How Much Does Lookahead Matter for Disambiguation? Partial Arabic Diacritization Case Study

Saeed Esmail*
School of Computer Science
Tel Aviv University, Tel Aviv, Israel
saeedesmail@mail.tau.ac.il

Kfir Bar*
School of Computer Science
College of Management Academic
Studies, Rishon LeZion, Israel
Basis Technology, MA, USA
barkfir@yahoo.com

Nachum Dershowitz*
School of Computer Science
Tel Aviv University, Tel Aviv, Israel
nachum@tau.ac.il

We suggest a model for partial diacritization of deep orthographies. We focus on Arabic, where the optional indication of selected vowels by means of diacritics can resolve ambiguity and improve readability. Our partial diacritizer restores short vowels only when they contribute to the ease of understandability during reading a given running text. The idea is to identify those uncertainties of absent vowels that require the reader to look ahead to disambiguate. To achieve this, two independent neural networks are used for predicting diacritics, one that takes the entire sentence as input and another that considers only the text that has been read thus far. Partial diacritization is then determined by retaining precisely those vowels on which the two networks disagree, preferring the reading based on consideration of the whole sentence over the more naïve reading-order diacritization.

For evaluation, we prepared a new dataset of Arabic texts with both full and partial vowelization. In addition to facilitating readability, we find that our partial diacritizer improves translation quality compared either to their total absence or to random selection. Lastly, we study the benefit of knowing the text that follows the word in focus toward the restoration of short

* All authors contributed equally.

Action Editor: Preslav Nakov. Submission received: 11 November 2021; revised version received: 19 July 2022; accepted for publication: 7 August 2022.

https://doi.org/10.1162/coli_a_00456

vowels during reading, and we measure the degree to which lookahead contributes to resolving ambiguities encountered while reading.

> L’Herbelot had asserted, that the most ancient Korans, written in the Cufic character, had no vowel points; and that these were first invented by Jahia–ben Jamer, who died in the 127th year of the Hegira.
>
> “Toderini’s History of Turkish Literature,” *Analytical Review* (1789)

1. Introduction

Ambiguity is part and parcel of natural language. It may manifest itself at the morphological level, the syntactic level, or at higher linguistic levels. For example, in the classic “garden path” sentence, “The old man the boat,” “old” can be a noun or an adjective, while “man” may be a noun or a 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.

We deal here with ambiguity at the morphological level, investigating the inclusion of optional disambiguating diacritics. We suggest a novel criterion for *partial* diacritization, namely, 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 (1996), 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, but these signs tend to be omitted in standard writing.

In Arabic, there are a number of such short-vowel diacritics, collectively named *harakat* حركات. Long vowels, on the other hand, are represented by a collection of *matres lectionis*, letters that otherwise serve as consonants (*alif*, *waw*, *ya*). Modern Hebrew is somewhat similar, but the use of *matres lectionis* is more haphazard. In both, full or almost-full vocalization (vowelization) is normally reserved for scripture and other archaic works, verse, works for children or beginners, and for loan words or foreign names.

A common characterization in modern psycholinguistics is that unvocalized 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 (Abu-Rabia 2001). But, when they are omitted, a reader needs to use some contextual information to resolve ambiguities in pronunciation and meaning.

Overall, fully vowelized Arabic text is considered much too complicated for ordinary reading and is rarely encountered. On the other hand, the lack of written short vowels in certain words, particularly homographs, may be detrimental to the ease of understandability and slows down reading. To resolve such pronunciation and sense ambiguities, it is often enough to add only one well-chosen short vowel. The issue we address is how to determine which added vowels are beneficial to the reader and which are excessive and undesirable.

The main motivation of this paper is to understand how to improve human intelligibility of Arabic texts, and potentially other languages with optional diacritics or punctuation, by automatically adding annotations that help resolve ambiguities encountered when one is reading normally. By adding just the minimally required vowels—for the language level of the reader, Arabic texts will hopefully be comprehended more easily.

For example, the word اكتشفت has a number of pronunciations with different meanings (e.g., "I/you/she/it discovered," "I/you *were* discovered," "she/it *was* discovered"), but with only two diacritics (*damma* and *sukun*) added on the key letters, reading اكتُشفتْ becomes easier ("she/it was discovered").

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. For example, here are two sentences (read right to left): مَنْ جَدَّ وَجَدَ "Hard work pays off," and مَنْ جَدُّ الوَلَدِ؟ "Who is the grandfather of the boy?" The second word is جد *jd* in both sentences, but it 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.

In other languages, Spanish and French, for example, a relatively small number of diacritics are required, and often serve to disambiguate. Anecdotally, even native speakers of such languages resort to a spellchecker to insert them ex post facto and save keystrokes thereby. Many Eastern European languages, on the other hand, make extensive use of diacritics. See Figure 1.

Classical Greek and Latin were often written *scripta continua*, without interword spaces. Likewise, many Eastern languages do not normally use spaces or punctuation within sentences. This, too, introduces a level of ambiguity, which an analogous notion of partial *punctuation* could help resolve.

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.

Our goal in this study is to improve the ease of understandability of Arabic text during reading by automatically generating diacritics, but only when they provide strong pronunciation hints to the reader. In some Arabic-language media outlets (e.g., *Nature* in Arabic[1]), partial diacritization is included to facilitate understandability. We propose a machine-learning model for partial diacritization. Deciding which vowels to recover is achieved by mimicking the way humans resolve pronunciation ambiguity. We train two networks for full diacritization, one taking the entire sentence into account, and another that considers only prior words. Partial diacritization is achieved by preserving those diacritics on which the two networks *disagree*, suggesting that without them disambiguation would require lookahead. Upon disagreement, we always take

[1] See arabicedition.nature.com.

