# Arabic Part of Speech Tagging “POS” with NetworkX

# Table of Contents:

# Introduction:

The problem statement arises from the persistent challenges associated with Part-of-Speech (POS) tagging for the Arabic language. Despite advancements in natural language processing (NLP) techniques, the complex morphology and dialectal variations inherent in Arabic pose obstacles to achieving accurate and context-aware POS tagging. Compounding this challenge is the scarcity of robust Arabic word embedding techniques, hindering the development of effective models. What I aim to accomplish with this project is to strategically explore techniques for Arabic word embeddings before delving into POS tagging model development. This iterative approach aims to establish a solid foundation for understanding Arabic linguistic structures, ultimately advancing the interpretability of POS tagging in diverse contexts.

## Project Overview:

The project unfolds through progressive steps, placing a strategic emphasis on addressing the limitations in Arabic word embeddings prior to developing the POS tagging model.

**1. Exploring Techniques for Arabic Word Embeddings:**

- Investigate and experiment with various techniques to develop Arabic word embeddings.

- Prioritize techniques that capture semantic nuances and account for the complex morphological structures of the Arabic language.

**2. Exploring Techniques for Arabic Word Embeddings:**

- Address the scarcity of Arabic word embeddings by using Arabic Word Embeddings techniques.

- Develop embeddings that capture semantic nuances and complexities present in Arabic.

**3. POS Tagging Model Development for Arabic:**

- Once robust Arabic word embeddings are established, proceed to develop POS tagging models tailored for Arabic.

- Leverage the refined embeddings to enhance the accuracy and interpretability of the POS tagging model.

**4. NetworkX Representation of POS tags**:

- Employ NetworkX to construct graphs representing the POS-tagged Arabic text.

- Enhance the visual representation of Arabic linguistic structures for improved interpretability.

By strategically placing the exploration of Arabic word embeddings at the forefront, the project aims to overcome the challenges posed by the complex Arabic language. This iterative approach ensures that subsequent steps, including POS tagging model development and network analysis, build upon a solid foundation of effective word embeddings tailored to the intricacies of Arabic.

# Data Description:

The Arabic-PADT UD treebank is derived from the Prague Arabic Dependency Treebank (PADT) and is composed of 7,664 sentences with 320,546 tokens. Originally in CoNLL-U format, the dataset has been transformed into a CSV format structured with 10 columns. The primary focus is on the following four columns:

**1. ID:**

- Status: Sentence-level units, obtained automatically and manually corrected.

**2. FORM:**

- Description: Unvocalized surface form; vocalized counterpart available in the MISC column as the Vform attribute.

**3. LEMMA:**

- Description: Plausible analyses from ElixirFM with manual disambiguation. Lemmas are vocalized, and word sense disambiguation provides English equivalents.

**4. UPOSTAG:**

- Description: Automatically converted from XPOSTAG; human-checked for consistency.

Dataset Characteristics:

- **Number of Instances (Rows):** 320,546

- **Number of Features (Columns):** 10

The dataset is further divided into training, testing, and validation sets, with the following token counts:

- **Training Set:** 254,317 tokens (nearly **80%**)

- **Testing Set:** 32,043 tokens (nearly **10%**)

- **Validation Set:** 34,186 tokens (nearly **10%**)

These distributions, expressed as percentages relative to the entire dataset, offer insights into the composition of each set within the overall dataset.

For our project, our primary focus will be on the **FORM** and **UPOSTAG** columns, as they play a crucial role in shaping our semantic analysis and subsequent tasks. These columns provide information about the unvocalized surface form and the automatically converted part-of-speech tags, respectively, serving as key elements for our data exploration and modeling efforts.

## Data Processing:

In preparing the Arabic-PADT UD treebank for modeling, several preprocessing steps were undertaken to ensure data integrity and suitability for semantic analysis. The following steps were implemented:

**1. Filtering Unnecessary Labels:**

- Irrelevant labels were filtered from the dataset, streamlining the information to only those pertinent for the subsequent tasks.

**2. Collation of Tokens to Form Sentences:**

- Tokenized entries were collated to form cohesive sentences, resulting in a total of 7,664 complete sentences.

**3. Tokenization with Train Data's Tokenizer:**

- Tokenization was performed individually on the training, testing, and validation sets using a tokenizer instance fitted exclusively on the training data. This approach mitigates the risk of data leakage, ensuring each set is processed independently.

**4. Padding for Model Suitability:**

- Padding was applied to the tokenized sets to homogenize the sequence lengths, rendering the data suitable for subsequent modeling.

**5. Embedding Weights Matrix Creation using AraVec:**

- AraVec, an Arabic word embedding technique, was applied to generate an embedding weights matrix. This matrix captures the semantic relationships between words, enhancing the model's understanding of the Arabic language nuances.

