Alserag: An Automatic Diacritization System for Arabic

Sameh Alansary[1,2(✉)]

[1] Bibliotheca Alexandrina, Alexandria, Egypt
sameh.alansary@bibalex.org
[2] Phonetics and Linguistics Department, Faculty of Arts, Alexandria University, Alexandria, Egypt

Abstract. Diacritization of written text has a significant impact on Arabic NLP applications. We present an approach to Arabic automatic diacritization that integrates morphological analysis with shallow syntactic analysis. The developed system (Alserag) is a rule based system. The system depends on three modules in order to provide fully diacritized Arabic words namely, morphological analysis module, syntactic analysis module and morph-phonological processing module. The results of the system were evaluated for accuracy against the reference using two metrics; diacritization error rate (DER) and word error rate (WER). The DER measurement was 8.68 % while WER measurement was 18.63 %. The system is benchmarked against three known diacritization systems; Harakat, Mishkal, and Aldoaly.

1 Introduction

Diacritizing Arabic written text is crucial for many NLP tasks, translation can be enumerated among a longer list of applications that vitally benefit from automatic diacritization [1–3]. Arabic diacritics are superscript and subscript diacritical marks (vocalization or voweling), defined as the full or partial representation of short vowels, shadda (gemination), nunation, and hamza [4]. Diacritization helps the reader in disambiguating the text or simply in articulating it correctly. As Arabic is a language where the intended pronunciation of a written word cannot be completely determined by its standard orthographic representation; it rather depends on a set of special diacritics. The absence of these diacritics in Arabic text increases lexical and morphological ambiguity, because one written form can have several vocalizations, each vocalization may have different meaning(s) [5, 6]. However, these diacritics are generally left out in most genres of written Arabic which results in widespread ambiguities in vocalizations and meaning.

Although native speakers are able to disambiguate the intended meaning and pronunciation from the surrounding context with minimal difficulty, it is not the case with automatic processing of Arabic which is often hampered by the lack of diacritics. Several applications can radically benefit from automatic diacritization, such as Text-to-speech (TTS), Part-Of-Speech (POS) tagging, Word Sense Disambiguation (WSD), and Machine Translation [6].

A.E. Hassanien et al. (eds.), *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016*, Advances in Intelligent Systems and Computing 533, DOI 10.1007/978-3-319-48308-5_18

Much work has been done on Arabic diacritization. The actually implemented systems can be divided into two categories [7]: Systems implemented by individuals as part of their academic activities and systems implemented by commercial organizations for realizing market applications. One of the advantages of the first type is that they present some good ideas as well as some formalization. The weak point about these systems is that they are mostly partial demo systems [7]. The following are examples of these systems: Vergyri and [8–11]. For the second category, the most representative commercial Arabic morphological processors are Sakhr, Xerox, and RDI [7]. There are also other available systems as Mishkal Arabic diacritizer[1], and Harakat Arabic diacritizer[2]; they are free Arabic diacritizers which are available online. Finally, on March Google has launched an innovative new Google Labs Arabic tool called Tashkeel, a tool that adds the missing diacritics to Arabic text. Unfortunately, the tool is not available now.

There is another system [12] that has integrated three different proposed techniques, each of which has its own strengths and weaknesses. They are lexicon retrieval, diacritized bigram and SVM statistical-based diacritizer. Most of the previous approaches cited above utilize different sequence modeling techniques that use varying degrees of knowledge from shallow letter and word forms to deeper morphological information. None of the previous systems make use of syntax with the exception of [13] which have integrated syntactic analysis; however, they are not rule based. In this paper, Alserag; an Arabic diacritizer, is proposed. Alserag is based on different steps: retrieval of unambiguous lexicon entries, disambiguating between the different stored possible solutions of the words to realize their internal diacritization through the morphological analysis step (the system tokenizes a text and provides a solution for each token and restore the appropriate internal diacritics from the dictionary), the syntactic processing step that is responsible for the case ending detection is based on shallow parsing and finally the morpho-phonological step that is developed to fulfill the requirements of vowel harmony and assimilation. Section 2 demonstrates the system architecture. Section 3 explains the different applied modules to fully diacritize texts. Section 4 evaluates the output and discusses the results and benchmarking process. Finally, Sect. 5 concludes the paper.

