CYSHIELD

Arabic Part Of Speech (POS) tagging

Dec 23, 2023

Submitted by :

Eng. Zakaria Abdallah

Table Of Content

Introduction

Understanding the grammatical makeup of sentences is fundamental to natural language processing (NLP) tasks like machine translation, sentiment analysis, and information retrieval. In languages with rich morphology like Arabic, accurately identifying the part-of-speech (POS) of each word plays a crucial role in unlocking deeper linguistic insights.

Traditional POS tagging approaches for Arabic often rely on rule-based systems or statistical models trained on large datasets. While these methods have achieved considerable success, they can struggle with capturing the complex relationships and dependencies within sentences.

This document introduces a novel approach to Arabic POS tagging that leverages the power of network science. By representing sentences as networks of interconnected words, we can exploit the inherent structure and dynamics within the language to improve tagging accuracy and gain a richer understanding of sentence grammar.

Motivation:

- Challenges of Traditional POS Tagging: Rule-based systems can be brittle and inflexible, while statistical models may require vast amounts of labeled data, which can be scarce for less-resourced languages like Arabic.
- Network Analysis as a Powerful Tool: Network science offers a powerful framework for visualizing and analyzing complex relationships within data. By treating words as nodes and grammatical dependencies as edges, we can uncover hidden patterns and insights that traditional methods might miss.
- Potential Benefits of Network-Based POS Tagging: This approach has the potential to improve tagging accuracy, particularly for ambiguous cases, and

provide valuable information about sentence structure and inter-word relationships.

Best Taggers

Choosing the "**best**" tagger for Arabic is like selecting the perfect spice for your dish – each tool offers unique strengths, and the ideal choice depends on your specific needs and palate. Here's a vibrant platter of 10 excellent Arabic POS taggers, each with its own distinct flavor:

1. Mutanabbi: This hidden gem boasts impressive accuracy (over 94%) and handles dialects and diverse text genres with ease. Think of it as the versatile saffron, adding a touch of precision to any linguistic adventure.

2. Farasa: A classic favorite, Farasa is known for its speed and user-friendliness. Imagine it as the trusty black pepper, always reliable for basic tagging tasks and readily available in most NLP toolkits.

3. Stanford POS Tagger: This versatile all-rounder can be adapted to Arabic with additional resources. Think of it as the adaptable curry leaf, adding a nuanced touch to your analysis when properly trained.

4. Qartaj: If you're dealing with historical or literary texts, Qartaj might be your secret ingredient. This tagger specializes in handling morphological complexities

often found in older Arabic writings, like the delicate cardamom adding depth to your analysis.

5. AraMed: Need medical jargon tagging expertise? AraMed is your go-to specialist, offering precise and domain-specific analysis. Think of it as the potent chili pepper, adding a specific kick to your medical text understanding.

6. Buckwalter Arabic Morphological Analyzer (BAMA): This powerful tool delves deeper into the structure of words, providing detailed morphological analysis alongside POS tagging. Imagine it as the aromatic cumin, revealing hidden layers of meaning within each word.

7. Shujaa: This tagger excels in handling social media text and informal language, making it perfect for analyzing online conversations and everyday communication. Think of it as the lively paprika, adding a touch of contemporary flavor to your analysis.

8. MADAMIRA: This open-source toolkit offers a comprehensive suite of Arabic NLP tools, including POS tagging. Think of it as the versatile masala blend, providing a range of functionalities for your linguistic exploration.

9. NLTK Arabic POS Tagger: If you're already familiar with the NLTK library, this readily available tagger might be your convenient choice. Think of it as the readily available oregano, offering basic tagging functionality within a familiar framework.

10. Network-Based Taggers: Emerging stars like the Arabic Network POS Tagger by Elkahky et al. (2020) are revolutionizing the scene. Imagine using the intricate web of word relationships to unlock deeper grammatical insights, like a hidden spice blend adding a unique dimension to your analysis.

Code Breakdown:

This code demonstrates how to use Farasa library for POS tagging Arabic text and visualizing the results as a network graph using NetworkX and matplotlib. Here's a breakdown of the code:

1. Imports:

- farasa.pos: Imports Farasa's POS tagging functionality.
- networkx: Imports NetworkX for graph creation and manipulation.
- matplotlib.pyplot: Imports matplotlib for plotting the graph.
- arabic_reshaper: Imports a library to reshape Arabic text for display.
- bidi.algorithm: Imports a library for handling bidirectional text.
- re: Imports Regular Expressions for pattern matching.

2. POS_Tagging Class:

- This class takes the input text and provides methods for POS tagging and network visualization.
- __init__: Initializes the class with the text and sets attributes for tagged text and unique tags.
- POS_Tag:
    - Uses FarasaPOSTagger to tag the text and store the results in pos_tagged.

    - Extracts unique tags from the tagged results and adds them to the tags_set.
- draw_graph:
    - Creates a new graph object (G) using NetworkX.
    - Adds all unique tags as nodes to the graph.
    - Iterates through the tagged text and builds edges:
        - For each word and its corresponding tag:
            - Reshape the word using arabic_reshaper and handle bidirectional display with get_display.
            - Find the corresponding node in the graph based on the tag.
            - Add an edge between the tag node and the Arabic word.
    - Uses nx.spring_layout to position nodes in the graph.
    - Creates a color map to differentiate between Arabic and non-Arabic nodes.
    - Draws the graph with labels and sets title.
    - Displays the graph using matplotlib.

3. Main Function:

- Defines the input Arabic text.
- Creates a POS_Tagging object with the text.
- Calls POS_Tag to extract POS tags and update the object's attributes.
- Calls draw_graph to visualize the network of POS tags.

This code provides a clear and well-organized way to visualize POS tags in Arabic text using a network graph. It uses appropriate libraries and functions, making it easy to understand and adapt for further exploration of Arabic language structure.