags or word forms. These rules get only the correct solution for the words context and consequently eliminate all wrong solutions. In addition, they may get only one morphological/diacritized form or get more than one correct morphological analysis after eliminating the wrong solutions. The extracted morphological rules are of 11 categories; Prepositional Phrases, Subjunctive Particles, Jussive Particles, Accusative Particles, Interrogative Particles, Pronouns, Verb Particles, Amendment Particle "l'kin", Time and Place Adverbs, Verbs and

Some Word Forms. In each category, a number of sub rules are extracted and implemented.

One of these rules states that the accusative particles ">an~a/PART", "li>an~a/PART" and "ka >an~a/PART" that have empty suffix may be followed by NOUN, PRON, PREP and some particles. Consequently, no ADJ, ADV, IV or PV (except 'layos/PV') follows these stems as Rule (1) shows:

Rule (1)

(Stem [Previous] % ">an~a/PART" & Suf [Previous] = "")
{ [Current]: @ Tag %"NOUN"
@ Tag = "PRON"
@ Tag = "PREP"
@ Stem = "layos/PV"
@ Stem = "lA/PART"
@ Stem = "mA/PART"
@ Stem = "lam/PART"
}

Ex. (1) shows that if the word form to be analyzed is "أفضل" ">fDl", in this case the Rule (1) will be applied and it will eliminate the PV, IV and ADJ forms of this word; "أَفْضَلَ" ">afoDala" "bestow", "أُفَضِّل" ">ufaD~il" "prefer", "أُفَضَّل" ">ufaD~al" "be prefered", and "أَفْضَل" ">afoDal" "better/best" and the nominal form of this word; "أَفْضَل" ">afoDal" "better/best" will be selected and assigned to this word.

Ex. (1) فهو يرى أن أفضل المقترحات هو المقترح الفيدرالي

fahuwa yaraY >an~a >afoDal AlomuqotaraHAt huwa AlomuqotraH AlofiydorAliy
He believes that the best proposals is the federal proposal

The realization of nominal case in Arabic is complicated by its orthography, which uses optional diacritics to indicate short vowel case morphemes, and by its morphology, which does not always distinguish between all cases. Additionally, case realization in Arabic interacts heavily with the realization of definiteness, leading to different realizations depending on whether the nominal is indefinite, i.e., receiving nunation (تنوين), definite through the determiner Al+ (ال+) or definite through being the governor of an EDAFAH possessive construction (إضافة) (Habash et al., 2007b). In addition, case ending realization in Arabic interacts in some cases with other information: **1. Word Patterns:** the diptote word patterns (الممنوع من الصرف) refer to a category of nouns and adjectives that have special case ending when they are indefinite since they do not take tanween. It must be noticed that when these words are definite, they take regular case ending diacritics.

2. Verb Transitivity: the transitivity of the verbs helps sometimes in detecting the subject (which receives nominative case ending) and object (which receives the subjunctive case ending).

3. Feminine Plural Word Forms: in Arabic, the object receives the case ending for the genitive case instead of the subjunctive case; this is in the case of those words end with "ات" "At/NSUFF" "جمع المؤنث السالم" (Fashwan, 2016). In order to detect the case ending diacritics, a prior step is done where some Arabic linguistic rules have been extracted and implemented to detect the definiteness of each word depending on its context or its selected morphological analysis. In addition, the stem pattern of each stem has been detected depending on its root, stem and lemma. Moreover, the transitivity of each verb has been detected depending on its lemma.

