From Arabic to Hebrew: Looking Ahead So We Don't Fall Behind in Diacritization

Lior Baruch, Ben Jaim Catalan

August 20, 2023

1 Introduction

In the rapidly evolving field of natural language processing, the challenge of disambiguating textual content is a critical area of investigation. This paper focuses on the implementation of a novel approach proposed by Esmail et al. [1], originally published in 2022. The primary objective of the original paper was to reduce ambiguity in text by adding partial diacritization, a method that has far-reaching implications for understanding complex languages.

The ambiguity addressed in this work is situated at the morphological level, involving the nuanced understanding of word forms and structures. The core idea is to incorporate just enough partial diacritization to enable clear comprehension of the words without the need to look ahead in the text. In essence, disambiguating diacritics are employed when the most likely interpretation of a word or phrase is ambiguous without them [2]. This innovative approach fosters greater readability and precision in text interpretation.

By attempting to reproduce and extend the work of Esmail et al., this paper contributes to the ongoing exploration of advanced techniques in text disambiguation. The subsequent sections detail the specific solution implemented, experimental results, and a comprehensive discussion of the findings, situating this study within the broader context of diacritization research.

1.1 Related Works

As far as we looked, we couldn't find a similar work other than Esmail et al. [1]

For just full diacritization there are several works and papers such as:

Shmidman et al. [3]

Gershuni-Pinter [4]

As part of our project, we used the Dicta dataset [5] containing over 100 books containing mainly of biblical and rabbinical texts. We trained over sentences, and since there were at least over 1m sentences and we had limited resources for this project, we didn't use all the dataset but about 300k sentences.

2 Solution

2.1 General approach

In our pursuit to enhance text disambiguation through partial diacritization, we have followed the work of Esmail et al. [1] and developed two distinct models:

1. Reading direction model: Emulating the natural reading direction of human readers, this model applies diacritization step by step as the text is processed. It employs Uni-Directional LSTM to mimic human reading patterns.

2. full sentence model: This model scans the entire sentence before proceeding with diacritization. It leverages BiLSTM to analyze the sentence structure and apply appropriate diacritical marks.

The synergy between these models lies in their combined inference process. Both models strive to fully restore the diacritization of a sentence, and at inference time, diacritization is assigned only to letters where there is disagreement between the models. In such cases, the diacritic from the Full Sentence Model is selected, capitalizing on its anticipated superior performance.

Generally speaking, the input is composed of a sequence of Hebrew characters $h_1, h_2, \ldots, h_n$ and the target is to predict a single label d_i for each character h_i,representing it's diacritic.

Data cleaning and tokenization

The Dicta dataset includes many books and writing with diactrics. Our first step was taking each text and splitting it into sentences (in our work sentences were split by "." or "\n").

Since we wanted to focus on the main diactrics which are as follows:

Name	Symbol	Name	Symbol
Sheva	ְ	Qubuts	ֻ
Patah	ַ	Hiriq	ִ
Qamatz	ָ	Hataf patah	ֲ
Tzeri	ֵ	Hataf Qamatz	ֳ
Segol	ֶ	Hataf Segol	ֱ
Holam	ֹ		

We also added the label for a character "<no_nikud>" which represnts that no diactric was added to the character.

After working with the dataset, we found out there are a bit more diactric combinations in the dataset, which we treated as more labels.

There are more diactrics which are character specific (Dagesh, shin dot, etc.), or refer to a more specific pronounciation (like vowel length), which we will leave for future work.

So for each sentence we removed all diactrics which are not part of the table above, and other marks such as " (Quotes), ', (,).

After words, for each sentence, we saved the full sentence, the sentece without the diactrics and in a separate list, the diactrics themselves.

For the tokenization process we used AlephBert's [6] tokenizer after editing it's vocabulary file manually to contain only tokens with at most one hebrew character (the original tokenizer tokenizes to words as well).

Similar to Esmail et al. [1] we also trained our model only on sentences with at most 100 characters (not including spaces), because of the robustness of data we also only used sentences of up to 100 characters for test and validation purposes.

