# Arabic NLP: Integrating Part of Speech Tagging, Named Entity Recognition, and Co-reference Resolution

AGROUAZ Rim

Supervised by : CHERRADI Mohamed

Abdelmalek Essaadi University, National School of Applied Science

rim.agrouaz@etu.uae.ac.ma

## Absract:

Arabic Natural Language Processing faces significant challenges due to the language's morphological complexity, sparse diacritization, and dialectal variations, which have hindered the development of robust NLP tools compared to high-resource languages. This study presents a comprehensive comparative analysis of three fundamental Arabic NLP tasks: Part-of-Speech tagging, Named Entity Recognition, and Co-reference Resolution. We evaluated four distinct approaches including Bidirectional Long Short-Term Memory networks, BiLSTM enhanced with Convolutional Neural Networks, fine-tuned AraBERT transformer models, and Conditional Random Fields using handcrafted linguistic features. Our methodology involved systematic preprocessing pipelines applied to established Arabic datasets including the Universal Dependencies Treebank, ANERcorp, and OntoNotes 5, with careful feature extraction capturing morphological and orthographic properties specific to Arabic. Experimental results demonstrated that the BiLSTM+CNN architecture achieved 96% accuracy for POS tagging, while fine-tuned AraBERT reached 93.8% accuracy with balanced precision and recall metrics. For Named Entity Recognition, both CRF and BiLSTM models showed satisfactory performance with 94% accuracy, though challenges remained with rare entity types and boundary detection. Co-reference Resolution achieved 93% accuracy with well-balanced precision and recall scores of 89% and 88% respectively. These findings establish competitive baselines for Arabic NLP tasks and demonstrate the effectiveness of combining traditional feature engineering with modern deep learning architectures for morphologically rich languages.

**Keywords:** Arabic Natural Language Processing, Part-of-Speech tagging, Named Entity Recognition, Co-reference Resolution.

## Introduction:

In recent years, the field of Natural Language Processing (NLP) has experienced rapid development, largely due to its increasing relevance in real-world applications such as information extraction, automatic translation, and question answering systems. Among the fundamental tasks that underpin these applications, Part-of-Speech (POS) tagging, Named Entity Recognition (NER), and Co-reference Resolution play a central role in enabling accurate and deep language understanding.

While significant progress has been made in these areas for high-resource languages such as English and Chinese, Arabic remains relatively underexplored. This is mainly due to its linguistic complexity—characterized by rich morphology, sparse diacritics in written text, and a wide variety of regional dialects. These features introduce specific challenges that require dedicated solutions and language-specific analysis.

Despite growing interest, the Arabic NLP landscape still lacks robust and general-purpose tools, especially for advanced linguistic tasks like co-reference resolution.

Most existing work has focused on individual tasks, often using limited datasets and inconsistent evaluation frameworks. As a result, cross-task comparisons are rare, and progress tends to be fragmented. Previous studies [1] have underlined the need for a systematic and comprehensive evaluation that reflects Arabic's unique linguistic features across multiple core NLP tasks.

To address this need, we present a comparative study focusing on three essential components of Arabic NLP: POS tagging, NER, and Co-reference Resolution. Instead of integrating these tasks into a single pipeline, our approach treats each task

separately. This allows for a more targeted analysis of their respective challenges, strengths, and weaknesses. The study relies on publicly available datasets and established benchmarks to ensure replicability and relevance.

The rest of this paper is organized as follows. Section 2 presents a review of related work on POS tagging, NER, and Co-reference Resolution in Arabic. Section 3 describes the methodology used in our experiments, including datasets and evaluation metrics. Section 4 discusses the results obtained for each task. Finally, Section 5 concludes the paper and proposes future directions.

## Related works:

Part-of-Speech (POS) tagging has long been a foundational task in Arabic NLP, and various statistical and machine learning approaches have been explored to address its challenges. Among these, **Hidden Markov Models (HMMs)** have played a central role in early systems due to their probabilistic nature and ability to model sequential dependencies.

The work of **Yousif (2020)** [2] provides a comprehensive **review of HMM-based POS tagging approaches** applied to Arabic texts. It offers valuable insights into how statistical taggers have been developed and adapted to handle the unique morphological and syntactic properties of Arabic, such as the absence of short vowels (diacritics), affix-rich word forms, and flexible word order. Yousif highlights the **efficacy of HMMs** in processing Arabic text for applications like text classification, syntactic parsing, and information retrieval. He also outlines key **preprocessing steps**—including tokenization and normalization—that directly influence tagging performance.