2 System Architecture

In this system, a rule-based approach was adopted. In this section, the different processes that took place in order to convert a plain text into a fully diacritized text will be described. Figure 1 presents the system's overall architecture, where the diacritization is achieved through 7 main phases: (i) Preprocessing which is responsible for auto-correcting the raw text and segmenting the Arabic text into sentences. (ii) Tokenization which is the process of splitting the natural language input into lexical items. (iii) Disambiguation which is a process of choosing the right internal diacritization for

[1] http://tahadz.com/mishkal.

[2] http://harakat.ae/.

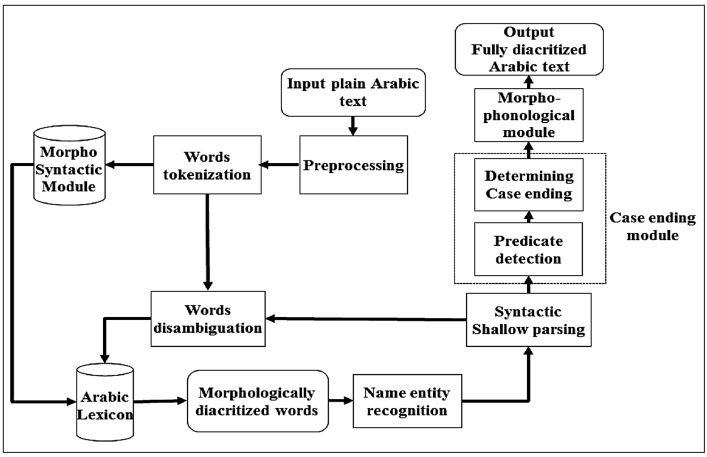

Fig. 1. Architecture of alserag system.

the word from the dictionary. (iv) Name entity recognition (stored in the dictionary and have been obtained from the UNLarium[3] [14]). (v) Syntactic shallow parsing which is an analysis of a sentence by identifying its constituents (NPs, JPs—etc.). (vi) Case ending module which is responsible for predicting the arguments of the predicate and assigning the diacritical marks that are attached to the ends of words to indicate their grammatical function. (vii) Morph-phonological module which is a series of rules that focus on the sound changes that take place in morphemes (minimal meaningful units) when they are combined to form words.

There are two engines that are used in Alserag, the first is Interactive ANalyzer (IAN)[4] which is used in the analysis process, it includes a grammar for natural language analysis. The syntactic processing is done automatically through the natural language analysis grammar, the second is dEep-to-sUrface natural language GENErator engine (EUGENE)[5] which is used in the generation process, It receives the analyzed input and provides a diacritized output without any human intervention, for more details see [15].

3 Development of the System Resources

Alserag depends on two resources; the Arabic diacritized dictionary and a set of linguistics rules. Each one will be described in details in the following subsections.

[3] http://www.unlweb.net/unlarium/index.php?unlarium=dictionary.

[4] it is a web application developed in Java and available at http://dev.undlfoundation.org/index.jsp.

[5] it is a web application developed in Java and available at http://dev.undlfoundation.org/index.

3.1 Dictionary

The Arabic diacritized dictionary is a dictionary where Arabic natural language words exist with their diacritics, along with the corresponding linguistic features which describe the Arabic word morphologically, syntactically and semantically. For example, the Arabic word "كتب" 'ktb' 'write' its diacritics "كَتَبَ" 'kataba' and a list of linguistic features such as part of speech, tense, transitivity, person, gender, number, etc. are included in the dictionary. The words in the Arabic diacritized dictionary are extracted from the Arabic dictionary in UNLarium. The process of diacritizing the entries mainly depends on two resources: BAMA and Alkhalil Arabic Morphological Analyzer.