Reading-Direction model

For this model we use a 4-layer unidirectional long short-term memory (LSTM) architecture that works on the character level, and predicts one label er input character. This is based on the same architecture proposed in Esmail et al. [1]

Full sentence model

For this model we use a 4-layer bidirectional LSTM (BiLSTM) architecture that works on the character level, and predicts one label per input character. We originally tried the same architecture as Esmail et al. [1] (transformer based), but didn't get sufficient results.

General architecture for both models

In general for a word $j = c_1 c_2 \dots c_n$, where c_i represents character embedding, for both architectures we first obtain

$$w_j = \text{BiLSTM}\left(c_1, c_2, \dots, c_n\right)$$

then we obtain for each character i, BiLSTM $([c_i; w_j])$ and LSTM $([c_i; w_j])$ for the full sentence model and reading direction model, respectively. And lastly, each of those results is sent to a fully connected layer to generate the final prediction (each model has it's own fully connected layer).

Training the models

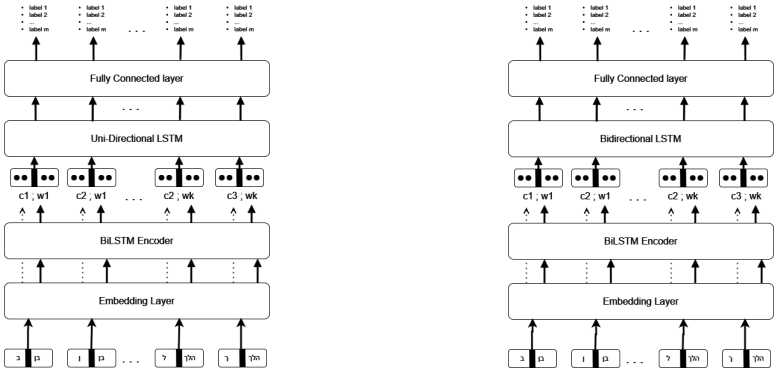
The training phase of both the Reading Direction and Full Sentence models was a critical component of our solution. To optimize the performance of each model, we embarked on an extensive hyperparameter tuning process using a grid search approach. Below are the details of the training strategy and the selected hyperparameters for each model.

Hyperparameter Tuning

We explored various configurations by tuning the following hyperparameters for both models:

- **Word and Character embedding dimension (as tuples):** (16,32), (32, 64), (64, 128)

Figure 1: Models Architecture



- **Hidden Size:** 512, 1024
- **Learning Rate:** 0.001, 0.0001
- **Weights For No Diactric ('<no_nikud>'):** 1, 0.5
- **Epochs:** 1-20 (selecting the epoch with the best validation accuracy)

Due to constraints in time and computing resources, the grid search was conducted on a randomly chosen subset of 5000 sentences from the dataset. The entire process took approximately 24 hours for each architecture, totaling 48 hours.

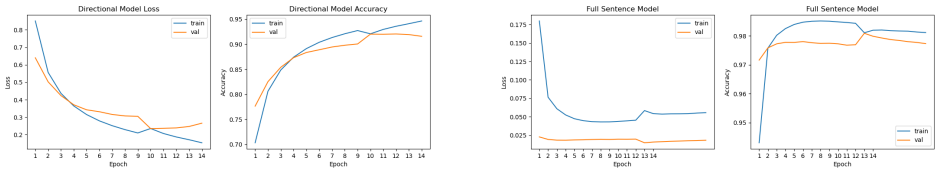
Best Model Configurations

- **Reading Direction Model**
  - Word Embedding: 32-dimensional
  - Character Embedding: 64-dimensional
  - Hidden Size: 512
  - Weight for No Diacritic: No weight
  - Learning Rate: 0.001
- **Full Sentence Model**
  - Word Embedding: 16-dimensional
  - Character Embedding: 32-dimensional
  - Hidden Size: 512
  - Weight for No Diacritic: No weight
  - Learning Rate: 0.001

With the optimal hyperparameters identified, we trained each model for about 15 epochs on a dataset comprising approximately 200,000 sentences, with around 50,000 reserved for validation. Each epoch ran for about 1h20m (GTX 3080).