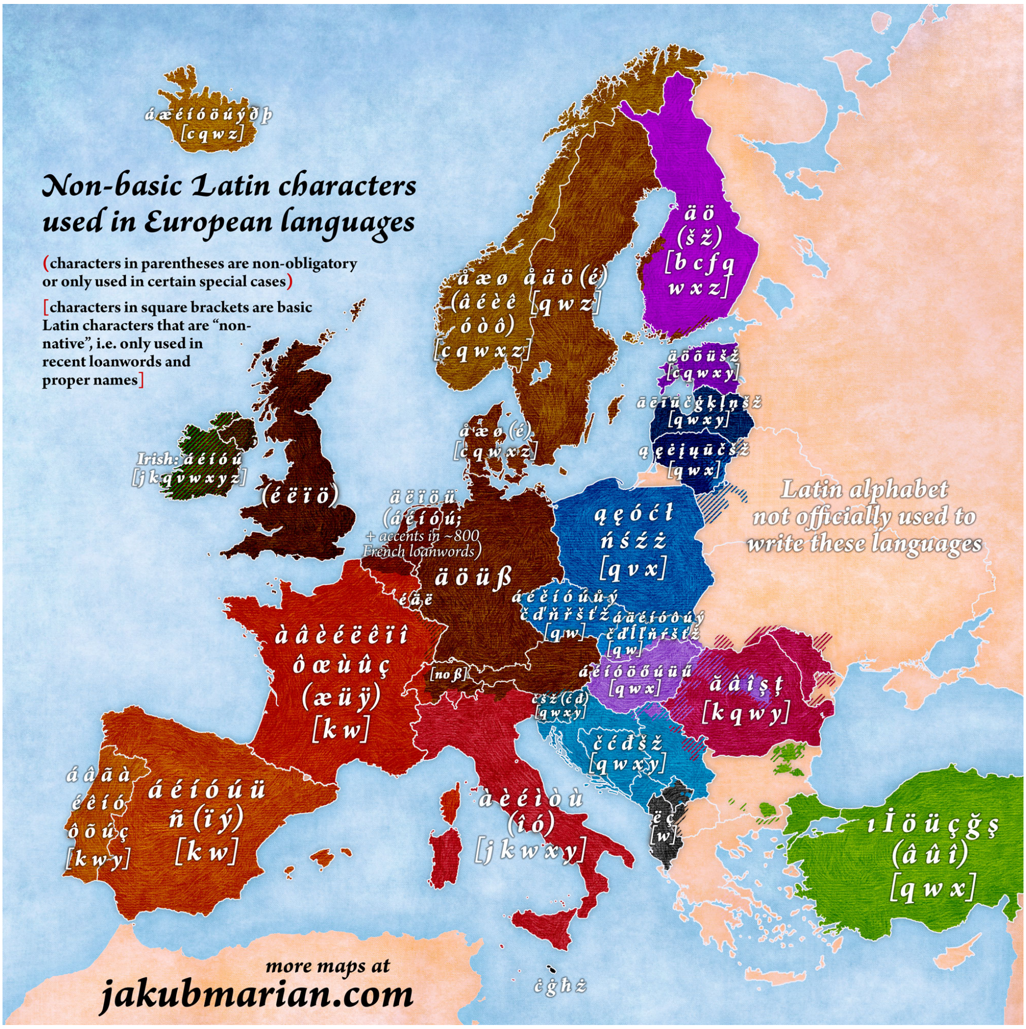


Figure 1
A map of the use of diacritics along with the Latin alphabet in European national languages. (Used with permission. Source: `https://jakubmarian.com/special-characters-diacritics-used-in-european-languages`)

the diacritic predicted by the network trained on the entire sentence to be the final assignment.

It has been claimed (Marcus 1978) 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. Our approach to diacritization is similar in using lookahead to determine what requires disambiguation.

We propose a new, dual neural-network architecture, designed to mimic human linear reading, on the one hand, and to model the impact of lookahead, on the other. By comparing the annotations of the two, one can determine what actually requires lookahead and what depends only on preceding text. We apply the dual-network to the specific problem of inferring sufficient optional diacritics to facilitate comprehension by human readers. The same approach can be applied to other reading aids, and can be tuned to the language level of the reader and the amount of lookahead presumed.

After surveying related work on Arabic in the next section, we describe the method chosen for partial diacritization in Section 4 and the datasets used for evaluation in Section 5. Section 6 presents the results of the experiments. It is followed by a short discussion in the final section.

2. Related Work

Comparing reading processes in languages of different orthography depths (Liberman et al. 1980; Frost, Katz, and Bentin 1987; Katz and Frost 1992) is still an active area of research. In particular, the contribution of short vowels to reading of Arabic has been studied. Whereas several studies report a positive contribution (Abu-Rabia 1995, 1996, 1997a,b, 1998a,b, 1999; Abu-Hamour, Al-Hmouz, and Kenana 2013; Taha 2016), a number (Ibrahim 2013; Asadi 2017) 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 (Abu-Rabia 2019) summarizes the conflicting results.

Bouamor et al. (2015) conducted a study of human annotation for minimal Arabic diacritization that showed a low inter-annotator agreement and demonstrated how subjective this task can be.

Various works apply deep neural networks to restoration of diacritics. Examples include Náplava et al. (2018) for Polish; Nuţu, Lőrincz, and Stan (2019) for Romanian; Hucko and Lacko (2018) for Slovak; Uzun (2018) for Turkish; and Nguyen et al. (2020), Hung (2018), Nga et al. (2019), Alqahtani, Mishra, and Diab (2019) for Vietnamese. A recent work (Stankevičius et al. 2022) employs a transformer-based (Vaswani et al. 2017) ByT5 network (Xue et al. 2022) for 12 European languages plus Vietnamese. Other transformer-based diacritization networks include Laki and Yang (2020), Dang and Nguyen (2020). For state-of-the-art vowelization of Hebrew, see Shmidman et al. (2020).

There is a large body of work on full Arabic diacritization. Early work took a more traditional machine-learning approach (Zitouni and Sarikaya 2009; Darwish, Mubarak, and Abdelali 2017); recent efforts are usually based on deep neural setups (Abandah et al. 2015; Belinkov and Glass 2015; Alqahtani, Mishra, and Diab 2019; Fadel et al. 2019a,b; Mijlad and Younoussi 2019; Mubarak et al. 2019; Abbad and Xiong 2020; AlKhamissi, ElNokrashy, and Gabr 2020). A few studies (Zalmout and Habash 2020; Alqahtani, Mishra, and Diab 2020) show the contribution of morphological data to diacritization. An encoder-decoder network using a Tacotron CBHG module (Wang et al. 2017) as part of the encoder has been introduced (Madhfar and Qamar 2020).

As may be expected, diacritization can help in morphological analysis (Habash, Shahrour, and Al-Khalil 2016) and with other natural language processing tasks (Alqahtani, Mishra, and Diab 2020). Recently, Alqahtani, Aldarmaki, and Diab (2019) evaluated the contribution of incomplete restoration of Arabic diacritics to a number of such 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. Fadel et al. (2019a) developed a deep recurrent neural network for diacritization, which was reported to positively contribute to neural machine translation, by encoding the diacritics on a parallel layer to the input characters. We will use their architecture as a downstream task in the evaluation of our model for partial diacritic restoration. A decade earlier, Diab, Ghoneim, and Habash (2007) measured the contribution of different partial diacritization schemes to a statistical Arabic-to-English translation system. They found that translation quality is not improved when the input is partially diacritized. At the same time, they showed that the translation quality significantly deteriorates when the input is provided fully diacritized.

All the same, we focus more on improving understandability during reading, as opposed to improving the accuracy of a downstream-task algorithm. Our goal is to generate diacritics only for letters (not necessarily all letters of a word) that resolve ambiguities encountered during continuous reading of a running text. We intentionally do not resolve ambiguities that can be handled with information already available while reading a sentence normally from the beginning forward. To the best of our knowledge, this is the first time that this goal is being addressed algorithmically. The closest related work is by Alnefaie and Azmi (2017), who developed an algorithm for partial diacritization of a sentence by filtering out inconsistent morphological analyses of the words given the sentence as context. In their case, each word is diacritized to distinguish the intended reading from the otherwise most likely sense, whereas we aim to provide only enough diacritics to disambiguate the word considering the preceding context—as is commonly done in actual texts. Their morphological analyses were retrieved using a lexicon that contains all potential analyses for the words in a given sentence. Naturally, diacritics can be assigned only to words that exist in the lexicon, a limitation that does not exist in our approach. Their evaluation was done manually on a set of sentences, which have not been made publicly available.