The diacritizing process begins with Buckwalter's analysis. Some words have only one solution, other words have more than one solution and some words couldn't be analyzed in Buckwalter. These are analyzed by Alkhalil which also suggests different solutions to some words. Then, these words are verified manually to select their correct diacritization. Not all of the Arabic diacritized dictionary entries are fully diacritized. Nouns, adjectives, subjunctive and indicative verbs are partially diacritized, since their case endings depend on the context. By default, a present tense verb is marked by a short /o/ (الضمة), in this case it is called indicative (المرفوع المضارع). However, if a present verb is preceded by certain particles, the verb will be marked by a short /a/ (الفتحة), and if the verb ends by one of the three suffixes (ين ون، ان،), the final (ن) will be deleted, in this case it is called subjunctive (المضارع المنصوب). Nevertheless, imperative verb forms and past verb forms are fully diacritized, because their case endings are not affected by the context. Some enhancements have to be made in the Buckwalter solutions. For example, some solutions have a missing vocalization "ً" before "ا" as in "عالم" 'EAlim' 'scientist', "مكتبات" 'makotabAt' 'libraries'. So, these missing vocalizations have been added manually.

3.2 The Linguistic Rules

Alserag depends on three modules in order to provide fully diacritized Arabic words namely, morphological analysis module, syntactic analysis module and morph-phonological processing module.

Morphological analysis: is responsible for the morphological analysis of Arabic words and assigning the correct POS and the internal diacritization of words which is achieved through two processes; tokenization process and disambiguation process. However, before the tokenization process began, a preprocessing phase should take place over the string stream to fix the most common spelling mistakes, if needed. First, the tokenization algorithm is based on the entries of the dictionary. It starts from left to right trying to match the longest possible string with dictionary entries. The process starts with preventing joined lexical items. Then, it identifies the different suffixes and prefixes that could be attached to each lexical category. Disambiguation rules apply over the natural language list structure to constrain word selection and to correctly disambiguate the POS. They have the following format:(node 1) (node 2) (…)(node n) = P; Where (node 1), (node 2) and (node n) are nodes, and P is an integer expressing the

possibility of occurrence. The engine is able to tokenize automatically some words correctly based on the dictionary and assign the correct POS to words. On the other hand, the larger the number of entries in the dictionary, the more the ambiguity during tokenization increases. For example, the sequence "بالفيضانات" 'by the floods' would be automatically segmented as "بال" 'bAl' (worn) + "فيضانات" 'fayaDAnAt' (floods), according to the longest match algorithm, given the fact that the dictionary includes [ART ال] 'Al' (the), [ADJ بال] 'bAl' (worn), [ب] 'bi' (by) and [N فيضانات] 'fayaDAnAt' (floods).

Rule in (1) states that adjectives can only be followed by a blank space (BLK), suffix (SFX) or to occur at the end of the sentence (STAIL), where (^) means not. So, [ب] + [ال] + [فيضانات] will be chosen as the appropriate combination.

1. (ADJ)(^SFX,^BLK,^STAIL) = 0;

If words have spelling mistakes or undergo morpho-syntactic changes, rules will investigate the morphological form of those words. For example, the most common mistake in Arabic writing is /Hamza/in the initial position as in "أنتزع" ">anotaziE". Rules will investigate the morphological pattern of the wrongly spelled word by the regular expression techniques. For example, if a five-letters word begins with the sequence "/(آ|إ|أ).ت../", the wrong written /Hamza/("أ", "إ" or "آ") will be modified to the correct "ا", according to the Arabic grammar, by the rule in (2), as in the pattern "افتعل" //ʔiftaʕala//. Then the correct diacritized form will be retrieved from the dictionary "اِنْتَزَعَ" ">inotazaE".

