Arabic POS Tagging

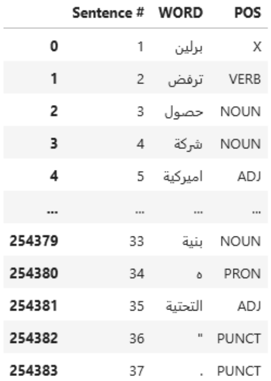
By/ Ahmed Essam

> *"Part of speech tagging is the process of selecting the most likely sequence of syntactic categories for the words in a sentence. It determines grammatical characteristics of the words, such as part of speech, grammatical number, gender, person, etc. In the case of Arabic language, this task is not trivial since most of the words are ambiguous as a result of the absence of vowels.*
>
> *For each word, we want at a minimum to identify its main lexical category (noun, verb etc.) and inflectional features (plural, past tense etc.) if any. We might also identify some quasi-semantic features (proper noun) or even specify a word sense relative to some lexicon."*

Arabic Word	POS Tag
الكتاب	Noun
يقرأ	Verb
الطالب	Noun
جميلة	Adjective
في	Preposition

Data Set:

At first I used the open source Arabic dataset UD_Arabic-PADT as it is benchmarked and well known dataset for pos tags but then we decided to generate other dataset in order to have a larger dataset that is more diverse.
So the data collection subteam scraped data from the internet and used some libraries in order to generate annotated data consisting of sentences where each word is assigned to a pos tag so the final dataset we used was that data consists of 36000 sentence.



	Sentence #	WORD	POS
0	1	برلين	X
1	2	ترفض	VERB
2	3	حصول	NOUN
3	4	شركة	NOUN
4	5	اميركية	ADJ
...	...	...	...
254379	33	بنية	NOUN
254380	34	ه	PRON
254381	35	التحتية	ADJ
254382	36	"	PUNCT
254383	37	.	PUNCT

Figure 1, Example of the dataset after parsing and converting to a dataframe.

Tools used:

Tool	Description
TensorFlow.Keras	A high-level API for building and training deep learning models. It simplifies the process of creating neural networks, including sequential models and recurrent neural networks like LSTM.
tqdm	A library for adding progress bars to Python iterations, making it easier to track the progress of long-running tasks.
conllu	A Python library for parsing CoNLL-U formatted files, which are commonly used for linguistic data.
pandas	A powerful data analysis and manipulation tool for Python. It's used to handle and process tabular data.
matplotlib	A plotting library for Python, used to create visualizations of data.
NumPy	A library for numerical computing in Python, providing efficient array operations and mathematical functions.
requests	A Python library for making HTTP requests, useful for fetching data from web APIs.
Genism	A Python library for topic modeling, document indexing, and similarity retrieval. It can be used for tasks like text similarity and document clustering.
TensorFlow.Keras.models	A module for building and training neural network models in Keras.
TensorFlow.Keras.layers	A module for creating different layers in neural networks, such as input layers, embedding layers, recurrent layers (LSTM), and dense layers.
TensorFlow.Keras.utils	A module for utility functions in Keras, including to_categorical for one-hot encoding.

Methodology:

- **Dataset preprocessing:**

1. Data Cleaning and Normalization:

- **Diacritization Removal:** Eliminating diacritics (vowel marks) to reduce noise and complexity.
- **Longation Reduction:** Simplifying elongated characters (e.g., "اا" to "ا") for consistency.
- **Text Normalization:** Applying other normalization techniques like removing extra spaces and special characters.

2. Tokenization:

- **Word Tokenization:** Breaking down text into individual words.
- **Tag Tokenization:** Assigning unique integer IDs to each unique word and tag.

3. Sequence Padding:

- **Length Normalization:** Ensuring all sequences have the same length by padding shorter sequences with zeros.

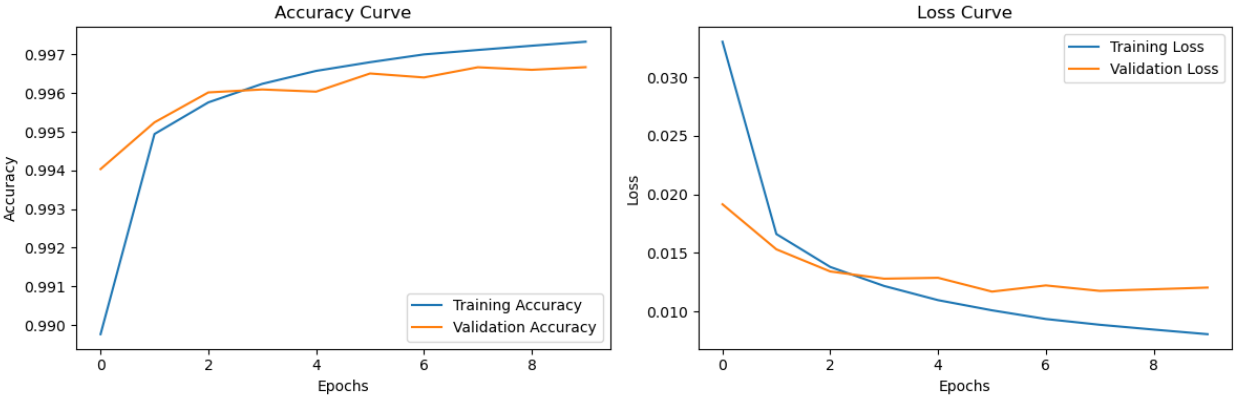
4. Word Embedding:

- **Pre-trained Embeddings:** Using pre-trained word embeddings like AraVec to provide semantic and syntactic information to the model.
- **Embedding Layer:** Incorporating the pre-trained embeddings into the model's architecture to improve performance.