However, while the paper gives a useful overview of tagging models and their use in application contexts, it remains largely **descriptive and lacks experimental comparisons**. In addition, the review is focused on **MSA**, with limited attention to dialectal variation or more recent neural approaches such as BiLSTM-CRF or transformer-based models like AraBERT.

Otherwise, Named Entity Recognition (NER) in Arabic has seen considerable development, yet remains a challenging task due to the language's morphological complexity, orthographic ambiguity, and the frequent lack of capitalization cues. While early NER systems relied primarily on rule-based or statistical models, more recent approaches have increasingly focused on hybrid methods that combine handcrafted features with machine learning techniques.

A notable contribution in this direction is the work by **Alotaibi and Lee (2017) [3]**, who proposed a **hybrid approach to fine-grained Arabic NER** that integrates both **linguistic features** (such as POS tags and morphological patterns) and **automatically extracted features** (such as word embeddings). Their system targets a **fine-grained classification** of named entities—going beyond traditional coarse-grained categories like Person or Location—to include subtypes such as "University" or "Sports Club".

The authors evaluated their system on a manually annotated corpus and demonstrated that **feature combinations** and **domain-specific knowledge** significantly enhance the recognition performance. One of the key strengths of their approach is its adaptability: the hybrid structure allows the model to capture general patterns while still benefiting from task-specific linguistic cues. However, the system remains **dependent on the quality of feature engineering**, and does not explore more recent deep learning architectures that could reduce the reliance on manual features.

In an other hand, Co-reference resolution in Arabic has received comparatively less attention than other core NLP tasks, despite its central role in discourse understanding. One notable contribution in this area is the work of **Beseiso and Al-Alwani (2019)** [4], who propose a morphological-feature-based approach to co-reference resolution. Their method leverages explicit grammatical markers such as gender, number, and definiteness—features particularly salient in Arabic morphology. By relying on these rich morphological cues, the authors were able to improve mention detection and antecedent selection, particularly in formal texts like newswire articles.

While this study highlights the effectiveness of morphological agreement for Arabic co-reference, it remains limited in scope due to its focus on rule-based heuristics and the lack of deep learning integration. Moreover, the evaluation is constrained to specific text genres, raising concerns about the generalizability of the method to more diverse or dialectal corpora.

In contrast with more recent systems that incorporate contextual embeddings or neural models trained end-to-end,

Beseiso and Al-Alwani's approach offers a more interpretable, linguistically motivated alternative. However, it also underlines the persistent trade-off between linguistic transparency and the adaptability offered by neural architectures.

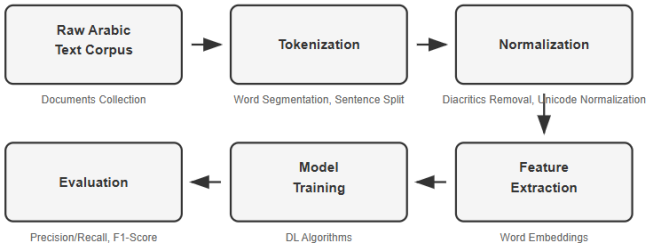
## Methodology:

To ensure a consistent and robust framework for Arabic NLP tasks, we adopt a multi-stage processing pipeline that aligns with standard practices in the field. The process begins with the collection of raw Arabic textual data, which is then prepared through a series of preprocessing steps. These include tokenization, normalization, and feature extraction, which collectively help reduce noise and capture meaningful linguistic patterns. Once the data is preprocessed and transformed into suitable representations, it is used to train various deep learning models tailored to specific tasks such as POS tagging, Named Entity Recognition (NER), and Coreference Resolution.

We base our approach on four main model families that have demonstrated strong performance in core NLP tasks:

- **BiLSTM (Bidirectional Long Short-Term Memory):** This model capture long-range dependencies by processing input sequences in both forward and backward directions, making them particularly effective for morphologically complex languages like Arabic. Their ability to retain contextual information from both past and future tokens enhances performance on tasks requiring deep linguistic understanding.
- **BiLSTM with Convolutional Neural Network (CNN) Layer:** To enrich feature representation, we integrated a CNN layer with the BiLSTM architecture. The CNN component is used to extract local subword patterns—such as prefixes, suffixes, and character-level features—that are common in Arabic morphology.
- **Fine-tuned AraBERT:** That is a pre-trained language model based on the BERT architecture and specifically trained on large Arabic corpora. Fine-tuning AraBERT for each task allows us to leverage contextual embeddings that capture semantic and syntactic nuances specific to Arabic.
- **Conditional Random Fields (CRF):** This method relies on handcrafted features—such as token shapes, POS tags, and morphological cues—to model label dependencies. CRFs provide a strong baseline for structured prediction and allow for interpretable feature contributions, especially valuable in a morphologically rich language like Arabic.