2. ([/..ت.(آ|إ|أ)/],^Hamza_modified)(%y,PUT): = ("1 >"\, Hamza_modified)(%y);

Second, disambiguation is concerned with preventing the wrong automatic lexical choices and obtaining the right internally diacritized words. Some linguistic indicators can help in solving the lexical ambiguity which are morphological and adjacency indicators.

Morphological indicators: affixation has an important role as the first level of part of speech disambiguation, as prefixes and suffixes are the smallest processing units rules can begin with. Prefixes can help as indicators in determining correct lexical choices. For example, in the word "لدفع" 'liDafoE', the noun "دَفْع" 'DafoE' (push) is chosen instead of the verb (V) "دَفَعَ" 'DafaE' (to push), since it is preceded by the preposition "ل" 'li' (to) by the rule in (3).

3. (P)(V) = 0;

Adjacency indicators: After disambiguating the POS on the word level, the role of the adjacent word will take its effect as the second level of disambiguation. In this level, disambiguating the part of speech could be controlled by many qualifiers.

Number and Gender qualifiers: as in "وهم يسمون" (and they call). According to the longest match algorithm, the engine will automatically choose the noun "وَهْم" 'wahom' (illusion). But, because it is followed by a plural verb "يُسَمُّونَ" 'yusam ~ uwna' (they call) and subject and verb should agree in number and gender in Arabic, this tokenization will be rejected and will be retokenized as "وَ" 'wa' (and) + "هُمْ" 'hum' (they) by the rule in (4).

4. (SHEAD) ([وهم],%x) (BLK) (V, ^NUM = %x) = 0;

Functional word qualifier: Particles could be used as indicators for disambiguating the part of speech, as there are particles for verbs and others for nouns. For example, the particle "أي" '>ay ~' (any) is a noun particle. Therefore, in the combination "أي شرط" (any condition), rule in (5) will reject the word "شرط" if it is chosen as a verb "شَرَّط" '$ar ~ aTa' (slit), since it is preceded by the particle (PTC) "أي" '>ay ~' (any). Then, it will backtrack it to the noun "شَرْط" '$aroT' (condition).

5. (PTC, "أيّ ") (BLK) (V) = 0;

The co-occurrence of specific words with words with specific semantic features is used as an indicator. The word "تقلع" has different internal diacritizations that depend on the different meanings, such as "تُقْلِع" 'tuqoliE' 'take off' with the semantic feature motion (MOT) and "تَقْلَعَ" 'taqolaE' 'strip' which has the semantic feature contact (CTC). If the verb "تقلع" 'taqolaE' 'strip' is followed by a noun such as "طائرة" 'TA} irap' 'airplane' which has the semantic feature artifact (ARF) (Nouns denoting man-made objects). Rule in (6) will reject "تَقْلَعَ" 'strip' "تُقْلِع" 'tuqoliE' 'take off'.

6. (V, SEM = CTC) (BLK)(ART)(N, SEM = ARF) = 0;

Syntactic analysis: Shallow parsing is considered necessary for case ending assignment. Transformation rules have been developed to group words under the different phrasal categories. The rules follow the very general formalism α: = β; where the left side α is a condition statement, and the right side β is an action to be performed over α. Phrasal grouping is necessary for identifying the sentence components and linking them by predicate. Then, the different functions of the sentence components can be identified and assigned the suitable case ending. This process will be illustrated in the following. Rules were developed to syntactically mark the phrasal units of the partially diacritized sentence in (7).

7. وَلِذَلِكَ لَمْ تَبْعَثْ اَلدِّرَاسَة اَلْمَدْرَسِيَّة لِتَارِيخ اَلْفَرَاعِنَة أَيّ شَوْق بَيْن اَلطَّلَبَة أَو اَلْخِرِّيجِين لِلْاِسْتِزَادَة

'wali*`lika lam taboEavo Ald ~ irAsap Almadorasiy ~ ap litAriyx AlfarAEinap ayo $awoq bay ~ na AlT ~ alobap > aw Alxir ~ iyjiyna lilAisotizAdap'

'Therefore, the school study for the Pharaohs history did not provoke any urge between the students or the graduates to increase.'