The meticulous preprocessing steps ensure that the dataset is appropriately shaped and ready for semantic analysis. By applying AraVec, the model gains access to enriched word embeddings, further empowering its ability to discern intricate semantic structures within the Arabic-PADT UD treebank

## Experimental Configurations:

For all the conducted experiments in Arabic POS tagging, a consistent set of configurations was applied to maintain uniformity across the models.

**Embedding Dimension:**

- Set the embedding dimension to 300 for all experiments. This dimensionality captures rich semantic information for Arabic POS tagging.

**RNN Cells:**

- Utilized 64 RNN cells in Vanilla RNN, RNN with Trainable Embeddings, and RNN with Pre-trained Embeddings.

**LSTM Cells:**

- For LSTM experiments, maintained 64 LSTM cells to capture long-term dependencies effectively.

**BiLSTM Cells:**

- Applied 64 Bidirectional LSTM (BiLSTM) cells to incorporate both forward and backward contextual information.

**Loss Function:**

- Employed Categorical Crossentropy as the loss function to measure the dissimilarity between predicted and actual POS tags.

**Optimizer:**

- Utilized the Adam optimizer to efficiently adjust weights during training, optimizing the learning process.

**Batch Size:**

- Configured the batch size to 128 for efficient mini-batch training, balancing computational efficiency and model performance.

**Epochs:**

- Set the number of epochs to 10 for each experiment, allowing the models to iteratively learn from the training data.

These consistent configurations across all experiments ensure a fair and comparable evaluation of each model's performance in Arabic POS tagging. The standardized settings contribute to the robustness and reliability of the experimental results.

# Baseline Experiment: Vanilla RNN for Arabic POS Tagging

## Goal:

Establish a baseline for Arabic Part-of-Speech (POS) tagging using a Vanilla Recurrent Neural Network (RNN) without leveraging pre-trained word embeddings. Randomly initialize embeddings and restrict weight updates during training.

## Steps:

1. Embedding Initialization:

- Initialize Embeddings: Create embeddings with random initialization for each word in the Arabic vocabulary.

2. Model Architecture:

- Construct Vanilla RNN: Implement a Vanilla RNN architecture without updating the embedding weights during training.

3. Training:

- Train the Model: Train the Vanilla RNN model on the Arabic POS dataset, ensuring embeddings remain fixed.

4. Evaluation on Test Data:

Assess Arabic POS tagging accuracy and contextual understanding by evaluating model performance without trainable embeddings on the test data. I calculated two evaluation metrics, the weighted F1 score is considered the main metric as it accounts for class imbalance, which is crucial for this classification task. Accuracy, while measured, is not as interpretable for such types of problems.

```
Weighted F1-Score: 82.36 %
Accuracy: 85.16 %
```

## Conclusion:

The Vanilla RNN baseline experiment provides insights into the capability of simple recurrent neural networks for Arabic POS tagging without leveraging pre-trained embeddings. The model is trained with randomly initialized embeddings, and their weights remain static throughout training.

# Baseline Experiment: RNN with Uninitialized Trainable Embeddings for Arabic POS Tagging

## Goal:

Extend the Vanilla RNN baseline by allowing embeddings to be trainable during training while still starting with random initialization for Arabic POS tagging.

## Steps:

1. Embedding Initialization:

- Initialize Trainable Embeddings: Create trainable embeddings with random initialization for each word in the Arabic vocabulary.

2. Model Architecture:

- Construct RNN with Trainable Embeddings: Implement an RNN architecture with trainable embeddings for Arabic POS tagging.

3. Training:

- Train the Model: Train the RNN model on the Arabic POS dataset, allowing embeddings to be updated during training.

4. Evaluation on Test Data:

Assess Arabic POS tagging accuracy and contextual understanding by evaluating model performance with trainable embeddings on the test data, the weighted F1 score is considered the main metric.

```
Weighted F1-Score: 97.41 %
Accuracy: 97.5 %
```

## Conclusion:

This experiment explores the impact of allowing embeddings to be trainable during training for Arabic POS tagging, providing a comparative analysis against the fixed embeddings in Vanilla RNN. Notably, the weighted F1 score exhibits a significant improvement, soaring from 82% to an impressive 97%.

# Baseline Experiment: RNN with Pre-trained Embedding Weights (AraVec) for Arabic POS Tagging

## Goal:

Enhance the baseline RNN experiment for Arabic POS tagging by utilizing pre-trained embedding weights from AraVec.

## Steps:

1. Embedding Initialization:

- Utilize AraVec: Load pre-trained embedding weights from AraVec for each word in the vocabulary.

2. Model Architecture:

- Implement RNN with Pre-trained Embeddings: Construct an RNN architecture using the pre-trained embedding weights for Arabic POS tagging.

3. Training:

- Train the Model: Train the RNN model on the Arabic POS dataset while allowing embeddings to be updated during training.