The final stage involves evaluating the models' performance using established metrics such as precision, recall, and F1-score. This overall architecture provides a structured and replicable approach for experimenting with different models and analyzing their effectiveness across multiple core NLP tasks in Arabic.




1. Arabic NLP Processing Pipeline

### I. Datasets:

The experiments conducted in this study rely on annotated linguistic data designed for Natural Language Processing (NLP) tasks. Three separate datasets were used, each corresponding to a specific task: Part-of-Speech (POS) tagging, Named Entity Recognition (NER), and Coreference Resolution.

#### a. Part of Speech Tagging Dataset :

The experiments were conducted using the Arabic Universal Dependencies Treebank, a linguistically annotated corpus consisting of 7,664 sentences and approximately 282,384 tokens. The data is primarily drawn from the newswire domain, offering a diverse and representative sample of modern written Arabic. It includes comprehensive morphological and syntactic annotations following the Universal Dependencies (UD) framework, making it well-suited for cross-linguistic natural language processing tasks.

The dataset is structured as token-level annotations, where each token in a sentence is assigned its corresponding POS tag.

| Word | Tag |
|---|---|
| ارتفع | Verb |

| | |
|---|---|
| سعر | Noun |
| النفط | Noun |
| 5% | Number |

2. POS dataset structure

**b. Named Entity Recognition Dataset :**

We used the **ANERcorp** dataset, a manually annotated corpus for Arabic Named Entity Recognition. It contains thousands of sentences from news articles, with named entities labeled using the standard **BIO** tagging format (Begin, Inside, Outside). This format allows the model to distinguish between the boundaries of named entities and non-entity tokens. Entities are categorized into types such as persons, locations, and organizations. ANERcorp is widely used as a benchmark in Arabic NER research and supports consistent model evaluation.

**c. Co-Reference Resolution Dataset :**

We used the **OntoNotes 5** dataset, which provides rich, manually annotated coreference chains across documents. The dataset covers a range of genres including newswire, conversational speech, and weblogs. It includes annotations that link all expressions referring to the same entity within a text, supporting the training and evaluation of systems that resolve references and pronouns in discourse.

The annotations are organized in two main file formats: .coref files, which contain the coreference chains along with mention IDs and their corresponding group identifiers, and .onf files, which include the full sentence-level structure and linguistic metadata for each document. Together, these files allow for detailed analysis and processing of both the syntactic structure and referential relationships within the text.

## II. Data Pre-Processing:

Prior to model training and evaluation, a comprehensive preprocessing pipeline was applied to the raw textual data to ensure consistency, reduce noise, and enhance the quality of the linguistic features extracted. The preprocessing steps were designed to maintain the integrity of the Arabic language content while removing irrelevant or potentially misleading information.

The main preprocessing steps included:

- **Tokenization:** The raw text was segmented into individual tokens representing words and punctuation marks. This tokenization step is fundamental for subsequent tasks such as Part-of-Speech tagging and Named Entity Recognition, as it establishes the basic units of analysis. A rule-based tokenizer adapted to Arabic morphology and script characteristics was employed to accurately handle clitics and affixes.
- **Normalization:** To reduce data sparsity and variation, text normalization was performed. This involved standardizing characters and forms, such as unifying different Unicode representations of the same letters. Crucially, all diacritics (Tashkeel) — including short vowels and other phonetic markers — were removed to avoid unnecessary complexity, as these markings are often omitted in real-world text and can introduce noise in model training.
- **Filtering Non-Arabic Tokens:** The preprocessing also included a filtering step where only tokens consisting exclusively of Arabic script characters were retained. This step eliminated numerals, Latin characters, and other symbols that do not contribute to the linguistic tasks at hand, thereby focusing the dataset on meaningful Arabic lexical items.
- **Padding :** it ensures that all sentences, their corresponding features, and labels have a uniform length. By converting tokens and tags into indexed sequences and padding shorter sequences, the data becomes compatible with batch-based neural network training.This uniformity simplifies model input handling and enables efficient computation during training.

