Automatic diacritization of Tunisian dialect text using Recurrent Neural Network

Abir Masmoudi
MIRACL Laboratory
University of Sfax
masmoudiabir@gmail.com

Mariem Ellouze Khmekhem
MIRACL Laboratory
University of Sfax
Mariem.Ellouze@planet.tn

Lamia Hadrich Belguith
MIRACL Laboratory
University of Sfax
l.belguith@fsegs.rnu.tn

Abstract

The absence of diacritical marks in the Arabic texts generally leads to morphological, syntactic and semantic ambiguities. This can be more blatant when one deals with under-resourced languages, such as the Tunisian dialect, which suffers from unavailability of basic tools and linguistic resources, like sufficient amount of corpora, multilingual dictionaries, morphological and syntactic analyzers. Thus, this language processing faces greater challenges due to the lack of these resources. The automatic diacritization of MSA text is one of the various complex problems that can be solved by deep neural networks today. Since the Tunisian dialect is an under-resourced language of MSA and as there are a lot of resemblance between both languages, we suggest to investigate a recurrent neural network (RNN) for this dialect diacritization problem. This model will be compared to our previous models models CRF and SMT (24) based on the same dialect corpus. We can experimentally show that our model can achieve better outcomes (DER of 10.72%), as compared to the two models CRF (DER of 20.25%) and SMT (DER of 33.15%).

1 Introduction

Modern Standard Arabic (MSA) as well as Arabic dialects are usually written without diacritics(24). It is easy for native readers to infer correct pronunciation from undiacritized words not only from the context but also from their grammatical and lexical knowledge. However, this is not the case for children, new learners and non-native speakers as they dont have a good mastery of rich language derivations. Moreover, the absence of diacritical marks leads to ambiguity that affects the performance of NLP tools and tasks. This may generally bring a considerable awkward ambiguity at the data-processing level for NLP applications. Hence, we can notice that automatic diacritization has been shown to be useful for a variety of NLP applications, such as Automatic Speech Recognition (ASR), Text-to-Speech (TTS), and Statistical Machine Translation (SMT)(24).

In this paper, we present our method to automatic dialectical Arabic diacritization. In fact, both previous experiences and works have shown that the use of Recurrent Neural Network (RNN) could give better results for such an MSA diacritization system as compared to the other approaches, like the Lexical Language Model (16), a hybrid approach combining statistical and rule-based techniques (28). For instance, the authors (1) demonstrated in their study that the RNN gave the least DER, compared to the other MSA diacritization works.

Based on the huge similarity between MSA and Tunisian dialect, we decided to benefit from its advantages by testing the RNN performance in the automatic diacritization of Tunisian Arabic dialects. To the best of our knowledge, it is the first work that investigates the RNN for the diacritization of the Tunisian dialect.

In this respect, we performed the task of restorating diacritical marks without taking into account any previous morphological or contextual analysis. Moreover, we diagnosed different aspects of the proposed model with various training options. The latter include the choice of transcription network (long short-term memory (LSTM) networks, bidirectional LSTM (B-LSTM)) and the impact of RNN sizes. The size of the neural network is a function of the number of hidden layers. Our goal is to choose the most pertinent layers

Proceedings of Recent Advances in Natural Language Processing, pages 730–739,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_085

in the Tunisian dialect based on the final findings provided by various experiments.

This model will be compared to our previous CRF and SMT models (24) by utilizing the same training and testing corpus.

The remaining of this paper is structured as follows: In Section 2 we describe the linguistic background of the Tunisian dialect, we try to show the complexity of diacritization tasks based on examples and we present the main level of diacritization. Section 3 introduces our proposed model and experiments. Section 4 provides an exhaustive experimental evaluation that illustrates the efficiency and accuracy of our proposed method. Section 5 summarizes the key findings of the present work and highlights the major directions for future research.

2 Linguistic background

2.1 Tunisian dialect

The language situation in Tunisia is "polyglossic", where distinct language varieties, such as the normative language (MSA) and the usual language (the Tunisian dialect) coexist (24).