In sentence (7), different NPs structures are established. The first is established by rule in (8a); it combines the definite article "ال" 'the' and the following noun to project a noun phrase (NP) "اَلدِّرَاسَة" 'Ald ~ irAsap' 'the school study', "اَلْفَرَاعِنَة" 'Alfar-AEinap' 'the Pharaohs', "الطَّلَبَة" 'AlT ~ alobap' 'the students' and "اَلْخِرِّيجِين" 'Alxir ~ iyjiyna" the graduates'. The second NP structure is formed by rule in (8b); it combines the indefinite noun "تَارِيخ" 'tAriyx' 'history' and the NP "اَلْفَرَاعِنَة" 'the Pharaohs', the composed NP "تَارِيخ اَلْفَرَاعِنَة" is automatically assigned with the features of its head "تَارِيخ" such as gender, number, animacy and semantic class that are necessary to describe the NP. The third NP structure consists of two coordinated elements and a conjunction; "الطَّلَبَة أو اَلخِرِّيجِين" 'AlT ~ alobap > aw Alxir ~ iyjiyna' 'the students or the graduates' by rule in (8c). Moreover, adverbial phrase (AP) consists of the

adverb "بين" 'bay ∼ na' 'between' and the coordination NP; "الطَّلَبَة أو الخرِّيجين" 'AlT ∼ alobap > aw Alxir ∼ iyjiyna' is established by rule in (8d). However, the AP "الخريجين بين الطلبة أو" 'between the students or the graduates' is considered as an optional argument in the sentence in (7). Next, prepositional phrases (PPs) will be established; the two previously composed NPs "الِاسْتِزَادَة" '>alisotizAdap' and "تَارِيخ اَلْفَرَاعِنَة" 'tAriyx AlfarAEinap' '' will be combined with the preceding preposition "ل" 'li' (to) to form the prepositional phrases (PPs) "للاستزادة" 'to increase' and "لتاريخ الفراعنة" 'for the Pharaohs history' by rule in (8e).

8. (a) (ART, %a)(%y, N): = ((%a)(%y), NP, ANI = %y, GEN = %y, NUM = %y, SEM = %y);
   (b)(^ART, %a)(%y, N)(NP, %x): = (%a)((%y, np)(%x), NP, ANI = %y, GEN = %y, NUM = %y, SEM = %y);
   (c)(NP, %a)(%y, COO)(NP, %x): = ((%a, np)(%y)(%x), NP, ANI = %a, GEN = %a, NUM = %a);
   (d) (ADV, %a)(NP, %x)(%j): = ((%a)(%x), AP)(%j);
   (e) (%x, P, ^pp)(%n, NP): = ((%x, pp)(%n), PP);

Different syntactic functions of the predicate arguments should be identified in order to assign the case ending after the shallow parsing stage. In (7), the arguments of the verb should be identified which will be illustrated in the following.

Verbs and their Arguments Diacritization: The sentence in (7) contains a verb, it is considered as the core of the sentence, since it is the verb that answers the three most important elements of any message - the what, who and when. In terms of the importance of the verb in the diacritization process, verb decides the case ending of the sentence elements. In sentence in (7), the verb "تَبْعَث" 'taboEavo' 'provoke' is a transitive verb that requires two arguments, one to function as a subject "الدِّرَاسَة" 'Ald ∼ irAsap' 'study' and another as an object "أيّ" 'ayo' 'any'. After identifying the phrasal constructions, grammar rules have been developed to assign the function and the case ending of the composed verb arguments by rule in (9a). The rule states that, if a verb is followed by two noun phrases and there is gender agreement between the verb and the following noun phrase (NP, GEN = %v), this noun phrase will be considered as the subject of the verb (SBJ). The second will be considered as the object (OBJ). Once the functions of the arguments have been determined, the case ending will be assigned to each noun phrase; the nominative case (NOM) will be assigned to the subject and the accusative case (ACC) will be assigned to the object.