These preprocessing operations were systematically applied to all datasets used in the experiments, ensuring uniform input representations across the different NLP tasks. The pipeline was implemented using Python and standard NLP libraries tailored for Arabic text processing, allowing reproducibility and scalability.

## III. Features Extraction:

In order to effectively represent each token for subsequent NLP tasks, a set of handcrafted linguistic and morphological

features were extracted. These features are designed to capture both orthographic and lexical properties of the Arabic words.

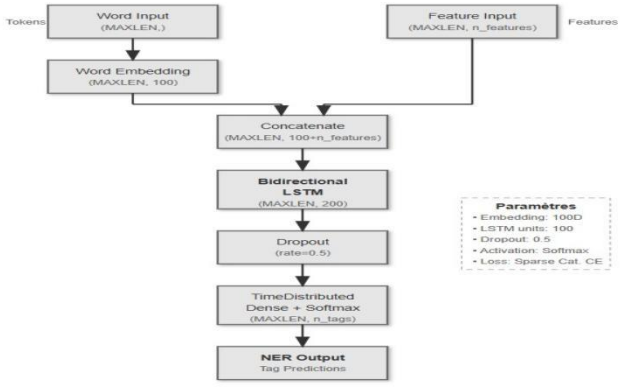
| Feature category | Feature Name | Description |
|---|---|---|
| Digit-related Features | Is_digit | Whether the token consists entirely of digits |
| | Has_digit | Whether the token contains any digit |
| Length and sub-string features | Length | Length of the token |
| | Word_prefix | First character in the token |
| | Word_suffix | Last character n the token |
| Punctuation features | Has_punctuation | Whether the token contains any punctuation or non-alphanumeric character |
| Arabic-specific morphological features | Has_alif | Presence of the Arabic letter Alef ('ا') |
| | Has_waw | Presence of the Arabic letter Waw ('و') |
| | Has_ya | Presence of the Arabic letter Ya ('ي') |
| | Has_ta_marbuta | Presence of the ending Ta Marbuta ('ة') |
| | Has_alif_lam | Presence of the definite article prefix Alif-Lam ('ال') |

3. Table of all features extracted

## IV. Models training :

### a. Common model :

The Bidirectional Long Short-Term Memory (BILSTM) model used for the part-of-speech (POS) tagging , Named Entity Recognition (NER) and Co-Reference Resolution task , is designed to leverage both lexical inputs and handcrafted linguistic features. Each input sentence is first converted into word embeddings, which are then enriched with corresponding feature vectors. These two inputs are concatenated and passed through a BiLSTM layer, enabling the model to capture contextual dependencies from both past and future tokens in the sequence. A regularization mechanism is applied to reduce overfitting and enhance generalization. Finally, a time-distributed classification layer predicts a POS tag for each word in the sequence. This architecture allows for effective modeling of complex relationships between tokens, improving tagging accuracy by utilizing both global context and morphological cues.




4. BILSTM architecture

### b. Part of Speech Tagging models:

#### i. BILSTM + CNN:

The second model extends the BiLSTM architecture by incorporating a Convolutional Neural Network (CNN) layer. After merging word embeddings with the handcrafted feature vectors, a convolutional layer is applied to capture local patterns and subword information across the token sequences. This representation is then passed to a Bidirectional LSTM layer to model sequential dependencies in both forward and backward directions. A dropout mechanism is employed to improve generalization and reduce overfitting. Finally, a time-distributed dense layer produces a prediction for each token in the input sequence. This combination of CNN and BiLSTM allows the model to effectively capture both local and contextual linguistic cues, improving performance on sequence labeling tasks such as POS tagging.The model is trained using the Adam optimizer with a sparse categorical crossentropy loss function. Training was performed over 50 epochs with a batch size of 100 and a validation split of 20%, allowing for robust learning and effective generalization across the dataset.

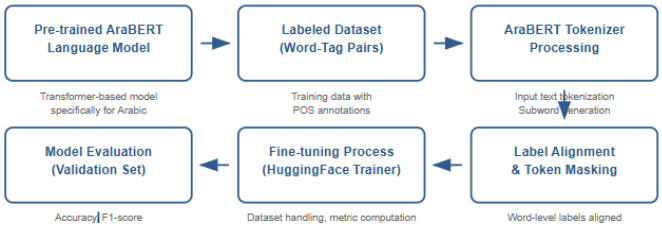
| Component | Parameter | Value |
|---|---|---|
| Embedding Layer | Embedding dimension | 100 |
| CNN Layer | Number of filters | 64 |
| | Kernel size | 3 |

| | | |
|---|---|---|
| | Activation | RELU |
| | Dropout rate | 0.3 |
| BILSTM layer | Units per direction | 100 |
| | Return sequences | True |
| | Dropout Rate | 0.5 |