After that, some Arabic linguistic rules have been extracted and implemented to detect the case ending depending on a window of -/+ 3 words around the focused word taking into consideration the context, the selected morphological analysis, definiteness feature, stem pattern and verb's transitivity. The extracted case ending rules are of 4 categories; rules for Detecting Case Ending (MOOD) of the Imperfect Verbs, rules for detecting the case ending of Noun Phrases, rules for detecting the case ending of Adverb phrases and rules for detecting the case ending of adjectival phrases. One of these rules states that if there is a noun phrase preceded by "أَمَّا" ">am~A" "as for/concerning" or "لَوْلَا" "lawolA" "if not" then the noun must be in nominative case taking into consideration the definiteness feature; if the noun is "INDEF" then the noun must receive nunation (Tanween Damma "ٌ"), but if the noun is "DEF" or "EDAFAH" then the noun must receive Damma "ُ" as Rule (2) shows:

Rule (2)

((Stem [Previous] = ">am~A/CONJ" (or) Stem [Previous] = "lawolA/CONJ") & NP/Tag [Cur-

rent] %“NOUN”)
(NP/Definiteness [Current] = “INDEF”)
{ [Current]: @ Case Ending/Syntactic Diacritic = “N”
}
(NP/Definiteness [Current] = “DEF” (or) NP/Definiteness [Current] = “EDAFAH”)
{ [Current]: @ Case Ending/Syntactic Diacritic = “u”
}

In Ex. (2), the first condition of Rule (2) is applied and the case ending diacritic of the word “صَدِيقٌ” “SadiyqN” “friend” is Tanween Damma “ٌ” ’N’. In Ex. (3), the second condition of Rule (2) is applied and the case ending diacritic of the word “صَدِيقُ” “Sadiysqu” “friend” is Damma “ُ” “u”.

Ex. (2) لَوْلَا صَدِيقٌ لِي أَخْبَرَنِي بِالْأَمْرِ
lawolA SadiyqN liy >axobaraniy biAlo>amori
Except for a friend told me about this matter

Ex. (3) أَمّا صَدِيقُ طُفُولَتِي فَلَهُ مِنِّي كُلُّ الْحُبِّ
>am˜A Sadiyqu Tufuwlatiy˙ falahu min˜iy kul˜u AloHub˜i
As for my childhood friend, all the love from me

Table 1 shows the number of the extracted linguistic rules for being used for both morphological and syntactic processing levels. For more details about these extracted rules, see (Fashwan, 2016).

Rules Type	Rules No.
Morphological Rules	178
Syntactic Rules	473
Definiteness Rules	46
Total No. of Rules	**697**

Table 1: Extracted Rules Number.

3.1.1 Building the Language Models

To reach the best morphological/diacritized form of the words to be analyzed statistically, there are three processes have been used (sub-section 4.1.3). According to these processes a number of Language Models (LM)s are built using the training data set:

- POS LM: a quad gram LM of Prefixes_Tags_Sufixes sequences used to detect the best POS if there are more than one tag for the word to be analyzed in relation to preceding and succeeding (if found) POS(s). It is used in building four smoothed language models using Good-Turing Discounting, Kneser-Ney Smooting, Witten-Bell Discounting and Katz Back-Off Smoothing (Manning and Schütze, 1999) to select the best technique.

- Word Form, Lemma, Tag and Stem LM: a bi-gram LM of Word_Lemma_Stem_Tag sequences used to detect the best lemma, tag or stem of the word to be disambiguated taking into consideration the word form itself.

- Word Form, Suffix Stem and Tag LM: a bi-gram LM of Word_Suffix_Stem_Tag sequences used to detect the best suffix of the word to be disambiguated in relation to preceding and succeeding (if found) suffixes taking into consideration the word form itself.

4 SHAKKIL Diacritization System

In this system, the diacritization problem will be handled through two levels; morphological processing level (for detecting the internal diacritics) and syntactic processing level (for detecting the case ending diacritics).

The morphological processing level depends on four layers. The first three layers are similar to BASMA’s (Alansary, 2015). The first layer is direct matching between the words that have only one morphological analysis and their diacritized forms, the second is disambiguating the input by depending on contextual morphological rules, and the third is disambiguating the input statistically by using machine learning techniques. However, the three layers in this algorithm are applied sequentially for the whole input, unlike BASMA’s system that applies the layers word by word. In each of these layers, a step towards the morphological diacritization of the input text is performed as figure 1 shows. Moreover, this algorithm makes use of the relations between the words and their contexts, whether the preceding or the succeeding words, but BASMA depends only on the morphological disambiguation of the preceding words. In addition to these three layers, another layer is used; the Out Of Vocabulary (OOV) layer.

