Arabic Autocomplete System - Documentation

# 1. Introduction

The Arabic Autocomplete System is a machine learning-based project aimed at predicting and suggesting the next possible word(s) in a sequence of Arabic text input. This system is built to facilitate faster and more efficient text input for Arabic speakers, particularly in digital communication platforms and smart keyboard applications.

# 2. Project Overview

This system employs a neural network-based language model trained on a large Arabic text dataset. It uses sequential modeling techniques to understand language patterns and provide accurate word predictions.

# 3. Dataset Information

The dataset used for training is a corpus of Arabic text collected from various sources such as Arabic news websites, Wikipedia articles, and literature. The dataset consists of approximately 1 million sentences and over 10 million words.

Preprocessing steps include:  
- Tokenization  
- Normalization (removing diacritics, unifying different forms of letters)  
- Removing punctuation and non-Arabic characters  
- Padding and truncating sequences to a fixed length

# 4. Model Details

The core of the system is built using an **n-gram language model**, which is a statistical model commonly used in natural language processing for predicting the next item in a sequence. This approach does not rely on deep learning but instead on the frequency of n-word sequences (n-grams) in the training data.

#### Model Architecture:

* **Type**: N-gram-based probabilistic language model
* **N-gram size**: Trigram (n=3), meaning the model uses the previous two words to predict the next word.
* **Tokenization**: The input text is tokenized into individual words or subword units.
* **Vocabulary Size**: Limited to the top 20,000 most frequent tokens to reduce sparsity.
* **Prediction Mechanism**: For each input context of n-1 words, the model looks up the most probable next word based on the frequency distribution in the training corpus.

#### Training Parameters:

* **Smoothing Technique**: Add-one (Laplace) smoothing applied to handle unseen n-grams.
* **Training Strategy**: Count-based learning from the dataset by constructing a dictionary of n-gram frequencies.
* **No gradient descent or backpropagation** is involved, as this is a statistical model.

#### Evaluation:

* **Validation Method**: A held-out validation set (20% of the dataset) is used to test accuracy and prediction quality.
* **Metrics**: Perplexity and prediction accuracy are used to evaluate the model’s performance.

# 5. Evaluation Metrics

The model is evaluated using Perplexity and Accuracy:  
- Perplexity: A measure of how well the model predicts the next word  
- Top-k Accuracy: Accuracy of predicting the correct word within the top k suggestions (e.g., top-1, top-3, top-5)

# 6. Model Limitations

- Limited to the vocabulary seen during training; out-of-vocabulary words are not handled well.  
- Accuracy may decrease for informal or dialectal Arabic.  
- Requires significant computational resources for training on large datasets.  
- Longer input sequences can lead to context dilution in LSTM models.

# 7. Future Work

- Integrating attention mechanisms or using transformer-based models (e.g., BERT, GPT) for better context handling.  
- Expanding the dataset to include dialects and conversational Arabic.  
- Deploying the model in a mobile-friendly format (e.g., TensorFlow Lite) for integration into smart keyboards.