As an official language, MSA is extensively present in multiple contexts, namely education, business, arts and literature, and social and legal written documents. However, the Tunisian dialect is the current mother tongue and the spoken language of all Tunisians from different origins and distinct social belongings. For this purpose, this dialect occupies a prominent linguistic importance in Tunisia.

Another salient feature of the Tunisian dialect is that it is strongly influenced by other foreign languages. In fact, it is the outcome of the interaction between Berber, Arabic and many other languages such as French, Italian, Turkish and Spanish. The manifestation of this interaction between these languages is obvious in introducing borrowed words. As a result, the lexical register of the Tunisian dialect seems to be more open and contains a very rich vocabulary.

The Tunisian dialect has other specific aspects. Indeed, since this dialects spoken rather than written or taught at school, there is neither grammatical, nor any orthographical or syntactic rules to be followed.

2.2 Challenges in the absence of diacritization in Tunisian dialect

The absence of diacritics signs in the Tunisian dialect texts often increases the morphological, syntactic and semantic ambiguity in the Tunisian dialect. Some of them are presented as follows:

- **Morphological ambiguity:** The absence of the diacritical marks poses an important problem at the association of grammatical information of the undiacritized word (24). For example, the word لعب /lEb/ admits several possible words that correspond to different solutions at the grammatical labeling level. We can find the plural noun "toys" and the verb "play" in 3rd person masculine, singular of passive accomplishment.

- **Syntactic ambiguity:** It should be noted that the ambiguities in the association of grammatical information, related to the undiacritic words, pose difficulties in terms of syntactic analysis (24). For example, the undiacritic expression كتب الولّيد عَالبيرو can admit two different diacritization forms that are syntactically accepted.

  - We find the diacritization form كتِب الولّيد عَالبيرو [The boy wrote on the desk] whose syntactic structure corresponds to a verbal sentence.
  - In addition, we find the diacritization form whose syntactic structure corresponds to a nominal sentence كتُب الولّيد عَالبيرو [The boy's books are on the desk].

- **Semantic ambiguity:**Tthe different diacritization of a word can have different meanings, even if they belong to the same grammatical category. For example, among the possible dicaritization of the word قريت /qryt/ we find the following dicaritization:

  - قريت /qryt/ [I read]
  - قَرِيت /qaryt/ [I taught].

  These two diacritic words have the same grammatical category: verb but they have two different meanings.

2.3 Diacritization level

The use of diacritic symbols in several instances is quite crucial in order to disambiguate homographic words. Indeed, the level of diacritization refers to the number of diacritical marks presented on a word to avoid text ambiguity for human readers. There are four levels of possible diacritization.

- **Full Diacritization:** this is the case where each consonant is followed by a diacritic. This type of diacritization is used mainly in classical Arabic, especially in religion-related books and educational writings.

- **Half Diacritization:** the objective of this category is to add diacritics of a word except the letters that depend on the syntactic analysis of the word. Often, it is the before last letter that depends on syntactic analysis of a word but it is not always the case due to the use of suffixes.

- **Partial Diacritization:** is the case of adding only lexical vowels. The latter can be defined as the vowels with which the meaning of words changes. The goal of marking these vowels is to remove ambiguity from the meaning of words.

- **No Diacritization:** This level is completely underspecified. The script is subject to ambiguity, especially with homographs (4).

3 Methodology and experiment step

In recent years, RNN has received a lot of interest in many NLP tasks of sequence transcription problems, such as speech recognition, handwriting recognition and diacritics restoration. So, we select the RNN to evaluate its performance on the diacritization of the Tunisian dialect. In this work, we adopted the full diacritization level, at which all diacritics should be specified in a word.