We are, unfortunately, unaware of the existence of any relevant resources that could help train a supervised machine-learning algorithm for partial diacritic generation.

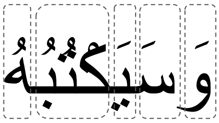
3. Arabic Morphology

Arabic, like most Semitic languages, enjoys a rich morphology. This includes verbal inflection (*binyan*, tense, mood, etc.), nominal cases, construct forms, prefixes for conjunctions, prepositions, the definite article, and more, and suffixes indicating gender, number (singular, dual, plural), and pronomial possessives. All together, these result in a high degree of ambiguity for Arabic words, with about 12 potential morphological analyses per word (without extra diacritics) on average (Habash 2010). An example of the morphological complexity of words is given in Figure 2. Arabic is written right to left. Letters often change form depending on their position (initial, medial, final) within the word.

4. Methodology

4.1 Morphological Ambiguity

Partial diacritization is the process of inferring a minimal subset of diacritics that is fundamental to disambiguate the context. This mission is not well configured, however, and there is no convention or explicit rules for 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 of words that a reader usually needs to look ahead in order to improve the ease of understandability.



Meaning	**Morpheme**
and	وَ
future indication	سَ
present tense, 3rd person, singular, masculine inflection	يَ
root letters of verb (*write*)	كْتُبُ
direct object pronoun, masculine, 3rd person, singular	هُ

Figure 2
An example of Arabic morphology.

To imitate human readers and the hurdles they face, we train two distinct neural models for the task of full restoration of diacritics: 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 by assigning letters with diacritics only when they cannot be trivially decoded using the content that has already been taken into account by the first unidirectional neural network. Therefore, we train both networks to restore diacritics in full, and at inference time we assign the diacritics predicted by the second model only to letters for which the two models made different predictions. We describe both models in greater detail below.

Generally speaking, the input is composed of a sequence of Arabic characters $c_1, c_2, \ldots, c_n$, and the target is to predict a single label d_i for each character c_i, representing its diacritic. Like previous work, we use the following set of labels to account for most of the diacritic types:

(a) Three short vowels (*harakat*): *fatḥah* /a/ ◌َ; *kasrah* /i/ ◌ِ; and *ḍammah* /u/ ◌ُ.

(b) Three nunations (*tanwin*s) to indicate case ending: ◌ً /an/; ◌ٍ /in/; and ◌ٌ /un/.

(c) Gemination: *shaddah* ◌ّ.

(d) *Sukūn* (a circle above a consonant) to indicate vowel absence.

(e) Six more labels for capturing various combinations consisting of geminated vowels or nunations.

(f) A final label, NO DIACRITIC, indicating the absence of any diacritics on the letter.

All told, there are 15 labels.

4.2 Reading-Direction Model

For the first model, we use a four-layer unidirectional long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) architecture that works on the character level, and predicts one label per input character. We choose LSTM because in this model we are interested in mimicking the short-term memory characteristic of reading as performed by humans. We are not interested only in restoring diacritics that can be deduced merely by looking ahead a few words. Rather, we also aim to restore diacritics that are difficult to determine due to a high level of contextual ambiguity.

Our assumption is that human readers lend their attention to whole words rather than letters; therefore, each input character is encoded with a non-pretrained embedding vector concatenated with the word-level embeddings of the containing word. Word-level embeddings are the output of another bi-directional LSTM (BiLSTM) that works on the containing word's characters. Formally, let word $j = c_1 c_2 \cdots c_n$ be composed of n characters c_i. Then, o_i is the encoded version of c_i:

$$w_j = \text{BiLSTM}(c_1, c_2, \ldots, c_n)$$

$$o_i = \text{LSTM}([c_i; w_j])$$

where w_j 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. The input characters are encoded using a nonpretrained embedding layer.

In addition to this LSTM architecture, we also experimented with an alternative, a character-based 6-layer transformer (Vaswani et al. 2017) encoder, each layer composed of 8 attention heads. To force the encoder to incorporate information only from characters that have been read so far, we masked all characters that follow the specific word that contains the processed character.

The LSTM architecture is shown in Figure 3(a). The alternate, transformer architecture is similar to the full-sentence model, which we describe next, and which is depicted in Figure 3(b).

4.3 Full-Sentence Model

To encode a full sentence before classification, we first experimented with several different architectures. A 6-layer transformer (Vaswani et al. 2017) working on the character level, each layer composed of 8 attention heads, followed by a fully connected layer, delivered the best accuracy for full diacritic restoration. As for the reading-direction model, we encode the input characters using a nonpretrained embedding layer. See Figure 3(b).

For training the models, we use sentences of up to 100 characters, not counting 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, each character gets a number of predictions, one per window. The final prediction is decided on by weighted maximum vote (weights are predicted probabilities).

4.4 Training and Hyperparameter Settings

For the reading-direction model, our best results are obtained by using a hidden size of 16 for the word-character BiLSTM, resulting in a 32-dimensional vector for word

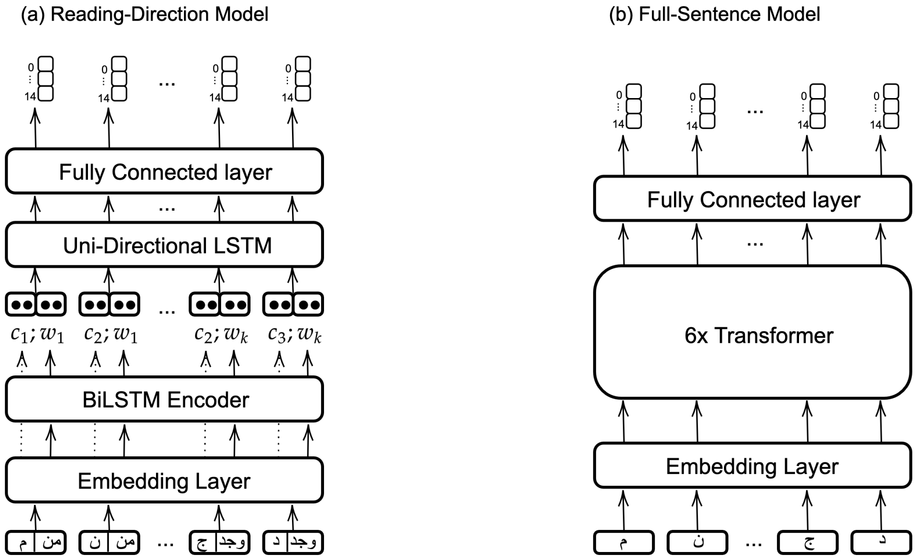


Figure 3
The architecture of the models we use for partial insertion of diacritics. (a) The simple reading-direction model. (b) The advanced, transformer model. Note that the input letters are shown left to right.