9. (a) (V, TSTD, %v)(NP, GEN = %v, ^CAS, %n)(PP, %a)(NP, ^CAS, %n2): = (%v) (SBJ, CAS = NOM, %n)(%a)(OBJ, CAS = ACC, %n2);
   (b) (لم, %a)(%x, PRS, ^MOO): = (%a)(MOO = JUS, %x);

In the rule in (9a), the nominative and the accusative cases have been assigned to the heads of the two composed NPs; the words "الدِّرَاسَة" (CAS = NOM) and "أيّ" (CAS = ACC). Rule in (9b) assigns the mode of the verb "تَبْعَث" as jussive (JUS) "تَبْعَثْ", because it is preceded by a jussive particle, as illustrated in (10).

10. وَلِذَلِكَ لَمْ تَبْعَثْ اَلدِّرَاسَةُ اَلْمَدْرَسِيَّة لِتَارِيخ اَلْفَرَاعِنَة أَيَّ شَوْق بَيْن اَلطَّلَبَة أَو اَلْخِرِّيجِين لِلْاِسْتِزَادَة

The modifiers are diacritized accordingly. The genitives such as "muDAf > ilayhi" 'مضاف إليه' and the constituents after prepositions do not depend on the case ending of the preceding elements. In sentence in (7), genitive case (GNT) is assigned to genitives, as "اَلْفَرَاعِنَة", "شَوْق" and "اَلطَّلَبَة", "اِسْتِزَادَة" and "تَارِيخ".

Adjectives, coordinated elements and nouns in apposition are assigned the same case ending of the preceding element. The adjective "اَلْمَدْرَسِيَّة" 'Almadorasiy ~ ap' is assigned with the same case ending of the preceding noun "اَلدِّرَاسَة" 'Ald ~ irAsap', so it is assigned nominative case "اَلْمَدْرَسِيَّةُ". As for the coordinated elements, as in "اَلْخِرِّيجِين اَلطَّلَبَة أو", the case of the NP "اَلطَّلَبَةِ" which is genitive is assigned to the NP "اَلْخِرِّيجِين". However, in Arabic, masculine plural noun ending with "ين" suffix, does not permit the genitive case marker (kasra), its genitive case is marked by fatha "َ" 'a'. The final diacritization for the sentence in (7) is as in (11).

11. وَلِذَلِكَ لَمْ تَبْعَثْ اَلدِّرَاسَةُ اَلْمَدْرَسِيَّةُ لِتَارِيخِ اَلْفَرَاعِنَةِ أَيَّ شَوْقٍ بَيْنَ اَلطَّلَبَةِ أَو اَلْخِرِّيجِينَ لِلْاِسْتِزَادَةِ

Nominal sentences Diacritization: Nominative case is directly assigned to the topic of the sentence (noun or noun phrase in the beginning of sentences), because it is considered as "مبتدأ" 'mobtadaa'. Since Arabic is a free word-order language, comment may precede topic in nominal sentences such as the sentence in (12).

12. فِي اَلْحَدِيقَة بَيْت 'A house in the garden'

The case of the topic "بَيْت" 'A house' can be detected in the system in the case of the prepositional phrase "فِي اَلْحَدِيقَة" "خبر شبه الجملة" 'in the garden' precedes it by rule (13).

13. (SHEAD, %x)(%c, PP)(NP, ^CAS, %y): = (%x)(%c)(%y, mobtadaa, CAS = NOM);

Rule in (13) states that if a prepositional phrase "فِي اَلْحَدِيقَةِ" comes in the beginning of the sentence is followed by a noun phrase "بَيْت", where (SHEAD) means beginning of the sentence. This noun phrase "بَيْت" is assigned with nominal case (NOM).