3.1 Recurrent neural networks

RNN can be mapped from a sequence of input observations to a sequence of output labels. The mapping is defined by activation weights and a non-linear activation function as in a standard MLP. However, recurrent connections allow to access past activations. For an input sequence x_1^T, RNN computes the hidden sequence h_1^T and the output sequence y_1^T by performing the following operations for $t = 1$ to T (13):

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

$$y_t = W_{hy}h_t + b_y \quad (2)$$

where H is the hidden layer activation function, W_{xh} is the weight matrix between input and the hidden layer, W_{hh} is the recurrent weight matrix between the hidden layer and itself, Why is the weight matrix between the hidden and output layers, b_h and b_y are the bias vectors of the hidden and output layers, respectively. In a standard RNN, H is usually an element-wise application of sigmoid function. Such a network is usually trained using the back-propagation through time (BPTT) training (27).

- **Long short-term memory: LSTM**

An alternative RNN called Long Short-Term Memory (LSTM) is introduced where the conventional neuron is replaced with a so-called memory cell controlled by input, output and forget gates in order to overcome the vanishing gradient problem of traditional RNNs (12). In this case, H can be described by the following composite function (13):

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (4)$$

$$c_t = f_t c_{t-1} + i_t \; tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (5)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (6)$$

$$h_t = o_t \; tanh(c_t) \quad (7)$$

where the σ is the sigmoid function. i, f, o and c correspond to, the input, forget, output gates and cell activation vectors respectively.

- **Bidirectional Long short-term memory: B-LSTM**

A BLSTM processes the input sequence in both directions with two sub-layers in order to account for the full input context. These two sub-layers compute forward and backward hidden sequences $\overrightarrow{h}$, $\overleftarrow{h}$ respectively, which are then combined to compute the output sequence y as follows (13):

$$\overrightarrow{h_t} = H(W_{x\overrightarrow{h}} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}) \quad (8)$$

$$\overleftarrow{h_t} = H(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}) \quad (9)$$

$$y_t = W_{\overrightarrow{h}y} \overrightarrow{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y \quad (10)$$

3.2 Model architecture

In our diacritization task, the basic idea is to attribute a corresponding diacritical label to each character. Hence, we apply RNN to model our sequence data, where a sequence of characters constitutes the input and the probability distribution over diacritics forms the output. Schematically, our RNN structure is employed in this work as the following figure:

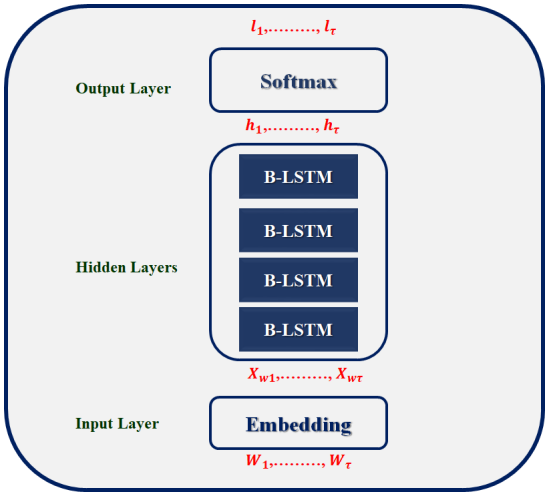

Figure 1: Architecture of our neural network

This can be statistically expressed in this way: Given that $W = (w_1 ... w_T)$, w indicates a sequence of characters, where each character is related to a label l_t. In this respect, a label may stand for 0, 1 or more diacritics. Furthermore, a real-valued vector x_w is a representation of each character w in the alphabet.

We can state that our neural network consists of 3 layers, namely an input layer, a hidden layer and an output layer. Each layer fulfils a particular purpose. In what follows, we will explain the advantages of each layer.

- **Input layer:** This level consists in mapping the letter sequence w to a vector sequence x. We have checked and prepared data of our corpus. In combining the gemination mark with another diacritic, each character in the corpus has a label corresponding to 0,1 or 2 diacritics. Character embedding input, which is initialized randomly and updated during training, means that each character in the input sentence is represented by a vector of d real numbers. It is worth pointing out that adding a linear projection after the input layer affects the learning of a new representation for the latter embedding.