The adopted syntactic algorithm is a rule based approach that detects the main constituents of the morphological analysis output and applies the suitable syntactic rules to detect the case ending.

4.1 Morphological Processing Level

4.1.1 Uni-Morphological Form Layer

This layer is only concerned with words that have only one diacritized form as well as one POS tag. For example, the word “الْاِحْتِلَال” “AliHotilAl” “occupation” has only one diacritized form with

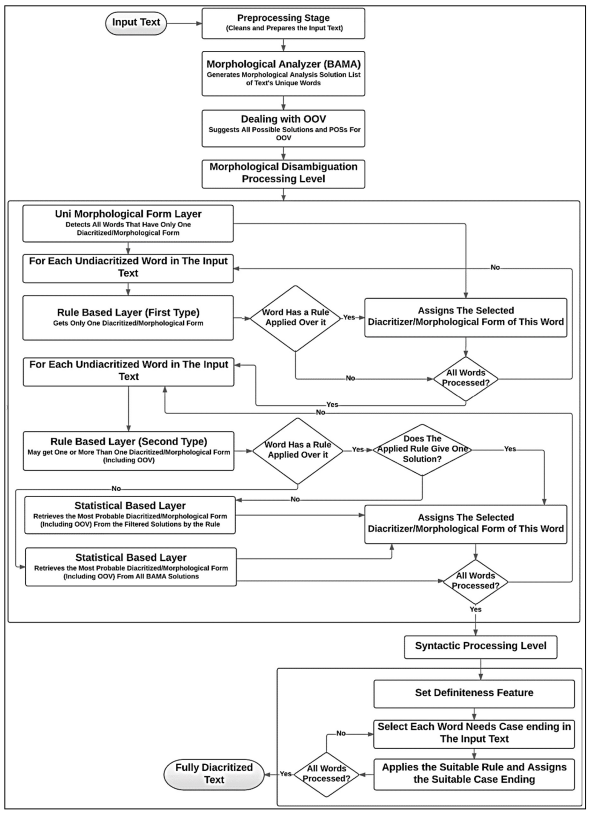

Figure 1: General Design of SHAKKIL System.

only one POS tag; NOUN

In this layer, the adopted morphological disambiguation algorithm does not disambiguate the words according to its word order in the text; however, it begins with matching directly all the words that have uni-morphological/diacritized form and assigns this analysis to the word. This layer is considered as the key for disambiguating other words in the other layers whether they are the preceding or the succeeding words. It may help in: **1)** disambiguating the other words in the rule-based layer if they are governed by a rule that provides one morphological/diacritized form, **2)** disambiguating the other words in the statistical-based layer or **3)** disambiguating the other words by the help of both the rule-based layer and the statistical-based layer if they are governed by a rule that provides more than one morphological/diacritized form.

4.1.2 Rule Based Morphological Disambiguation Layer

The main goal of implementing the morphological rules is to help in eliminating the wrong solution and making the searching problem easier while selecting the best POS or a complete morphological analysis of the non-disambiguated words and hence detecting the internal diacritization. The extracted and implemented rules in this layer are of two types:

1. Rules always help in providing only one morphological/diacritized form for some non-disambiguated words without the need to use the statistical-based layer and word text orders. Most of the rules of this type are concerned with the relation between some non-disambiguated word forms and the preceding or succeeding words.

2. Rules may provide one or more than one solution depending on the word form solutions' variation. If the rule provides only one analysis solution, then this analysis is assigned to the word directly and there will be no need for applying the statistical-based layer. However, if the applied rule provides more than one solution (morphological/diacritized form) the statistical-based layer will be applied in order to get the best solution, depending on the solutions provided by the rules. In the case of having more than one solution for a certain word after applying the rule, following the text word order is a must. The system will depend on a window of -/+ 3 analyzed words around this word to obtain the best solution through the statistical-based layer, as sub-section 4.1.3 shows. The disambiguated words in this layer by rules only may help the statistical-based layer in disambiguating the succeeding word, if it is not analyzed yet. In addition, the disambiguated words through the statistical-based layer may be governed by a rule that helps in disambiguating the succeeding word if it is not analyzed yet.