4. Evaluation on Test Data:

Assess Arabic POS tagging accuracy and contextual understanding by evaluating model performance on the test data, the weighted F1 score is considered the main metric.

```
Weighted F1-Score: 97.19 %
Accuracy: 97.26 %
```

## Conclusion:

"The RNN with pre-trained embedding weights from AraVec aims to demonstrate the impact of leveraging external linguistic knowledge in enhancing Arabic POS tagging. However, it is noteworthy that no notable advancements were observed, as the last RNN model with trainable embeddings proved to be sufficient for the given data. To further validate these findings, we also employed two additional models for a comprehensive assessment.

# Next Experiments: LSTM and BiLSTM Models with Pre-trained Embedding Weights for Arabic POS Tagging

### Experiment 2: LSTM Model with Pre-trained Embedding Weights for Arabic POS Tagging

**Goal:**

Develop an LSTM model for Arabic POS tagging using pre-trained embedding weights from AraVec to capture long-term dependencies in semantic tasks.

**Steps:**

1. Embedding Initialization:

- Utilize AraVec: Load pre-trained embedding weights from AraVec for each word in the vocabulary.

2. Model Architecture:

- Implement LSTM with Pre-trained Embeddings: Construct an LSTM architecture using the pre-trained embedding weights for Arabic POS tagging.

3. Training:

- Train the Model: Train the LSTM model on the Arabic POS dataset with updating the embeddings while training.

4. Evaluation on Test Data:

- Evaluate POS Tagging Performance: Assess the LSTM model's ability to capture long-term dependencies and semantic nuances in Arabic POS tagging by evaluating model performance the test data, the weighted F1 score is considered the main metric.

```
Weighted F1-Score: 97.3 %
Accuracy: 97.38 %
```

**Conclusion:**

The LSTM experiment extends the baseline by introducing long-term dependency capture through a more complex architecture for Arabic POS tagging. However, it is important to note that there are no notable improvements in the weighted F1 score compared to the last two models. In light of this, we will explore potential enhancements with our upcoming model.

# Experiment 3: BiLSTM Model with Pre-trained Embedding Weights for Arabic POS Tagging

**Goal:**

Develop a Bidirectional LSTM (BiLSTM) model for Arabic POS tagging using pre-trained embedding weights from AraVec to capture both forward and backward contextual information.

**Steps:**

1. Embedding Initialization:

- Utilize AraVec: Load pre-trained embedding weights from AraVec for each word in the vocabulary.

2. Model Architecture:

- Implement BiLSTM with Pre-trained Embeddings: Construct a BiLSTM architecture using the pre-trained embedding weights for Arabic POS tagging.

3. Training:

- Train the Model: Train the BiLSTM model on the Arabic POS dataset dataset with updating the embeddings while training.

4. Evaluation on Test Data:

- Evaluate POS Tagging Performance: Assess the BiLSTM model's ability to capture bidirectional contextual information and enhance semantic understanding in Arabic POS tagging.

```
Weighted F1-Score: 97.66 %
Accuracy: 97.71 %
```

**Conclusion:**

The BiLSTM experiment explores the impact of bidirectional information flow in capturing contextual nuances, contributing to a more sophisticated semantic analysis for Arabic POS tagging. These baseline and subsequent experiments establish a solid foundation for further exploration and advancements in Arabic POS tagging capabilities. Notably, there was a slight improvement in the weighted F1 score, indicating progress in the model's performance.

**Challenges:**

I employed the Python programming language and utilized Jupyter Notebook for this project. Throughout the implementation, I faced a few noteworthy challenges. These included addressing ambiguity and homography in POS tagging, coping with limited annotated data, navigating the lack of standardized POS tagging schemes, overcoming these challenges required a combination of advanced modeling techniques and data preprocessing to ensure accurate and robust Arabic POS tagging.

# Overall Conclusion:

The evolution of our Arabic POS tagging model from its initial stages to the latest BiLSTM experiment represents a significant advancement in enhancing the accuracy and contextual understanding of Arabic part-of-speech tagging.

**1- Baseline Experiment:**

The baseline experiment established the foundation by implementing an RNN with various architectures, serving as the initial benchmark for Arabic POS tagging. This approach provided an initial understanding of the challenges and requirements of the task, paving the way for subsequent enhancements.

**2-LSTM Experiment:**

The experiment introduced an LSTM architecture, aiming to capture long-term dependencies and enhance Arabic POS tagging. Despite these efforts, there were no notable advancements in the weighted F1 score compared to the previous models.

**3. BiLSTM Experiment:**

- The experiment incorporated bidirectional information flow through a BiLSTM architecture. Despite the expectations of improved contextual analysis, there were slightly notable advancements in the weighted F1 score compared to the previous models.

Throughout these experiments, the goal was to advance the Arabic POS tagging model's performance by incorporating different techniques and linguistic knowledge. The incremental steps underscored the challenges and complexities of the task while also highlighting the importance of choosing suitable architectures and embedding strategies. The journey showcases the continual effort to enhance the model's accuracy and contextual understanding, paving the way for future refinements and improvements in Arabic POS tagging capabilities.