- **Hidden layer:** This layer consists in mapping the vector sequence x to a hidden sequence h. Several types of hidden layers have been used to choose the best performance and the best result in the automatic diacritization of the Tunisian dialect. Hence, these experiments were based on LSTM different layers ranging from one layer to multiple B-LSTM layers.

- **Output layer:** This last layer focuses on mapping each hidden vector h_t to a probability distribution over labels l. In this layer, we use a softmax activation function to produce a probability distribution over output alphabet at each time step.

$$P(l|w_t) = \frac{exp(y_t[l])}{\sum_{l'} exp(y_t[l'])} \quad (11)$$

where $y_t = W_{hy} h_t + b_y$ and $y_t[l]$ is the l^{th} element of y_t.

3.3 Experience

As mentioned above, in order to train the RNN to achieve high accuracy, we apply our experiment based on several training options. These options include the choice of the number of layers in the hidden layer. The aim of this experiment is to determine which options will give optimal accuracy. Indeed, we applied these experiences, in which several types of hidden layers were tested. These layers are ranging from one LSTM layer to multiple B-LSTM layers.

The network is trained using Gradient Descent optimizer with learning rate 0.0003 and a mini-batch size of 200. For dropout, a rate of 0.2 is used both on embedded inputs and after each type of hidden layers; either LSTM or B-LSTM. Weights are randomly-initialized with normal distribution of zero mean and 0.1 standard deviation and weight updates after every batch. The loss function is the cross-entropy loss summed over all outputs. The GPU used is Nvidia GTX 580 that has 16 streaming multiprocessors and 1.5 GB of memory

4 Results and discussion

4.1 Evaluation Metric

In order to measure our model performance, an evaluation metric, known as Diacritic Error Rate (DER) is generally used. DER indicates how many letters have been incorrectly restored with their diacritics. The DER can be calculated as follows (24):

$$DER = \frac{(1 - |TS|)}{|TG|} * 100 \qquad (12)$$

Where $|TS|$ is the number of letters assigned correctly by the system, and $|TG|$ is the number of diacritized letters in gold standard texts.

4.2 Datasets

This section shows a breakdown of different sizes of our data sets, which were gathered from various sources. So far, we have used four existent types of corpora for our teamwork.

- We made use of our TARIC corpus (Tunisian Arabic Railway Interaction Corpus) (24). The latter collected information about the Tunisian dialect used in a railway station. This corpus was recorded in the ticket offices of the Tunis railway station. We recorded conversations in which there was a request for information about the train schedules, fares, bookings, etc. This corpus consists of several dialogues; each dialogue is a complete interaction between a clerk and a client. All the words are written using the Arabic alphabet with diacritics. The diacritics indicate how the word is pronounced. The same word can have more than one pronunciation.

- The second corpus is called STAC (Spoken Tunisian Arabic Corpus)(35). This corpus is a representation of spontaneous discourses in Tunisian dialect. This dialect corpus of transcribed discourses deals with multiple themes, such as social affairs, health, religion, etc.

- We utilized another type of corpus that is the result of a conversion tool from Latin written texts (also called Arabizi) into Arabic scripts following the CODA conversion. The Arabizi corpus is collected from social media like Facebook, Twitter and SMS messaging (22).

- In order to solve the problem of the lack of resources for the Tunisian dialect, we have chosen to gather corpora from blog sites written in this dialect using Arabic alphabets (24). (For more details see (24))

As mentioned above, the Tunisian dialect differs from MSA and it does not have a standard spelling because there are no academies of Arabic dialect. Thus, to obtain coherent learning data, it is necessary to utilize a standard spelling. Indeed, there are words with many forms. For example, the word رزرفسيون /reservation/ can be written in four different ways: رَازرفسيون, رَازَارفسيون and ريزرفسيون.

