Arabic Text Diacritization

Deep Learning Approach

Problem Statement

كُتُبٌ
Books

كَتَبَ
Wrote

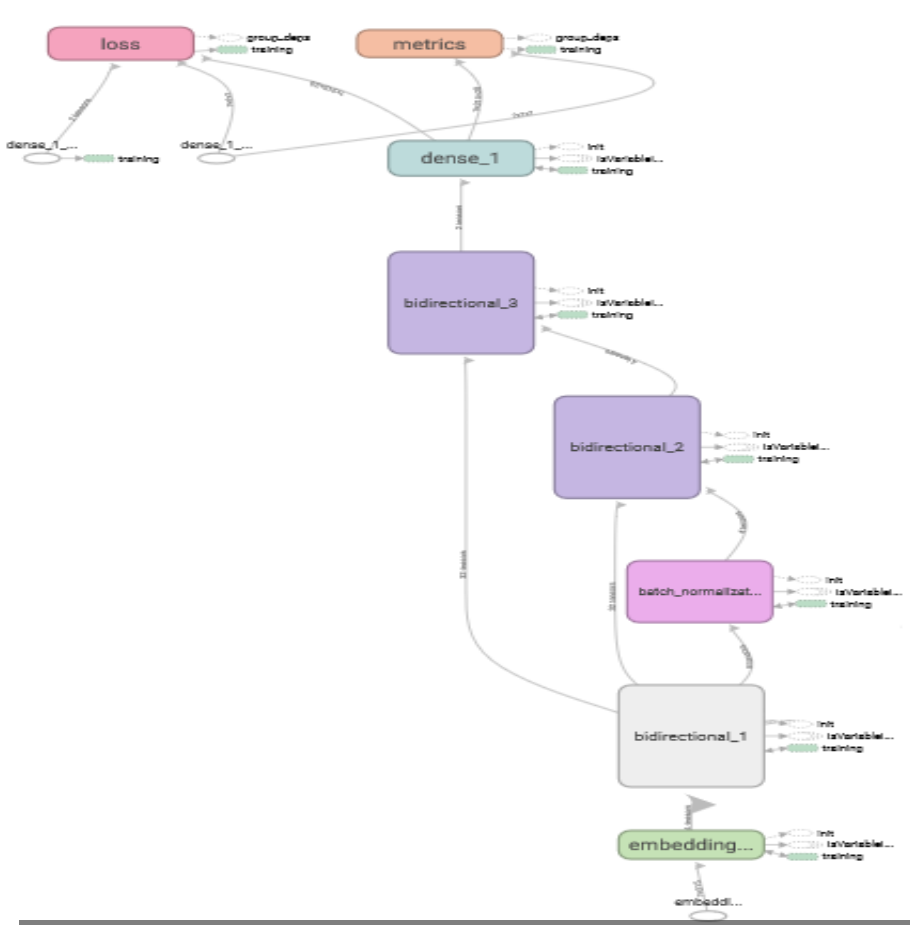
كتّب
Battalions

Problem Statement

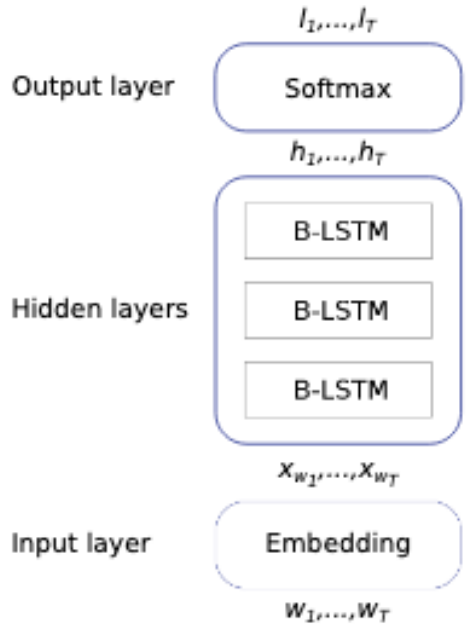
- In Arabic and many other languages, diacritics are added to the characters of a word in order to convey certain information about the meaning of the word as a whole and its place within the sentence
- Arabic Text Diacritization is an important problem with various applications such as text to speech
- Diacritics are important in Arabic it changes the meaning of the word and changes the pronunciation as well

