How Much Does Lookahead Matter for Disambiguation?

Partial Arabic Diacritization Case Study

A thesis

submitted in partial fulfilment

of the requirements for the Degree of

Master of Science

by

Saeed Esmail

This research has been carried out under the supervision

of

Dr. Kfir Bar &

Prof. Nachum Dershowitz

Tel Aviv University

Blavatnik School of Computer Science

March 2021

Abstract

We suggest a model for partial diacritization of deep orthographies, focusing on Arabic as a case study. Our partial diacritizer restores short vowels only when they contribute to the ease of understandability of a given running text. Our model is based on two independent neural networks, one that takes the entire sentence as an input, and another that considers as input only the text that has been read so far. Partial diacritization is achieved by keeping only those vowels on which the two networks disagree. For evaluation, we measure the contribution of the restored short vowels to an Arabic-to-English neural machine translation system; our model shows a 1.36% improvement in translation quality over a baseline model. We suggest a novel criterion for partial diacritization, viz. just enough to obviate the need for lookahead for disambiguation. Additionally, we argue how much lookahead context is required for resolving ambiguities in reading.

Acknowledgments

”مَنْ عَلَّمَني حَرْفًا صِرْتُ لَهُ عَبدًا“

“For him, who has taught me a single letter, I will be a servant.”

First and foremost, I wish to express my deepest gratitude to my advisors Dr. Kfir Bar and Prof. Nachum Dershowitz for believing in me to accomplish this work, for their dedicated work and guidance across all this journey, I have learned a lot from them.

Their patience, motivation, and immense knowledge had made me enjoy and learn a lot during this research. I could not have imagined having better advisors and mentors for my M.Sc. study, and without their persistent help, the goal of this thesis would not have been realized.

I also want to thank my family for their continuous and unparalleled love, help and support.

Table of Contents

List of Figures

List of Tables

1

Introduction

Ambiguity is part and parcel of natural language. It may manifest itself at the morphological level, the syntactic, or at higher linguistic levels. For example, in the classic "garden path" sentence, "The old man the boat", "old" can be a noun or adjective, while "man" may be a noun or verb. The point is that the prima facie more likely reading of "old man" as adjective-noun is found to be untenable by the end of the sentence, and the reader must retrace her steps and reinterpret the morphology and syntax to understand the intended meaning. Though ambiguity may be deliberate—as in poetry—it is usually desirable to keep it to a minimum. Classical Greek and Latin were often written scripta continua, sans interword spaces. Likewise, many Eastern languages do not normally use spaces or punctuation within sentences. This, too, introduces a level of ambiguity, which partial punctuation could help resolve.

We deal here with ambiguity at the morphological level, investigating the inclusion of optional disambiguating diacritics. We suggest a novel criterion for partial diacritization, viz. just enough to obviate the need for lookahead for disambiguation—to the extent possible. In other words, disambiguating diacritics are called for when the most likely interpretation—considering only what precedes in reading order—is erroneous.

Semitic languages form a branch of languages originating in the Middle East and include, among others, Arabic, Hebrew, and Aramaic. Most of the writing systems (*orthographies*) of those languages omit some or all vowels from their alphabet. Daniels and Bright [23], in their sixfold classification of writing systems, call such scripts *abjads*. The missing vowels are typically covered by a set of diacritics, serving

as a phonetic guide, which tend to be omitted in standard writing. Full or almost-full vocalization is normally reserved for scripture and other archaic works, verse, works for children or beginners, and for loan words or foreign names. In Arabic, for example, there are a number of such short-vowel diacritics, collectively named *harakat* حَرَكَات. Long vowels are represented by a collection of *matres lectionis*, letters that otherwise serve as consonants (*alif*, *waw*, *ya*).

A common characterization in modern psycholinguistics is that Arabic and Hebrew have *deep* orthography since there is no one-to-one mapping between phonemes and graphemes. At the opposite end of the spectrum are languages with *shallow* (or *phonemic*) orthographies, such as Finnish and Maltese (a Semitic language), for which it is usually easy to pronounce any word given its letters. Arabic orthography is considered shallow when short vowels are present [11]. But, when they are omitted, a reader needs to use some contextual information to resolve ambiguities in pronunciation and meaning. In English, diacritics are optional (except in some borrowed words and expressions such as "coup d'état") and rarely used today (diaeresis on "naïve"; circumflex on "rôle"); they may indicate pronunciation but are not needed for understanding. Also, some commas and hyphens are optional punctuation in English, but can help one parse the sentence properly. In many other languages, however, diacritics are essential and are never omitted. Anecdotally, even native speakers of such languages resort to a spellchecker to insert them ex post facto and save keystrokes thereby.