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 between the layers with 20% drop probability and the Adam optimizer, configured with a learning rate of 10^{-4}. Batch size is 512. The reading-direction model has 7,505,778 parameters in total.

For the full-sentence and reading-direction transformer models, we use 6 layers for both encoder and decoder, 8 attention heads, and hidden size 512. We use 10%-drop dropout and ReLU as activation function. The full-sentence model has a total of 25,308,690 parameters.

We trained the reading-direction LSTM model for 10 epochs and reached optimal results on a validation test after 8; each epoch took about 12 hours on average. The transformer models reaches optimality after 6 epochs. (It takes about 55 minutes to run 10 epochs on a GeForce GTX 1080 Ti.)

5. Datasets

We make use of two datasets, one mostly consisting of classical Arabic texts and the other of tweets in modern standard Arabic.

5.1 Tashkeela

To train both models, we use the Tashkeela corpus (Zerrouki and Balla 2017), comprising more than 65M words, fully diacritized. It mostly contains classic works written in Classical Arabic (CA), the forerunner of Modern Standard Arabic (MSA), the main

Table 1
Word and line counts of our corpus, before and after cleaning, broken down into CA+HQ and MSA.

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

language used today in formal settings (in contradistinction to spoken Arabic, with its many mutually unintelligible regional varieties). Modern and Classical Arabic have much in common, but oftentimes they use different grammatical structures and vocabularies. Only 1% of the texts in Tashkeela are written in MSA.

We preprocess the corpus in a way similar to Fadel et al. (2019b); additionally, we replace a few rare letters by their natural equivalents (e.g., Farsi *yeh* ی into Arabic *yeh* ي and Farsi *peh* پ into Arabic *beh* ب). Furthermore, in Tashkeela, not all letters carry diacritics. Because 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. (2019b) removed lines below a rate of 80%. Because it appears that Modern Arabic has a lower rate than Classical Arabic, we remove lines below 50%, to have more MSA text during training. This left us with 20K of MSA text (corresponding to 604K words) out of the original 50K lines (801K). Additionally, we added the Holy Quran (HQ), fully diacritized, to the CA corpus. Table 1 summarizes the number of words and lines in our corpus.

5.2 Arabic Treebank: Part 3 v1.0

To compare our full-sentence model with two related studies, we train and evaluate it on the Arabic Treebank: Part 3 v1.0 (LDC catalog number LDC2004T11, ISBN 1-58563-298-8). The treebank contains approximately 300,000 Arabic word tokens with annotation that includes complete vocalization with case endings. Following previous work, we use the original train/test split for training and evaluation.

5.3 Tweets

Additionally, we collected 75 tweets written in MSA,[2] taken from official accounts that cover news in different domains. As a first step, the tweets were fully diacritized by a professional native linguist, whom we hired specifically for this task. As a second step, the fully diacritized tweets were manually processed by three native MSA speakers, keeping 19%, 25%, and 26% of the diacritics, chosen to facilitate fluency. The native speakers were instructed to read the non-diacritized version of each tweet, and to keep the diacritics of those letters for which they failed to read fluently. The knowledge of MSA of all annotators is at mother-tongue level. Tables 2 and 3 show examples for two of the tweets. We will make this dataset publicly available.

[2] From `www.twitter.com`.

Table 2
Example of a tweet from our dataset, provided with different diacritic-assignment conditions. This maxim translated into English: “A person can beg for love, buy it, receive it as a gift, or find it in the alley—but it cannot be robbed.” The 1st row (O) has the original tweet (note that there are also dots and other marks that are nowadays integral parts of the signs of the alphabet); the 2nd row (G) shows diacritics fully assigned by our native linguist; the 3rd (F) has the full diacritization predicted by the full-sentence transformer model (the two differences are highlighted in red); the 4th row (H) shows partial diacritization assigned by one of our native annotators (25 diacritics, highlighted in blue); the last row (P) shows partial diacritization automatically generated by our model-disagreement algorithm (11 diacritics, highlighted in blue). Only one difference from (G) remains, highlighted in red. See the text for details.

O	يمكن للمرء أن يستجدي الحب أو يشتريه، أن يناله هبة، أو يعثر عليه في الزقاق، لكن سلب الحب غير ممكن
G	يُمْكِنُ لِلْمَرْءِ أَنْ يَسْتَجْدِيَ الْحُبَّ أَوْ يَشْتَرِيَهُ، أَنْ يَنَالَهُ هِبَةً، أَوْ يَعْثُرَ عَلَيْهِ فِي الزُّقَاقِ، لَكِنَّ سَلْبَ الْحُبِّ غَيْرُ مُمْكِنٍ
F	يُمْكِنُ لِلْمَرْءِ أَنْ يَسْتَجْدِيَ الْحُبَّ أًوْ يَشْتَرِيَهُ، أَنْ يَنَالَهُ هِبَةً، أَوْ يَعْثُرَ عَلَيْهِ فِي الزُّقَاقِ، لَكِنَ سَلْبَ الْحُبِّ غَيْرُ مُمْكِنٍ
H	يُمْكن للمرءِ أنْ يَسْتَجْدِيَ الحُب أو يشتريَه، أنْ يَناله هِبَةً، أو يَعْثر عليه في الزُّقاق، لكنَّ سَلْبَ الحُبِّ غير مُمكن
P	يمكن للمرء أن يستجديَ الحُب أو يشتريَهُ، أنْ يناله هبَةً، أو يَعثر عليه في الزقاق، لكنَ سلْب الحُب غير ممكن

Table 3
Another example of a tweet from our dataset, provided with different diacritic-assignment conditions. This tweet translated into English: “Attention Deficit Hyperactivity Disorder is a difficult disorder that many people deal with, and it is not exclusive to childhood, as some people think, but could affect adults as well.” The rows are as in the previous table.

O	إضطراب فرط الحركة وتشتت الانتباه من المشاكل الصعبة التي يتعرض لها بعض الأشخاص، وليست مقتصرة على مرحلة الطفولة فقط كما يظن البعض، لكن قد تصيب الكبار أيضا.
G	إِضْطِرَابُ فَرْطِ الْحَرَكَةِ وَتَشَتُّتِ الْإِنْتِبَاهِ مِنَ الْمَشَاكِلِ الصَّعْبَةِ الَّتِي يَتَعَرَّضُ لَهَا بَعْضُ الأَشْخَاصِ، وَلَيْسَتْ مُقْتَصِرَةً عَلَى مَرْحَلَةِ الطُّفُولَةِ فَقَطْ كَمَا يَظُنُّ الْبَعْضُ، لَكِنْ قَدْ تُصِيبُ الْكِبَارَ أَيْضًا.
F	إِضْطِرَابُ فَرْطِ الْحَرَكَةِ وَتَشَتُّتِ الإِنْتِبَاهِ مِنَ الْمَشَاكِلِ الصَّعْبَةِ الَتِي يَتَعَرَّضُ لَهَا بَعْضُ الأَشْخَاصِ، وَلَيْسَتْ مُقْتَصِرَةً عَلَى مَرْحَلَةِ الطُّفُولَةِ فَقَطْ كَمَا يَظُنُّ الْبَعْضُ، لَكِنْ قَدْ تُصِيبُ الْكِبَارَ أَيْضًا.
H	إِضْطِراب فَرْط الحركة وتشتُّتِ الإِنتباه منَ المشاكل الصعبة التي يَتعرَّض لها بعضُ الأشخاص، وليستْ مُقْتصِرةً علَى مرحلةِ الطفولة فقطْ كما يَظنُّ البعض، لكنْ قد تُصيب الكبارَ أيضًا.
P	إضطرابُ فرطِ الحركة وتشَتُّتِ الإنتباه منَ المشاكل الصعبة التي يتعرض لها بعض الأشخاص، وليست مقتصرة على مرحلةِ الطُّفولة فقط كما يَظُن البعض، لكن قد تُصيب الكبار أيضا.

