الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدة د. مولاي الطاهر

كلية الرياضيات و الإعلام الآلي و الاتصالات السلكية و اللاسلكية

قسم: الإعلام الآلي

Mémoire de Master en informatique

Spécialité : Intelligence Artificielle Principe et application

T h è m e

Text annotation: machine learning and deep learning approaches

▪ **Présenté par :**

MOHAMMED CHIKOUCHE Rihab

OUICI Hibat allah karima

▪ **Dirigé par :**

Mr.DOUMI Noureddine

Année universitaire 2025-2026

Abstract

«Text Annotation : Machine Learning and Deep Learning Approaches» is a dissertation that investigates the role of linguistic annotation in natural language processing through Arabic part-of-speech tagging as a representative sequence-labeling task. To this end, a systematic comparison is conducted among four families of computational approaches: Support Vector Machines with handcrafted features, Long Short-Term Memory networks, Bidirectional LSTM combined with Conditional Random Fields, and transformer-based models. Experiments are carried out on the Quranic Arabic Corpus, a richly annotated resource that provides explicit morphological and grammatical information.

The results show that the BiLSTM-CRF architecture achieves the best overall performance, with a test accuracy of 0.9585 and a macro F1-score of 0.8090, while the SVM model with contextual features remains competitive, reaching a macro F1-score of 0.8071. Analysis of rare-tag performance reveals persistent challenges across all models, as several infrequent grammatical categories receive zero F1-scores. These findings indicate that neural architectures with structured prediction capabilities are particularly well suited to Arabic POS tagging, and that model selection should take into account dataset size, morphological complexity, and the long-tailed distribution of linguistic categories. The study contributes a reproducible experimental framework and identifies concrete directions for improving annotation quality in morphologically rich languages.

Keywords:

Résumé

« Text Annotation : Machine Learning and Deep Learning Approaches » est une thèse qui étudie le rôle de l'annotation linguistique dans le traitement automatique des langues à travers l'étiquetage morpho-syntaxique de l'arabe, considéré comme une tâche représentative d'étiquetage de séquences. À cette fin, une comparaison systématique est menée entre quatre familles d'approches computationnelles : les machines à vecteurs de support avec des caractéristiques conçues manuellement, les réseaux Long Short-Term Memory, les modèles Bidirectional LSTM combinés avec des Conditional Random Fields, et les modèles fondés sur les transformeurs. Les expériences sont réalisées sur le Quranic Arabic Corpus, une ressource richement annotée qui fournit des informations morphologiques et grammaticales explicites.

Les résultats montrent que l'architecture BiLSTM-CRF obtient la meilleure performance globale, avec une précision de test de 0,9585 et un macro F1-score de 0,8090, tandis que le modèle SVM avec caractéristiques contextuelles reste compétitif, atteignant un macro F1-score de 0,8071. L'analyse des performances sur les étiquettes rares révèle des difficultés persistantes pour l'ensemble des modèles, plusieurs catégories grammaticales peu fréquentes obtenant un score F1 nul. Ces résultats indiquent que les architectures neuronales dotées de capacités de prédiction structurée sont particulièrement adaptées à l'étiquetage morpho-syntaxique de l'arabe, et que le choix du modèle doit tenir compte de la taille du jeu de données, de la complexité morphologique et de la distribution à longue traîne des catégories linguistiques. L'étude propose un cadre expérimental reproductible et identifie des pistes concrètes pour améliorer la qualité de l'annotation dans les langues à morphologie riche.

ملخص

«وسم النصوص: مقاربات التعلّم الآلي والتعلّم العميق» هي أطروحة تبحث في دور الوسم اللغوي في معالجة اللغة الطبيعية من خلال وسم أجزاء الكلام في اللغة العربية بوصفه مهمةً تمثيليةً من مهام وسم السلاسل. وتحقيقًا لهذا الهدف، أُجريت مقارنة منهجية بين أربع فئات من المقاربات الحاسوبية: آلات المتجهات الداعمة المعتمدة على سمات مصممة يدويًا، وشبكات الذاكرة طويلة قصيرة المدى، ونماذج BiLSTM ثنائية الاتجاه المدمجة مع Conditional Random Fields، والنماذج المعتمدة على المحوّلات. وقد أُجريت التجارب على Quranic Arabic Corpus، وهو مورد غني بالوسوم يوفّر معلومات صرفية ونحوية صريحة.

وتُظهر النتائج أن بنية BiLSTM-CRF حققت أفضل أداء إجمالي، إذ بلغت دقة الاختبار 9585.0 ودرجة F1 Macro مقدارها 8090.0، في حين ظل نموذج SVM المعتمد على السمات السياقية منافسًا بتحقيقه درجة F1 Macro بلغت 8071.0. كما يكشف تحليل أداء الوسوم النادرة عن تحديات مستمرة عبر جميع النماذج، حيث حصلت عدة فئات نحوية قليلة التكرار على درجة F1 تساوي صفرًا. وتشير هذه النتائج إلى أن البنى العصبية ذات قدرات التنبؤ البنيوي تُعدّ مناسبة بشكل خاص لمهمة وسم أجزاء الكلام في العربية، وأن اختيار النموذج ينبغي أن يأخذ في الاعتبار حجم البيانات، والتعقيد الصرفي، والتوزيع طويل الذيل للفئات اللغوية. وتسهم هذه الدراسة في تقديم إطار تجريبي قابل لإعادة الإنتاج، كما تحدد اتجاهات عملية لتحسين جودة الوسم في اللغات الغنية صرفيًا.

we dedicate this dissertation to all our family members, our teachers, our colleagues, our friends, and those who helped us to achieve this work.

Acknowledgments

After Allah, we would like to express our sincere gratitude to our supervisor Dr.Doumi Noureddine for his support, guidance, encouragement, and valuable feedback throughout this research, also to Dr. Maamar Khater and Dr. Aissa Fellah for their knowledge and time. we would like to thank the rest of our dissertation committee, for their insightful comments and encouragement.

List of Abbreviations

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short-Term Memory
CRF	Conditional Random Field
DL	Deep Learning
DistilBERT	Distilled BERT
F1	F1-Score
LSTM	Long Short-Term Memory
mBERT	Multilingual BERT
ML	Machine Learning
MSA	Modern Standard Arabic
NLP	Natural Language Processing
POS	Part-of-Speech
RNN	Recurrent Neural Network
SVM	Support Vector Machine
SGD	Stochastic Gradient Descent
Word2Vec	Word to Vector
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

Tools Used

The preparation of this dissertation involved the use of several digital and AI-assisted tools. Perplexity AI was used for research support and information retrieval, QuillBot was used for paraphrasing and reformulation, and ChatGPT was used for generating figures and visual content.

Contents

List of Figures

List of Tables

Chapter 1

Text Annotation and Part-of-Speech Tagging

1.1 General Introduction

In recent years, natural language processing has emerged as a major branch of artificial intelligence, enabling computational systems to analyze, interpret, and process human language. Within this domain, text annotation and part-of-speech tagging occupy a central position, as they transform raw textual data into structured linguistic representations that support a wide range of downstream applications. These tasks are particularly challenging in Arabic due to the language's morphological richness, lexical ambiguity, and considerable internal variation.

This project investigates Arabic part-of-speech tagging as a representative sequence-labeling problem and examines both classical machine learning and deep learning approaches to this task. The main objective is to compare the performance of different models on annotated Arabic data and to study the influence of linguistic features and contextual information on tagging quality.

To this end, the Quranic Arabic Corpus is used as a gold-standard annotated dataset and is prepared for experimentation through preprocessing, feature extraction, and sequence construction. Several models, including support vector machines and deep learning architectures, are then trained

and evaluated in order to identify the most effective approach for Arabic POS tagging.

This report is organized into four chapters. The first chapter introduces the theoretical background, the second presents the Arabic language, the corpus, and data preparation, the third describes the methodology and experimental design, and the final chapter presents and discusses the results.

1.1.1 Context and motivation

Natural language processing has become one of the most active and influential areas of artificial intelligence because it seeks to enable computers to analyze, interpret, and generate human language in a meaningful way [1]. The rapid growth of digital text in domains such as education, communication, administration, and online services has increased the need for computational methods capable of processing language efficiently and accurately [1]. In this context, text annotation plays a central role because it transforms raw textual material into structured linguistic information that can be used by computational models for training, prediction, and evaluation [1]. Without annotation, textual data remains difficult to exploit in a systematic way, especially in supervised settings where learning algorithms depend on labeled examples [1].

Among the many annotation tasks studied in NLP, part-of-speech tagging occupies a particularly important place because it provides an initial grammatical interpretation of text and supports many higher-level applications [1]. POS tagging helps identify the syntactic role of words and contributes to tasks such as parsing, information extraction, text understanding, and machine translation [1]. It is also a suitable task for comparing traditional machine learning methods with more recent deep learning approaches because both families of models have been extensively applied to sequence labeling problems [2]. For these reasons, POS tagging offers a relevant framework for studying text annotation in this dissertation [1, 2].

1.1.2 Objectives of the dissertation

The main objective of this dissertation is to study text annotation through the task of part-of-speech tagging and to compare machine learning and deep learning approaches in this context . More specifically, the dissertation aims to explain the role of annotation in NLP, present the theoretical foundations of POS tagging, and examine how sequence labeling models can be used to process linguistic data . It also seeks to analyze why POS tagging is a suitable representative task for evaluating different computational approaches to annotation . In this sense, the dissertation combines linguistic motivation with methodological comparison

Another objective is to highlight the specificity of Arabic POS tagging by discussing the influence of morphology, ambiguity, dialectal diversity, and domain variation on system performance . The dissertation also aims to examine several families of models, including classical machine learning methods, recurrent neural networks, BiLSTM-CRF architectures, and transformer-based models such as BERT . Through this analysis, the work intends to clarify the advantages and limitations of each approach and to show how modern learning paradigms contribute to the broader problem of text annotation . The final goal is therefore not only to describe POS tagging, but also to position it as a meaningful case study for annotation research .