Overall, a fully vowelized Arabic text is considered too complicated to read easily. On the other hand, the lack of written short vowels in certain words, particularly homographs (strings that have multiple pronunciations or meanings), may be detrimental to the ease of understandability. Moreover, to resolve such pronunciation ambiguities, it is often enough to add only one short vowel. For example, the word اكتشفت has a number of pronunciations with different meanings (e.g., "I discovered", "You discovered", "It/she was discovered"), but with only one short vowel (*dammah* /u/) added, reading اكتُشفت ("It") becomes easier. This example illustrates that, in Arabic, the voice of a verb is distinguished only by short vowels. In many cases, deciding on the correct pronunciation of a word requires looking at the following words in the text, and not only at preceding ones. We claim that, when only information from prior words is needed to resolve any ambiguity of a given word, then the short vowels may be safely omitted, since by the time that word is encountered, the reader has already collected what is necessary for disambiguation. In the following example, the first words are identical, the second word جد *jd* in both sentences takes a different diacritic on the last letter, which results in a completely different meaning. Only with the third

word is a reader able to resolve the ambiguity of the second one:

مَنْ جَدَّ وَجَدَ

Transliteration: mano jad~a wajada

Translation: who works hard succeeds

مَنْ جَدُّ الوَلَدِ؟

Transliteration: mano jad~u alwaladi?

Translation: who is the grandfather of the boy?

The high level of ambiguity in Arabic results in having about 12 analyses per word on average. Table 1.1 highlights this richness.

In some Arabic media outlets (e.g., *Nature* in Arabic [`arabicedition.nature.com`]), *partial* diacritization is used to facilitate understandability. It has been claimed [37] that to handle ambiguities as in garden-path sentences, which make understandability during reading more difficult, it is necessary to parse natural language either nondeterministically or by a deterministic parser with *lookahead* (LR(k)) capabilities.

We propose a machine-learning model for partial diacritization. Deciding which vowels to recover is achieved by mimicking the way a human resolves pronunciation ambiguity. We train two networks for full vowelization, one taking the entire sentence into account, and another that considers only prior words. Partial diacritization is achieved by preserving those vowels that the two networks *disagree* on, suggesting that without them disambiguation would require lookahead.

Diacritized	**POS**	**English**
jad~	Noun	Grandfather
jad~	Verb	Became lucky
jad~a	Verb	Toil (Work hard)
jid~	Noun	Earnestness
jud	Verb	Be generous
jid	Verb	Find

Table 1.1: A subset of all the possible analyses for the two letters word جد *jd*

2

Related Work

Comparing reading processes in languages of different orthography depths is an active area of research [34, 28, 33]. Specifically, the contribution of short vowels to reading of Arabic has been studied. Whereas several studies report a positive contribution [4, 5, 6, 7, 9, 8, 10, 3, 43], a number [32, 18] have shown a decrease in reading fluency (measured as the time to correctly read a text) and accuracy (the percentage of words correctly pronounced), due to the visual load and complexity of short-vowel diacritics in Arabic. A recent review [12] summarizes the conflicting results.

Various works apply deep neural networks to diacritic restoration. Examples include [39] for Polish; [42] for Romanian; [30] for Slovak; [44] for Turkish; and [41, 31, 40, 15] for Vietnamese.

There is a large body of work on full Arabic diacritization. Early works took a more traditional machine-learning approach [49, 24]; recent efforts are usually based on deep neural setups [2, 13, 25, 27, 38, 16, 19, 1]. A few works [47, 17] show the contribution of morphological data to diacritization. Recently, an encoder-decoder network using a Tacotron CBHG module [46] as part of the encoder was introduced [36].

Fadel et al. [27] developed a deep recurrent neural network for diacritization, which was reported to positively contribute to neural machine translation (NMT), by encoding the diacritics on a parallel layer to the input characters. In our work, we use their NMT architecture for evaluation of our model for partial diacritic restoration.

Recently, Alqahtani et al. [14] evaluated the contribution of incomplete restoration of Arabic diacritics to

a number of downstream tasks. Estimating the errors introduced by a full diacritization algorithm, their approach is to restore the diacritics only for ambiguous words, which is what they refer to as *selective* diacritic restoration. In our work, we focus more on improving understandability during reading, as opposed to improving the accuracy of a downstream-task algorithm. Our goal it to restore diacritics only for letters (not necessarily all letters of a word) that resolve ambiguities during reading of a running text. To the best of our knowledge, this is the first time that this goal is being addressed. We are unaware of the existence of relevant resources that may help us train a supervised machine-learning algorithm for partial diacritic restoration. Bouamor et al. [20] conducted a study of human annotation for minimal diacritization, which shows a low inter-annotator agreement and how subjective this task can be.