In this work, spelling transcription guidelines CODA (Tunisian Dialect writing convention), (36), were adopted. CODA is a conventionalized orthography for Dialectal Arabic. In CODA, every word has a single orthographic representation. It uses MSA-consistent and MSA-inspired orthographic decisions (rules, exceptions and ad hoc choices). CODA preserves, also, dialectal morphology and dialectal syntax. CODA is easily learnable and readable. CODA has been designed for the Egyptian Dialect (11) as well as the Tunisian Dialect (36) and the Palestinian Levantine Dialect (20). For a full presentation of CODA and an explanation of its choices, see ((11), (36)).

The normalization step is essential because it presents a key point for the other steps of our method. Among the normalisation Tunisian Dialect words we have:

- Number ”sixteen” is written as ستطَاش.

- To define the future, we must follow the following form: بَاش + verb, for example: بَاش نمشي.
- To define the negation, we must follow the following form: مَا + verb.

Let's remember that the Tunisian Dialect is characterized by the presence of foreign words, such as for instance: French, English, Spanish, Italian, etc. To transcribe these words, Arabic alphabets have been used. These foreign words have a unique form, for example: رتور [Back], ترَان [train]...

At the end of this step, we obtain a standardized corpus. The figure 2 represents a corpus extract before the normalization step.

Figure 2: Corpus extract before the normalisation step

The figure 3 represents a corpus extract after the normalization step.

Figure 3: Corpus extract after the normalisation step

Since there are no automatic diacritization tools for the Tunisian dialect, and because the MSA tools are unable to treat this dialect due to the differences between the MSA and the Tunisian dialect, we were obliged to diacritize the corpus manually.

Below we provide the most important characteristics in Table 1.

Table 1: Characteristics of our corpus.

	# **statements**	#**words**
TARIC	21,102	71,684
STAC	4,862	42,388
Arabizi	3,461	31,250
Blogs	1582	27,544
Total	31,007	172,866

We aimed to decently create training, development and test sets in order to judge our diacritization models. We outlined the available datasets for the language under investigation. We randomly selected 23,255 sentences for training, 1,550 for development and 6,202 for testing.

Table 2 reports some quantitative information for the datasets.

Table 2: Tunisian dialect diacritization corpus statistics.

	Train	Dev	Test
# Statements	23,255	1,550	6,202
# words	129,649	8,643	34,574
# Letters	64,8247	43,216	172,867

4.3 Result

In this section, we present the evaluation outcome of our established diacritization models. We use DER as an evaluation metric. The adopted RNN has from 1 to 4 hidden layers, each with 250 neurons. This number is chosen after different experiments. We come up with the conclusion that a smaller number of neurons (less than 250) have an impact on accuracy rate and a greater number do not improve it in a significant way. Table 3 gives an overview of the RNN models outcomes in terms of diacritic error rate (DER).

Table 3: Diacritization Error Rate Summary for the Tunisian dialect RNN model

Model	DER
LSTM	13.86%
B-LSTM	12.31%
2-layer B-LSTM	11.53%
3-layer B-LSTM	10.72%
4-layer B-LSTM	10.83%

Table 3 shows the effect of using LSTM and B-LSTM models, and the number of hidden layers on the DER. According to this table, results show a DER of 13.56% for LSTM and 12.31% for B-LSTM. Based on the results of our RNN, we detected an enhancement of 1.55% in DER of the B-LSTM model as compared to LSTM model. This means that B-LSTM is more performant than LSTM.

Moreover, we noticed that increasing the number of B-LSTM layers from one hidden layer to three layers improves accuracy. But, we applied the 3-layer BLSTM because its accuracy is not only closer but also fasther than the 4-layer BLSTM. Indeed, the training time rises from 3:52 to 6:78 hours when the number of layers progresses monotonically from 3 to 4 and the testing time icreases from 3.65 to 5.41 minutes. Hence, the 3-layer B-LSTM configuration was adopted.

To conclude, a 3-layer BLSTM models achieved the best results.