However, these nominal phrases cases change if Anna and her sisters precede them. In the example, "إنَّ فِي اَلْحَدِيقَة بَيْتا" '<in ~ a fiy AlHadiyqap bayotAF', the NP "بَيْت" 'bayot' 'house' became accusative.

Morpho-phonological process: Many morpho-phonological alternations occur in Arabic due to the concatenative nature of Arabic morphology, the interaction between morphological and phonological processes is usual. There are two cases where morpho-phonological change is necessary; vowel harmony and assimilation necessity. Vowel harmony takes place in the diacritization process (i.e. phonological). For example, a morpho-phonological rule is necessary for "له" 'lahu' 'for him' that consists of two morphemes "ل" 'li' + "هُ" 'hu', to change the vowel "َ" 'i' in "ل" to "ـَ", 'a' to be more harmonious with the vowel "ُ" 'u' on the suffix "هُ". Moreover, the phonological Arabic system doesn't permit the "moon letters" '(ا-ب-غ-ح-ج-ك-و-خ-ف-ع-ق-ي-م-ه)' to be assimilated with the /l/ of the definite article "ال" 'Al' 'the', as they are not near in the place of articulation, but they can assimilate with the other Arabic alphabets which are

called “sun letters”. When the definite article is followed by a sun letter, the /l/ of the Arabic definite article al- assimilates to the initial consonant of the following noun, resulting in a doubled consonant (phonologically) which is orthographically expressed by putting a shaddah ‘ّ’ on the consonant after /ل/. For example, the word “الصباح” ‘the morning’ before applying the morpho-phonological rule, is diacritized as “الصَبَاح” ‘AlsabAH’. Another rule adds a shaddah before the vowel (“َ”, “ُ” or “ِ”), if the diacritical mark is on a sun letter, it would be diacritized “الصَّبَاح” ‘Als ∼ abAH’.

4 Evaluation and Benchmarking

The corpus has been selected from the International Corpus of Arabic (ICA). The selected corpus size is 400,000 Modern Standard Arabic words; they are divided into 300,000 words as tuning data and 100,000 words as testing data. The selected texts are from different sources; Newspapers, Net Articles and Books representing the following genres; politics: 148,211, miscellaneous: 100,253, child stories: 57,174, economy: 34,930, society: 32,955 and sports: 26,477.

The results were evaluated automatically for accuracy against the reference which is a fully diacritized texts by Arabic linguist using the following two metrics; diacritization error rate (DER) which is the proportion of characters with incorrectly restored diacritics. Word error rate (WER) which is the percentage of incorrectly diacritized white-space delimited words: in order to be counted as incorrect, at least one letter in the word must have a diacritization error.

These two metrics were calculated as: (1) all words are counted excluding numbers and punctuators, (2) each letter in a word is a potential host for a set of diacritics, and (3) all diacritics on a single letter are counted as a single binary (True or False) choice. Moreover, the target letter that is not diacritized is taken into consideration, as the output is compared to the reference.

In addition to calculating DER and WER, the evaluation system calculates internal diacritics and case ending separately. Alserag results were compared with the output of other three known diacritization systems; Harakat, Mishkal, and Aldoaly as they are the only available systems. The outputs of these three systems were evaluated using the same data. Table 1 shows benchmarking of the whole data of Alserag among the other three systems.

According to the results obtained by the benchmarking process, our system scored the least error rate followed by Harakat and Mishkal and finally Aldoaly which scored over 80 % error rate. Future plan is associated with improving parsing phase as it is the main source of problems that raised DER and WER. In addition, it is planned to perform the evaluation and benchmarking of Alserag by using the same dataset of LDC (Arabic Treebank) used by more robust systems such as Sakhr, RDI and Microsoft system in the evaluation process so at least we can compare results published by such systems.

Table 1. Benchmarking of the whole data of alserag among the other three systems.