3

Methods and Results

3.1 Data

Tashkeela Corpus

To train both models, we use the Tashkeela corpus [48], comprising more than 75M words, fully diacritized. It mostly contains classic works written in Classical Arabic (CA), the forerunner of Modern Standard Arabic (MSA), the main language used today in formal settings (in contradistinction to spoken Arabic, with its many mutually-unintelligible regional varieties) and what we interested in. MSA and CA have much in common, but oftentimes they use different grammatical structures and vocabularies. The MSA texts that were extracted from the Internet represent 1.15% of the corpus, while the major part was collected from Shamela Library, which represents the other 98.85% of the corpus, obtained from 97 books, collected mainly from Islamic classics. Also, it is worth noting that Tashkeela is one of two datasets that are used by the majority of works, and the only freely available corpus. The second dataset is LDC's Arabic Treebank [35].

Preprocessing

We preprocess the corpus in a way similar to [25], for solving the following issues:

- Whitespaces separating ending diacritics from their word.
- Non-Arabic characters with misplaced diacritics.
- Multiple diacritcs for single letter.
- Non-diacritized files/lines.

	CA and HQ		MSA	
	Original	**Clean**	**Original**	**Clean**
Words	66.5M	65.8M	801K	604K
Lines	1.7M	1.5M	50K	20K

Table 3.1: Word and line counts of Tashkeela corpus, before and after cleaning, broken down into CA and MSA.

- HTML tags/URLs/English letters.

additionally, we replace a few rare letters by their natural equivalents (e.g., Farsi yeh ى into Arabic yeh ي and Farsi peh پ into Arabic beh ب). Furthermore, in Tashkeela, not all letters carry diacritics. Since we train our model one line at a time, we delete from the corpus lines that have a low rate of diacritics per letter to maintain relatively high support of all labels. Fadel et al. [25] removed lines below a rate of 80%. Since it appears that MSA has a lower rate than CA, we remove lines below 50%, to allow more MSA during training. We added the Holy Quran (HQ), fully diacritized 6,236 lines (157,935 words), to the CA corpus. Table 3.1 summarizes the number of words and lines we have in our corpus. Also, the alphabet was extended to include not only Arabic letters, but Arabic numbers (0,1,2,…), Eastern Arabic Numbers (...١،٢،٣), and Arabic punctuation marks. Other characters were replaced by a special non-letter character. Overall, we have 83 characters.

Tweets Corpus

Additionally, we collected 75 Twitter tweets[1] (1,075 words) in MSA, taken from official accounts that cover news in politics, health, sports, technology and Literature. Tweets were fully diacritized by a native linguist, and then manually processed by another native speaker who kept only about 25% to achieve fluency. It was used to test both full and partial models on MSA with different distribution. *We make this dataset publicly available.*

[1] `www.twitter.com`

3.2 Method

Partial diacritization is the process of inferring a minimal subset of diacritics that is fundamental to disambiguate the context. Yet, this mission is not well configured and there is no convention or explicit rules how to accomplish it. We distinguish between ambiguous words that may be resolved using *previously* seen context, and ambiguous words that need some of the context that follows in order to improve resolution while reading. The former are easier to resolve when reading; therefore, we try to restore the diacritics only for letters for which readers often need the context that follows.

To imitate this, we train separately two neural models for full restoration: One encodes information obtained in reading direction, not crossing the predicted word, ignoring what comes after. The second scans the entire sentence before diacritizing it in full; therefore, it is assumed that this model has a better chance to predict the correct diacritic. The idea is to provide pronunciation hints to the reader only when they cannot be trivially decoded using the content that has already been attended by the first unidirectional neural network. We represent the source data as a sequence of characters, and the target data as a sequence of diacritics; this representation reduces vocabulary size and avoids OOV ("out of vocabulary") words.