4.4 Error Analysis

In order to reveal the weaknesses of our automatic diacritic restoration RNN models, we analyzed all errors that are mainly due to the following reasons:

- We noticed that these errors are due to the presence of foreign words in our corpus.
- Some error words with prefixes, or suffixes or both can be significantly perceived. It is hard to diacritize these complex words in a correct way, as the in flection diacritical mark is related to the last letter of the stem rather than to the last letter of the suffix.
- Errors due to form/spelling diacritization errors. Errors caused by ”Shadda” (consonant doubling), or Tanween (nunation). We perceived that restoring ”shadda” is harder than restoring the other diacritics.
- Errors due to missing and incorrect short vowels (i.e. lexical diacritics).

We have manually checked 150 error samples of our input RNN model. The following figure shows an example of 4 sample sequences that have errors. The words that have errors are underlined and in red.

Sample	Target sequence	Output sequence
1	قَرّيتْ صبَاحْ	قْرِيتْ صبَاحْ
2	لاَ مزَالْ برْنَامجْ فِي اللِيلْ	لاَ مزَالْ برْنَامجْ فِي اللِيلْ
3	شْرَاتْ لِوِلْدْهَا الصْغِيرْ لُعْبَةْ	شْرَاتْ لُوِلْدْهَا الصْغِيرْ لُعْبَةْ
4	اَكْسِيرَاسْ ثْرَانْ	اَكْسِيرَاسْ ثْرَانْ

Figure 4: Sample sequences with errors

In about 21% of the samples, we have remarked that the absence of ”shadda” in some words can lead to a semantic ambiguity of the verb. For instance, sample 1 shows that target verb قَرّيت [I taught] is output as قرِيت [I read]. These two diacritic words have the same grammatical category: verb but they have two different meanings.

Diacritization errors in test samples can cause about 4% of errors. For example, sample 2 displays that the "Fatha" in the word لَا was mistakenly entered after the last letter rather than the first letter.

Some error words with prefixes, suffixes or both can be significantly perceived, in about 41% of the samples. In another illustration, the error word in sample 3 has both the prefix "la" لِ and the pronoun suffix "haA" هَا.

We also noticed a significant fraction of error words (34%) due to the presence of foreign words in our corpus as in sample 4..

4.5 Comparison with State-of-art Systems

In this section, we compare our proposed RNN model with two other models, namely SMT and a discriminative model as a sequence classification task based on CRFs (24). These two models were realized in our previous works in order to carry on the dialect restoration for the Tunisian dialect. To achieve this comparison, we employed the same dialectical corpus and evaluation metrics.

Concerning the first model, we regarded the diacritization problem as a simplified phrase-based SMT task. The source language is the undiacritic text while the target language is the diacritic text. The basic idea of SMT is to analyze automatically existing human sentences called parallel corpus in order to build translation model. The alignment from words without diacritics to words with diacritics is a monotonic mapping. To do this, we employed moses (21) a SMT tool.Word alignment was done with GIZA++ (25). We implemented a 5-gram language model using the SRILM toolkit (31). We decoded using Moses (21).

In the second model, we decided to get the diacritical marks restoration by focusing on diacritization based on grammatical information. We intended to build dependency relations between words and "POS" tags and to perceive their effects on word diacritizations. In fact, we proposed to scrutinize the integration of grammatical information "POS" for the diacritization with the aid of Conditional Random Fields (CRF)(24).

The following table reviews the accuracy results to restore diacritics automatically from our previous published researches in the Tunisian dialect field.

Table 4: Diacritization results of related work (CRF and SMT models) and our RNN model

Model	DER
3-layer B-LSTM	10.72%
CRF	20.25%
SMT	33.15%

As depicted in Table 4, our RNN model (3-layer B-LSTM) provides the best results(DER of 10.72%) compared to both SMT (DER of 33.15%) and CRF (DER of 20.25%) models.

5 Conclusion

The absence of short vowels gives rise to a great ambiguity which influences the results of such NLP applications. An outcome of this study was the development of RNN diacritic restoration model for Tunisian dialect. To the best of our knowledge, this is the first work that deals with the problem of Tunisian dialect diacritizers using RNN.