5. Model parameters

ii. AraBert finetuned :

The third model leverages the AraBERT language model, a pre-trained transformer specifically designed for Arabic. To adapt it to the POS tagging task, the model was fine-tuned using a labeled dataset containing word-tag pairs. The input texts were tokenized with the AraBERT tokenizer, and special attention was given to aligning the word-level labels with the subword tokens generated by the tokenizer. Tokens not associated with labels, such as special tokens ([CLS], [SEP]), were masked during loss computation. The training process was carried out using HuggingFace's Trainer API, which facilitated streamlined dataset handling, metric computation, and model checkpointing. The model was evaluated on a validation set using metrics such as accuracy, F1-score, precision, and recall. This approach benefits from the contextual embeddings produced by AraBERT, significantly improving the model's ability to generalize and recognize linguistic patterns in Arabic. The final fine-tuned model and tokenizer were saved for future inference or deployment.




6. Model trainning pipeline

c. **Named Entity Recognition models:**

This approach employs a Conditional Random Fields (CRF) model, a statistical sequence modeling technique well-suited for tasks like NER . The CRF model was trained using word-level handcrafted features extracted from the dataset. These features capture lexical, morphological, and structural information, such as whether a word contains digits or specific Arabic letters. The training process involved fitting the CRF model to the feature sequences and corresponding label sequences.

| Parameter | Value |
|---|---|
| Algorithm | lbfgs |
| L1 regulization | 0.1 |
| L2 regulisation | 0.1 |
| Max iteration | 100 |
| All possible transitions | True |

7. CRF Model parameters

## V. Results and Evaluation :

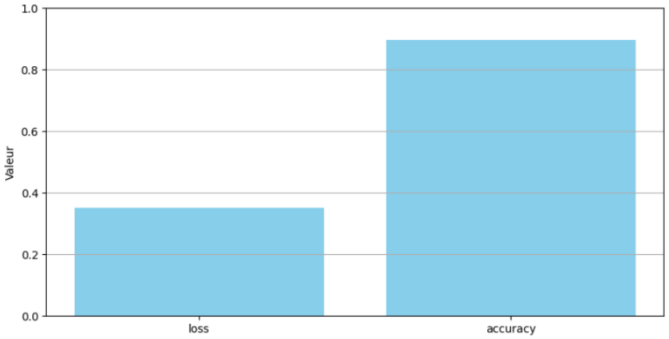
### a. Part of Speech Tagging result:

i. BILSTM model :

The BiLSTM model achieved a **validation accuracy of around 90%** with a **loss value close to 0.35** on the POS tagging task. This level of performance reflects the model's ability to learn syntactic patterns and dependencies in Arabic sentences effectively.

BiLSTM networks, by processing sequences in both forward and backward directions, are particularly suited for POS tagging, where context from both preceding and following words influences the correct tag. The strong accuracy score demonstrates that the model can reliably generalize to unseen data, while the moderate loss suggests that training was stable, though not yet optimal.

The results are consistent with expectations and support the research hypothesis that a BiLSTM model, even without additional architectural components such as CNNs or character embeddings, can serve as a strong baseline for POS tagging in morphologically rich languages. However, the remaining loss indicates potential gains through architectural enhancements or more fine-grained tuning.