6. Evaluation Results

Before considering the results of our method for partial diacritization, we check that the full-sentence network achieves reasonable accuracy. We also examine how the amount of lookahead affects fluency.

Table 4
DER / WER results (in %) on the Tashkeela test set, as defined by Fadel et al. (2019b). Results are reported under different conditions, when including or excluding case-ending errors or the NO DIACRITIC label.

	Including NO DIACRITIC		**Excluding** NO DIACRITIC	
	with case	**w/o case**	**with case**	**w/o case**
Fadel et al. (2019a)	2.60 / 7.69	2.11 / 4.57	3.00 / 7.39	2.42 / 4.44
AlKhamissi, ElNokrashy, and Gabr (2020)	**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 5
DER / WER results (in %) on the Arabic Treebank Part 3 (Arabic Treebank Part 3 v1.0: catalog number LDC2004T11 and ISBN 1-58563-298-8) test set, as defined by Zitouni, Sorensen, and Sarikaya (2006). Results are reported under different conditions, when including or excluding case-ending errors. All previous studies on this dataset included the NO DIACRITIC label for evaluation purposes.

	With CASE ENDING		**Without** CASE ENDING	
	DER	**WER**	**DER**	**WER**
Zitouni, Sorensen, and Sarikaya (2006)	5.5	18.0	2.5	7.9
Habash and Rambow (2007)	4.8	14.9	2.2	**5.5**
Our full-sentence model	**3.7**	**12.2**	**2.0**	5.7

6.1 Full Restoration

To compare our full-sentence transformer model with state-of-art systems, we preprocess the Tashkeela+HQ dataset and use exactly the same train/test split as Fadel et al. (2019b). That means that, for the following experiment, we use the same diacritics per character rate of 80% as used in Fadel et al. (2019b). As is customary, we measure (1) diacritic error rate (DER), the percentage of misclassified letters, and (2) word error rate (WER), words with at least one misclassified letter. Following others, we evaluate our models under several varying conditions. Generally speaking, case endings (only when handled by nunations; other case-ending forms exist) are deemed more difficult than basic vowels, since they are syntax related; therefore, we also report results when discounting such nunation mistakes. Similarly, we also report results ignoring mistakes in predicting NO DIACRITIC, since the Tashkeela dataset often omits legitimate diacritics, and the predicted diacritic may in fact be correct. We only account for Arabic letters; foreign-script characters and digits are ignored.

Tables 4 and 5 compare the results of our transformer model for full diacritization with the best known models on the Tashkeela+HQ corpus, under the different conditions we mentioned above, and on the Arabic Treebank Part 3 corpus, respectively. For the latter, we use the original train/test split for training and evaluating our full-sentence model.

Even though our goals did not include delivering a system for full diacritization, our transformer model is almost on par with the state of the art.

Because 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

Table 6
Error rates DER / WER (in %) on different validation sets.

Validation Dataset	Model	Including NO DIACRITIC		Excluding NO DIACRITIC	
		with case	w/o case	with case	w/o case
50%-filtered validation	Reading direction (LSTM)	13.0 / 34.0	9.3 / 26.1	8.3 / 25.4	5.3 / 16.9
MSA validation	Full sentence	6.9 / 26.6	7.0 / 23.0	6.1 / 16.0	5.5 / 11.5
Tweets	Reading direction (LSTM)	16.2 / 47.5	11.1 / 33.1	14.0 / 44.9	9.5 / 30.3
	Reading direction (Transformer)	15.4 / 36.7	11.3 / 26.9	12.4 / 33.1	9.0 / 23.6
	Full sentence	**9.2 / 27.7**	**6.3 / 18.1**	**7.2 / 23.6**	**5.0 / 15.0**

model. The first row of Table 6 shows the results of the simple reading-direction model trained and evaluated on this split.

Our ultimate goal is to generate partial diacritics by using the predictions of the full-sentence model only when the two models disagree. Therefore, we would like to make our full-sentence transformer model more capable than the simple reading-direction model. To improve performance of our full-sentence model on MSA, we train the model in two phases: (1) pre-training with CA+HQ texts (with 85%/15% train/validation split), and then (2) fine-tuning with MSA texts (with 90%/10% train/validation split), to end with weights that handle MSA better than CA. This fine-tuning training style gave us 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 6 shows the final results of the two-phase fine-tuned model on the MSA-only validation set. Finally, we evaluate all three models—the two simple reading-direction models trained on the 50%-filtered training set as well as the full-sentence two-phase model—on our MSA-only fully-diacritized tweets (last three rows). The full-sentence model clearly performs better across all metrics, as it should. The third row in Table 2 shows the diacritics predicted by our full-sentence model. As can be seen, the accuracy for this tweet is nearly perfect, including nunations and geminations.

The transformer-based reading-direction model performs slightly better than the LSTM one. However, as we show in the following section, the LSTM model actually works better for the partial restoration task at hand.

We performed some error analysis on the tweets collection. Recall and precision per type are reported in Table 7. For easier reading, in Table 8 we aggregate the diacritic types under four main categories: vowels (including *fatḥah*, *kasrah*, *ḍammah*, and their corresponding geminated versions), nunations (including their geminated versions), sukun, and NO DIACRITIC. (We ignore the two instances of *shaddah*.) Both precision and recall are calculated by comparing the models' prediction with the gold-standard tweets. Overall, we see that the full-sentence transformer model performs better than the reading-direction model across all categories. Restoring nunation diacritics seems to

Table 7
Model performance per-type, including precision and recall of both models, full sentence and reading direction (LSTM), on the tweet collection. # is the number of instances of each type in the tweet dataset.

Type	Full Sentence		Reading Direction		#
	Precision	Recall	Precision	Recall	
fatḥah	92	92	85	86	1,493
kasrah	92	92	85	82	875
ḍammah	89	86	77	70	427
Geminated *fatḥah*	65	91	57	82	113
Geminated *kasrah*	69	79	63	60	53
Geminated *ḍammah*	67	85	50	54	26
fatḥatān	86	76	62	60	25
kasratān	70	76	39	49	49
ḍammatān	68	70	52	48	27
Geminated *fatḥatān*	0	0	0	0	0
Geminated *kasratān*	56	100	50	60	5
Geminated *ḍammatān*	0	0	0	0	1
shaddah	0	0	0	0	2
sukun	91	91	85	88	642
no diacritic	99	96	97	95	2,290

Table 8
Aggregated version of Table 7. For more information, see the text.