In order to choose the best configuration of the RNN network, we did several preliminary experiments with different training options. These options concern the hidden layer where we tested the impact of the change of the neural network size and the topology on its performance. Several types of hidden layers are tested, ranging from one layer LSTM to multiple B-LSTM layers. The best accuracy is obtained when using the 3-layer B-LSTM model (DER of 10.72%). We compared our RNN diacritization model with two major models, namely a SMT and CRF models (24). These two models were realized in our previous works in order to carry on the dialect restoration for the Tunisian dialect. During this comparison, we employed the same dialectical corpus and evaluation metrics. About 9.53% DER improvement of RNN model was achieved over the best reported CRF model.

We have two future plans for the diacritization problems of Tunisian dialect. The first plan consists in expanding a rule-based diacritizer system and integrating it into our RNN model in order to ameliorate the outcomes. The second plan focuses on providing morphological analysis of such

words to the RNN in order to achieve higher accuracy. The presence of significant fraction of errors in complex words that contain prefixes, suffixes, or both open up new perspectives for future research.

Abandah, G., Graves, A., Al-Shagoor, B., Arabiyat, A., Jamour, F. and Al-Taee. M. 2015. Automatic diacritization of arabic text using recurrent neural networks. International Journal on Document Analysis and Recognition (IJDAR).

Alotaibi, Y. A., Meftah, A. H. and Selouani. S.A. 2013. Diacritization, Automatic Segmentation and Labeling for Levantine Arabic Speech. In Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE).

Alqudah,S., Abandah,G., Arabiyat, A., 2017.Investigating Hybrid Approaches for Arabic Text Diacritization with Recurrent Neural Networks. 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies.

Al-Badrashiny, M., Hawwari, A. and Diab, M. 2017. A Layered Language Model based Hybrid Approach to Automatic Full Diacritization of Arabic. Third Arabic Natural Language Processing Workshop.

Ameur, M. Moulahoum, Y. and Guessoum, A. 2015. Restoration of Arabic Diacritics Using a Multilevel Statistical Model. In IFIP International Federation for Information Processing.

Ayman, A. Z., Elmahdy, M., Husni, H. and Al Jaam, J. 2016. Automatic diacritics restoration for Arabic text. International Journal of Computing & Information Science..

Azmi, A. and Almajed, R. 2015. A survey of automatic Arabic diacritization techniques. Natural Language Engineering, 21, pages:477495.

Belinkov, Y. and Glass. J. 2015. Arabic diacritization with recurrent neural networks. Conference on Empirical Methods in Natural Language Processing.

Bouamor, H., Zaghouani, W., Diab, M., Obeid, O., Kemal, O., Ghoneim, M. and Hawwari, A. 2015. A pilot study on Arabic multi-genre corpus diacritization annotation. The Second Workshop on Arabic Natural Language Processing.

Diab, M., Ghoneim, M. and Habash. N. 2007. Arabic Diacritization in the Context of Statistical Machine Translation. In Proceedings of MTSummit, Denmark.

Diab, M., Habash, N., Owen, R.: Conventional Orthography for Dialectal Arabic. In Proceedings of the Language Resources and Evaluation Conference , Istanbul,2012

Gers,F. : Long short-term memory in recurrent neural networks, Ph.D. dissertation, Department of Computer Science, Swiss Federal Institute of Technology, Lausanne, EPFL, Switzerland, 2001.

Graves, A., Mohamed, A., Hinton, G: Speech recognition with deep recurrent neural networks, IEEE International Conference on Acoustics, Speech, and Signal Processing, Canada,2013.

Fashwan, A., Alansary, S. 2017. SHAKKIL: an automatic diacritization system for modern standard Arabic texts. The Third Arabic Natural Language Processing Workshop (WANLP).

Habash, N., Shahrour, A., and Al-Khalil, M. 2016, Exploiting Arabic Diacritization for High Quality Automatic Annotation, the Tenth International Conference on Language Resources and Evaluation, LREC 2016.

