A Novel Deep-learning based Approach for Automatic Diacritization of Arabic Poems using Sequence-to-Sequence Model

Mohamed S. Mahmoud[1], Nermin Negied[2]
Technical University of Munich, School of Informatics, Germany[1]
School of Communication and Information Engineering, Zewail City, Giza, Egypt[2]
School of Engineering and Applied Science, Nile University, Giza, Egypt[2]

***Abstract*—Over the last 10 years, Arabic language have attracted researchers in the area of Natural Language Processing (NLP). A lot of research papers suddenly emerged in which the main work was the processing of Arabic language and its dialects too. Arabic language processing has been given a special name ANLP (Arabic Natural Language Processing). A lot of ANLP work can be found in literature including almost all NLP applications. Many researchers have been attracted also to Arabic linguistic knowledge. The work expands from Basic Language Analysis to Semantic Level Analysis. But Arabic text semantic analysis cannot be held without considering diacritization, which can greatly affect the meaning. Many Arabic texts are written without diacritization, and Diacritizing them manually is a very tiresome process that may need an expert. Automatic diacritization systems became a demand as an initial step for processing Arabic text for any Arabic Language Processing application as Arabic diacritization is very important to get a readable and understandable Arabic text. For this reason, many researchers recently worked on building systems and tools that automatically diacritize un-diacritized Arabic texts. This work presents a novel deep learning-based sequence-to-sequence model to diacritize un-diacritized Arabic poems. The proposed model was tested and achieved high diacritization accuracy rate.**

Keywords*—*Text diacritization; deep learning; sequence-to-sequence; regex; tokenization; ANLP

I. Introduction

Arabic language is spoken by more than 360 million people, and its speakers are distributed in the Middle East, and many other neighboring regions. It is noteworthy to know that it is one of the six official languages of the United Nations. Some ANLP work in literature includes sentiment analysis [1], machine translation [2], dialect identification [3], Named Entity Recognition NER [4], question answering [5], summarization [6], etc. What makes Arabic special and challenging at the same time is the diacritization. Diacritics play an important role for understanding the meaning of Arabic statements. Undiacritized Arabic sentence may be ambiguous and difficult to understand. Arabic text is naturally written in some percentage of diacritics to avoid misleading meaning. The percentage of these diacritics depends on the context or the domain. For example, religious text like the Holy Quran is fully diacritized to minimize chances of understanding it incorrectly. Hadeeth books are also diacritized with high percentage. So are kids' educational books. Classical literature and poetry tend to be partially diacritized as well. However, newspapers and other genre are rarely diacritized. Naturally Arabic text consists of two classes of symbols: letters and diacritics. Letters comprise long vowels such as A, y, w as well as consonants. Diacritics on the other hand comprise short vowels, gemination markers, nunation markers, as well as other markers (such as hamza, the glottal stop which appears in conjunction with a small number of letters, dots on letters, elongation, and emphatic markers) which in all, if present, render an exact precise reading of a word. The remaining of this paper is organized as follows: A literature survey of the recent work in this area can be found in the following section. Section III describes the approach proposed by this paper. Experimental work is demonstrated in Section IV. Section V presents the Results and Discussion. Conclusion and suggestions for future work can be found in Section VI.

II. Literature Review

A lot of Arabic Natural Language Processing ANLP work has been done in literature. For instance, Farghaly and Shaalan started their investigations in 2009. Their research showed the significance of the Arabic language and its properties. Their work also discussed the challenging characteristics of the language like agglutination and morphological non-concatenation [7]. Habash in 2010 [8] has wrote a comprehensive survey about Arabic language characteristics and features. The survey wasn't limited only to Modern Standard Arabic (MSA) but also, it's dialects. Habash's survey extended to talk about different orthographic, morphological and syntactic aspects of this language in details. Five years later another ANLP comprehensive survey of Shoufan and Alameri [9] has been published. The authors proposed a general categorization of dialectical Arabic language processing into four main classes: (1) Basic Language Analyses (BLA). (2) Building Resources (BR), (3) Language Identification (LI) and (4) Semantic-Level Analysis (SemA). Their survey can be considered as a reference for knowing relevant contributions that address a specific ANLP aspect for Arabic dialects.