4.1.3 Statistical Based Morphological Disambiguation Layer

As mentioned before, to reach the best morphological/diacritized form of the words to be analyzed statistically, there are three processes have been used. It must be noted that these processes are not used in all cases while disambiguating the word to be analyzed.

The first one is for getting the best POS score from analysis solutions of the word to be analyzed in relation to the preceding and succeeding (if found) POS depending on some smoothing techniques. The second one is for getting the best stem, tag or lemma score with relation to the word form itself. It is used in two cases:

1) When the POS of the word's analyses are the same. **2)** When the POS model detects best POS and it is found that this POS has more than one lemma or diacritized form.

The third one is for getting the best suffix score

from analysis solutions for the word to be analyzed. It is used when the word to be disambiguated has more than one suffix. The top scoring solution of the word is then selected.

4.1.4 Dealing with OOV Words

For predicting the OOV words, a prior step is taken; preprocessing the stems of the training data. The stems of the training data are used to get a list of unique 4307 diacritized patterns with their templates and frequencies. The patterns are prepared by converting the consonants in the stem to placeholder while keeping the vowels, hamazat hamazat (“أ” “>”, “إ” “<”, “ؤ” “&”, “ئ” “}”, ... etc.) and weak letters “حروف العلة” “واي” “wAy”. In addition, POS of patterns are taken into consideration as figure 2 shows.

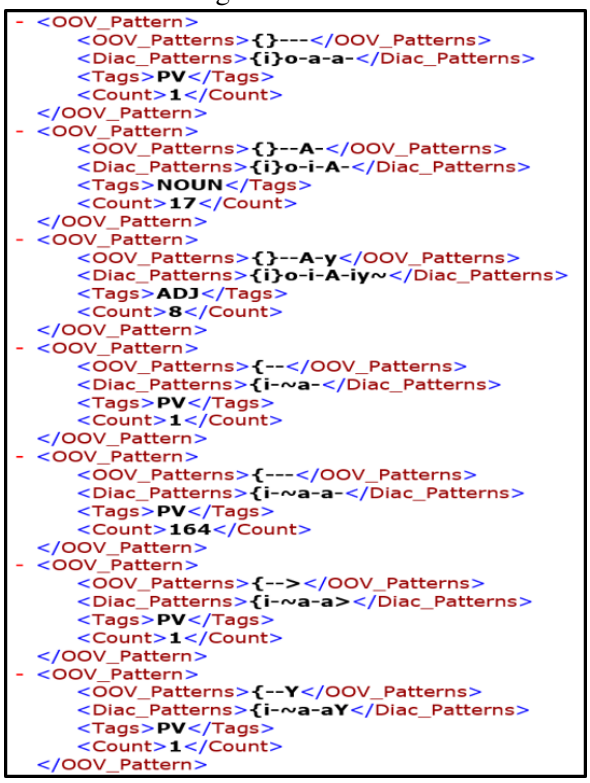

Figure 2: Patterns List with their Diacritized Patterns and Tags.

The POS helps, in some cases, in limiting the scope of the search of the matched pattern, where, for example, if the OOV word has been detected as having “ال” “Al” at the beginning of it, this means the system should search for the detected pattern in the patterns of nouns or adjectives.

While detecting the input text analysis solutions, each word is checked by the system to determine whether it has analyses solutions from BAMA or it is OOV. When the word form is checked as OOV, the system switches to the OOV module. In this module, the system tries to get all word’s possible morphological constituents (a combination of prefixes, stem and suffixes). Then, it uses the list of detected stems and gets their counterpart diacritized patterns.