Type	Full Sentence		Reading Direction		#
	Precision	Recall	Precision	Recall	
Vowel	96	98	94	94	2,987
Nunation	72	79	47	62	107
Sukun	85	88	77	70	642
no diacritic	99	96	97	95	2,290

Table 9
Reading-direction (LSTM) performance results, using the predictions of the full-sentence model as gold. The precision and recall are given by aggregated types, similar to Table 8.

Type	Precision	Recall	#
Vowel	96	94	3,039
Nunation	54	65	116
Sukun	89	91	648
no diacritic	97	98	2,223

be the most challenging task, especially for the reading-direction model. The geminated types are more difficult to predict than the non-geminated ones.

Additionally, in Table 9 we compared the prediction performance of the reading-direction model with the predictions of the full-sentence model, used as the gold standard. Following our partial restoration design, we keep those diacritics on which the two models disagree; therefore, this comparison shows that our partial-restoration approach keeps a relatively large percentage of the nunations and only a relatively small percentage of the vowels.

Table 10
Results of partial restoration on tweets. Metrics are the kappa coefficient for the 15-label task (left) and F1 for the diacritic assignment task (right), that is, checking if a letter is assigned a diacritic or not.

Diacritization	Annotator 1	Annotator 2	Annotator 3
Random partial restoration	0.11 / 0.77	0.10 / 0.80	0.11 / 0.77
Our partial restoration (LSTM)	**0.26** / **0.82**	**0.22** / **0.84**	**0.22** / **0.81**
Our partial restoration (Transformer)	0.15 / 0.80	0.13 / 0.83	0.16 / 0.80

6.2 Partial Restoration

Both models are combined and used for generating partial diacritics as part of our model-difference approach. As mentioned above, diacritics are assigned only to letters for which the two networks *disagree* with regard to their predicted labels; in such cases, we output the predicted label of the full-sentence model, completely ignoring the label of the simplistic reading-direction model.

To evaluate, we match the predicted partial diacritics with those manually assigned to the tweets by native speakers. Our system decided to keep about 12% of the restored diacritics, while the native speakers kept 19%–26%. For a baseline, we randomly select 12% of the diacritics that were manually assigned by Annotator 1, who has the highest level of agreement with the model. Table 10 shows the improvement achieved by our model-difference method, using both the LSTM and transformer architectures, with the kappa coefficient for the 15-label partial diacritic restoration task (recall that one of the labels represents NO DIACRITIC), and F1 for a binary-classification task of diacritic assignment (with positive label being diacritic assignment). We can clearly see that our partial diacritization algorithm performs better than a baseline random selection of diacritics. It adds fewer diacritics than a human might, so recall is somewhat low, while precision is relatively high (because full-sentence diacritization is good), for an F1 score of 0.80–0.84. Subjectively, the results are quite pleasing for a native reader.

Although the transformer-based reading-direction model does a better job at diacritization than does the LSTM model, it has the opposite effect when it comes to partial restoration. The transformer reading-direction model agrees with the full-sentence model more often than the LSTM model does, resulting in fewer desirable diacritics being restored, perhaps because it is taking distant contextual information into account more than a typical reader does.

The last row of Table 2 shows the partial diacritics predicted by our model-disagreement algorithm. One can see that there are 11 predicted diacritics, which are mostly aligned with the gold-standard version provided in the row above it, containing 20 diacritics. The correctly predicted *fatha* over the final *ya* in the fourth word يستجديَ is a nice example for when the subjunctive mood is used, which has to end with a *fatha* due to the word أن that comes before. This information was not memorized well by the simple reading-direction model, even though it was encoded during processing. Another example is the *sukūn* over the last letter of the eighth word أنْ. The simple reading-direction model predicted *fatha*, as in أَنَّ, which is what usually comes before a noun. However, looking ahead one word, the full-sentence model was able to predict the correct diacritic, *sukūn*, which is usually used before verbs. See Table 3 for another example.

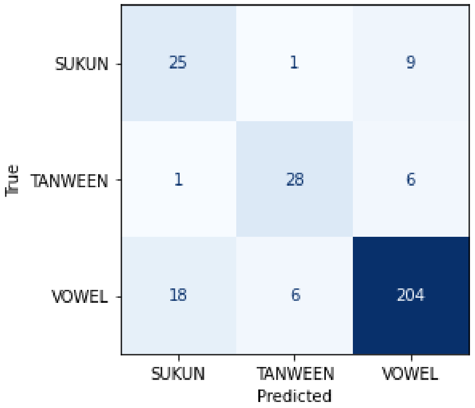


Figure 4
Confusion matrix for the partial-restoration algorithm (excluding the NO DIACRITICS label), evaluated on the tweet dataset.

We have done some error analysis and learned that both of our models, reading-direction and full-sentence, make more mistakes on proper names than on any other word types, resulting in less accurate partial restoration. Additionally, case endings are usually more challenging to predict than in-word diacritic marks—for both models. We provide the aggregated confusion matrix of our partial restoration model in Figure 4. (Mistakes involving the NO DIACRITIC label are not included so as to focus on diacritic assignment mismatches.)

6.3 Impact on Translation

Fadel et al. (2019a) suggested evaluating diacritic restoration by measuring their contribution to a diacritic-sensitive Arabic-to-English neural-network translation system. Even though our goal in this study has nothing to do with improving a downstream natural-language-processing task, we decided to follow the same approach as another extrinsic evaluation method for our partial diacritization algorithm. Therefore, we train the same translation system on one million sentence pairs, for which we restore diacritics in full with our full-sentence transformer model, and evaluate it on a test set under different conditions of diacritic assignment, including partial diacritization using our model-combination approach, random partial diacritization using the same

Table 11
BLEU scores of an Arabic-to-English machine translation system, using different levels of diacritics. The results achieved by Fadel et al. (2019a) are given in the last row.

Diacritization	BLEU Score
No diacritics	33.48
Random partial diacritization	33.34
Our partial diacritization	33.75
Full-sentence full diacritization	34.25
(Fadel et al. 2019a)	34.34

Table 12
DER / WER results of the full-sentence transformer on the tweet dataset, under different lookahead windows.

Lookahead (# words)	Including NO DIACRITIC		Excluding NO DIACRITIC	
	with case	w/o case	with case	w/o case
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

diacritic-per-character rate, no diacritics at all, and full diacritization provided by our full-sentence transformer model. The evaluation results as BLEU scores (Papineni et al. 2002) are summarized in Table 11. This shows a small increased precision with our partial diacritic restoration, which kept about 10% of the diacritics over the two baselines. Machine translation benefits more, of course, when the input comes with all diacritics, since it was trained under the same conditions.