Moving to automatic Arabic diacritization attempts found in literature, Diab and Habash [10] studied the impact of Arabic diacritization on statistical machine translation (SMT). Their research confirmed that the SMT performance is positively affected by the presence of diacritization to the extent that would make it robust to slightly high OOV rates. They reported that diacritization step is needed before translation attempts to achieve high SMT performance. In 2008, Shaalan el al [11] used SVM to diacritze undiacrtized Arabic text and they achieved accuracy rate of 95.3% and 82% F-measure, but their technique only adds diacritics to the last letter of the word, while the diacritization of internal letters can affect the meaning, for example عمّان means (Amman – Captial of Jourdan) and عُمان (Oman - Arabian Country). In 2016, Bouamor et al [12] limited their study to three types of diacritical marks: short vowels, nunation, and shadda (gemination). They were mainly interested in reducing the time consumed in Diacritizing undiacritized Arabic text, so they conducted a pilot study for a minimum diacritization scheme. Their proposed scheme mainly encoded the most relevant differentiating diacritics to reduce confusability among words that are homographs but not homophones. They finally confirmed that it is difficult to build such a scheme because of subjectivity and they promised to dig deeper in future work. In 2017, Darwish et al [13] built an Arabic diacritizer using a Viterbi decoder to diacritize every word including stems and all morphological patterns. Their diacritizer also used transliteration mining combined with sequence labeling to diacritize named entities that have English transliterations. They also used SVM but combined with some filters, and they achieved good results for case ending diacritization. In the same year, Fashwan and Alansari [14] used morphological and syntactic processing to discretize Arabic texts, in which Morphology-dependent that selects the best internal diacritized form of the same spelling, whereas Syntax-dependent that detects the best syntactic case of the word within a given sentence using the parsing tree of that sentence. In their discussion section they stated that they need to improve their system through adding more morphological and syntactic Arabic rules. In 2018, Alosaimy et al [15] have created a tool for accurate diacritization of highly cited texts by automatically "borrowing" diacritization from other citations using n-grams matching, and they used the well-known sunnah book 'Riyad Al Salheen' to test their tool. They reported a high accuracy in their conclusion, but their tool only works with highly cited texts. Darwish et al [16] in the same year, limited their study to Moroccan and Tunisian Arabic, using deep neural networks DNN, and they reported a small error rate for both languages. In 2019, Fadel et al [17], used Feed-Forward Neural Network (FFNN) and Recurrent Neural Network (RNN), to diacritize Arabic texts. Their models were trained using 50 epochs and tested only on the available benchmark datasets, and they reported a good diacritization results. M. Madhfar, A. M. Qamar in 2021 [18], used simple baseline deep learning model, encoder-decoder model, and encoder model that is used in text-to-speech system. The authors mainly studied the impact of presence and absence of diacritics on a text-to-speech, and they concluded that diacritics play a major role in the successfulness of almost all ANLP applications. They also stated that there is a big difference between Modern Standard Arabic (MSA) and Classical Arabic (CA) language, the fact that makes the problem more challenging and data demanding. Abuali and Kurdy in 2022 [19], studied the relationship between the full diacritization of the Arabic text and the quality of the speech synthesized in screen readers, and they came up with a conclusion that it is mandatory to diacritize Arabic text before converting it to audio. Thompson and Alshehri [20] in 2022, stated that training a diacritizer to diacritize and translate Arabic text is much better than training the model to only diacritize the text as it resolves a lot of ambiguity issues. The authors used the standard data splits of Diab et al. [21], and they achieved a low error rate of 4.8% on the Penn Arabic Tree-bank datasets.

III. DATASET

In this work, we collected a variety of textual sentences along with their diacritized version. The sentences varied in lengths and contexts (talking about different topics) in Arabic Language. Fig. 1 depicts the histogram of the different sentence lengths in our data (number of characters is used to measure sentence length). As you can see from the figure, nearly all the data are concentrated at lengths less than half of the maximum length in both the un-diacritized (Yellow) and diacritized sentences (Blue) which is a good reason to use the half of the maximum length as a padding length in the data preprocessing stage. More about the padding and data preprocessing will be explained in the next section.