The selected pattern is used to retrieve the suitable diacritic for the stem. Moreover, the system chooses the POS tag of the diacritized pattern and assign it to the diacritized stem where each selected solution is added to text’s solutions. While working in the morphological disambiguation processing level, if the OOV word has more than one matched POS tag, the system detects the best one depending on step one and two of subsection (4.1.3).

It must be noted that there are no out of vocabulary (OOV) words in MASAR data since they are analyzed manually as if they are analyzed by BAMA and then added in BAMA’s dictionaries so that they would be analyzed correctly the next time they are used.

4.2 Syntactic Processing Level

The task of the syntactic processing level is to predict the syntactic case of a sequence of morphologically diacritized words given their POS tags, definiteness, stem pattern and/or transitivity and hence assigning the suitable case ending diacritics. Some limitations violate the rules for setting the case ending of syntactic diacritic, since the rules are limited to use a window of -/+3 words before the focused word.

Before diacritizing the word syntactically, its POS tag is checked first. Using the POS tag of the word, it is decided how the syntactic diacritization of this word should be handled. As mentioned before, the extracted rules in this level simulates one of the language processing approaches that computes a basic analysis of sentence structure rather than attempting full syntactic analysis; shallow syntactic parsing. It is an analysis of a sentence which identifies the constituents (noun groups, verb groups, prepositional groups adjectival groups, adverbial groups), but does not specify their internal structure, nor their role in the main sentence.

The extracted rules for detecting the imperfect verb case ending, the case ending of noun, the case ending of adjectives, the case ending of adverbs and the case ending of some proper nouns (sub-section 3.1) have been implemented in the current proposed system, taking into consideration the phrases in which each of the previous cate-

gories occur.

5 Evaluation

A blind data set that is not part of the development of the system is then used to evaluate the system versus the gold annotated data. Two error rates are calculated: diacritic error rate **(DER)** which indicates how many letters have been incorrectly restored with their diacritics, and word error rate **(WER)** which indicates how many words have at least one diacritic error. Table 2 shows the total error rate of the system as a whole and WER and DER for each layer in morphological processing level and the percentage of words that has been detected by the help of each layer using different machine learning techniques.

The comparison between the different smoothing techniques shows that Witten-Bell discounting and Kneser-Ney smoothing techniques results are close; however, Kneser-Ney smoothing technique is the best one in detecting the best morphological analysis (internal diacritic) and case ending diacritics in the current proposed system. Consequently, the Kneser-Ney smoothing technique is used while comparing the results of the current systems with other state of the art systems.

System	Diacritized Words%	Good-Turing & Katz Back-off		Witten-Bell		Kneser-Ney	
		DER	WER	DER	WER	DER	WER
1st Layer	55.86%	0.01%	0.02%	0.01%	0.02%	0.01%	0.02%
2nd Layer	9.53%	0.06%	0.15%	0.06%	0.15%	0.06%	0.15%
3rd Layer	28.75%	1.59%	3.85%	1.26%	3.09%	1.20%	2.86%
2nd&3rd Layers	5.86%	0.53%	1.32%	0.50%	1.37%	0.61%	1.53%
Morphological Level	**100%**	**2.19%**	**5.34%**	**1.83%**	**4.63%**	**1.88%**	**4.56%**
Case Ending	—	—	**10.25%**	—	**9.93%**	—	**9.36%**
Overall Results	—	—	**15.59%**	—	**14.56%**	—	**13.92%**

Table 2: System Evaluation Results.

5.1 Error Analysis

When checking the results, we find that the first two layers of the proposed system have the lowest WER and DER, then, the third layer. The statistical based layer only gives the highest WER and DER. In what follows, some error analysis in each layer is reviewed:

- In the uni-morphological layer, it has been found that the error rate is a result of some possessive nouns or present verbs that have affixed possessive. In such words, the case ending diacritic is assigned within the word, not at its end. For example, the word "يُمَارِسهَا" "yumArishA" "he + practice/pursue/exercise", has three moods with three different cases ("ُ" "u", "َ" "a", " ٥" "o") within the word form. When the system fails to detect the suitable case ending diacritic according to the context, the blind testing process counts this wrong diacritic as a morphological (internal error) not a syntactic error. If the testing process has been done for the internal diacritics and syntactic case separately, the results are expected to be enhanced.