6.4 The Contribution of Lookahead

To measure the contribution of forward-looking context to fluency of reading, we ran additional experiments with the full-sentence transformer model, placing successively larger limits on the number of words following the current word that the transformer may encode. Table 12 summarizes DER and WER on the tweet dataset for each instance; the limit on lookahead is indicated in the first column. Under all evaluation conditions, the transformer model benefits from more and more lookahead in order to fully diacritize the current word. One word lookahead has a dramatic impact. But after that, there are diminishing returns. See Figure 5.

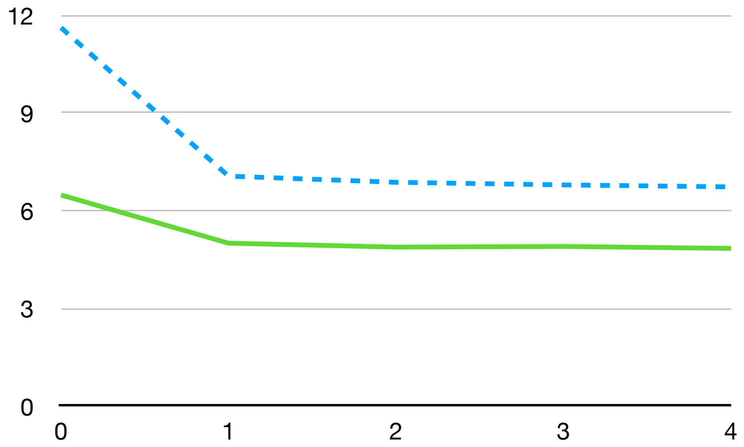


Figure 5
Diminishing returns for diacritic error rate in % (excluding the NO DIACRITICS label) for lookahead in the range 0–4 words. Solid (green) line ignores errors of case endings; dashed blue counts them. Compare Table 12.

7. Conclusions

We have proposed a novel criterion for partial diacritization of Arabic and have implemented it as the difference between two neural networks that restore diacritics in full. One network uses only context that has already been read, and the other benefits from seeing the entire sentence prior to prediction. For evaluation, we manually diacritized a set of tweets written in Modern Arabic and then selectively marked those diacritics that contribute most to disambiguation during reading. Using this dataset, as well as a diacritic-sensitive neural-network machine translation system, we found our model-difference approach to be superior to the baseline method. We proffer this dataset to the community.

It bears keeping in mind that downstream machine translation and BLEU-score evaluation are less than ideal for measuring inherent ambiguity, especially since neural machine translation systems still leave much to be desired for languages like Arabic, with or without diacritics.

We have also quantified 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. In future work, we hope to compare fluency with different levels of lookahead-based vowelization.

We have chosen Arabic here as a convenient case study. Partial vowelization is of practical value for that language. We would like to expand our model-difference method to additional languages and to additional aspects of disambiguation, such as optional punctuation. The same idea could also be used to suggest cases in English—or other languages—where sentence structure could be simplified to ease reading comprehension.

Acknowledgments

We thank the annotators for their assistance and the reviewers for their suggestions.

References

Abandah, Gheith A., Alex Graves, Balkees Al-Shagoor, Alaa Arabiyat, Fuad Jamour, and Majid Al-Taee. 2015. Automatic diacritization of Arabic text using recurrent neural networks. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(2):183–197. https://doi.org/10.1007/s10032-015-0242-2

Abbad, Hamza and Shengwu Xiong. 2020. Multi-components system for automatic Arabic diacritization. In *European Conference on Information Retrieval*, pages 341–355. https://doi.org/10.1007/978-3-030-45439-5_23

Abu-Hamour, Bashir, Hanan Al-Hmouz, and Mohammed Kenana. 2013. 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. https://doi.org/10.1080/19404158.2013.852980

Abu-Rabia, Salim. 1995. 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. https://doi.org/10.1080/0270271950160401

Abu-Rabia, Salim. 1996. The role of vowels and context in the reading of highly skilled native Arabic readers. *Journal of Psycholinguistic Research*, 25(6):629–641. https://doi.org/10.1007/BF01712413

Abu-Rabia, Salim. 1997a. 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. https://doi.org/10.1111/1467-9817.00026

Abu-Rabia, Salim. 1997b. 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.

Abu-Rabia, Salim. 1998a. Attitudes and culture in second language learning among Israeli-Arab students. *Curriculum and Teaching*, 13(1):13–30. https://doi.org/10.7459/ct/13.1.03

Abu-Rabia, Salim. 1998b. Reading Arabic texts: Effects of text type, reader type and vowelization. *Reading and Writing*, 10(2):105–119. https://doi.org/10.1023/A:1007906222227

Abu-Rabia, Salim. 1999. 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. https://doi.org/10.1023/A:1023291620997, PubMed: 9949716

Abu-Rabia, Salim. 2001. The role of vowels in reading Semitic scripts: Data from Arabic and Hebrew. *Reading and Writing*, 14(1):39–59. https://doi.org/10.1023/A:1008147606320

Abu-Rabia, Salim. 2019. The role of short vowels in reading Arabic: A critical literature review. *Journal of Psycholinguistic Research*, 48(4):785–795. https://doi.org/10.1007/s10936-019-09631-4, PubMed: 30719613

AlKhamissi, Badr, Muhammad ElNokrashy, and Mohamed Gabr. 2020. Deep diacritization: Efficient hierarchical recurrence for improved Arabic diacritization. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 38–48.

Alnefaie, Rehab and Aqil M. Azmi. 2017. Automatic minimal diacritization of Arabic texts. *Procedia Computer Science*, 117:169–174. https://doi.org/10.1016/j.procs.2017.10.106

Alqahtani, Sawsan, Hanan Aldarmaki, and Mona Diab. 2019. Homograph disambiguation through selective diacritic restoration. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 49–59. https://doi.org/10.18653/v1/W19-4606

Alqahtani, Sawsan, Ajay Mishra, and Mona Diab. 2019. Efficient convolutional neural networks for diacritic restoration. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1442–1448. https://doi.org/10.18653/v1/D19-1151

Alqahtani, Sawsan, Ajay Mishra, and Mona Diab. 2020. A multitask learning approach for diacritic restoration. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8238–8247. https://doi.org/10.18653/v1/2020.acl-main.732

Asadi, Ibrahim A. 2017. Reading Arabic with the diacritics for short vowels: Vowelised but not necessarily easy to read. *Writing Systems Research*, 9(2):137–147. https://doi.org/10.1080/17586801.2017.1400493

Belinkov, Yonatan and James Glass. 2015. Arabic diacritization with recurrent neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2281–2285. https://doi.org/10.18653/v1/D15-1274

Bouamor, Houda, Wajdi Zaghouani, Mona Diab, Ossama Obeid, Kemal Oflazer, Mahmoud Ghoneim, and Abdelati Hawwari. 2015. A pilot study on Arabic multi-genre corpus diacritization. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 80–88. https://doi.org/10.18653/v1/W15-3209

Dang, Trung Duc Anh and Thi Thu Trang Nguyen. 2020. TDP–A hybrid diacritic restoration with transformer decoder. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pages 76–83.