Reading-Direction We use a four-layer unidirectional Long Short-Term Memory (LSTM) [29] architecture that works on the character level, and predicts one label per input character. Like in previous works, we use the following set of labels, to account for most of the diacritic types. Essentially, we cover the vowels: *fatḥah* /a/ ◌َ, *kasrah* /i/ ◌ِ, *ḍammah* /u/ ◌ُ; *tanwin*s (nunations) to indicate case ending: ◌ً /an/, ◌ٍ /in/, ◌ٌ /un/; *shaddah* ◌ّ for gemination; and *sukūn* to indicate vowel absence. We add 6 labels for capturing combinations of *shaddah* vowels or nunations, and finally a label indicating no diacritics. Overall, we have 15 labels. Each input character is encoded with a non-pretrained embeddings vector concatenated with the embeddings of the containing word. Word-level embeddings are the output of another bi-directional LSTM (BiLSTM) that works on the word's characters. More formally, let word j be composed of n letters $l_1, l_2, \ldots, l_n$. Then, o_i is the encoded version of l_i:

$$
\begin{aligned}
c_i &= \mathrm{EMB}(l_i) \\
w_j &= \mathrm{BiLSTM}(c_1, c_2, \ldots, c_n) \\
o_i &= \mathrm{LSTM}([c_i; w_j]),
\end{aligned}
$$

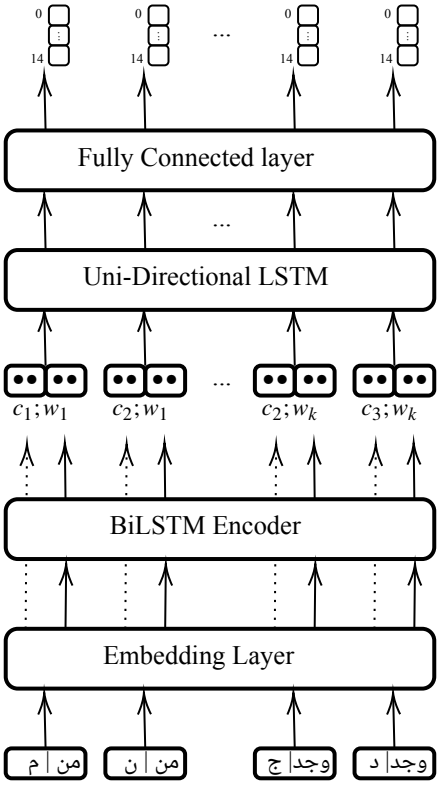

Figure 3.1: The Reading-Direction architecture, with a character embedding layer and word encoder.

where w_i is the concatenation of the two last outputs from both of the BiLSTM's sides. Every o_i is then sent to a fully-connected layer for generating the final prediction.

For the reading-direction model, we use 15% of the processed data for validation and the rest for training. Our best results are achieved by using a hidden size of 16 for the word-character BiLSTM, resulting in 32-dimensional vector for word embedding, which we concatenate with another 32-dimensional vector representing a character. This 64-dimensional vector is the input for the 4-layer unidirectional LSTM with hidden size of 512, followed by a fully connected layer of 15 output labels. We use dropout with 20% drop probability and the Adam optimizer, configured with a learning rate of 10^{-4}. Batch size is 512, We trained the model for 10 epochs and reached optimal results on a validation test after 8.

Full-Sentence To encode a full sentence before classification, we experimented with a few architectures; first attempt was to use similar architecture to Reading-Direction described above, but with a Bi-Directional LSTM instead. Next attempt was a Transformer [45] working on a character level. The

Transformer consists of a stack of 6 encoders, followed by a stack of 6 decoders. Each layer of the stacks composed of 8 attention heads, and hidden size of 512 delivered the best accuracy for full diacritics restoration.

For training both models, we split the data into lines, then we use sentences of up to 100 characters (excluding diacritics), longer sentences are handled during training and testing by using a 100-character overlapping window that moves forward one word at a time. During testing, every character gets a number of predictions, one per window. The final prediction is done by maximum voting.

We use 10%-drop and ReLU for the dropout and activation layers that follow each multi-head attention and linear layers in encoder/decoder components. Training first on CA and then fine-tuning on MSA. The transformer model reaches optimality on CA after 5 epochs, and on MSA after 6 epochs.

Automatic-Evaluation As mentioned before, partial diacritization of a word aiming to make it readable and unambiguous is a subjective task, and relies on reading comprehension ability and eloquence of the reader. Therefore, we make an attempt at designing a new automatic evaluation metric: Our first attempt was to use Google's Arabic-to-English machine translation, hoping it would benefit from diacritics provided with the input text. Unfortunately, we could not identify a meaningful difference in the translation quality, even when we tried different computational approaches for comparing the English translations, like Pre-trained InferSent [22] and the Google Sentence Encoder [21].

The next attempt uses Arabic morphological analyses, which given a sentence return for each word its all possible prefix-stem-suffix segmentations. we tried given our partial diacritization to keep only relevant analyses with respect to diacritization, and measure the difference by counting the number of analyses that differ.