- In the rule-based layer, the errors in this layer happen for some reasons; the first is when the word that is governed by a rule that makes the succeeding or preceding word to be diacritized wrongly. In EX. (4), the word "أَن" ">n" is diacritized wrongly as "أَنَّ" ">an˜a" not "أَنْ" ">ano" according to this context. This leads to diacritize the word "ظَلَّ" "Zal˜a" as "ظِلَّ" "Zil˜a" affected by the rule mentioning that only nouns occurs after "أَنَّ". The same problem appeared in the uni-morphological form layer appears in this layer. In addition, when the applied rule gives more than one available solution and the statistical based layer is used to select the best solution, the statistical based layer may choose a wrong one.

Ex. (4) بَعْدَ أَنَّ ظِلَّ قُرُونًا عَدِيدَةً

baEoda >an˜a Zil˜a quruwnAF EadiydapF
After that shadow many centuries

- In the statistical-based layer, although it can predict the correct diacritized form in most of cases even if the same word appears in the same sentence with two different POS and diacritics, it cannot, in other cases, predict the suitable diacritic form. In Ex. (5) the same word "طالب" "TAlb" have been diacritized with two different diacritics according to the context, where the first word is a verb while the second one is a noun.

Ex. (5) طَالَبَ وَزِير التَّرِبِيَة وَالتَّعْلِيم بِالْاِهْتِمَام بِكُلّ طَالِب عَلَى حَدّ سَوَاء

TAlaba waziyr Alt arobiyap waAlt aEoliym biAl{ihotimAm bikul TAlib EalaY Had sawA'
The Minister of Education demanded with the interest for each student alike

In some other cases, the statistical-based layer can predict the correct POS but it fails in detecting the best lemma that helps to differentiate among the different word forms diacritization. In Ex. (6), the system fails to detect the correct diacritized form of the word "المحافظة" "AlmhAfZp" where it should be "الْمُحَافِظَة" "AlmuHAfiZap" "the + governess".

Ex. (6) وَقَامَتْ الْمُحَافَظَة بِإِصْدَار قَرَار بِتَعْيِين نَائِب لَهَا

wa qAmato AlomuHAfaZap bi<iSodAr qarAr bitaEoyiyn nA}ib lahA

The issuance of the province's decision to appoint a deputy for her

Moreover, the statistical based layer may fail to detect both correct POS and the correct diacritized form of the word to be analyzed. In Ex. (7), the input word "علي" "Ely" wrongly diacritized with wrong POS where the system edited it to "عَلَى" "EalaY/PREP" while it should be in this sentence as "عَلِيّ" "Ealiy˜/NOUN_PROP". This wrong POS and diacritization leads to have wrong case ending for both this word and the succeeding one.

Ex. (7) عَلَى طِفْلٍ صَغِيرٍ يَبْلُغُ مِنْ الْعُمْرِ عَشَرَةَ أَعْوَامٍ

EalaY TifolK SagiyrK yabolug mino AloEumori Ea$arapa >aEowAmK

A small child at the age of ten years

The problem of detecting the proper nouns is similar to the previous problem where there are a lot of words in Arabic that may be used as proper nouns and nouns or adjectives. Predicting these words wrongly leads to have wrong case ending for both the word and the succeeding one.

- In the syntactic processing level, Assigning a wrong POS, transitivity or definiteness to a word, may lead to wrong syntactic case. In Ex. (8), the word "هدف" "hadaf" "goal/target" has assigned definiteness as "EDAFAH" as a result of considering "وحيد" "waHiyd" "alone" as "NOUN_PROP". This problem leads to set the case ending as ' ' "i" not ' ' "K".