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

Darwish, Kareem, Hamdy Mubarak, and Ahmed Abdelali. 2017. Arabic diacritization: Stats, rules, and hacks. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 9–17. https://doi.org/10.18653/v1/W17-1302

Diab, Mona, Mahmoud Ghoneim, and Nizar Habash. 2007. Arabic diacritization in the context of statistical machine translation. *Proceedings of MT Summit*, 7 pages. https://aclanthology.org/2007.mtsummit-papers.20.pdf

Fadel, Ali, Ibraheem Tuffaha, Bara' Al-Jawarneh, and Mahmoud Al-Ayyoub. 2019a. Neural Arabic text diacritization: State of the art results and a novel approach for machine translation. In *Proceedings of the 6th Workshop on Asian Translation, WAT@EMNLP-IJCNLP 2019*, pages 215–225. https://doi.org/10.18653/v1/D19-5229

Fadel, Ali, Ibraheem Tuffaha, Bara' Al-Jawarneh, and Mahmoud Al-Ayyoub. 2019b. Arabic text diacritization using deep neural networks. In *Proceedings of the 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, pages 1–7. https://doi.org/10.1109/CAIS.2019.8769512

Frost, Ram, Leonard Katz, and Shlomo Bentin. 1987. Strategies for visual word recognition and orthographical depth: A multilingual comparison. *Journal of Experimental Psychology: Human Perception and Performance*, 13(1):104. https://doi.org/10.1037/0096-1523.13.1.104

Habash, Nizar and Owen Rambow. 2007. Arabic diacritization through full morphological tagging. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 53–56. https://doi.org/10.3115/1614108.1614122

Habash, Nizar, Anas Shahrour, and Muhamed Al-Khalil. 2016. Exploiting Arabic diacritization for high quality automatic annotation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4298–4304.

Habash, Nizar Y. 2010. *Introduction to Arabic Natural Language Processing*, volume 10 of *Synthesis Lectures on Human Language Technologies* (Graeme Hirst, ed.). Morgan & Claypool Publishers. https://doi.org/10.1007/978-3-031-02139-8

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735, PubMed: 9377276

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

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

Ibrahim, Raphiq. 2013. Reading in Arabic: New evidence for the role of vowel signs. *Creative Education*, 4(4):248–253. https://doi.org/10.4236/ce.2013.44036

Katz, Leonard and Ram Frost. 1992. The reading process is different for different orthographies: The orthographic depth hypothesis. In *Advances in Psychology*, volume 94. Elsevier, pages 67–84. https://doi.org/10.1016/S0166-4115(08)62789-2

Laki, László János and Zijian Gyozo Yang. 2020. Automatic diacritic restoration with transformer model based neural machine translation for east-central European languages. In *Proceedings of the 11th International Conference on Applied Informatics (ICAI)*, pages 190–202.

Liberman, Isabelle Y., Alvin M. Liberman, Ignatius Mattingly, and Donald Shankweiler. 1980. Orthography and the beginning reader. In J. F. Kavanagh and R. L. Vinezky, editors, *Orthography, Reading, and Dyslexia*. University Park Press, Baltimore, MD, chapter 10, pages 137–153.

Madhfar, Mokthar Ali Hasan and Ali Mustafa Qamar. 2020. Effective deep learning models for automatic diacritization of Arabic text. *IEEE Access*, 9:273–288. https://doi.org/10.1109/ACCESS.2020.3041676

Marcus, Mitchell Philip. 1978. *A Theory of Syntactic Recognition for Natural Language*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Mijlad, Ali and Yacine El Younoussi. 2019. Arabic text diacritization: Overview and solution. In *Proceedings of the 4th International Conference on Smart City Applications*, SCA '19, pages 1–7. https://doi.org/10.1145/3368756.3369088

Mubarak, Hamdy, Ahmed Abdelali, Hassan Sajjad, Younes Samih, and Kareem Darwish. 2019. 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. https://doi.org/10.18653/v1/N19-1248

Náplava, Jakub, Milan Straka, Pavel Straňák, and Jan Hajič. 2018. Diacritics restoration using neural networks. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 1566–1573.

Nga, Cao Hong, Nguyen Khai Thinh, Pao-Chi Chang, and Jia-Ching Wang. 2019. Deep learning based Vietnamese diacritics restoration. In *2019 IEEE International Symposium on Multimedia (ISM)*, pages 331–334. https://doi.org/10.1109/ISM46123.2019.00074

Nguyen, Kiet, Vu Nguyen, Anh Nguyen, and Ngan Nguyen. 2020. A Vietnamese dataset for evaluating machine reading comprehension. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2595–2605. https://doi.org/10.18653/v1/2020.coling-main.233

Nuţu, Maria, Beata Lőrincz, and Adriana Stan. 2019. Deep learning for automatic diacritics restoration in Romanian. In *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 235–240. https://doi.org/10.1109/ICCP48234.2019.8959557

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. https://doi.org/10.3115/1073083.1073135

Shmidman, Avi, Shaltiel Shmidman, Moshe Koppel, and Yoav Goldberg. 2020. Nakdan: Professional Hebrew diacritizer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 197–203. https://doi.org/10.18653/v1/2020.acl-demos.23

Stankevičius, Lukas, Mantas Lukoševičius, Jurgita Kapočiūtė-Dzikienė, Monika Briedienė, and Tomas Krilavičius. 2022. Correcting diacritics and typos with a ByT5 transformer model. *Applied Sciences*, 12(5):2636. https://doi.org/10.3390/app12052636

Taha, Haitham. 2016. Deep and shallow in Arabic orthography: New evidence from reading performance of elementary school native Arab readers. *Writing Systems Research*, 8(2):133–142. https://doi.org/10.1080/17586801.2015.1114910

Uzun, Aysenur. 2018. Diacritic restoration using neural network. Technical report, Computer Engineering, Istanbul Technical University.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008.

Wang, Yuxuan, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Z. Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Robert A. J. Clark, and Rif A. Saurous. 2017. Tacotron: Towards end-to-end speech synthesis. In *Proceedings of INTERSPEECH*, pages 4006–4010. https://doi.org/10.21437/Interspeech.2017-1452

Xue, Linting, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306. https://doi.org/10.1162/tacl_a_00461

Zalmout, Nasser and Nizar Habash. 2020. Joint diacritization, lemmatization, normalization, and fine-grained morphological tagging. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8297–8307. https://doi.org/10.18653/v1/2020.acl-main.736

Zerrouki, Taha and Amar Balla. 2017. Tashkeela: Novel corpus of Arabic vocalized texts, data for auto-diacritization systems. *Data in Brief*, 11:147–151. https://doi.org/10.1016/j.dib.2017.01.011, PubMed: 28224131

Zitouni, Imed and Ruhi Sarikaya. 2009. Arabic diacritic restoration approach based on maximum entropy models. *Computer Speech & Language*, 23(3):257–276. https://doi.org/10.1016/j.csl.2008.06.001

Zitouni, Imed, Jeffrey S. Sorensen, and Ruhi Sarikaya. 2006. Maximum entropy based restoration of Arabic diacritics. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 577–584. https://doi.org/10.3115/1220175.1220248