The collected data contains 2000 sentences being represented as an ordered set of characters in the input and their equivalent 2000 sentences characters and their diacritization in the output Table I. This research, unlike related work in literature covers all types of diacritization of Arabic language (Harakat) such as: Fataha, Kasara, Damma, Skoon, Fathatan, Kasratan, Dammatan and Shaddah.

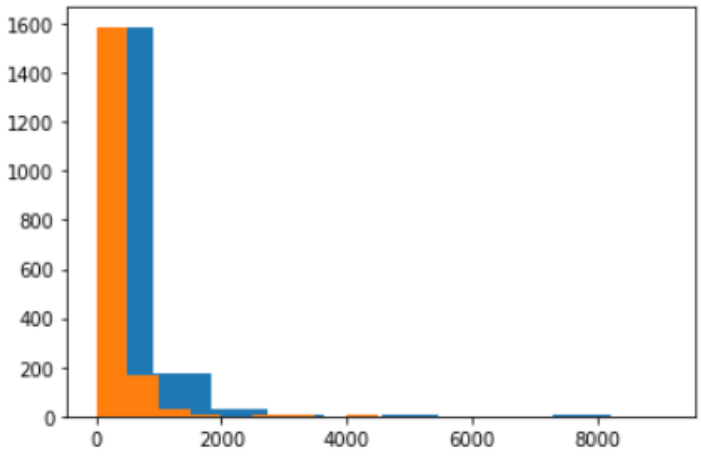

Fig. 1. Histogram of the lengths of the sentences diacritized (Blue) and un-diacritized (Yellow).

TABLE I. EXAMPLE OF INPUT / OUTPUT TEXT SEQUENCE

Input Sequence	كذلك كان من أهداف حركة الاستشراق خدمة مخططات اليهود في هدم الإسلام والتمكين لهم في فلسطين عن طريق تشويه التاريخ العام، وتأكيد حق اليهود في فلسطين، وذلك ما يشير إليه المرحوم الدكتور محمد البهي بقوله: إن هؤلاء أقبلوا على الاستشراق لأسباب دينية، وهي محاولة إضعاف الإسلام والتشكيك في قيمته وإثبات فضل اليهود على الإسلام بادعاء أن اليهودية هي مصدر الإسلام الأول، ولأسباب سياسية تتصل بخدمة الصهيونية فكرة أول، ثم دولة ثاني. ولا يغيب عن بالنا أن المستشرقين هم الذين طرحوا على إنجلترا فكرة إنشاء وطن قومي لليهود في فلسطين في مؤتمر لندن المنعقد في سنة سبع وتسعمئة وألف
Output Sequence	كَذَلِكَ كَانَ مِنْ أَهْدَافِ حَرَكَةِ الِاسْتِشْرَاقِ خِدْمَةُ مُخَطَّطَاتِ الْيَهُودِ فِي هَدْمِ الْإِسْلَامِ وَالتَّمْكِينُ لَهُمْ فِي فِلَسْطِينَ عَنْ طَرِيقِ تَشْوِيهِ التَّارِيخِ الْعَامِّ، وَتَأْكِيدِ حَقِّ الْيَهُودِ فِي فِلَسْطِينَ، وَذَلِكَ مَا يُشِيرُ إِلَيْهِ الْمَرْحُومُ الدُّكْتُورُ مُحَمَّدُ الْبَهِيُّ بِقَوْلِهِ: إِنَّ هَؤُلَاءِ أَقْبَلُوا عَلَى الِاسْتِشْرَاقِ لِأَسْبَابٍ دِينِيَّةٍ، وَهِيَ مُحَاوَلَةُ إِضْعَافِ الْإِسْلَامِ وَالتَّشْكِيكُ فِي قِيمَتِهِ وَإِثْبَاتُ فَضْلِ الْيَهُودِ عَلَى الْإِسْلَامِ بِادِّعَاءِ أَنَّ الْيَهُودِيَّةَ هِيَ مَصْدَرُ الْإِسْلَامِ الْأَوَّلُ، وَلِأَسْبَابٍ سِيَاسِيَّةٍ تَتَّصِلُ بِخِدْمَةِ الصَّهْيَوْنِيَّةِ فِكْرَةً أَوَّلَ، ثُمَّ دَوْلَةً ثَانِيَ. وَلَا يَغِيبُ عَنْ بَالِنَا أَنَّ الْمُسْتَشْرِقِينَ هُمُ الَّذِينَ طَرَحُوا عَلَى إِنْجِلْتِرَا فِكْرَةَ إِنْشَاءِ وَطَنٍ قَوْمِيٍّ لِلْيَهُودِ فِي فِلَسْطِينَ فِي مُؤْتَمَرٍ لَنْدَنَ الْمُنْعَقِدِ فِي سَنَةِ سَبْعٍ وَتِسْعِمِئَةٍ وَأَلْفٍ