Related Work

- There exist two approaches to Arabic Text Diacritization:
  - Traditional rule-based approach
  - Machine learning approach



Shakkala Project



Arabic Diacritization with RNN

Related Work

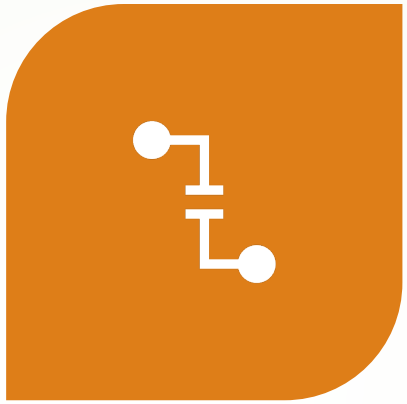
Related Work

- Expanding on Machine Learning approach
- If we have time after deploying our proposed solution and reach a reasonable results, we want to expand our work to the field of NLP and Machine Translation while Arabic-text-Diacritization is essential for applications such as Text-to-Speech

Proposed Solution

- Use convolution neural network approach in addition to RNN
- Use word embedding instead of character embedding

Original Models



FEED-FORWARD NEURAL NETWORK MODEL

RECURRENT NEURAL NETWORK MODEL

Feed-Forward Model

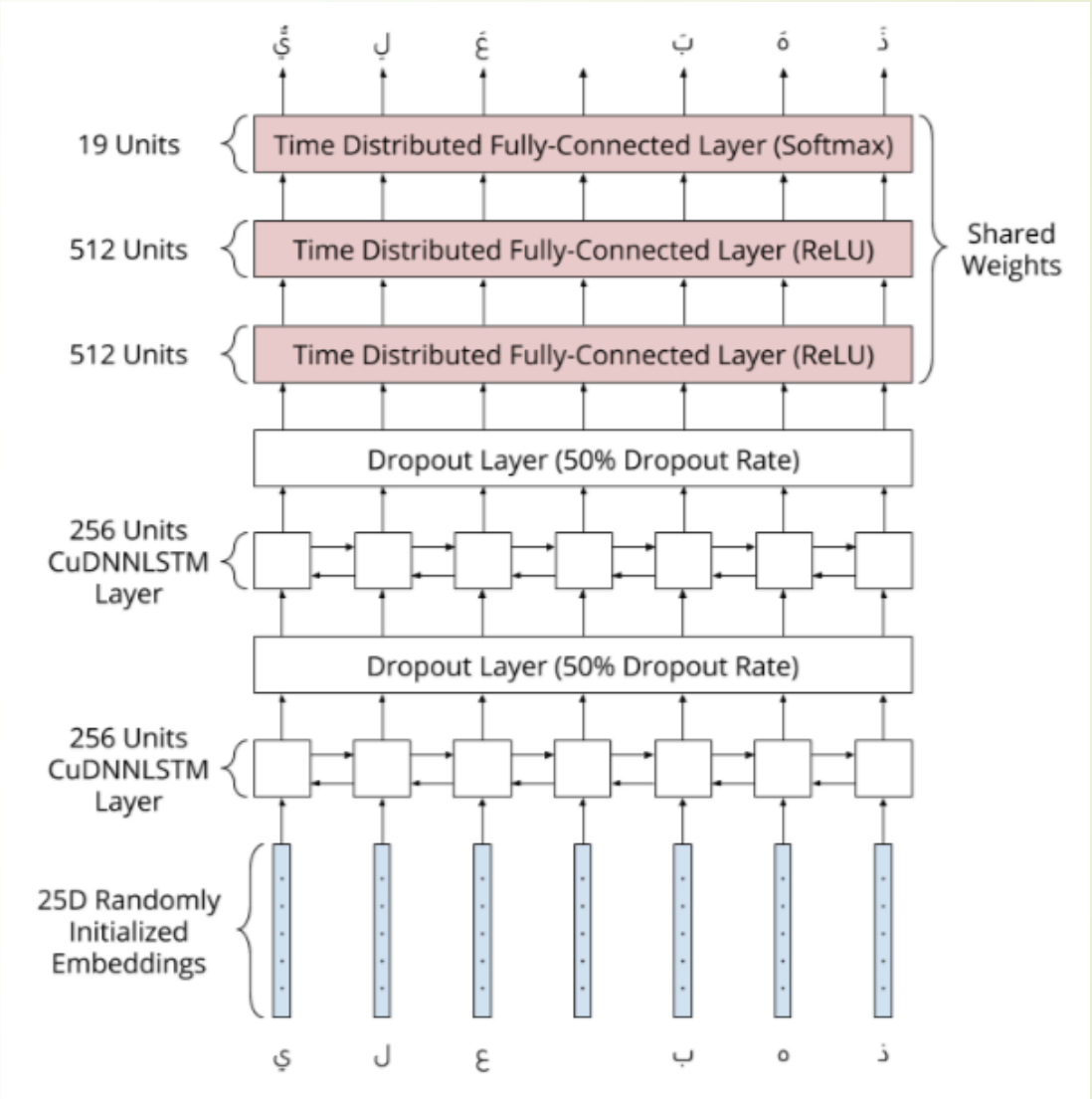
- The model has five hidden layers with 728K trainable parameters
- 100-hot input embeddings layer to learn feature vectors for each character through the training process
- Performed much better than the best rule-based Diacritization, Mishakl DER: 13.78% vs FFNN Embeddings model DER: 4.06%