Habash, N. and Rambow, O. 2007. Arabic diacritization through full morphological tagging. The Conference of the North American Chapter of the Association for Computational Linguistics.

Hamed, O and Zesch, T. 2017. A Survey and Comparative Study of Arabic Diacritization Tools. Journal of Language Technology and Computational Linguistics, volume 32, number 1.

Harrat, S., Abbas, M., Meftouh, K., Smaili, K., Bouzareah, E.N.S. and Loria, C. 2013. Diacritics restoration for Arabic dialect texts. 14th Annual Conference of the International Speech Communication.

Hifny, Y. 2012. Higher order n-gram language models for Arabic diacritics restoration. In Proceedings of the 12th Conference on Language Engineering.

Jarrar, M., Habash, N., Akra, D. and N. Zalmout, N.: Building a Corpus for Palestinian Arabic: a Preliminary Study :In Proceedings of the Arabic Natural Language Processing Workshop, EMNLP, Doha,2014

Koehn, P. Hoang; H. Birch, A. Callison-Burch, C. Federico, M. and Bertoldi, N. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. ACL 2007, demonstration session.

Masmoudi, A., Habash, N., Khmekhem, M., Esteve, Y. and Belguith, L.: Arabic Transliteration of Romanized Tunisian Dialect Text: A Preliminary

Investigation, Computational Linguistics and Intelligent Text Processing - 16th International Conference, CICLing 2015, Cairo, Egypt, p.608-619, (2015).

Masmoudi, A., Bougares,F., Ellouze, M., Esteve, Y., Belguith, L.: Automatic speech recognition system for Tunisian dialect. Language Resources and Evaluation 52(1): 249-267 (2018).

Masmoudi, A., Mdhaffer,S., Sellami, R. Belguith, L.: Automatic Diacritics Restoration for Tunisian Dialect. TALLIP2018: Transactions on Asian and Low-Resource Language Information Processing.

Och, F. and Ney, H. 2003. A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics, 29(1):19–52.

Rashwan, M., Al Sallab, A., Raafat, H. and Rafea, A. Deep Learning Framework with Confused Sub Set Resolution Architecture for Automatic Arabic Diacritization. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2015.

Rumelhart, D., Hinton,G., and Williams, R. Learning representations by back-propagating errors, Nature, no. 323, pp. 533 536, 1986.

Said, A., El-Sharqwi, M., Chalabi, A , and Kamal, E. A hybrid approach for Arabic diacritization, Application of Natural Language to Information Systems, pp. 53-64, Jun 2013.

Shaalan, K., Abo Bakr, M. and Ziedan. I. 2009. A hybrid approach for building Arabic diacritizer. In Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages.

Shaalan, K., Abo Bakr, H. and Ziedan. I. 2008. A statistical method for adding case ending diacritics for Arabic text. In Proceedings of Language Engineering Conference.

Stolcke A. 2002. SRILM an Extensible Language Modeling Toolkit. Proceedings of ICSLP.

Zaghouani, W., Bouamor, H., Hawwari, A., Diab, M., Obeid, O., Ghoneim, M., Alqahtani, S and Oflazer, K. 2016. Guidelines and framework for a large-scale Arabic diacritized corpus. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016).

Zitouni, I., Sorensen, J. and 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 Association for Computational Linguistics.

Zitouni, I. and Sarikaya, R. 2009. Arabic Diacritic Restoration Approach Based on Maximum Entropy Models. In Journal of Computer Speech and Language.

Zribi, I., Ellouze, M., Belguith, L.H. and Blache, P. 2015. Spoken Tunisian Arabic Corpus "STAC": Transcription and Annotation. Res. Comput. Sci. 90.

Zribi I., Boujelben R., Masmoudi A., Ellouze M., Belguith L., Habash N.: A conventionnal Orthography for Tunisian Arabic, LREC'2014, Reykjavik, Iceland, (2014).