Eventually, we decided to use Translation-Over-Diacritization (TOD) [26]. The basic idea of TOD is to use basic Encoder-Decoder sequence to sequence (seq2seq) model with additive attention, to train two identical Arabic-to-English machine translations, with and without Arabic diacritization. We extend this idea to include also partial diacritization. The dataset contains 1M Arabic-English sentence pairs.

3.3 Results

Full Restoration To compare our transformer model with the state-of-art systems, we preprocess the dataset and use the same train/test split as did Fadel et al. [25]. As customary, we measure:

1. Diacritic error rate (DER): the percentage of misclassified letters.
2. Word error rate (WER): the percentage of words with at least one misclassified letter. (Foreign-script characters and digits are ignored for both.)

Generally speaking, case endings are deemed more difficult than basic vowels, since they are syntax related; therefore, we also report results when discounting case-ending mistakes. Similarly, we report on results also while ignoring mistakes in predicting the "no diacritic" label. Table 3.2 compares the results of our transformer model for full diacritization with the best known models, under different conditions as reported in previous works. Our transformer model is on par with the state of art.

DER / WER	**Including 'no diacritic'**		**Excluding 'no diacritic'**	
	w/ case ending	**w/o case ending**	**w/ case ending**	**w/o case ending**
Fadel et al. [27]	2.60 / 7.69	2.11 / 4.57	3.00 / 7.39	2.42 / 4.44
AlKhamissi et al. [13]	**1.83** / **5.34**	**1.48** / **3.11**	**2.09** / **5.08**	**1.69** / **3.00**
Our Full-Sentence Model	3.57 / 8.52	2.32 / 5.44	3.42 / 8.26	2.23 / 5.32

Table 3.2: DER / WER results (in %) on the Tashkeela test set, as defined by [25]. Results are reported under different conditions of case endings and when including or excluding the "no diacritic" label.

Partial Restoration Since we care more about MSA than CA, we use a lower cleaning threshold (50%) and generate a new 85/15% train/validation split for the simple reading-direction model (first row of Table 3.3). Following others, we evaluate our models under varying conditions as before. To improve performance of our full-sentence transformer model on MSA, we split the data differently, and use only MSA texts for validation—about 10% of the corpus. Then we train the model in two phases: (1) pre-training with CA+HQ texts, and then (2) fine-tuning with MSA texts, to end with weights that handle MSA better than CA. This fine-tuning training style gave an improvement of 1.5% in word error over the same model that was trained on the entire training set in one phase. The second row in Table 3.3 shows the final results of the two-phase fine-tuned model on the MSA-only validation set. Finally, we evaluate both models, the simple reading-direction model as well as the full two-phase model, on our MSA-only fully-diacritized tweets (last two rows):

	Model	w/ case	w/o case	w/ case	w/o case
		w/ "no diacritic"		w/o "no diacritic"	
Valid.	Reading	13.0/34.0	9.3/26.1	8.3/25.4	5.3/16.9
	Sentence	6.9/26.6	7.0/23.0	6.1/16.0	5.5/11.5
Tweet	Reading	16.2/47.5	11.1/33.1	14.0/44.9	9.5/30.3
	Sentence	**9.2/27.7**	**6.3/18.1**	**7.2/23.6**	**5.0/15.0**

Table 3.3: Error rates DER/WER (in %) on MSA test set.

Both models are used for generating partial diacritics as part of our model-difference approach. To evaluate, we match the predicted partial diacritics with those manually assigned to the tweets by a native speaker. Our system decided to keep about 12% of the restored diacritics, while the native speaker kept 25%. For a baseline, we randomly select 12% of the manually assigned diacritics. Table 3.4 shows the improvement achieved by our model-difference method:

Diacritization	Kappa IAA	F1
Random partial restoration	0.11	0.77
Our partial restoration	**0.26**	**0.82**

Table 3.4: Results of partial restoration on tweets. Metrics are Kappa coefficient for the 15-label task, and F1 for the diacritic assignment task, that is, checking if a letter is assigned a diacritic or not.

Fadel et al. [27] suggested evaluating diacritic restoration by measuring their contribution to a diacritics-sensitive Arabic-to-English NMT system. We train the same NMT system on 1M sentence pairs, for which we restore diacritics with our full-sentence transformer model, and evaluate it on a test set under different conditions of diacritic assignment. The evaluation results are shown in 3.5, This shows a small increased accuracy with our partial diacritic restoration, which kept about 10% of the diacritics, over the two baselines. MT benefits more, of course, when the input comes with all diacritics.