IV. THE PROPOSED APPROACH

Using our collected diacritized dataset, the proposed methodology exploits techniques in Deep Learning and Text Analysis to diacritize un-diacritized Arabic text. The objective is to take an input un-diacritized text and output its equivalent diacritized text. Unlike most of work in literature that used regular text analysis or statistical text classification to partially diacritize un-diacritized Arabic text, we used sequence-to-sequence model to diacritize text through mapping the input sequence of un-diacritized characters into diacritized ones. In our work, we clean the input and output texts and generate a parallel map between the two texts as a first stage to be able to train the deep learning model as a second stage. A further explanation of both stages can be found in the following subsections:

A. Data Preprocessing

In the data preprocessing stage, several text processing techniques were utilized. At first, data cleaning was used to remove irregular characters from the data such as "\ufeff" and "\u200f" which appeared so often because the data was stored in a bytes-like fashion. Then, regular expressions were used to remove any irregular character to Arabic language from the data including but not limited to English characters, question marks, dashes, dollar signs, asterisks, etc. Fig. 2 shows the regex we used to remove all the strange characters from the data. After that we tokenized the input and output text into their letters and diacritization (character wise tokenization). The character tokenization was done to make the model predicting the diacritic per character not per word. Following tokenization, we padded the sequences to a maximum length which we chose to be half of the maximum length and because of limitations on computational power. After the padding, we extracted all the unique tokens from the input text and output diacritized text which was 36 (Arabic characters count) and 44 characters (Characters + diacritics) respectively. The strength of the regex we used was the main reason to be able to extract the unique characters of the data and to make it ready for use. We then used these unique characters to do one hot encoding of the data to be able to use them for deep learning training. Fig. 3 depicts the steps of the text preprocessing stage.

Regex	[?|$|.|!|:|%|#|^|@|~|،|؛|؟|-|"|(|,|)|\xa0|/|_|-|-|-|_|-]

Fig. 2. Regex symbols used in this work to clean data.

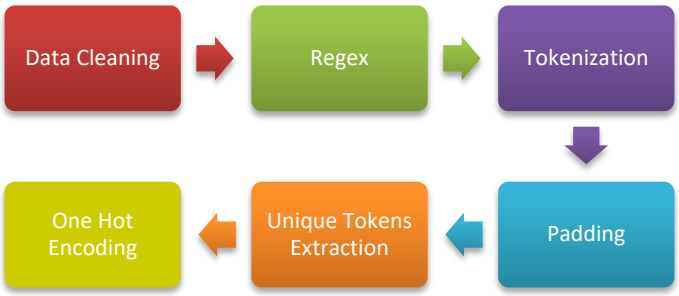

Fig. 3. The basic steps of text preprocessing stage.

B. Sequence-to-Sequence LSTM Model

After the text preprocessing stage, a sequence-to-sequence language model was built. The input to the model was the padded sequence one hot encoding of characters obtained from the text preprocessing stage. The sizes of the input data array and output data array were (4000, 4994, 36) and (4000, 9124, 44) respectively. The 4000 represents the total number of sentences we had that is why it is fixed in both input and output arrays. 4994 and 9124 indicate the padding length of the inputs and outputs respectively (half of the maximum length of both). Lastly, the 36 and 44 represent the number of unique tokens in the input and output respectively. The sequence-to-sequence model can generally be seen as encoder-decoder model. The encoder model has an input layer and an LSTM layer of 200 nodes. The decoder model has also input layer, LSTM layer of 200 nodes and output dense layer of the 44 output nodes (the unique output tokens). The activation functions used was SoftMax in the output layer. The model has in total 394,444 trainable parameters. We used Adam optimizer, and categorical_cross_entropy loss. 25% of the data were used for validation with 200 epochs. We used LSTM layer because of its ability to capture long sequence dependencies which is of great help in language modeling. The results obtained are discussed in the next section.