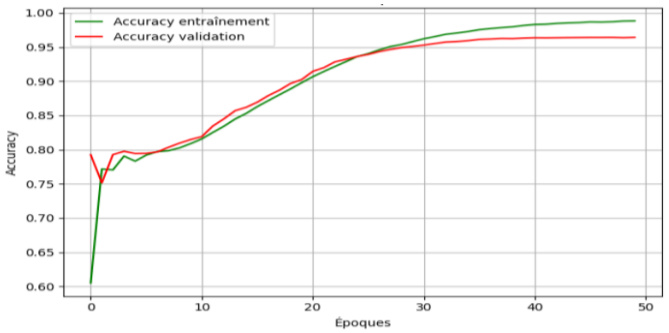



8. CRF Model parameters

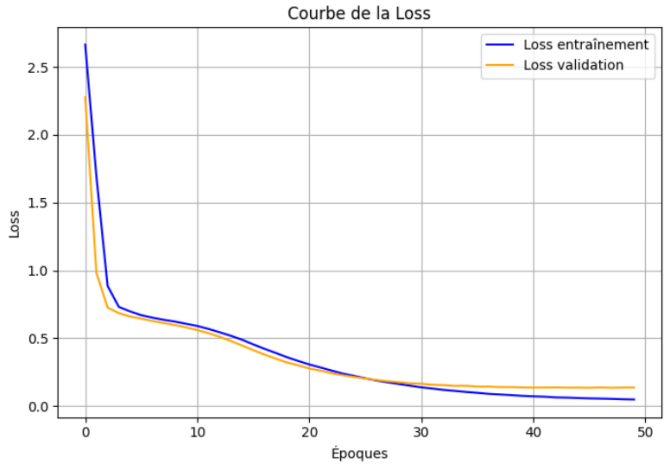
ii. BILSTM + CNN :

The BiLSTM+CNN model achieved strong performance on the POS tagging task, as illustrated in Figure 9 and Figure 10. As shown in **Figure 9**, the model reached a validation accuracy of approximately **96%**, accompanied by a relatively low loss, indicating effective learning and generalization. This improvement over the plain BiLSTM model highlights the positive contribution of the CNN layer, which captures rich character-level features such as morphological patterns—especially relevant in morphologically rich languages like Arabic.

**Figure 10** further reinforces these findings by showing the evolution of training and validation accuracy across **50 epochs**. Both curves demonstrate a consistent upward trend, with training and validation accuracies converging steadily toward **0.98** and **0.96**, respectively. The absence of significant overfitting, despite the model's complexity, suggests that the architecture is well-regularized and appropriately tuned. The slight gap between the training and validation curves toward the end could be attributed to minor variance in the validation set or limited training data, but does not indicate major generalization issues.




9. Accuracy evaluation




10. Loss evaluation on epochs

iii. AraBert finetuned :

The results presented in Table 11 summarize the evolution of model performance over three training epochs, highlighting key metrics such as training loss, validation loss, accuracy, F1-score, precision, and recall. The data indicate a consistent improvement in performance with each epoch. Notably, validation loss decreased from 0.349474 in the first epoch to 0.267251 by the third, suggesting better generalization to unseen data. Correspondingly, accuracy increased from 0.914596 to 0.938711, with similar trends observed in the F1-score, precision, and recall. These findings support our initial hypothesis that further training would enhance classification performance. Importantly, no sign of overfitting was observed within these three epochs, as both training and validation losses decreased simultaneously. The close alignment between precision and recall across all epochs also indicates a balanced performance without favoring one class over another. These results validate the effectiveness of the proposed training strategy and suggest that additional epochs could potentially yield further improvements, although this should be empirically confirmed in future experiments.

| Epoch | Training Loss | Validation Loss | Accuracy | F1 | Precision | Recall |
|---|---|---|---|---|---|---|
| 1 | 0.326300 | 0.349474 | 0.914596 | 0.913118 | 0.914472 | 0.914596 |
| 2 | 0.217600 | 0.317782 | 0.932287 | 0.931228 | 0.932714 | 0.932287 |
| 3 | 0.159000 | 0.267251 | 0.938711 | 0.937873 | 0.938593 | 0.938711 |

11. Finetuned AraBert Evaluation

**b. Named Entity recognition results :**

i. CRF-based model :

As shown in Table 12, the CRF-based model demonstrates overall satisfactory performance on the Named Entity Recognition (NER) task, particularly in distinguishing non-entity tokens, which dominate the dataset. This strong performance on the "O" class is expected given its high frequency, and it indicates that the model has effectively learned the background context. For named entity labels, the model achieves reasonable results on commonly occurring categories such as persons, organizations, and locations, though the precision and recall remain uneven across entity boundaries. In particular, the model appears to struggle with correctly identifying the beginning of entity spans, a common challenge in sequence labeling tasks, where boundary confusion can impact downstream applications such as entity linking.