Ex. (8) بَعْدَ عَدِمِ النَّجَاحِ فِي إِحْرَازِ هَدَفِ وَحِيدْ

baEoda Eadami Aln˜ajaHi fiy <iHorAzi Hadafi waHiydo

After the lack of success in achieving Waheed's goal

5.2 Comparing the System with Other State of the Art Systems

In order to have an objective evaluation of the system, the same testing data (LDC's Arabic Treebank) that was used in the other systems was used to compare the results. It is a part of Arabic Tree Bank part 3 (ATB3) from "An-Nahar" Lebanese News Agency. It consists of 91 articles (about 52.000 word) covering the period from October 15, 2002 to December 15, 2002. In the testing process, 51.63% of the words are diacritized in the first layer, 5.56% of the words are diacritized by rule-based layer only, 8.26% of the words are diacritized by both rule-based and statistical-based layers, 32.99% of the words are diacritized by statistical-based layer only, and finally 1.56% of the word are diacritized in OOV layer. In OOV layer, the system could predict the words with WER of 11.2% and DER of 6.7%. Table 4 summarizes the results of the current proposed system

	Total WER		Ignoring Last	
Systems	**WER**	**DER**	**WER**	**DER**
(Zitouni et al., 2006)	17.30%	5.10%	7.90%	2.50%
(Habash et al., 2009)	13.60%	NA	5.20%	NA
(Rashwan et al., 2015)	12.90%	NA	NA	NA
(Abandah et al., 2015)	11.68%	NA	3.54%	1.28%
(Metwally et al., 2016)	13.70%	NA	4.3%	NA
(Chennoufi and Mazroui, 2016b)	NA	NA	1.86%	0.71%
Current System	**14.78%**	**4.11%**	**4.81%**	**1.93%**

Table 3: Summary of the Comparison between the State-Of-The-Art Systems.

The comparison indicates that (Abandah et al., 2015), (Rashwan et al., 2015) and (Habash et al., 2009) outperform the current system's results. However, the results are still close to (Metwally et al., 2016).

6 Conclusion

In this work, we depend on Arabic morphological rules as well as different machine learning techniques for detecting the morphological diacritics (internal diacritics). In addition, we adopt a rule-based approach that depends on POS morphological sequences, definiteness classifier, stem pattern and verb transitivity for detecting the case ending diacritics. Evaluation of the proposed system is made in comparison with other best state of the art systems. The best WER of the morphological diacritization achieved by the system is 4.81% and the best syntactic diacritization achieved is 9.97% compared to the best-published results. Since this work is in progress, these results are expected to be enhanced by extracting more Arabic linguistic rules (morphological and syntactic), adding more semantic features, using different machine learning techniques for morphological and syntactic processing levels and implementing the improvements by working on larger amounts of data. For enhancing the OOV results more patterns with more features need to be handled.

References

Gheith A. Abandah, Alex Graves, Balkees Al-Shagoor, Alaa Arabiyat, Fuad Jamour, and Majid Al-Taee. 2015. Automatic diacritization of arabic text using recurrent neural networks. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(2):183–197.

Mohamed Attia Mohamed Elaraby Ahmed. 2000. A large-scale computational processor of the arabic morphology, and applications. Master's thesis, Faculty of Engineering, Cairo University Giza, Egypt.

Sameh Alansary. 2015. Basma: Bibalex standard arabic morphological analyzer. In *15th International Conference on Language Engineering*. The Egyptian Society of Language Engineering (ESOLE).

Sameh Alansary. 2016a. Alserag: An automatic diacritization system for arabic. In *16th International Conference on Language Engineering*. The Egyptian Society of Language Engineering (ESOLE).

Sameh Alansary. 2016b. Masar: A morphologically annotated gold standard arabic resource. In *16th International Conference on Language Engineering*. The Egyptian Society of Language Engineering (ESOLE).

Tim Buckwalter. 2004. Buckwalter arabic morphological analyzer version 2.0. linguistic data consortium, university of pennsylvania, 2002. ldc cat alog no.: Ldc2004l02. Technical report, ISBN 1-58563-324-0.