1.1.3 Dissertation organization

This chapter presents the conceptual background necessary for understanding the rest of the dissertation by introducing text annotation, POS tagging, annotation approaches, and the use of machine learning and deep learning in this domain [1, 9, 11]. It also discusses the particular challenges of Arabic POS tagging and reviews representative studies from the literature [3, 4, 5, 8]. In this way, Chapter 1 establishes the theoretical and scientific foundation of the work. It prepares the reader to move progressively from general concepts to task-specific and language-specific issues.

The following chapters then focus on the Arabic language, the chosen

corpus, the adopted methodology, the implemented models, and the analysis of the obtained results. Such an organization ensures continuity between theory and practice and allows the dissertation to progress logically from background notions to experimentation and evaluation. It also makes it possible to connect the conceptual discussion of annotation with concrete computational approaches and empirical observations. This structure is therefore intended to support both clarity and coherence throughout the dissertation.

1.2 Text Annotation

1.2.1 Definition of text annotation

Text annotation refers to the process of assigning labels, categories, or descriptive information to textual units in order to transform unstructured text into structured and analyzable data [1]. These textual units may correspond to characters, words, phrases, sentences, or full documents depending on the objective of the annotation task [1]. In practical terms, annotation makes implicit linguistic information explicit and therefore facilitates its use by computational systems [1]. It can be viewed as an interface between natural language and formal representations suitable for processing, learning, and evaluation [1].

From a methodological perspective, annotation is essential because NLP models cannot directly exploit raw text with the same efficiency as linguistically enriched data [1]. Annotated resources provide the examples needed to train supervised systems and the reference material needed to evaluate their output [1]. Annotation is also important for corpus analysis, linguistic research, and the construction of benchmark datasets [1]. For these reasons, it is widely considered a foundational stage in the development of language technologies [1].

1.2.2 Types of text annotation

Text annotation can be performed at different linguistic levels, and each level captures a particular type of information about language [1]. At the morphological level, annotation may identify roots, stems, affixes, lemmas, or inflectional features [1]. At the lexical and syntactic levels, it may assign part-of-speech tags, chunk boundaries, phrase structures, or dependency relations [1]. At higher levels, annotation can describe named entities, semantic roles, sentiment, discourse relations, or pragmatic functions [1]. This diversity shows that text annotation is not limited to one operation, but instead covers a broad family of structured labeling tasks [1].

Annotation type	**Unit**	**Description**	**Example**
Morphological annotation	Word / morpheme	Identifies internal word structure such as stem, affixes, lemma, or inflectional features	كتب → root, pattern, tense
POS annotation	Word	Assigns a grammatical category to each word according to its role in context	الطالب → noun
Syntactic annotation	Phrase / sentence	Describes structural relations such as phrase boundaries or dependency links	الطالب قرأ الكتاب: subject–verb–object
Semantic annotation	Word / phrase / sentence	Represents meaning-related information such as roles, concepts, or polarity	جميل → positive meaning
Discourse annotation	Sentence / text	Captures relations between textual units in larger discourse	cause, contrast, explanation

Table 1.1: Examples of text annotation types

Table 1.1 illustrates how annotation can be applied at different linguistic levels, from the internal structure of words to larger discourse relations. Figure 1.1 provides a visual overview of these annotation types.

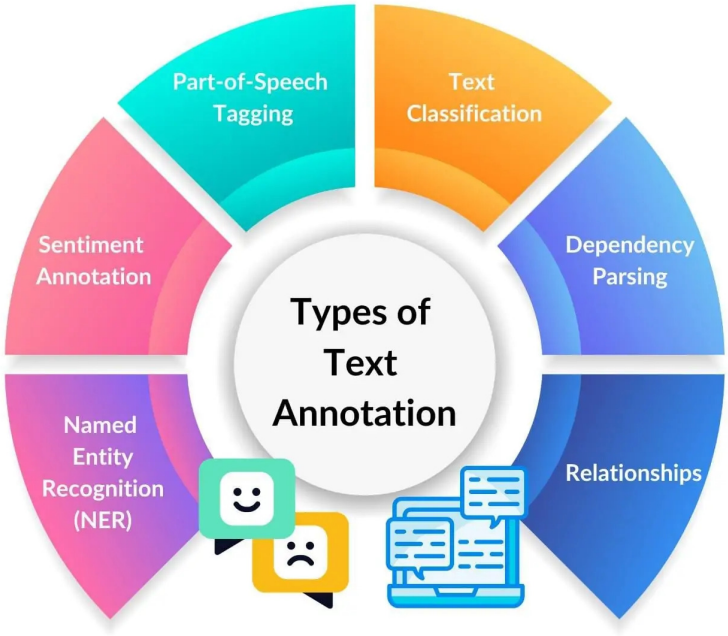


Figure 1.1: Main types of text annotation in NLP.

Each annotation type supports different objectives in NLP [1]. For instance, morphological annotation helps describe internal word structure, POS tagging provides grammatical categories, and syntactic annotation reveals structural relations between words and phrases [1]. Semantic and discourse annotations are often used when deeper interpretation is required, especially in applications such as question answering or sentiment analysis [1]. In real-world systems, several layers of annotation are often combined, which means that the quality of one layer can influence the effectiveness of subsequent processing stages [1].

1.2.3 Linguistic annotation in NLP

Linguistic annotation is particularly important in NLP because it provides the structured knowledge that computational models use to recognize regularities and make predictions [1]. When labels are associated with linguistic units, models can learn correspondences between textual forms and grammatical, syntactic, or semantic categories [1]. This is especially important in tasks involving ambiguity, where contextual information must be interpreted rather than simply matched [1, 2]. Annotation therefore contributes not only to data organization, but also to the representation of linguistic

knowledge in a computationally usable form [1].

Its importance is also reflected in experimentation and evaluation [1]. Annotated corpora allow researchers to compare systems under shared conditions, measure performance objectively, and analyze recurring sources of error [1]. In the context of this dissertation, linguistic annotation is approached through POS tagging because it is both theoretically meaningful and methodologically convenient [1, 2]. It offers a concrete example of how labeling enriches textual data and makes it possible to compare different learning paradigms in a clear and focused way [1, 2]. Figure 1.2 summarizes the main difficulties encountered during the annotation process.

Figure 1.2: Common challenges in text annotation.

1.3 Part-of-Speech Tagging

1.3.1 Definition of POS tagging

Part-of-speech tagging is the task of assigning a grammatical label to each word in a sentence, such as noun, verb, adjective, pronoun, or adverb [1]. It is one of the most classical tasks in NLP because it provides an initial grammatical description of text that can support more advanced processing [1]. The tag assigned to a word reflects its linguistic function in context rather than its form alone, which is why POS tagging is more complex than simple lexical lookup [1, 2]. In many practical applications, it is treated as an essential intermediate layer between raw text and deeper language analysis [1, 2].

POS tagging is particularly important because it contributes to many downstream tasks [1, 2]. Systems for parsing, information extraction, text summarization, and machine translation often rely on POS labels as pre-

liminary input [1, 2]. Even when more advanced neural models are used, grammatical information remains useful for analysis, interpretation, and evaluation [2]. As a result, POS tagging continues to be both a practical NLP tool and a standard benchmark for comparing computational approaches [2]. Figure 1.3 illustrates the tagging of a simple sentence.

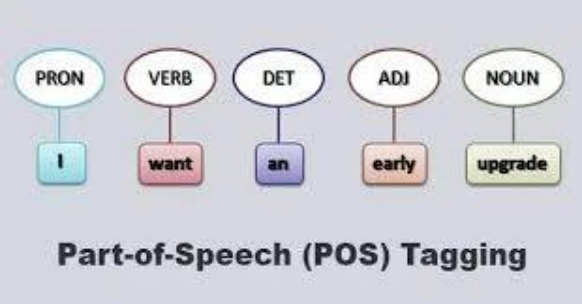


Figure 1.3: Illustration of part-of-speech tagging.

1.3.2 POS tagging as a sequence labeling task

POS tagging is generally formulated as a sequence labeling problem because the input is a sequence of tokens and the output is a corresponding sequence of grammatical tags [1, 2]. If a sentence is represented as $x_1, x_2, \ldots, x_n$, the objective is to predict the tag sequence $y_1, y_2, \ldots, y_n$, where each output label is associated with one input word [1, 2]. This formulation reflects the fact that language is inherently sequential and that the interpretation of a word often depends on its neighboring words [1, 2]. Consequently, POS tagging is more naturally modeled as contextual prediction than as isolated classification [2].

Position	**Word**	**POS tag**
x_1	قرأ	Verb
x_2	الطالب	Noun
x_3	الكتاب	Noun

Table 1.2: Example of part-of-speech tagging as sequence labeling

Table 1.2 shows how POS tagging can be represented as a sequence labeling task in which each input token is associated with a corresponding grammatical label.