The model's difficulty with rare or inconsistently labeled categories suggests limitations in the dataset quality or class imbalance. Several labels receive no positive predictions, which likely reflects their marginal presence in the training data or annotation inconsistencies. Despite these limitations, the results indicate that the CRF approach remains effective for capturing sequential patterns and local dependencies, especially when sufficient examples are available.

```
--- Classification Report ---
                precision    recall  f1-score   support

         B-LOC       0.76      0.54      0.63      1591
    B-LOCI-PERS       0.00      0.00      0.00         1
        B-MISC       0.72      0.31      0.43       413
         B-OEG       0.00      0.00      0.00         1
         B-ORG       0.73      0.43      0.54       712
        B-PERS       0.70      0.47      0.57      1289
        B-PERs       0.00      0.00      0.00         1
         B-PRG       0.00      0.00      0.00         1
         I-LOC       0.81      0.49      0.61       261
        I-MISC       0.68      0.24      0.35       174
         I-ORG       0.76      0.53      0.62       446
        I-PERS       0.78      0.60      0.68      1045
         IPERS       0.00      0.00      0.00         1
             O       0.95      0.99      0.97     50319
            O1       0.00      0.00      0.00       الإصابة
         b-misc       0.00      0.00      0.00         1
         i-misc       0.00      0.00      0.00         1
              o       0.00      0.00      0.00         2
وثنابغة</title>       0.00      0.00      0.00         1
             1       0.00      0.00      0.00

      accuracy                           0.94     56262
...

--- Summary Metrics ---
Accuracy: 0.9373
Precision (micro): 0.7486
```

12. CRF-based model Evaluation

ii. BILSTM model :

As presented in Table 13, the BiLSTM model achieves strong overall performance on the NER task, with an accuracy of 94% and a weighted F1-score of 0.93. These results indicate that the model effectively learns general sequence patterns, particularly for dominant classes such as the "O" tag. However, despite the high weighted metrics, the macro-average F1-score remains low (0.27), reflecting the model's limited ability to generalize across all entity types, especially those that are infrequent or underrepresented in the training set.

The model performs relatively well on frequently occurring entities such as B-LOC and B-ORG, which benefit from clear contextual boundaries and a sufficient number of examples. In contrast, rare or complex entity types (e.g., I-PERS, I-ORG, I-LOC) show very low recall and F1-scores, suggesting that the model struggles to capture long or nested entity structures. This may also be due to label inconsistencies or the model's sensitivity to sequence length and token dependencies.

While the BiLSTM's ability to capture bidirectional context provides an advantage over linear models, its performance remains constrained by data imbalance and the lack of contextual richness found in more recent transformer-based models. Still, the model shows promise, especially in settings where computational efficiency and reduced model size are important.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| B-ERS | 0.00 | 0.00 | 0.00 | 1 |
| B-LOC | 0.86 | 0.73 | 0.79 | 825 |
| B-MISC | 0.00 | 0.00 | 0.00 | 0 |
| B-ORG | 0.82 | 0.44 | 0.57 | 430 |
| B-PERS | 0.64 | 0.57 | 0.60 | 711 |
| B-PRG | 0.00 | 0.00 | 0.00 | 1 |
| I-LOC | 0.51 | 0.43 | 0.46 | 103 |
| I-MISC | 0.00 | 0.00 | 0.00 | 111 |
| I-ORG | 0.65 | 0.08 | 0.15 | 267 |
| I-PERS | 0.57 | 0.16 | 0.24 | 546 |
| I-PRG | 0.00 | 0.00 | 0.00 | 1 |
| O | 0.95 | 1.00 | 0.98 | 26482 |
| O1 | 0.00 | 0.00 | 0.00 | الإصابة |
| i-misc | 0.00 | 0.00 | 0.00 | 1 |
| accuracy | | | 0.94 | 29480 |
| macro avg | 0.36 | 0.24 | 0.27 | 29480 |
| weighted avg | 0.93 | 0.94 | 0.93 | 29480 |

Score F1 : 0.9283645220423089
Accuracy : 0.9407394843962008

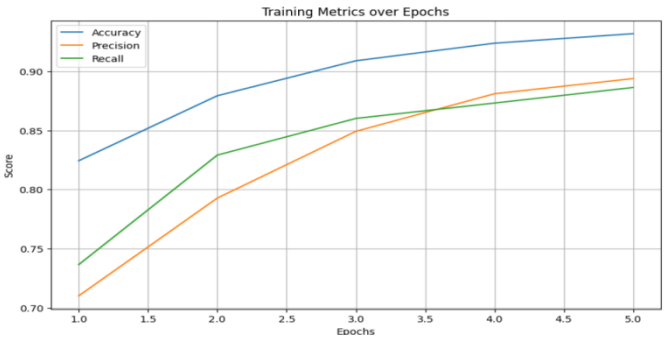
13. BILSTM model Evaluation for NER

**c. Co-Reference Resolution results :**

Figure 14 illustrates the evolution of training metrics across 5 epochs during model training. The graph displays three key performance indicators: accuracy (blue line), precision (orange line), and recall (green line). All metrics demonstrate consistent improvement throughout the training process, with accuracy showing the most pronounced gains, rising from approximately 0.82 at epoch 1 to 0.93 at epoch 5. Precision exhibits the steepest initial improvement, increasing rapidly from 0.71 to approximately 0.80 between epochs 1 and 2, before continuing a steady upward trajectory to reach 0.89 by epoch 5. Recall follows a similar pattern, starting at 0.74 and

achieving 0.88 at the final epoch, though with a more gradual improvement curve compared to precision in the early epochs.