V. RESULTS AND DISCUSSION

This section demonstrates the experimental work done to evaluate our model. We used Google Colab GPU using a high RAM station to train the model. Training the 200 epochs took more than 6 hours. Due to the very long time of training, we used the first 50 epochs to show the accuracy results that was 92.12. However, after training the model for the complete 200 epochs later, we obtained an accuracy rate of 96.83%. The accuracy was a good metric for testing the model predictions

because the padding length didn't use lots of zeros because most of the data were concentrated in that length. However, we also used loss as an evaluation metric. The Model loss for the first 50 epochs is recorded in the figures below (Fig. 4). It reached below 0.19 for training and below 0.22 for validation. After the end of the 200 epochs, the loss obtained was 0.074 and the validation was 0.080. It seems clearly that the model isn't overfitting and obtaining excellent results.

To test the model, we needed to convert the sentence into all its corresponding characters and feed them into the trained model and recollect the characters predicted to get the output correct sequence. The results are clearly outperforming previous methods and are a good potential for the use of sequence-to-sequence modeling for automatic text diacritization. We can see that the main reasons for obtaining these excellent results are due to the correct data cleaning/preprocessing using regex and ell the text preprocessing steps. In addition, the padding length does not allow the use of lots of zeros in the sequence the thing that made the data represented in an excellent manner. In addition, the size of the validation data along with the absence of correlation between the features (because of one-hot encoding) allowed the model not to be overfitting. One of the main strengths of this work also, was that we extracted the exact unique tokens of both input and output which allowed the sequence-to-sequence model to perform excellently and outperform preceding sequence-to-sequence models. In the most recent research in literature RNNs and LSTMs were used but without the text preprocessing steps and size settings, they failed in obtaining high results using sequence-to-sequence models due to their inability to extract the exact unique tokens of diacritics and Arabic language.

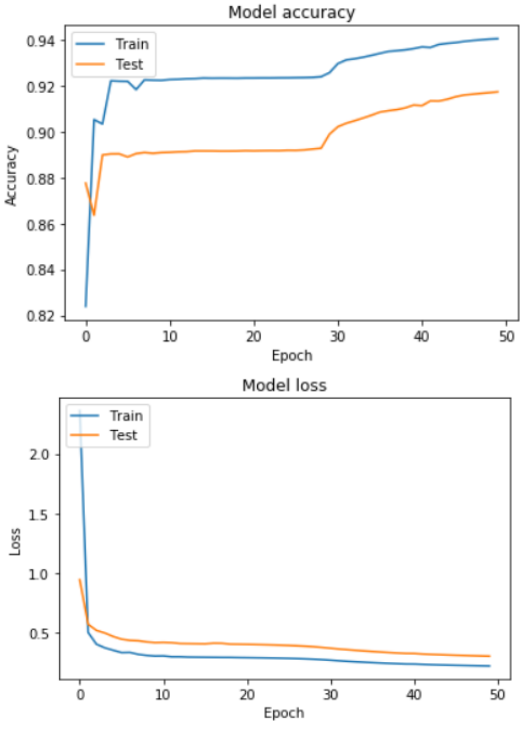

Fig. 4. The Accuracy/Loss vs epochs for the model training for the first 50 epochs.

VI. Conclusion and Future Work

This work achieved a high accuracy rate and low loss rate that outperforms results found in literature using sequence-to-sequence model to diacritrize Arabic text. The main strength points in this work compared to related work in literature are:

1) The coverage of all Arabic diacritics rather than focusing on certain diacritics.

2) Diacritizing the whole word rather than diactrizing the end of it, and

3) The good text preprocessing and the extraction of unique tokens before using the LSTMs to diacritize the un-diacritized text.

As a future work, we are planning to gather more data for further evaluation of our proposed model. We are also planning for using transfer learning to evaluate different deep learning models applied for the same problem.

References

[1] Ahmed Emam, "Arabic Sentiment Analysis: A Survey", International Journal of Advanced Computer Science and Applications, December 2015.

[2] Jezia Zakraoui, Saleh Moutaz, Somaya Al-ma'adeed, and Jihad Mohamad Aljaam "Arabic Machine Translation: A Survey With Challenges and Future Directions", IEEE Access, PP(99), December 2021.