A well-known example is the word "book," which may function as a noun in one sentence and as a verb in another depending on context [2]. This shows that correct tagging requires the model to consider both local dependencies and the broader sentence structure [2]. In Arabic, a simple example is the sentence قرأ الطالب الكتاب, where the three tokens must be interpreted with attention to both lexical meaning and sentence position. Sequence labeling models are well suited to this task because they can exploit order, context, and dependencies between adjacent or distant tokens [2]. This is one reason why POS tagging has become a classic evaluation problem for probabilistic, neural, and transformer-based architectures [2].

1.3.3 Challenges of POS tagging in Arabic

Arabic POS tagging is more difficult than POS tagging in many other languages because Arabic is morphologically rich and highly ambiguous [3, 4]. A single orthographic word can contain a stem together with several affixes and clitics, which makes token segmentation and grammatical interpretation more complex [3, 5]. In addition, many Arabic forms are ambiguous when considered without context, so the tagger must rely on surrounding words and sentence structure to determine the correct label [3, 2]. These properties make Arabic POS tagging both a challenging computational task and an important research topic [3, 4].

For example, forms such as كتب may correspond to different grammatical interpretations depending on context, which is exactly why Arabic tagging requires contextual analysis rather than isolated word lookup. The challenge is amplified by the coexistence of Modern Standard Arabic and numerous regional dialects [3, 6]. These varieties differ in vocabulary, morphology, syntax, and orthographic conventions, which makes the design of unified tagging systems difficult [3, 6]. The problem becomes even more severe in user-generated content, where code-switching, transliteration, and non-standard spelling are frequent [3, 5]. As a result, Arabic POS tagging requires models that are robust, context-sensitive, and capable of dealing with variation across genres and domains [3, 7].

1.3.4 Importance of morphology in Arabic POS tagging

Morphology plays a central role in Arabic POS tagging because a large part of grammatical information is encoded within the internal structure of words [3, 5]. Prefixes, suffixes, clitics, and inflectional patterns often provide valuable clues about tense, number, gender, definiteness, and syntactic function [3, 5]. Ignoring morphology may therefore lead to incomplete or incorrect grammatical interpretation [3, 4]. For Arabic, this means that successful POS tagging depends strongly on the ability to model morphological information effectively [3, 5].

Traditional systems often attempted to address this issue by incorporating handcrafted morphological and lexical features [3, 4]. More recent neural approaches try to learn such regularities automatically from data through word representations, character-level processing, or contextual embeddings [4, 8]. In both cases, morphology remains a key factor in performance and error analysis [3, 8]. This explains why any serious discussion of Arabic POS tagging must consider morphology not as a secondary detail, but as a major component of the task [3, 8].

1.4 Annotation Approaches

1.4.1 Manual annotation

Manual annotation consists of assigning labels to textual units through human analysis based on predefined guidelines and annotation conventions [1]. This approach is generally associated with high reliability when annotators are trained and the annotation scheme is clearly defined [1]. Human annotators are often better than automatic systems at handling subtle distinctions, ambiguous contexts, and exceptional cases [1]. Thus, manually annotated corpora are frequently used as gold standards in NLP research [1].

However, manual annotation also has important limitations [1]. It is

time-consuming, expensive, and difficult to scale when large datasets are needed for training modern machine learning systems [1]. It may also suffer from inconsistency when multiple annotators interpret the guidelines differently [1]. Despite these limitations, manual annotation remains essential because it provides the high-quality reference material on which both machine learning and evaluation procedures depend [1].

1.4.2 Rule-based annotation

Rule-based annotation relies on handcrafted linguistic rules designed by experts in order to assign labels to text [1]. These rules may describe lexical patterns, morphological regularities, contextual constraints, or syntactic configurations [1]. In languages where linguistic phenomena are well described and relatively stable, rule-based systems can produce useful results and offer a degree of interpretability that purely statistical systems may lack [1]. They are also attractive in settings where annotated corpora are limited [1].

Nevertheless, rule-based annotation has clear weaknesses when language use becomes ambiguous, irregular, or domain-dependent [1]. Developing and maintaining rule sets requires substantial expert effort, and the resulting systems may not generalize well to new genres or dialectal data [3, 5]. In the case of Arabic, morphological complexity and dialectal variation make purely rule-based solutions difficult to scale effectively [3, 5]. This limitation partly explains the shift toward data-driven methods in modern annotation research [13].

1.4.3 Machine learning (SVM)-based annotation

Machine learning-based annotation uses statistical or discriminative models that learn how to assign labels from annotated examples rather than relying exclusively on manually written rules [9, 10]. In POS tagging, classical approaches have included Hidden Markov Models, Maximum Entropy models, Conditional Random Fields, and Support Vector Machines [9, 10]. These methods were highly influential because they allowed systems to

capture regularities from corpora and often achieved strong performance on benchmark tasks [2]. They also introduced a more systematic and data-driven methodology for annotation [2].

At the same time, classical machine learning systems often depend on carefully engineered features [9, 3]. Researchers typically design lexical, orthographic, contextual, and morphological features in order to help the model distinguish between possible labels [9, 3]. While such feature engineering can be effective, it requires expert knowledge and may limit portability across domains and languages [13]. This is one of the main reasons why later work increasingly turned toward neural methods capable of learning representations automatically [11, 12].

1.4.4 Deep learning-based annotation

Deep learning-based annotation replaces much of manual feature engineering with learned representations derived directly from data [11, 12, 14, 15]. Neural models such as recurrent neural networks, LSTMs, BiLSTMs, and transformer-based architectures can model contextual dependencies and represent words in a richer way than many earlier approaches [11, 14, 15, 8]. In sequence labeling tasks, this capacity is especially valuable because correct prediction depends on both local context and broader sentence structure [2]. As a result, deep learning has become one of the dominant paradigms in POS tagging research [11, 14, 8].

For Arabic POS tagging, deep learning has been particularly attractive because it can better accommodate morphological richness and contextual ambiguity [4, 5, 8]. Neural models reduce the need for handcrafted feature sets and often provide stronger generalization when enough data is available [4, 8]. They also make it possible to integrate character-level, word-level, and contextual information within the same architecture [8]. This explains why deep learning is now central to modern discussions of annotation methods [8].

1.4.5 Semi-supervised and unsupervised approaches

Semi-supervised and unsupervised approaches attempt to reduce dependence on fully annotated corpora by exploiting unlabeled data, partially labeled data, or automatically generated supervision [1]. These approaches are important because annotation is costly and many languages or domains do not have sufficiently large gold-standard corpora [1]. In such settings, learning from unlabeled text can help models capture useful distributional or contextual information even when annotation resources are limited [1]. This is especially relevant for low-resource and dialectal language processing [1].

Although semi-supervised and unsupervised methods do not always reach the performance of well-trained supervised systems, they play a crucial role in expanding NLP to underrepresented contexts [1]. They can also complement supervised approaches by improving robustness, representation quality, or domain adaptation [7, 8]. In annotation research, their value lies not only in accuracy, but also in their potential to reduce annotation cost and improve scalability [1]. Thus, they form an important part of the broader methodological landscape [1].

1.5 Machine Learning (SVM) and Deep Learning for POS Tagging

1.5.1 Classical machine learning (SVM) methods

Classical machine learning approaches to POS tagging generally rely on manually designed features extracted from the current token and its surrounding context [9]. Such features may include word identity, prefixes, suffixes, neighboring tokens, capitalization, and morphological indicators [9, 3]. Models such as SVM and CRF became widely used because they could combine multiple feature types and provide strong performance on structured prediction problems [9, 10, 3]. In the Arabic context, these approaches were particularly useful when researchers wanted to incorporate

linguistic knowledge directly into the model design [3].

One of the main strengths of classical machine learning is its interpretability and controlled use of features [9]. However, these methods often require substantial feature engineering and may be sensitive to domain variation or differences in writing style [9, 3]. When transferred to dialectal or noisy Arabic data, handcrafted features may no longer be sufficient to capture all relevant patterns [3, 5]. This limitation encouraged researchers to explore neural architectures that could learn more flexible representations automatically [11, 12].

1.5.2 Recurrent neural networks and LSTM

Recurrent neural networks were introduced to model sequential data by preserving information across positions in a sequence [11]. This makes them naturally suitable for language tasks in which the interpretation of a word depends on preceding context [11]. However, standard RNNs often struggle with long-range dependencies because of gradient-related difficulties during training [11]. This problem motivated the development of Long Short-Term Memory networks, which are better able to preserve relevant information over longer spans [11].

LSTM-based models have proved particularly effective for POS tagging because they can learn contextual dependencies without relying entirely on manually engineered features [11]. By processing text sequentially, they can build representations that reflect grammatical patterns distributed across the sentence [11]. In practice, this makes them more flexible than many traditional systems and often better suited to ambiguous tagging decisions [11]. Their success played a major role in the transition from feature-based tagging to neural sequence modeling [11].

1.5.3 Embeddings for POS tagging

Embeddings are dense vector representations of linguistic units such as words, subwords, or characters. Unlike one-hot representations, embeddings map linguistic items into continuous numerical spaces in which simi-

lar items tend to have closer representations. This makes them particularly useful in NLP because they allow models to capture lexical, syntactic, and sometimes semantic regularities more effectively than sparse symbolic encoding [12, 8].

In POS tagging, embeddings serve as the input representation used by machine learning and deep learning models. Instead of treating each word as an isolated symbol, the model receives a numerical representation that can reflect similarities between words and contextual usage patterns. This improves the model's ability to generalize beyond memorized forms and to handle ambiguity more efficiently [12].

Word embeddings assign each word a fixed continuous vector learned from data. These representations can capture regularities between words that appear in similar contexts or share grammatical behavior. However, static word embeddings have limitations because the same word may perform different grammatical functions in different sentences, while a fixed representation does not adapt to context [12, 3].