- **Reading Direction Model:** The best epoch for this model was identified at epoch 12, signifying the point where the model achieved optimal validation performance.
- **Full Sentence Model:** Similarly, the Full Sentence Model's best performance was achieved at epoch 13.

Figure 2: Models Results



These configurations and training epochs represent our best efforts to balance model complexity, performance, and efficiency, leading to robust models tailored for the task of diacritization restoration.

3 Experimental results

In order the evaluate our models performence, we used the Diacritization Error Rate (DER) / Word Error Rate (WER) metrics, expressed as percentages. These metrics were defined by Fadel et al. (2019b) and align with the evaluation approach of Esmail et al.

Reading direction model

	Including NO NIKUD	Excluding NO NIKUD
Our model	7.98 / 21.84	12.3 / 19.3

(add confusion matrices)

Full sentence model

	Including NO NIKUD	Excluding NO NIKUD
Our model	1.93 / 5.24	2.97 / 4.61
Dicta	3.23 / 5.89	- / -
Nakdimon	3.63 / 10.25	- / -

* Results of Dicta and Nakdimon taken from Gershuni-Pinter [4]

For the Partial Restoration, since we don't have any human annotator in order to make a real comparison with our results, we leave this for future work, but examples are provided below to illustrate how it helps to lessen ambiguity in the sentences:

	Original sentence	Full sentence model pred.	Reading direction model pred.	Final result
1.	וּבְמַסָּךְ פָּאָה בַּיִּאַר מִילַת מֵירוּחַ עִנְיָנוֹ כְּמוֹ מֵירוֹק שֶׁמְּמִירָק הַתָּבָן מִן הַתְּבוּאָה	וּבְמַסָּךְ פָּאָה בַּיִּאַר מִילַת מֵירוּחַ עִנְיָנוֹ כְּמוֹ מֵירוֹק שֶׁמְּמִירָק הַתָּבָן מִן הַתְּבוּאָה	וּבְמַסָּךְ פָּאָה בַּיִּאַר מִילַת מִירוּחַ עִנְיָנוֹ כְּמוֹ מִירוּק שֶׁמְּמִירָק הַתָּבָן מִן הַתְּבוּאָה	ובמס פאה ביאר מילת מֵירוח ענינו כמו מֵירוֹק שמֵירָק התָּבָן מן התבואה
2.	וּמְדַיְּקִין לְשׁוֹן הַגְמִי אִי טְעִין לַיְתֵי ב סָהֲדֵי דְּדָרוּ בֵּי יְמָמָא וְלֵילְיָא דְּלִכְאוֹרָה גַּם לְפִי הָרַמְבַּם לָמָּה זֶה שֶׁאוֹמֵר יָבִיא עֵדִים כו	וּמְדַיְּקִין לְשׁוֹן הַגְמִי אִי טְעִין לַיְתֵי ב סָהֲדֵי דְּדָרוּ בֵּי יְמָמָא וְלֵילְיָא דְּלִכְאוֹרָה גַּם לְפִי הָרַמְבַּם לָמָּה זֶה שֶׁאוֹמֵר יָבִיא עֵדִים כו	וּמְדַיְּקִין לְשׁוֹן הַגְמִי אִי טְעִין לְיְתֵי ב סָהֲדֵי דְּדָרוּ בֵּי יְמָמָא וְלֵילְיָא דְּלִכְאוֹרָה גַּם לָפֵי הָרַמְבַּם לָמָּה זֶה שֶׁאוֹמֵר יָבִיא עֵדִים כו	ומדויק לשון הגמ אי טעין לַיתי ב סָהֲדֵי דדרו בי יְמָמָא ולֵילְא דלכאורה גם לפי הרמבם למה זֶה שאומר יביא עֵדִים כו
3.	שֶׁהַפָּגָם הוּא מִלְשׁוֹן פְּגִימַת הַסַּכִּין שֶׁנַּעֲשָׂה חו כְּמוֹ פְּגָמַי וְנִקְבֵי בְּכֵלִים דזא כו כָּנוֹדָע במש בקש שֶׁעַל הַמִּטָּה אִם פְּגָמְתִּי בָּאוֹת י כו	שֶׁהַפָּגָם הוּא מִלְשׁוֹן פְּגִימַת הַסַּכִּין שֶׁנַּעֲשָׂה חו כְּמוֹ פְּגָמַי וְנִקְבֵי בְּכֵלִים דזא כו כָּנוֹדָע במש בקש שֶׁעַל הַמִּטָּה אִם פְּגָמְתִּי בָּאוֹת י כו	שֶׁהַפָּגָם הוּא מִלְשׁוֹן פְּגִימַת הַסַּכִּין שֶׁנַּעֲשָׂה חוֹ כְּמוֹ פְּגָמַי וְנִקְבֵי בְּכלִים דזא כו כנוֹדָע במש בקש שֶׁעלְ הַמִּטָּה אִם פְּגָמתִי בְּאוֹת י כו	שהפָּגָם הוא מלשון פגימת הסַכין שנעשה חו כמו פְּגָמַי ונקבי בכֵלים דזא כו כָּנודע במש בקש שעַל המטָה אם פָּגמְתי בָּאות י כו