Amine Chennoufi and Azzeddine Mazroui. 2016a. Impact of morphological analysis and a large training corpus on the performances of arabic diacritization. *International Journal of Speech Technology*, 19(2):269–280.

Amine Chennoufi and Azzeddine Mazroui. 2016b. Morphological, syntactic and diacritics rules for automatic diacritization of arabic sentences. *Journal of King Saud University-Computer and Information Sciences*.

Mona Diab, Mahmoud Ghoneim, and Nizar Habash. 2007. Arabic diacritization in the context of statistical machine translation. In *Proceedings of MT-Summit*.

Amany Fashwan and Sameh Alansary. 2016. A rule based method for adding case ending diacritics for modern standard arabic texts. In *16th International Conference on Language Engineering*. The Egyptian Society of Language Engineering (ESOLE).

Amany Fashwan. 2016. Automatic diacritization of modern standard arabic texts: A corpus based approach. Master's thesis, Faculty of Arts, University of Alexanderia, Egypt.

Nizar Habash and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 573–580. Association for Computational Linguistics.

Nizar Habash and Owen Rambow. 2007a. Arabic diacritization through full morphological tagging. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 53–56. Association for Computational Linguistics.

Nizar Habash, Ryan Gabbard, Owen Rambow, Seth Kulick, and Mitchell P. Marcus. 2007b. Determining case in arabic: Learning complex linguistic behavior requires complex linguistic features. In *EMNLP-CoNLL*, pages 1084–1092.

Nizar Habash, Owen Rambow, and Ryan Roth. 2009. 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*, volume 41, page 62.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*, volume 999. MIT Press.

Aya S. Metwally, Mohsen A. Rashwan, and Amir F. Atiya. 2016. A multi-layered approach for arabic text diacritization. In *Cloud Computing and Big Data Analysis (ICCCBDA), 2016 IEEE International Conference on*, pages 389–393. IEEE.

Mohsen A. Rashwan, Mohammad Al-Badrashiny, Mohamed Attia, and Sherif Abdou. 2009. A hybrid system for automatic arabic diacritization. In *The 2nd International Conference on Arabic Language Resources and Tools*.

Mohsen A. Rashwan, Al-Badrashiny Mohamad, Mohamed Attia, Sherif Abdou, and Ahmed Rafea. 2011. A stochastic arabic diacritizer based on a hybrid of factorized and unfactorized textual features. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1):166–175.

Mohsen A. Rashwan, Ahmad A. Al Sallab, Hazem M. Raafat, and Ahmed Rafea. 2014. Automatic arabic diacritics restoration based on deep nets. *ANLP 2014*, page 65.

Mohsen A. Rashwan, Ahmad A. Al Sallab, Hazem M. Raafat, and Ahmed Rafea. 2015. Deep learning framework with confused sub-set resolution architecture for automatic arabic diacritization. *IEEE Transactions on Audio, Speech, and Language Processing*, 23(3):505–516.

Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 2008. 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, pages 117–120. Association for Computational Linguistics.

Khaled Shaalan, Hitham M. Abo Bakr, and Ibrahim Ziedan. 2008. A statistical method for adding case ending diacritics for arabic text. In *Proceedings of Language Engineering Conference*, pages 225–234.

Khaled Shaalan, Hitham M. Abo Bakr, and Ibrahim Ziedan. 2009. A hybrid approach for building arabic diacritizer. In *Proceedings of the EACL 2009 workshop on computational approaches to semitic languages*, pages 27–35. Association for Computational Linguistics.

Anas Shahrour, Salam Khalifa, and Nizar Habash. 2015. Improving arabic diacritization through syntactic analysis. In *EMNLP*, pages 1309–1315.

Imed Zitouni, Jeffrey S. Sorensen, and Ruhi Sarikaya. 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 Association for Computational Linguistics*, pages 577–584. Association for Computational Linguistics.