Diacritization	BLEU Score
No Diacritics	33.48
Random Partial Diacritization	33.34
Our Partial Diacritization	33.75
Full-Sentence Full Diacritization	34.25

Table 3.5: BLEU scores of an Arabic-to-English TM, using different levels of diacritics.

The Contribution of Lookahead To measure the contribution of forward-looking context to reading, we ran additional experiments with the transformer model, placing successively larger limits on the num-

ber of words following the current word that the transformer may encode. Table 3.6 summarizes DER and WER on the tweets dataset for each instance; the limit on lookahead is indicated in the first column. The inference was done on the same trained full-sentence model (with unrestricted lookahead), under all evaluation conditions, the transformer model benefits from more and more lookahead in order to fully diacritize the current word. For future work, we suggest to train separate models for each restriction.

Lookahead	**Including 'no diacritic'**		**Excluding 'no diacritic'**	
	w/ case ending	**w/o case ending**	**w/ case ending**	**w/o case ending**
0	13.70 / 43.81	7.87 / 23.62	11.63 / 39.96	6.49 / 20.40
1	9.06 / 27.47	6.33 / 18.63	7.07 / 23.31	5.01 / 15.50
2	8.89 / 27.06	6.22 / 18.42	6.88 / 23.00	4.89 / 15.30
3	8.83 / 26.53	6.24 / 18.31	6.80 / 22.48	4.91 / 15.19
4	8.77 / 26.43	6.18 / 18.21	6.74 / 22.37	4.85 / 15.09

Table 3.6: DER / WER results of the full-sentence transformer on the tweets dataset, under different lookahead windows.

4

Discussion

We propose a criterion for partial diacritization of Arabic and implement it as a combination of two neural networks that restore diacritic marks in full. One uses only context already read, and one benefits from seeing the entire sentence prior to prediction. For evaluation, we manually diacritized a set of tweets written in MSA and then selectively marked those diacritics that contribute most to disambiguation during reading. Using this dataset, as well as a diacritic-sensitive NMT system, we found our model-difference approach to be superior to the baseline method. We also quantify the impact of lookahead window size on disambiguating pronunciation—measured by correctness of diacritics; the density of automatic partial vowelization of our method could be adjusted to obviate only more distant lookahead. We suggest a novel criterion for partial diacritization, viz. just enough to obviate the need for lookahead for disambiguation We chose Arabic here only as a convenient case study; we plan to expand the method to additional languages.

Bibliography

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

[2] Hamza Abbad and Shengwu Xiong. Multi-components system for automatic Arabic diacritization. In *European conference on information retrieval*, pages 341–355. Springer, 2020.

[3] Bashir Abu-Hamour, Hanan Al-Hmouz, and Mohammed Kenana. The effect of short vowelization on curriculum-based measurement of reading fluency and comprehension in Arabic. *Australian Journal of Learning Difficulties*, 18(2):181–197, 2013.

[4] Salim Abu-Rabia. Learning to read in Arabic: Reading, syntactic, orthographic and working memory skills in normally achieving and poor Arabic readers. *Reading Psychology: An International Quarterly*, 16(4):351–394, 1995.

[5] Salim Abu-Rabia. The role of vowels and context in the reading of highly skilled native Arabic readers. *Journal of Psycholinguistic Research*, 25(6):629–641, 1996.

[6] Salim Abu-Rabia. The need for cross–cultural considerations in reading theory: The effects of Arabic sentence context in skilled and poor readers. *Journal of Research in Reading*, 20(2):137–147, 1997.

[7] Salim Abu-Rabia. Reading in Arabic orthography: The effect of vowels and context on reading accuracy of poor and skilled native Arabic readers in reading paragraphs, sentences, and isolated words. *Journal of Psycholinguistic Research*, 26(4):465–482, 1997.

[8] Salim Abu-Rabia. Attitudes and culture in second language learning among Israeli-Arab students. *Curriculum and Teaching*, 13(1):13–30, 1998.

[9] Salim Abu-Rabia. Reading Arabic texts: Effects of text type, reader type and vowelization. *Reading and Writing*, 10(2):105–119, 1998.

[10] Salim Abu-Rabia. The effect of Arabic vowels on the reading comprehension of second-and sixth-grade native Arab children. *Journal of Psycholinguistic Research*, 28(1):93–101, 1999.

[11] Salim Abu-Rabia. The role of vowels in reading Semitic scripts: Data from Arabic and Hebrew. *Reading and Writing*, 14(1):39–59, 2001.