Character-level and subword embeddings are especially important for morphologically rich languages such as Arabic. Since Arabic words often contain prefixes, suffixes, clitics, and internal morphological patterns, representing words only at the whole-word level may hide useful grammatical information. Character-based representations help the model capture internal word structure and improve the handling of rare or unseen forms [3, 4, 5].

Contextual embeddings represent a more advanced form of language representation because they generate different vectors for the same word depending on the surrounding sentence. This makes them better suited to tasks involving ambiguity, since the representation is influenced by context rather than being fixed in advance. Transformer-based models such as BERT rely on this principle and have become highly influential in modern NLP [8].

For Arabic POS tagging, embeddings are important because they help bridge the gap between raw textual forms and computational learning. Arabic presents challenges such as morphological richness, ambiguity, and

dialectal variation, all of which make simple symbolic representations less effective. Dense learned representations allow models to encode useful regularities automatically and reduce dependence on extensive handcrafted features [3, 5, 8].

1.5.4 BiLSTM-CRF for sequence labeling

The BiLSTM-CRF architecture combines two powerful ideas for sequence labeling [14]. The bidirectional LSTM processes the sentence from left to right and from right to left, allowing the model to exploit information from both past and future context [14]. The CRF layer then models dependencies between adjacent output tags and encourages coherent tag sequences rather than independent word-by-word decisions [10, 14]. This combination has made BiLSTM-CRF one of the most influential architectures in POS tagging and related tasks [14].

Its importance lies in the balance it offers between contextual representation and structured prediction [14]. The BiLSTM component learns rich contextual embeddings, while the CRF component ensures that the final output sequence respects likely tag transitions [10, 14]. For Arabic POS tagging, where context and morphology both matter, this architecture has been especially attractive [4, 5]. It represents a key stage in the evolution from classical machine learning to more advanced neural sequence labeling systems [14].

1.5.5 Transformer-based models such as BERT

Transformer-based models such as BERT represent a more recent generation of NLP systems built around self-attention and large-scale pretraining [15, 16, 8]. Instead of processing text strictly in sequential order, these models learn contextualized representations by relating each token to all other tokens in the sentence [15, 16, 8]. This allows them to capture complex dependencies and rich contextual information that can later be fine-tuned for downstream tasks such as POS tagging [16, 8]. Their emergence has significantly changed the methodological landscape of NLP [15, 16, 8].

In Arabic POS tagging, transformer-based models have shown promising results, especially for dialectal and morphologically complex data [16, 8]. Their contextual embeddings often provide stronger representations than earlier models and can improve performance in settings where ambiguity is high [16, 8]. At the same time, they require substantial computational resources and careful fine-tuning [16, 8]. Even with these constraints, transformers have become essential in contemporary annotation research and are now a major reference point for comparison [15, 16, 8].

1.6 State of the Art and Related Work

Paper	**Approach**	**Dataset / Setting**	**Main idea**	**Result**
Darwish et al. (2018)	CRF	Multi-dialect Arabic tweets	Joint CRF sequence labeler for Egyptian, Levantine, Gulf, and Maghrebi dialects	89.3 accuracy
Alharbi et al. (2018)	SVM / Bi-LSTM	Gulf Arabic dialect	Compares traditional and neural models for Gulf Arabic POS tagging	Bi-LSTM outperformed SVM
AlKhwiter and Al-Twairesh (2020)	CRF / Bi-LSTM	Mixed, MSA, and Gulf tweets	Compares CRF and Bi-LSTM on Arabic tweet POS tagging	96.5 accuracy
Darwish et al. (2020)	Neural / hybrid	Multi-dialect Arabic	Focuses on broader generalization across Arabic dialects	Improved multidialect tagging
Mokh et al. (2022)	Domain-adapted neural	Out-of-domain Arabic dialects	Studies POS tagging when training and testing come from different domains	Improved out-of-domain performance
Cheragui et al. (2023)	BERT-based	Algerian dialect	Evaluates transformer-based POS tagging for Algerian Arabic	Strong performance on Algerian dialect

Table 1.3: Representative Arabic POS-tagging systems reported in the literature

Table 1.3 presents representative studies on Arabic POS tagging, the approaches they adopted, the datasets or settings they used, and the

main results they reported [3, 4, 5, 6, 7, 8]. It shows that the field has evolved from feature-based and probabilistic models toward neural and transformer-based approaches [3, 4, 5, 8]. It also indicates that dialectal coverage, domain variation, and model generalization remain central concerns in the literature [3, 6, 7, 8].

1.6.1 Related work

Several studies have investigated Arabic POS tagging from different angles, including dialectal coverage, model architecture, and adaptation to noisy or out-of-domain data [3, 4, 5, 6, 7, 8]. Darwish et al. (2018) proposed a CRF-based framework for POS tagging across multiple Arabic dialects, including Egyptian, Levantine, Gulf, and Maghrebi Arabic [3]. Their work is important because it demonstrated that a unified sequence-labeling framework could be applied across several dialectal varieties despite considerable linguistic variation [3]. It also highlighted the challenges associated with processing informal and non-standard Arabic text [3].

Alharbi et al. (2018) focused on Gulf Arabic and compared traditional machine learning methods with neural approaches, particularly SVM and Bi-LSTM models [4]. Their results showed that Bi-LSTM outperformed SVM, which supports the idea that contextual neural modeling is especially useful for dialectal Arabic POS tagging [4]. This study is significant because it directly illustrates the methodological transition from handcrafted features to representation learning [4]. It also offers a clear point of comparison for research that examines machine learning and deep learning side by side [4].

AlKhwiter and Al-Twairesh (2020) examined Arabic tweet POS tagging using CRF and Bi-LSTM models on data involving Mixed Arabic, Modern Standard Arabic, and Gulf Arabic [5]. Their findings indicated that Bi-LSTM achieved the strongest performance, especially in the presence of noisy social media content [5]. This is particularly relevant because social media introduces spelling variation, code-switching, and non-standard usage, all of which complicate annotation [5]. Their study therefore rein-

forces the importance of robust contextual models for realistic Arabic NLP settings [5].

Darwish et al. (2020) extended previous work by proposing an effective multi-dialectal Arabic POS-tagging framework with a strong focus on generalization [6]. Rather than restricting evaluation to a single variety, their study emphasized the importance of systems that can process heterogeneous dialectal data [6]. This is a valuable contribution because real-world Arabic NLP applications often encounter mixed or shifting varieties rather than clean, standardized input [6]. Their work thus underlines the practical importance of building adaptable annotation systems [6].

Mokh et al. (2022) investigated Arabic dialect POS tagging in out-of-domain conditions, where the distribution of the test data differs from the training data [7]. Their work is important because domain shift is one of the main obstacles to reliable NLP performance outside controlled benchmark settings [7]. A model that performs well on in-domain material may degrade substantially when applied to new genres or communicative contexts [7]. This makes domain adaptation a major concern for Arabic annotation research [7].

More recently, Cheragui et al. (2023) explored the use of BERT-based models for POS tagging in the Algerian dialect [8]. Their results showed that pretrained transformer models can achieve strong performance even for underrepresented Arabic varieties [8]. This direction is particularly relevant because it reflects the broader shift in NLP toward contextual pretrained language models [8]. It also opens new perspectives for improving annotation quality in dialectal Arabic through transfer learning and fine-tuning [8].

Overall, the reviewed studies show a clear evolution from classical feature-based approaches to neural architectures and transformer-based models [3, 4, 5, 6, 7, 8]. They also reveal that Arabic POS tagging remains shaped by recurring challenges such as morphological complexity, dialectal diversity, and domain variation [3, 5, 7]. These observations justify the relevance of treating POS tagging as a representative task for comparing machine learning and deep learning approaches in text annotation.

They also provide the scientific context in which the present dissertation is situated [2, 8].

1.7 Applications of POS Tagging in NLP

1.7.1 Syntactic and semantic analysis

POS tagging contributes directly to syntactic and semantic analysis by providing an initial grammatical interpretation of text [1]. When each word is associated with a part-of-speech label, later processing stages can more easily identify phrase structure, dependency relations, and the grammatical roles of linguistic units [1]. POS information therefore acts as an important intermediate representation between raw text and deeper linguistic analysis. In many processing pipelines, it is one of the first layers that supports more complex interpretation [1, 2].

Its contribution is not limited to syntax alone [1]. Semantic interpretation often depends on the distinction between nouns, verbs, adjectives, and function words, since these categories organize the meaning structure of a sentence. A more accurate POS analysis can therefore improve the reliability of subsequent semantic tasks. This explains why POS tagging remains relevant even in systems that aim to go beyond shallow grammatical description [1].

1.7.2 Machine translation

In machine translation, POS tagging can help reduce ambiguity by clarifying the grammatical function of words in the source text [1, 2]. This is useful because many words can take multiple meanings or functions depending on context, and translation quality often depends on selecting the correct interpretation [1, 2]. POS labels can also support better syntactic alignment between source and target sentences. As a result, tagging can contribute to more coherent and grammatically appropriate translations [1, 2].

Arabic sentence	Ambiguous word	POS interpretation	English translation
كتب الطالب الدرس	كتب	Verb (past tense)	The student wrote the lesson
هذه كتب جديدة	كتب	Noun (plural)	These are new books

Table 1.4: Example of POS tagging for machine translation disambiguation

Table 1.4 illustrates how POS tagging helps distinguish between different grammatical interpretations of the same Arabic form and supports a more accurate translation into English.