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 100, 25)	2075
flatten (Flatten)	(None, 2500)	0
dropout (Dropout)	(None, 2500)	0
dense (Dense)	(None, 250)	625250
dense_1 (Dense)	(None, 200)	50200
dense_2 (Dense)	(None, 150)	30150
dense_3 (Dense)	(None, 100)	15100
dense_4 (Dense)	(None, 50)	5050
dense_5 (Dense)	(None, 15)	765

Total params: 728,590
Trainable params: 728,590
Non-trainable params: 0

RNN Model

- They used two Bidirectional CuD-NN Long Short-Term Memory (BiCuDNNLSTM) as well as 256 hidden units per layer
- Using less units will increase the error rate
- Conclusions they reached:
  - the depth is not as important as the size of each layer
  - Best results produces with model with 2 layers with 512 hidden units each



Deploying the Original Model

- We managed to deploy all models proposed by the paper (feed-forward neural network and RRN models)
- Paper's contributors were very helpful and responsive they respond to our issues very fast
- We faced computational issues while training models because the model need high performance capabilities, while Google Collaboratory provide us a GPU that terminates while training.
- We overcame this issue by saving checkpoints of the model and start training again from the saved before model

Proposed Solution Progress

- We developed a convolutional neural network model and get the initial results of the model.
- Next, we want to fine-tuning the model architecture and train on more epochs
- Then we will apply word embedding instead of the already used character embedding

Evaluation Used

- Diacritic Error Rate (DER): the percentage of misclassified Arabic characters
- Word Error Rate (WER): the percentage of Arabic words which have at least one misclassified Arabic character

	With case ending	Without case ending	With case ending	Without case ending
DER	Including no diacritic		Excluding no diacritic	
%	1.98	1.59	2.30	1.84

	With case ending	Without case ending	With case ending	Without case ending
WER	Including no diacritic		Excluding no diacritic	
%	5.95	3.47	5.77	3.41

Initial Results

- Recurrent Neural Network results with 25 epochs only

Initial Results

- We started training our model, but we faced some issues with Google Colab. So, to meet the deadline we did not include our initial results

Next steps and Timeline

- First, we are going to continue deploying our proposed solution
- Then, Continue training and tuning the model till we reach a reasonable performance and accuracies
- Get ready for the second milestone of the project exporting our results and compare it with the literature approaches
- If we have time as mentioned before we want to expand on the Arabic-English Machine Translation problem
- Finally, Finalize our work to the project delivery phase