[12] Salim Abu-Rabia. The role of short vowels in reading Arabic: A critical literature review. *Journal of Psycholinguistic Research*, 48(4):785–795, 2019.

[13] Badr AlKhamissi, Muhammad N ElNokrashy, and Mohamed Gabr. Deep diacritization: Efficient hierarchical recurrence for improved Arabic diacritization. *arXiv preprint*, 2020. arXiv:2011.00538.

[14] Sawsan Alqahtani, Hanan Aldarmaki, and Mona Diab. Homograph disambiguation through selective diacritic restoration. *arXiv preprint*, 2019. arXiv:1912.04479.

[15] Sawsan Alqahtani, Ajay Mishra, and Mona Diab. Efficient convolutional neural networks for diacritic restoration. In *EMNLP*, 2019.

[16] Sawsan Alqahtani, Ajay Mishra, and Mona Diab. Efficient convolutional neural networks for diacritic restoration. In *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[17] Sawsan Alqahtani, Ajay Mishra, and Mona Diab. A multitask learning approach for diacritic restoration, 2020.

[18] Ibrahim A Asadi. Reading Arabic with the diacritics for short vowels: vowelised but not necessarily easy to read. *Writing Systems Research*, 9(2):137–147, 2017.

[19] Yonatan Belinkov and James Glass. Arabic diacritization with recurrent neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2281–2285, 2015.

[20] Houda Bouamor, Wajdi Zaghouani, Mona Diab, Ossama Obeid, Kemal Oflazer, Mahmoud Ghoneim, and Abdelati Hawwari. A pilot study on Arabic multi-genre corpus diacritization. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 80–88, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3209. URL `https://www.aclweb.org/anthology/W15-3209`.

[21] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder, 2018.

[22] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data, 2018.

[23] Peter T. Daniels and William Bright. *The World's Writing Systems*. Oxford University Press on Demand, 1996.

[24] Kareem Darwish, Hamdy Mubarak, and Ahmed Abdelali. Arabic diacritization: Stats, rules, and hacks. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 9–17, 2017.

[25] Ali Fadel, Ibraheem Tuffaha, Mahmoud Al-Ayyoub, et al. Arabic text diacritization using deep neural networks. In *2019 2nd international conference on computer applications & information security (ICCAIS)*, pages 1–7. IEEE, 2019.

[26] Ali Fadel, Ibraheem Tuffaha, Bara' Al-Jawarneh, and Mahmoud Al-Ayyoub. Neural Arabic text diacritization: State of the art results and a novel approach for machine translation. In Toshiaki Nakazawa, Chenchen Ding, Raj Dabre, Anoop Kunchukuttan, Nobushige Doi, Yusuke Oda, Ondrej Bojar, Shantipriya Parida, Isao Goto, and Hidaya Mino, editors, *Proceedings of the 6th Workshop on Asian Translation, WAT@EMNLP-IJCNLP 2019, Hong Kong, China, November 4, 2019*, pages

215–225. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-5229. URL https://doi.org/10.18653/v1/D19-5229.

[27] Ali Fadel, Ibraheem Tuffaha, Bara' Al-Jawarneh, and Mahmoud Al-Ayyoub. Neural Arabic text diacritization: State of the art results and a novel approach for machine translation. *arXiv preprint*, 2019. arXiv:1911.03531.

[28] Ram Frost, Leonard Katz, and Shlomo Bentin. Strategies for visual word recognition and orthographical depth: a multilingual comparison. *Journal of Experimental Psychology: Human Perception and Performance*, 13(1):104, 1987.

[29] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

[30] A. Hucko and P. Lacko. Diacritics restoration using deep neural networks. In *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, pages 195–200, 2018. doi: 10.1109/DISA.2018.8490624.

[31] Bui T. Hung. Vietnamese diacritics restoration using deep learning approach. In *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*, pages 347–351, November 2018. doi: 10.1109/KSE.2018.8573427.

[32] Raphiq Ibrahim et al. Reading in Arabic: New evidence for the role of vowel signs. *Creative Education*, 4(04):248, 2013.

[33] Leonard Katz and Ram Frost. The reading process is different for different orthographies: The orthographic depth hypothesis. In *Advances in Psychology*, volume 94, pages 67–84. Elsevier, 1992.

[34] Isabelle Y Liberman, Alvin M Liberman, Ignatius Mattingly, and Donald Shankweiler. Orthography and the beginning reader. *Orthography, Reading, and Dyslexia*, pages 137–153, 1980.