	Alserag	Harakat	Mishkal	Aldoaly
Int.	5.77%	43.30%	32.53%	80.92%
C.E	14.77%	16.23%	31.15%	89.72%
DER	8.68%	37.63%	32.24%	82.76%
WER	18.63%	43.49%	65.00%	97.87%

5 Conclusion

The paper presents an automatic diacritization system Alserag that is developed based on the rule- based approach which is considered as our contribution to the subject of automatic diacritization. All of the other available systems that were mentioned are statistical based. The results of the system were evaluated against the reference. The DER was 8.68 % while WER measurement was 18.63 %.

References

1. Smr, O.: Yet Another Intro to Arabic NLP (2005). http://ufal.mff.cuni.cz/∼smrz/ANLP/anlp-lecture-notes.pdf
2. Rashwan, M., Abdou, S., Rafea, A.: Stochastic arabic hybrid diacritizer. In: IEEE Transactions on Natural Language Processing and Knowledge Engineering, pp. 1–8 (2009)
3. Attia, M., Rashwan, M.A.A., Al-Badrashiny, M.A.S.A.A.: Fassieh®, a semi-automatic visual interactive tool for morphological, PoS-tags, phonetic, and semantic annotation of arabic text corpora. IEEE Trans. Audio Speech Lang. Process. 17(5), 916–925 (2009)
4. Maamouri, M., Bies, A., Kulick, S.: Diacritization: A Challenge to Arabic Treebank Annotation and Parsing. Linguistic Data Consortium, University of Pennsylvania, USA (2006)
5. Bouamor, H., Zaghouani, W., Diab, M., Obeid, O., Oflazer, K., Ghoneim, M., Hawwari, A.: A pilot study on arabic multi-genre corpus diacritization annotation. In: Proceedings of the Second Workshop on Arabic Natural Language Processing, pp. 80–88. c2014 Association for Computational Linguistics, Beijing, China (2015)
6. EL-Desoky, A., Fayz, M., Samir, D.: A smart dictionary for the arabic full-form words. IJSCE 2(5) (2012). ISSN: 2231-2307
7. Al Badrashiny, M.: Automatic Diacritizer for Arabic Text. A Thesis Submitted to the Faculty of Engineering. Cairo University in Partial Fulfillment of the Requirements for the Degree of master of science in electronics and electrical communication (2009)
8. Vergyri, D., Kirchhoff, K.: Automatic diacritization of arabic for acoustic modeling in speech recognition. In: COLING Workshop, Geneva, Switzerland (2004)
9. Ananthakrishnan, S., Narayanan, S., Bangalore, S.: Automatic diacritization of arabic transcripts for asr. In: Proceedings of ICON-2005, Kanpur, India (2005)
10. Zitouni, I., Sorensen, J.S., Sarikaya. R.: Maximum entropy based restoration of arabic diacritics. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL), Workshop on Computational Approaches to Semitic Languages, Sydney-Australia (2006)

11. Habash, N., Rambow, O.: Arabic diacritization through full morphological tagging. In: Proceedings of the 8th Meeting of the North American Chapter of the Association for Computational Linguistics (ACL), (HLT-NAACL) (2007)
12. Shaalan, K., Abo Bakr, H.M., Ziedan, I.: A hybrid approach for building Arabic diacritizer. In: Semitic 2009 Proceedings of the EACL Workshop on Computational Approaches to Semitic Languages (2009)
13. Shahrour, A., Khalifa, S., Habash, N.: Improving arabic diacritization through syntactic analysis. In: Proceedings of EMNLP, Lisbon (2015)
14. Alansary, S.: MUHIT: A multilingual harmonized dictionary. In: The 9th Edition of the Language Resources and Evaluation Conference, Reykjavik, Iceland, 26–31 May 2014
15. Alansary, S.: A Suite of Tools for Arabic Natural Language Processing: A UNL Approach, the special session on Arabic Natural Language Processing: Algorithms, Resources, Tools, Techniques and Applications, (ICCSPA 2013), Sharjah, UAE (2013)