- Models Selection:

After literature review I choose model architecture employed a deep bidirectional LSTM network. The input layer processes sequences of fixed length, and the embedding layer maps words to dense vectors. The bidirectional LSTM layers capture long-range dependencies in both forward and backward directions, enabling the model to learn contextually rich representations. Finally, a time-distributed dense layer with softmax activation is used to predict the part-of-speech tag for each word in the input sequence.

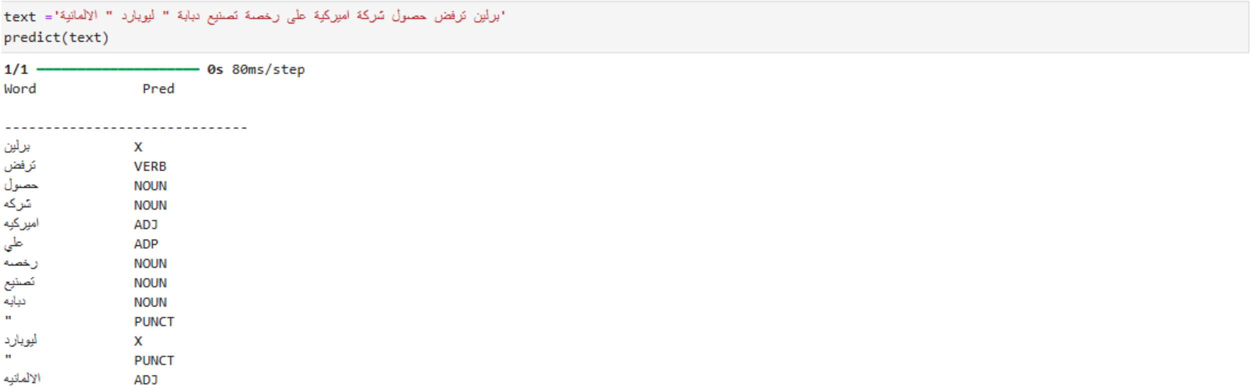
Results:



Class	Precision	Recall	F1-Score	Support
0	0.98	0.80	0.88	343
1	1.00	0.99	0.99	3205
2	0.96	0.90	0.93	3307
3	0.98	1.00	0.99	7408
4	0.86	0.80	0.83	333
5	1.00	1.00	1.00	3586
6	1.00	0.90	0.94	1212
7	0.95	0.91	0.93	4784
8	0.98	0.96	0.97	894
9	0.92	0.32	0.48	37
10	0.99	0.87	0.93	182
11	1.00	0.94	0.97	50
12	1.00	0.00	0.00	1
13	1.00	0.99	0.99	1686
14	0.96	0.96	0.96	14893
15	0.00	1.00	0.00	0

Accuracy			**0.96**	**41921**
Macro avg	0.91	0.83	0.80	41921
Weighted avg	0.97	0.96	0.96	41921

Testing the model:



```
text ='برلين ترفض حصول شركة اميركية على رخصة تصنيع دبابة " ليوبارد " الالمانية'
predict(text)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 80ms/step
Word                 Pred

------------------------------
برلين               X
ترفض               VERB
حصول               NOUN
شركه               NOUN
اميركيه             ADJ
علي                ADP
رخصه               NOUN
تصنيع              NOUN
دبابه               NOUN
"                  PUNCT
ليوبارد             X
"                  PUNCT
الالمانيه           ADJ
```

Conclusion:

The results show that the model performs exceptionally well for most classes, with notable accuracy in identifying common POS tags such as verbs, nouns, and adjectives. Specifically, the model achieved high precision and recall for categories like VERB, NOUN, and ADJ, demonstrating its robustness in handling a variety of word forms in Arabic. For example, class 5 (NOUN) and class 3 (VERB) showed outstanding results, with precision and recall close to 1, reflecting strong prediction performance.

Challenges:

Preprocessing Arabic text for POS tagging is a multifaceted challenge that requires handling complex linguistic features such as morphological richness, tokenization issues, absence of diacritics, and ambiguous word meanings. Addressing these challenges often requires specialized tools and approaches that are tailored to the Arabic language, including techniques for stemming, normalization, and disambiguation. Additionally, handling modern, informal, and code-switched Arabic poses a significant challenge, especially in the context of social media and other non-standard text forms. The success of any Arabic POS tagging system depends on overcoming these preprocessing hurdles to ensure the accuracy and effectiveness of subsequent models.