[3] Ahmed Abdelali, Hamdy Mubarak, Younes Samih, Sabit Hassan, and Kareem Darwish "Arabic Dialect Identification in the Wild", Arabic natural langauge processing tools, May 2020.

[4] Chadi Helwe and Shady Elbassuoni, "Arabic named entity recognition via deep co-learning", Artificial Intelligence Review Journal, Springer, June 2019.

[5] Mariam Biltawi, Sara Tedmori, and A. Awajan, "Arabic Question Answering Systems: Gap Analysis", IEEE Access, 2021.

[6] Samira Lagrini and Redjimi Mohamed, "A New Approach for Arabic Text Summarization", The The Fourth International Conference on Natural Language and Speech Processing (ICNLSP), February 2021.

[7] Aly Farghaly and Khaled Shaalan, "Arabic Natural Language Processing: Challenges and Solutions", ACM Transactions on Asian Language Information Processing, Vol. 8, No. 4, Article 14, Pub. date: December 2009.

[8] Nizar Y. Habash, "Introduction to Arabic natural language processing", Synthesis Lectures on Human Language Technologies, 2010.

[9] Abdulhadi Shoufan and Sumaya Alameri, "Natural Language Processing for Dialectical Arabic: A Survey", Proceedings of the Second Workshop on Arabic Natural Language Processing Conference, 2015.

[10] Diab, M., Ghoneim, M., & Habash, N. "Arabic diacritization in the context of statistical machine translation", In Proceedings of MT-Summit.In Proceeding of the MT-Summit, Copenhagen, Denmark. 2007.

[11] Abo Bakr H. M., Shaalan K., and Ziedan I., "A Statistical Method for Adding Case Ending Diacritics for Arabic Text", The Eighth Conference on Language Engineering, ESOLEC'2008, Page 225--234, Cairo, Egypt, December 17--18 2008.

[12] Bouamor, H., Zaghouani, W., Diab, M., Obeid, O., Oflazer, K., Ghoneim, M., & Hawwari, "On Arabic Multi-Genre Corpus Diacritization", Qatar Foundation Annual Research Conference Proceedings, Issue 1 (Vol. 2016, No. 1, p. ICTPP2921), Hamad bin Khalifa University Press (HBKU Press), 2016.

[13] Darwish, K., Mubarak, H., & Abdelali, A. "Arabic diacritization: Stats, rules, and hacks", Proceedings of the Third Arabic Natural Language Processing Workshop (pp. 9-17), 2017.

[14] Fashwan, A., and Alansary, S., “SHAKKIL: an automatic diacritization system for modern standard Arabic texts”. Proceedings of the Third Arabic Natural Language Processing Workshop (pp. 84-93), 2017.

[15] Alosaimy, Abdulrahman, and Eric Atwell. "Diacritization of a Highly Cited Text: A Classical Arabic Book as a Case.", 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR). IEEE, 2018.

[16] Abdelali, A., Attia, M., Samih, Y., Darwish, K., & Mubarak, H., “Diacritization of maghrebi arabic sub-dialects”, arXiv preprint arXiv:1810.06619, 2018.

[17] Fadel, A., Tuffaha, I., Al-Jawarneh, B., & Al-Ayyoub, M., “Neural Arabic Text Diacritization: State of the Art Results and a Novel Approach for Machine Translation”, arXiv preprint arXiv:1911.03531, 2019.

[18] M. Madhfar and A. M. Qamar, “Effective Deep Learning Models for Automatic Diacritization of Arabic Text”, IEEE Access Journal, 2021.

[19] Batool Abuali and Mohamad-Bassam Kurdy, “Full Diacritization of the Arabic Text to Improve Screen Readers for the Visually Impaired”, Journal of Advances in Human-Computer Interaction, 2022.

[20] Brian Thompson and Ali Alshehri, “Improving Arabic Diacritization by Learning to Diacritize and Translate”, Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022), pages 11 – 21, May 2022.

[21] Mona Diab, Nizar Habash, Owen Rambow, and Ryan Roth, “Ldc arabic treebanks and associated corpora: Data divisions manual”, arXiv preprint arXiv:1309.5652., 2013.