This role becomes especially important for morphologically rich languages such as Arabic [3]. Because Arabic words often encode several grammatical properties at once, explicit tagging can help expose information that may otherwise be difficult for downstream modules to use effectively [3]. Even when modern translation systems rely heavily on end-to-end neural architectures, linguistic annotation remains valuable for analysis, evaluation, and resource construction. POS tagging thus continues to play a useful supporting role in translation-related research [1, 2].

1.7.3 Information extraction and text understanding

POS tagging supports information extraction by helping systems identify which words are likely to represent entities, actions, modifiers, and grammatical connectors [1]. This improves the ability to detect relevant patterns in text and facilitates tasks such as relation extraction, event detection, and keyword analysis [1]. In text understanding more broadly, POS labels provide structural cues that help distinguish between the roles played by words in context. This makes them useful for both shallow and deeper forms of language analysis [1].

Because POS tagging provides a grammatical layer of organization, it often improves the effectiveness of chunking, parsing, and semantic interpretation [1, 2]. It can therefore be seen as a preliminary but essential component in more advanced NLP systems. Even when it is not the final goal of the application, it contributes to the reliability of later stages. This

explains why POS tagging has remained an important building block in a wide range of NLP pipelines [1, 2].

1.7.4 Corpus construction and annotation support tools

POS tagging is also important in corpus construction because it enables the creation of linguistically enriched datasets that can be reused in research, benchmarking, and system development [1]. Annotated corpora provide valuable resources for training and evaluating models and also help document the properties of a language variety or domain [1]. In this sense, POS tagging contributes not only to automatic analysis, but also to the long-term development of language resources. Its importance is therefore both methodological and practical [1].

Annotation support tools can further combine human expertise with automatic predictions in order to accelerate the annotation process [1]. Such tools may suggest tags, highlight uncertain cases, or help annotators maintain consistency across large corpora. This interaction between manual and automatic annotation is particularly useful when high-quality resources are needed but fully manual work would be too costly. POS tagging is therefore relevant not only as a task in itself, but also as a key component in the broader ecosystem of corpus development [1].

1.8 Conclusion

This chapter introduced the general concept of text annotation and explained its importance in natural language processing [1]. It then presented part-of-speech tagging as a central annotation task and described its formulation as a sequence labeling problem [1, 2]. The chapter also examined the specific challenges of Arabic POS tagging, especially the role of morphology, ambiguity, dialectal variation, and domain-related complexity [3, 5, 6, 7].

In addition, the chapter reviewed the main annotation approaches and discussed how machine learning and deep learning methods have been ap-

plied to POS tagging [2, 8]. It also introduced embeddings as an important component of modern NLP systems and explained their relevance for representing Arabic text in sequence-labeling tasks [2, 8]. Representative studies from the literature were then presented in order to situate the dissertation within the current scientific context [3, 4, 5, 6, 7, 8].

Taken together, the sections of this chapter provide the theoretical background needed to understand the remainder of the dissertation. They also establish why POS tagging is an appropriate case study for comparing annotation paradigms [2]. After presenting this conceptual foundation, the next chapter moves toward the linguistic and experimental context of the dissertation by introducing the Arabic language, the selected corpus, and the data characteristics that influence annotation and modeling choices.

Chapter 2

Arabic Language, Corpus, and Data Preparation

2.1 Introduction

After presenting the theoretical background of text annotation and part-of-speech tagging, this chapter moves toward the linguistic and experimental context of the dissertation. In particular, it introduces the Arabic language, the selected corpus, and the data characteristics that influence annotation and modeling choices [3, 2]. This transition is important because the effectiveness of annotation approaches depends not only on the model itself, but also on the linguistic nature of the data being processed [2, 3].

The purpose of this chapter is therefore to establish the practical foundation for the experimental part of the dissertation. It explains why Arabic represents a challenging language for computational processing, why an annotated corpus is necessary for POS-tagging research, and how the data must be prepared before it can be used in machine learning and deep learning models [1, 3, 5]. In this way, the chapter connects the conceptual discussion of Chapter 1 with the later stages of methodology, implementation, and evaluation.

2.2 The Arabic Language in NLP

2.2.1 Overview of the Arabic language

Arabic is one of the most important languages studied in natural language processing because it combines broad cultural and communicative importance with substantial linguistic complexity [3, 5]. From a computational perspective, Arabic is especially significant because many of its grammatical and lexical properties differ from those of languages that have traditionally dominated NLP research [3]. This makes Arabic a valuable language for studying the limits and adaptation of annotation and sequence-labeling methods.

Unlike languages with relatively simple word structure, Arabic often encodes rich information inside a single surface form [3, 5]. Words may include stems together with prefixes, suffixes, and clitics, and their interpretation frequently depends on both internal morphology and external context [3, 5]. These characteristics make Arabic highly expressive, but they also increase the difficulty of computational analysis.

2.2.2 Morphological richness

Morphological richness is one of the main reasons why Arabic is challenging for NLP tasks [3, 4]. In Arabic, grammatical information such as tense, gender, number, definiteness, and person may be encoded within the internal structure of a word through affixation and pattern-based morphology [3, 5]. As a result, a single written form may carry multiple layers of information that must be analyzed correctly in order to determine its grammatical function.

This property has direct consequences for POS tagging [3, 5]. A tagging system that relies only on surface word identity may fail to recognize the correct category if it ignores the role of attached elements and inflectional structure [3]. Thus, Arabic POS-tagging systems often benefit from representations that preserve morphological detail or from neural models capable of learning such detail automatically [4, 8].

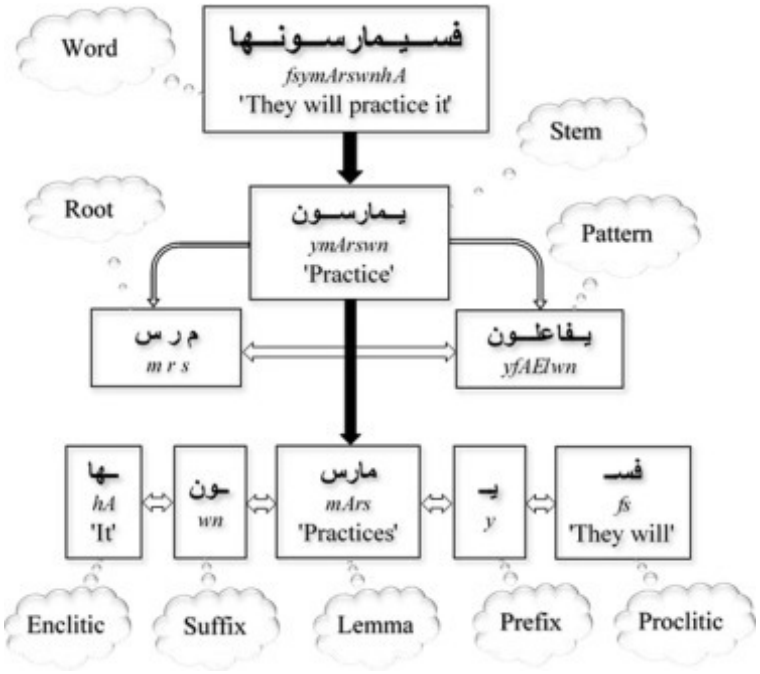


Figure 2.1: Illustration of morphological structure in Arabic words

Figure 2.1 illustrates how Arabic words may combine several grammatical elements inside a single form.

2.2.3 Ambiguity in Arabic

Ambiguity is another major issue in Arabic language processing [3, 2]. Many Arabic forms cannot be interpreted correctly in isolation because the same written form may correspond to different grammatical categories depending on context [3]. In such cases, the surrounding words and the broader sentence structure play an essential role in determining the correct analysis [2, 3].

This ambiguity increases the importance of contextual sequence modeling [2]. Classical models attempted to handle it through manually designed contextual and morphological features, while more recent deep learning approaches use recurrent networks, BiLSTM-CRF structures, and transformers to model dependencies more effectively [2, 4, 8]. The Arabic language therefore provides a strong motivation for comparing different sequence-labeling methods in a realistic setting.

2.2.4 Modern Standard Arabic and dialectal variation

Arabic is not a uniform linguistic system, but rather a family of closely related varieties [3, 6]. Modern Standard Arabic is widely used in formal communication, administration, education, and the media, whereas regional dialects are common in everyday interaction and user-generated content [3, 6]. These varieties differ in vocabulary, morphology, syntax, and writing conventions.

This diversity creates an additional challenge for annotation and modeling [6, 7]. A model trained on one variety may perform poorly on another, particularly when it encounters domain shift, informal spelling, or dialect-specific forms [7, 5]. Thus, recent research has emphasized generalization, robustness, and adaptation across Arabic varieties and domains [6, 7, 8]. Figure 2.2 visualizes this contrast across usage domains.