[35] Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. The penn arabic treebank: Building a large-scale annotated arabic corpus. *NEMLAR Conference on Arabic Language Resources and Tools*, 01 2004.

[36] Mokhtar Madhfar and Ali Mustafa Qamar. Effective deep learning models for automatic diacritization of Arabic text. *IEEE Access*, 2020.

[37] Mitchell Philip Marcus. *A Theory of Syntactic Recognition for Natural Language*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1978.

[38] Hamdy Mubarak, Ahmed Abdelali, Hassan Sajjad, Younes Samih, and Kareem Darwish. Highly effective Arabic diacritization using sequence to sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2390–2395, 2019.

[39] Jakub Náplava, Milan Straka, Pavel Straňák, and Jan Hajič. Diacritics restoration using neural networks. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL `https://www.aclweb.org/anthology/L18-1247`.

[40] Cao Hong Nga, Nguyen Khai Thinh, Pao-Chi Chang, and Jia-Ching Wang. Deep learning based Vietnamese diacritics restoration. In *2019 IEEE International Symposium on Multimedia (ISM)*, pages 331–334, Los Alamitos, CA, December 2019. IEEE Computer Society. doi: 10.1109/ISM46123.2019.00074. URL `https://doi.ieeecomputersociety.org/10.1109/ISM46123.2019.00074`.

[41] Kiet Nguyen, Vu Nguyen, Anh Nguyen, and Ngan Nguyen. A Vietnamese dataset for evaluating machine reading comprehension. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2595–2605, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.233. URL `https://www.aclweb.org/anthology/2020.coling-main.233`.

[42] M. Nuțu, B. Lőrincz, and A. Stan. Deep learning for automatic diacritics restoration in Romanian. In *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 235–240, September 2019.

[43] Haitham Taha. Deep and shallow in Arabic orthography: New evidence from reading performance of elementary school native arab readers. *Writing Systems Research*, 8(2):133–142, 2016.

[44] Aysenur Uzun. Diacritic restoration using neural network. Technical report, Computer Engineering, Istanbul Technical University, May 2018. URL https://raw.githubusercontent.com/aysnrgenc/TurkishDeasciifier/master/diacritic-restoration-recurrent.pdf.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[46] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint*, 2017. arXiv:1703.10135.

[47] Nasser Zalmout and Nizar Habash. Joint diacritization, lemmatization, normalization, and fine-grained morphological tagging. *arXiv preprint*, 2019. arXiv:1910.02267.

[48] Taha Zerrouki and Amar Balla. Tashkeela: Novel corpus of Arabic vocalized texts, data for auto-diacritization systems. *Data in Brief*, 11:147 – 151, 2017. ISSN 2352-3409. doi: https://doi.org/10.1016/j.dib.2017.01.011. URL http://www.sciencedirect.com/science/article/pii/S2352340917300112.

[49] Imed Zitouni and Ruhi Sarikaya. Arabic diacritic restoration approach based on maximum entropy models. *Computer Speech & Language*, 23(3):257–276, 2009.

הפקולטה למדעים מדויקים
ע"ש ריימונד ובברלי סאקלר
אוניברסיטת תל אביב

כמה חשוב לחד משמעיות להביט קדימה
חקר מקרה: ניקוד ערבית חלקי

חיבור זה
הוגש כחלק מהדרישות לקבלת תואר
מוסמך אוניברסיטה

על ידי

סעיד איסמאעיל

העבודה הוכנה בהנחיית
ד"ר כפיר בר
פרופ׳ נחום דרשוביץ

אוניברסיטת תל אביב
בית הספר למדעי המחשב

אדר התש"פא

תקציר

אנו מציעים מודל לניקוד חלקי של אורתוגרפיות עמוקות, עם התמקדות בערבית. המנקד חלקי שלנו משחזר תנועות רק כאשר הן תורמות להבנת הנקרא בטקסט.
המודל שלנו מבוסס על שתי רשתות נוירונים בלתי תלויות, הרשת הראשונה מעבדת כל המשפט לפני שמתחילה לנקד, לעומת השנייה אשר לוקחת בחשבון רק הטקסט שנקרא עד עכשיו ובהתבסס עליו מנקדת את האות הנוכחית.
ניקוד חלקי מתקבל על ידי שמירת ניקוד שעליו שתי הרשתות אינן מסכימות בלבד.
לצורך בדיקת המודל, אנו מודדים את התרומה של התנועות המשוחזרות, למודל תרגום מערבית לאנגלית; המודל שלנו מראה שיפור של 1.36% באיכות התרגום בהשוואה למודל בסיסי.