Computer Engineering Department
Faculty of Engineering
Cairo University

CMPElective
4 Fall 2023

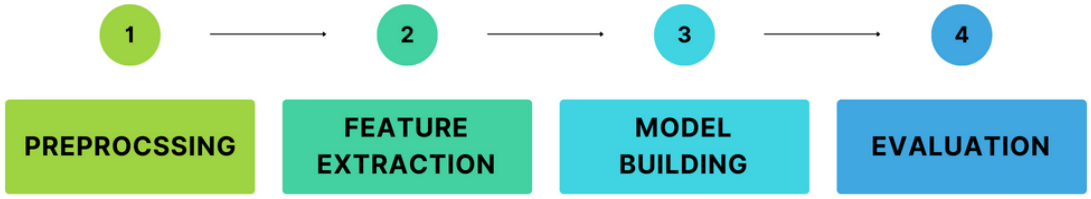
Natural Language Processing

Project Document

Arabic text diacritization

Name	Sec	BN
Waleed Hesham	2	36
Mahmoud Reda	2	20
Heba Ashraf	2	32
Donia Gameel	1	24

1. Project Pipeline



2. Detailed Description of Each Phase

i. Data Preprocessing

- Cleaning Text Data: remove characters which isn't an Arabic character nor a diacritic
- Tokenization: The NLTK library's word_tokenize function was used for tokenizing the cleaned text into words.
- Extracting Arabic Characters and Diacritics: generate lists of characters without diacritics (**characters**), diacritics corresponding to each character (**Diacritics_characters**), words without diacritics (**words**), and diacritics of the last character in each word (**Diacritics_words**).
- Reading the dataset: reading lines from the dataset, cleaning each line, tokenizing it, and subsequently extracting Arabic characters and diacritics.

ii. Feature Extraction

- One-Hot Vectors: Representing each character as a one-hot vector, capturing the presence or absence of each element in a fixed-size vector.
- Word2Vec Embeddings: Creating dense vector representations for words using Word2Vec, capturing semantic relationships between words.
- Trainable Embeddings: Utilizing trainable embeddings involves dynamically learning vector representations for words during the training of a neural network.
- Bag-of-Words (BOW): It involves creating vectors that represent the frequency or presence of each element (word or character) in a fixed-size vocabulary.

- **Word-level BOW:**
- **Character-level BOW**

iii. Model Training:

- **LSTM Model with One-Hot Encoding Vector for Each Character:**
  - Utilizes LSTM architecture with one-hot encoding vectors representing individual characters.
- **Bidirectional LSTM Model with One-Hot Encoding Vector for Each Character:**
  - Extends the LSTM model to a bidirectional variant, capturing context information from both directions.
- **Bidirectional LSTM Model with One-Hot Encoding Vector for Each Character and Trainable Embedding for Each Word:**
  - Combines bidirectional LSTM with trainable embeddings, allowing the model to dynamically learn vector representations for words.
- **Bidirectional LSTM Model with One-Hot Encoding Vector for Each Character and Word2Vec for Each Word:**
  - Integrates bidirectional LSTM with Word2Vec embeddings, capturing semantic relationships between words.
- **Bi-Directional LSTM Model with Character Embedding as Features and Character Diacritics as Labels:**
  - Utilizes a bidirectional LSTM where character embeddings serve as input features, and character diacritics act as labels for the model to predict.

3. Evaluation

Model	Score
LSTM model with one hot encoding vector for each character	51.56% (validation)
Bidirectional LSTM model with one hot encoding vector for each character	51.56% (validation)

Bidirectional LSTM model with one hot encoding vector for each character and trainable embedding for each word	60% (validation)
Bidirectional LSTM model with one hot encoding vector for each character and word2vec for each word	68% (validation)
Bi-Directional LSTM model with character embedding as features and character diacritics as labels.	99.073(validation) 93.39% (first leaderboard)

4. Model for Kaggle Submission

We used Bi-Directional LSTM model with character embedding as features and character diacritics as labels. The model has the following parameters:

Loss function: cross entropy

Output layer activation: linear function (we make argmax to the model output)

Vocab size: number of Arabic letters + 2 (space + padding symbol)

Embedding dimension: 250

Hidden size: 100

Number of layers: 2

Dropout: 0.5

Learning Rate: 0.01

Train Batch size: 128

Evaluation Batch size: 256

Epochs: 5