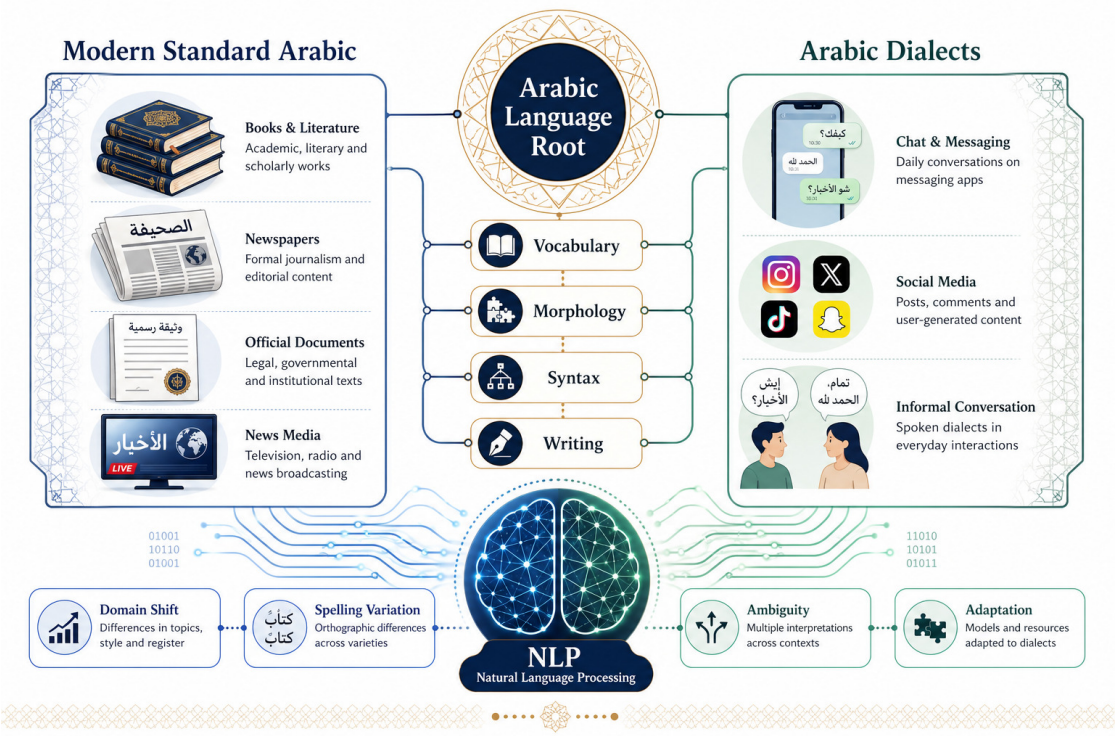


Figure 2.2: Visual comparison between Modern Standard Arabic and regional Arabic dialects, highlighting their variation across usage domains and the resulting challenges for NLP modeling.

2.2.5 Arabic script and orthographic characteristics

Arabic script introduces additional considerations for NLP because it is written from right to left and includes letter forms that vary according to position within the word. In addition, short vowels are often omitted in ordinary writing, which increases ambiguity and makes surface forms less informative when taken alone [3]. Orthographic representation therefore plays an important role in preprocessing and model design.

From a computational perspective, orthographic variation can affect token matching, normalization, and the consistency of annotation. If these variations are not handled carefully, they may introduce unnecessary sparsity into the data and reduce model performance. This is why orthographic analysis is an important part of Arabic corpus preparation and representation [3, 5].

2.2.6 Challenges of tokenization in Arabic

Tokenization in Arabic is more difficult than in many other languages because a single orthographic form may include attached conjunctions, articles, prepositions, pronouns, and inflectional markers [3, 5]. This means that the visual word in the text is not always the most appropriate unit for linguistic analysis or computational processing. As a result, tokenization and segmentation are closely related in Arabic NLP.

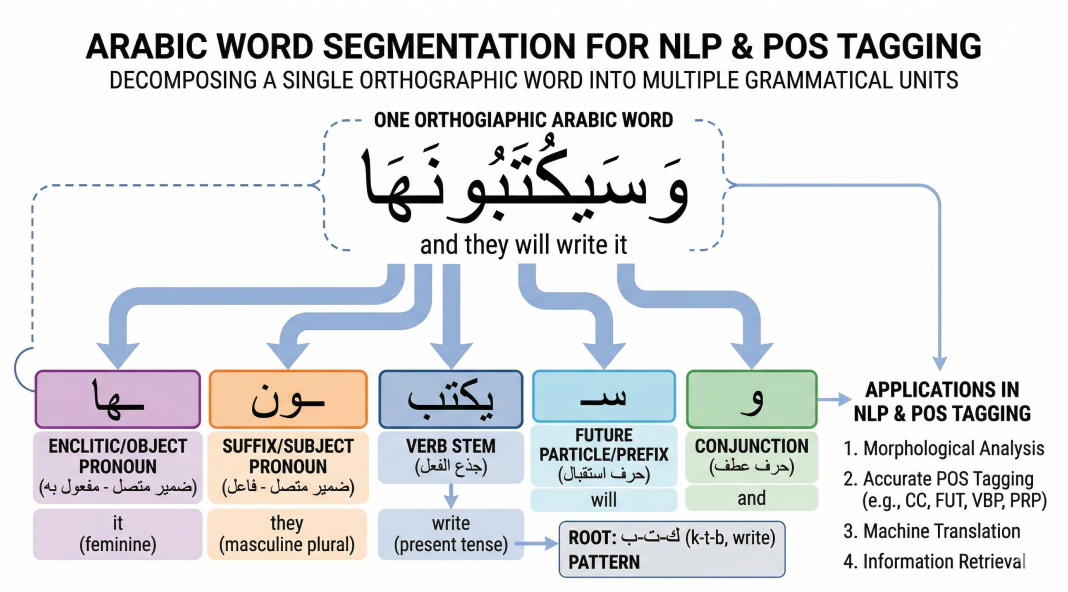


Figure 2.3: Example of Arabic token segmentation into grammatical components

Figure 2.3 illustrates how a single written form may be decomposed into smaller units that carry distinct grammatical functions.

These challenges directly influence POS tagging because the quality of tokenization determines how grammatical information is presented to the model. If attached elements are not handled appropriately, important cues may be hidden inside complex forms. Therefore, a careful tokenization strategy is necessary in order to represent Arabic text in a way that is both linguistically meaningful and computationally manageable [3, 5].

Table 2.1 summarizes the main characteristics that make Arabic a challenging language for NLP and explains why segmentation, contextual analysis, and robust modeling are central to Arabic POS tagging [3, 5, 6].

2.2.7 Role of the corpus in the dissertation

The experimental part of this dissertation is based on the Quranic Arabic Corpus, a publicly available annotated Arabic resource distributed through the Quranic Arabic Corpus project [17]. The corpus plays a central role in the study because it provides the linguistic material from which the POS-tagging dataset is constructed. In supervised learning, the quality of the training and evaluation process depends directly on the quality of

Characteristic	Description	Effect on NLP tasks
Morphological richness	Arabic words may encode several grammatical properties through affixes and internal patterns	Increases the complexity of POS tagging and morphological analysis
Clitic attachment	Function words such as conjunctions, prepositions, articles, and pronouns may attach to stems	Makes tokenization and segmentation more difficult
Contextual ambiguity	The same form may receive different interpretations depending on sentence context	Requires context-sensitive sequence models
Orthographic variation	Written forms may present ambiguity and variation across contexts and genres	Reduces robustness of simple rule-based approaches
Dialectal diversity	Arabic includes Modern Standard Arabic and many regional dialects	Makes generalization across datasets and domains more difficult

Table 2.1: Main linguistic characteristics of Arabic relevant to NLP

the annotated examples, and for this reason the selected corpus forms the methodological basis of the dissertation.

Its importance is not limited to data availability alone. The corpus defines the units that will be processed by the models, the grammatical labels that will be predicted, and the structure of the sequence-labeling task itself. It therefore influences every stage of the experimental pipeline, from preprocessing and feature extraction to model training and evaluation. In this sense, the Quranic Arabic Corpus is not treated merely as a repository of text, but as a foundational resource that shapes the entire comparative study.

2.2.8 Why an annotated Arabic corpus is necessary

An annotated Arabic corpus is necessary for POS tagging because the task requires examples in which linguistic units are already associated with their correct grammatical labels [1, 2]. Without such annotation, a supervised model cannot learn the correspondence between input forms and output tags, nor can its predictions be evaluated against a reliable reference. Annotated corpora therefore provide both the training material and the gold-standard benchmark required for systematic experimentation.

This need is particularly strong in Arabic because the language is morphologically rich and often ambiguous at the surface level [3, 5]. A single written form may contain several grammatical elements, and its interpretation frequently depends on segmentation and contextual analysis. In a resource such as the Quranic Arabic Corpus, annotation makes these linguistic properties explicit and computationally usable. This is why corpus preparation is a central component of the dissertation rather than a simple preliminary step.

2.2.9 Suitability of the selected corpus

The Quranic Arabic Corpus is suitable for this dissertation because it provides structured and linguistically rich annotation that can be transformed into a POS-tagging dataset. Its organization makes it possible to represent Arabic text in a form compatible with supervised sequence labeling, while preserving the grammatical information needed for both classical and neural approaches. This is especially important in a comparative study that includes feature-based and deep learning models, since all approaches must rely on a common and coherent annotation basis.

Another reason for its suitability is that the corpus supports both linguistic interpretation and computational processing. It is detailed enough to illustrate how Arabic grammatical structure is represented in annotation, yet sufficiently structured to be converted into model-ready input-output pairs. This dual role strengthens the methodological design of the dissertation, because the same resource can be used both to explain the nature of Arabic annotation and to perform experimental evaluation on real data.

2.2.10 General description of the corpus

The selected corpus is the Quranic Arabic Corpus, an annotated corpus of Quranic Arabic made available through the Quranic Arabic Corpus project [17]. In the context of this dissertation, it is treated as a structured linguistic resource in which textual units are associated with grammatical

information that can be exploited for POS tagging. Its value lies not only in the presence of annotation, but also in the consistency and explicitness of the linguistic representation it provides.