Figure 3: Confusion Matrices

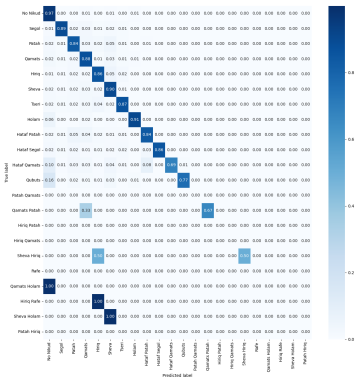


Figure 4: Reading Direction Confusion Matrix

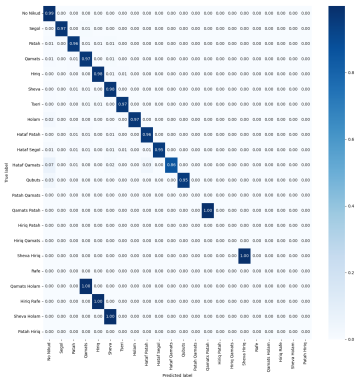


Figure 5: Full Sentence Confusion Matrix

4 Discussion

4.1 Reproduction of Previous Work

In this study, we embarked on the ambitious task of reproducing the work of Esmail et al. in the context of Hebrew. Our efforts yielded promising results, demonstrating the adaptability of the original methodology.

4.2 Challenges and Limitations

Due to time constraints and a lack of compatible Hebrew datasets with diacritics, our work was limited to Dicta's dataset, containing texts from the late 19th to early 20th centuries. This constraint prevented us from training and testing our models on modern texts, a challenge we plan to address in future work. Additionally, our attempt to implement the full-sentence model with the same architecture as Esmail et al. encountered difficulties, leading us to opt for the BiLSTM architecture.

4.3 Achievements

Despite these challenges, our models achieved results quite close to those of Esmail et al. We attribute this success in part to the close linguistic characteristics of Arabic and Hebrew, both Semitic languages. Additionally, our exploration of partial diacritization yielded encouraging findings. From the few examples we selected, it seems that partial diacritization effectively contributed to our goal of reducing ambiguity, indicating a promising direction for further refinement and application.

4.4 Future Directions

We envision numerous future applications and enhancements for this work. For example, in the realm of education, our models could assist children or new immigrants learning hebrew transitioning from reading with full diacritics to no diacritics. Further research might also explore how this approach can be extended to professional texts, such as books, newspapers, and other written media. This expansion into various text domains could lead to broader applicability and impact, enriching our understanding of diacritization in modern language processing.

5 GitHub Link

GitHub repository

References

[1] Esmail et al. “How Much Does Lookahead Matter for Disambiguation? Partial Arabic Diacritization Case Study”
Paper link

[3] Shmidman et al. “Nakdan: Professional Hebrew Diacritizer”
Paper link

[4] Gershuni-Pinter “Restoring Hebrew Diacritics Without a Dictionary”
Paper link

[5] Dicta dataset
Dataset link

[2] Usage and examples of partial diactrics in hebrew (article is in hebrew)
Hebrew partial diactrics

[6] AlephBert GitHub page
GitHub