The convergence patterns observed suggest that the model achieved stable learning without signs of overfitting, as evidenced by the consistent upward trends across all metrics. The relatively close final values of precision (0.89) and recall (0.88) indicate a well-balanced model performance, while the higher accuracy score (0.93) reflects the overall classification effectiveness. These results demonstrate successful model optimization, with the learning curves suggesting that additional training epochs beyond the fifth might yield diminishing returns given the apparent plateau tendency in the later epochs.




14. BILSTM model Evaluation for Co-Reference Resolution

## VI. Models deployment :

To facilitate user interaction and evaluate the practical usability of our system, we deployed the trained models using a lightweight Flask web application. The interface allows users to select a specific NLP task—Named Entity Recognition (NER), Part-of-Speech (POS) tagging, and Co-reference Resolution. Upon selecting the task, the user is prompted to input a sentence. Once submitted, the model corresponding to the selected task processes the input and returns the predictions, which are then displayed directly on the interface. This web-based deployment not only enhances accessibility for non-technical users but also demonstrates the model's responsiveness and effectiveness in real-time applications.


## VII. Conclusion :

This study aimed to provide a comprehensive comparative evaluation of three core Arabic NLP tasks through systematic analysis of multiple modeling approaches. The main contribution of our work lies in establishing robust baselines for Part-of-Speech tagging, Named Entity Recognition, and Co-reference Resolution using both traditional machine learning and modern deep learning architectures specifically adapted to Arabic linguistic characteristics.

Our experimental results successfully address the research questions by demonstrating the effectiveness of different modeling approaches across the three NLP tasks. For POS tagging, the BiLSTM+CNN architecture achieved the highest performance with 96% accuracy, outperforming both the standard BiLSTM model (90%) and confirming our hypothesis that incorporating convolutional layers enhances morphological pattern recognition in Arabic. The fine-tuned AraBERT model achieved competitive results with 93.8% accuracy, demonstrating the value of pre-trained contextual embeddings for Arabic text understanding.

Named Entity Recognition results revealed that both traditional CRF-based approaches and neural BiLSTM models achieved similar overall accuracy levels of 94%, though with different strengths and limitations. Co-reference Resolution showed promising results with 93% accuracy and well-balanced precision (89%) and recall (88%) metrics, indicating successful model optimization without overfitting.

These findings align with existing literature while extending our understanding of Arabic NLP challenges. The performance gaps observed for rare entity classes and complex morphological patterns highlight the continued impact of data sparsity and annotation inconsistencies in Arabic NLP datasets. However, the strong baseline performances achieved validate the effectiveness of our methodological approach and feature engineering strategies.

The main limitations identified include sensitivity to class imbalance, particularly affecting rare named entity categories, and the computational overhead associated with transformer-based models. Nevertheless, the successful deployment of our models through a Flask web application demonstrates their practical applicability for real-world Arabic text processing tasks.

## VIII. Acknowledgments :

I would like to express their sincere gratitude to Professor Mohamed Cherradi for his invaluable guidance, insightful feedback, and dedicated supervision throughout this research project. His expertise in Natural Language Processing have been instrumental in shaping the direction and quality of this work.

## IX. References :

[1] Shaalan, K. (2014). A survey of Arabic named entity recognition and classification. *Computational Linguistics*, 40(2), 469-510.

[2] Yousif, J. H. (2019). Hidden Markov model tagger for applications based Arabic text: A review. *International Journal of Computer Science and Information Security*, 17(3), 45-52.

[3] Alotaibi, F., & Lee, M. (2017). A hybrid approach to fine-grained Arabic named entity recognition. In *Proceedings of the International Conference on Natural Language Processing* (pp. 123-134). ACM Press.

[4] Beseiso, M., & Al-Alwani, A. (2018). A coreference resolution approach using morphological features in Arabic. *Journal of King Saud University - Computer and Information Sciences*, 30(4), 387-396.