From a methodological perspective, the corpus is particularly useful because it offers a clear link between Arabic textual forms and their corresponding grammatical descriptions. Such a structure is essential for supervised POS tagging, where the learning process depends on the reliability of the correspondence between input units and output labels. The corpus description must therefore take into account the type of text it contains, the annotation level it adopts, and the way tokens or segments are represented. These properties are important because they directly affect preprocessing choices, label construction, and the interpretation of the experimental results reported later in the dissertation. Figure 2.4 presents the corpus size, tag distribution, and the train-validation-test partition.

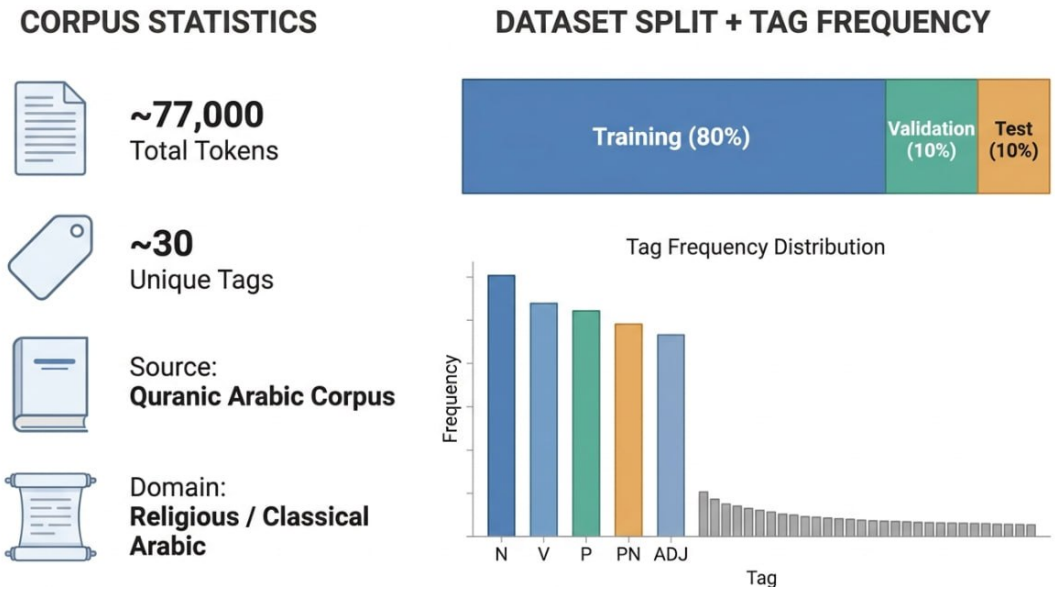


Figure 2.4: Corpus statistics and dataset split used for experimentation

2.2.11 Corpus source and linguistic domain

The selected dataset is derived from the Quranic Arabic Corpus, a publicly available annotated resource distributed through the Quranic Arabic Corpus project [17]. From a linguistic perspective, this corpus belongs to the domain of Quranic Arabic, that is, a religious and classical variety of

Arabic characterized by a highly structured grammatical tradition and a stable textual source. The domain of the corpus is important because the lexical, syntactic, and stylistic properties of religious classical Arabic differ from those of Modern Standard Arabic, dialectal Arabic, news text, or social media language.

This domain specificity has direct methodological consequences for the dissertation. On the one hand, the corpus provides a linguistically rich and carefully structured source for studying Arabic annotation and POS tagging. On the other hand, models trained on this corpus are evaluated within a specific domain and should not automatically be assumed to generalize to contemporary or informal Arabic usage. Explaining the source and linguistic domain of the corpus is therefore necessary for understanding both the strengths of the dataset and the limits of the experimental setting.

2.2.12 Advantages and limitations of the corpus

The Quranic Arabic Corpus offers several advantages for the present dissertation. First, it provides explicit and structured grammatical annotation that is well suited to supervised POS-tagging experiments. Second, it represents Arabic in a linguistically informed way, which is particularly valuable in a language where morphology, clitic attachment, and ambiguity strongly affect grammatical analysis. Third, the corpus can support both methodological explanation and experimental implementation, since it allows the researcher to examine annotation structure while also converting the data into a model-ready sequence-labeling format.

At the same time, the corpus also has limitations that must be acknowledged. Because it is based on Quranic Arabic, it represents a specific linguistic domain rather than the full range of Arabic varieties used in modern communication. Its annotation scheme is therefore highly valuable for grammatical analysis, but the resulting models may be less directly transferable to domains such as newswire, social media, or dialectal text. A clear discussion of these limitations is important because it helps interpret

the experimental results in a realistic and methodologically transparent way.

2.3 Corpus Annotation Structure

2.3.1 Annotated units

In the Quranic Arabic Corpus, the basic annotated unit is not always identical to the visible orthographic word as it appears in the text. This is an important property of the dataset because Arabic words often contain attached elements that carry distinct grammatical functions. As a result, the corpus adopts a linguistically motivated representation in which the analysis may distinguish between components that are merged in the surface written form.

This design is particularly relevant for POS tagging. If grammatical information is distributed across attached elements rather than contained in a single unanalyzed word, then the annotation unit must reflect that structure in order to preserve linguistic accuracy. For the purposes of this dissertation, the notion of annotated unit is therefore central, since it determines how the corpus is transformed into computational examples and how the input-output relation of the tagging task is defined.

2.3.2 Segmentation and token structure

Segmentation plays a central role in the representation of the Quranic Arabic Corpus because the surface form of Arabic often combines conjunctions, particles, prepositions, articles, stems, and pronoun markers within a single written word. The corpus is valuable in this respect because it preserves a more detailed linguistic structure than would be obtained from a simple whitespace-based tokenization. This makes the dataset especially informative for Arabic NLP, where grammatical interpretation often depends on the internal composition of the word.

From a computational point of view, segmentation has both benefits

and consequences. A segmented representation makes grammatical structure more explicit and may improve the ability of the model to capture relevant patterns. At the same time, it increases the number of units to be processed and may lead to a more complex sequence-labeling task. Thus, segmentation in the corpus is not merely a descriptive property, but a methodological factor that directly influences preprocessing, sequence construction, and model behavior.

2.3.3 Part-of-speech labels

The core annotation layer used in this dissertation is the set of part-of-speech labels provided by the Quranic Arabic Corpus. These labels define the target categories that the models must predict and therefore determine the output space of the supervised learning problem. In methodological terms, the POS inventory of the corpus is essential because it establishes how grammatical distinctions are encoded and how the tagging task is formulated computationally.

Such labels are particularly useful in Arabic because the grammatical function of a form is not always transparent from the surface structure alone. In the Quranic Arabic Corpus, POS labels provide an explicit grammatical interpretation that can be used consistently across the dataset. This consistency is important for both machine learning and deep learning approaches, since the reliability of the predicted output depends directly on the quality and coherence of the underlying annotation scheme.

2.3.4 Morphological features in the corpus

In addition to part-of-speech labels, the Quranic Arabic Corpus provides rich morphological information that increases the descriptive value of the dataset. Depending on the representation adopted, this may include features related to person, gender, number, tense, definiteness, lemma, root, and other aspects of grammatical structure. Such information is particularly important in Arabic because a large part of grammatical meaning is encoded morphologically rather than through separate function words.

These morphological features are relevant to the dissertation even when the main predictive task remains POS tagging. They make it possible to interpret the internal structure of annotated forms more precisely and can support feature engineering in the classical machine learning setting. They also increase the analytical value of the corpus by allowing the researcher to relate model behavior to specific grammatical properties. In this way, the corpus serves not only as a source of POS labels, but also as a richer linguistic resource that supports explanation, feature design, and error analysis.

2.3.5 Sequence representation

To prepare the corpus for POS tagging, the data must be represented as ordered sequences in which each input unit is aligned with an output label [2]. This representation reflects the fact that POS tagging is a sequence-labeling problem rather than an isolated classification task [2]. Preserving token order is therefore essential for both training and prediction.

Sequence representation is particularly important in Arabic because context often determines the interpretation of ambiguous forms [3, 2]. A sequence-based dataset enables the model to exploit dependencies between neighboring units and to learn grammatical patterns distributed across the sentence. This is why sequence organization is one of the key stages in the preparation of the experimental data.

2.3.6 Example of annotated sentence

A concrete example of annotated data is useful because it shows how theoretical concepts are instantiated in the corpus. Instead of describing annotation only abstractly, an example makes it possible to observe how tokens, POS labels, and morphological features interact at the sentence level. This helps the reader understand the structure of the data that will later be processed by the models.

Such examples are particularly valuable in Arabic because segmentation and morphology may produce annotation units that are not immediately

obvious from the original written form. Presenting one illustrative sentence with its associated labels therefore improves both clarity and methodological transparency.

Token	**Segment**	**POS label**	**Possible feature**
قرأ	قرأ	Verb	Past tense
الطالب	ال + طالب	Noun	Definite, singular
الكتاب	ال + كتاب	Noun	Definite, singular

Table 2.2: Illustrative example of an annotated Arabic sentence

Table 2.2 gives a simple illustration of how an Arabic sentence may be represented in annotated form for POS-tagging purposes. It highlights the relation between the written token, its segmented structure, the assigned POS category, and selected grammatical information.

2.4 Data Preparation

2.4.1 Objectives of preprocessing

Before the corpus can be used in machine learning and deep learning experiments, it must be transformed into a suitable computational representation. The purpose of preprocessing is to extract the relevant units, retrieve the associated POS labels, reduce inconsistency, and organize the data into a form that can be used for training, validation, and testing. This stage is essential because annotated corpora are not always stored in a format directly compatible with sequence-modeling pipelines.

Preprocessing is particularly important in Arabic NLP because decisions made at this stage directly affect how morphology and segmentation are represented [3, 5]. If important information is lost during data preparation, the model may fail to capture key grammatical distinctions. Thus, preprocessing should be viewed as a methodological stage that shapes the experimental problem itself.

2.4.2 Tokenization and segmentation

Tokenization consists of dividing the text into processing units, whereas segmentation concerns the decomposition of complex forms into smaller grammatical parts [3, 5]. In Arabic, these two processes are closely connected because many orthographic words contain multiple attached elements [3]. The way the corpus handles these elements has a direct influence on the complexity and quality of the POS-tagging dataset.

A coarse word-level representation may simplify the task, but it may also hide relevant morphological information. A more fine-grained segmented representation preserves structure more accurately, though it can increase the number of units and the difficulty of the learning problem. The selected tokenization strategy must therefore reflect the balance between linguistic fidelity and computational practicality.

2.4.3 Label extraction

Once the processing units are defined, the corresponding POS labels must be extracted from the annotation [1, 2]. This requires identifying the relevant grammatical category for each unit and ensuring that the labels are represented consistently across the dataset. Label extraction must be handled carefully because inconsistent tags can introduce noise and reduce model performance.

In some corpus settings, additional grammatical features may accompany the POS labels [3]. Although the main task of the dissertation is POS tagging, such information can still be useful for analysis and interpretation, and in some cases for feature design in classical models. The extraction stage therefore determines not only the target labels, but also the range of linguistic information made available to the experimental pipeline.

2.4.4 Normalization

Normalization aims to reduce superficial inconsistencies in the textual representation while preserving linguistically meaningful distinctions. In Ara-

Tag	Meaning
N	Noun
V	Verb
P	Preposition
PN	Proper noun
ADJ	Adjective
PRON	Pronoun
DEM	Demonstrative pronoun
REL	Relative pronoun
T	Time adverb
LOC	Location adverb
NEG	Negative particle
CONJ	Coordinating conjunction
SUB	Subordinating particle
ACC	Accusative particle
VOC	Vocative particle
INT	Interrogative particle

Table 2.3: Primary part-of-speech tags used in the Quranic Arabic Corpus

bic, this may involve handling orthographic variants or harmonizing the representation of forms that differ only in ways that are not central to the task. A well-designed normalization strategy improves consistency without removing relevant morphological or grammatical information.

This step is especially important because Arabic writing can exhibit variation across texts, genres, and annotation sources [3, 5]. If such variation is left untreated, the model may waste capacity learning accidental differences rather than genuine grammatical regularities. Normalization therefore contributes to cleaner input and more reliable training.

2.4.5 Sequence construction

After tokenization, label extraction, and normalization, the data must be organized into coherent sequences. This step is necessary because POS tagging depends on contextual information and is most naturally modeled at the sentence level rather than as isolated token-label pairs [2]. The final dataset should therefore preserve the order of the units and their contextual

relations.

This organization is particularly useful for models such as CRF, BiLSTM, BiLSTM-CRF, and transformer-based architectures, all of which exploit sequential or contextual dependencies [10, 14, 15, 16, 4, 8]. In Arabic, where ambiguity is often resolved by context, sequence construction is an essential part of effective preprocessing.

2.4.6 Dataset splitting

Once the sequences are constructed, the dataset must be divided into training, validation, and test subsets [2]. This division enables the model to be trained on one portion of the data, tuned on another, and evaluated objectively on unseen examples. Such a procedure is necessary to measure genuine generalization rather than memorization.

For Arabic POS tagging, careful splitting is especially important because some categories or structures may be less frequent than others [7]. An unbalanced split may distort evaluation and produce misleading conclusions about model quality. A consistent and well-designed partition therefore supports fair comparison between alternative methods.

2.4.7 Challenges during preprocessing

Preprocessing Arabic data may present several challenges, including inconsistent annotation, segmentation ambiguity, orthographic variation, and imbalance across labels. Some forms may be difficult to tokenize consistently, while others may carry multiple possible grammatical interpretations depending on context. These issues complicate the construction of a clean sequence-labeling dataset and must be managed carefully during corpus preparation.

Another challenge concerns the balance between linguistic detail and modeling practicality. Preserving rich annotation may provide useful information, but it can also increase data sparsity and make the learning problem more difficult. A well-designed preprocessing strategy therefore

seeks to retain the information that matters most for POS tagging while reducing avoidable complexity [3, 5].

Stage	**Purpose**	**Expected result**
Data reading	Load the annotated corpus into a computational environment	Accessible raw data
Tokenization / segmentation	Define the linguistic units used for tagging	Word-level or segmented units
Label extraction	Retrieve POS categories from annotation	Input-output pairs for learning
Normalization	Reduce avoidable textual inconsistency	Cleaner and more consistent representation
Sequence construction	Organize units into ordered sentences or sequences	Sequence-labeling dataset
Dataset splitting	Separate training, validation, and test data	Reliable experimental evaluation

Table 2.4: Main preprocessing stages for Arabic POS-tagging data

Table 2.4 summarizes the main preparation stages required to transform an annotated Arabic corpus into a dataset suitable for POS-tagging experiments [2, 3].

2.5 Methodological Importance of the Chapter

2.5.1 Connection to the experimental study

This chapter provides the linguistic and data-related foundation required for the experimental part of the dissertation. The later comparison between machine learning and deep learning approaches can only be interpreted properly if the structure of Arabic, the nature of the corpus, and the preprocessing choices are clearly understood in advance. In this sense, the present chapter prepares the reader for the methodological and empirical chapters that follow.

The representation of the corpus influences not only implementation, but also the meaning of the results. Different segmentation strategies, label granularities, and normalization choices can affect model behavior

and therefore shape the comparison between approaches. By explaining these issues explicitly, the dissertation ensures continuity between linguistic analysis and computational experimentation.

2.5.2 Importance for interpretation of results

Understanding the corpus and the preparation pipeline is also necessary for interpreting the strengths and weaknesses of the models. If a particular approach performs better or worse, the explanation may depend partly on the nature of the annotation, the distribution of labels, or the handling of morphological detail. Experimental results should therefore be interpreted in relation to the linguistic and structural properties of the data.

This point is especially important in Arabic NLP because the interaction between morphology, ambiguity, and variation can strongly influence tagging performance [3, 5, 7]. The present chapter thus plays both a descriptive and an interpretive role. It clarifies the data conditions under which the models are evaluated and provides the basis for a more meaningful analysis of later results.

2.6 Conclusion

This chapter introduced the Arabic language as the linguistic framework of the dissertation and highlighted the main characteristics that make Arabic challenging for natural language processing, especially morphological richness, ambiguity, clitic attachment, dialectal variation, and orthographic complexity [3, 5, 6]. It also explained the importance of the selected annotated corpus and showed how segmentation, POS-label assignment, morphological features, and sequence representation support the formulation of Arabic POS tagging as a supervised sequence-labeling task [1, 2].

In addition, the chapter described the main stages of data preparation, including tokenization, segmentation, label extraction, normalization, sequence construction, dataset splitting, and preprocessing challenges [2, 3]. These steps form the practical basis of the experimental study and pre-

pare the data for the machine learning and deep learning models discussed in the following chapters [4, 8]. The next chapter can therefore focus on the methodological framework adopted for implementation, training, and evaluation.

Chapter 3

Methodology and Experimental Design

3.1 Introduction

This chapter presents the methodological framework adopted in this dissertation for the design of an Arabic part-of-speech tagging system [1, 2]. After the previous chapter introduced the Arabic language and the corpus, the present chapter explains how the task is formulated, how the dataset is prepared for computational modeling, and how the selected machine learning and deep learning methods are organized for comparison [1, 2, 3]. The chapter therefore serves as the bridge between the linguistic description of the corpus and the experimental evaluation presented later in the dissertation [2].

The methodology of this study is based on a supervised learning setting in which annotated linguistic data is used to train models that assign part-of-speech labels to Arabic words [1, 2]. In accordance with the objectives of the dissertation, the methodological design is structured around two major directions: a classical machine learning approach based on Support Vector Machines and a set of deep learning approaches based on embeddings and neural sequence models [9, 12, 4, 8]. This organization makes it possible to compare feature-based and representation-based paradigms within the same POS-tagging task [2, 8].

This chapter is divided into four main parts. It first introduces the research framework and defines the task from a methodological perspective [1, 2]. It then explains how the corpus is prepared and transformed into datasets suitable for machine learning and deep learning [3, 5]. After that, it presents the SVM-based approach and the feature sets adopted for it, before describing the deep learning methods used in the study, including embeddings, LSTM, BiLSTM-CRF, and BERT-based architectures [9, 11, 14, 15, 16, 4, 8].

3.2 Problem Formulation and Comparative Design

3.2.1 Task definition

The task addressed in this dissertation is Arabic part-of-speech tagging formulated as a supervised sequence-labeling problem [1, 2]. Given an input sequence of Arabic words or segments, the objective is to assign to each unit its corresponding grammatical label, such as noun, verb, adjective, pronoun, or particle [1]. From a computational point of view, the problem can be represented as a mapping from an input sequence $X = (x_1, x_2, \ldots, x_n)$ to an output label sequence $Y = (y_1, y_2, \ldots, y_n)$, where each output tag corresponds to one input token [2].

Although POS tagging is generally modeled as sequence labeling, the present study adopts two complementary methodological perspectives. In the machine learning setting, each token is represented through an explicit feature vector and classified according to its corresponding part-of-speech label [9]. In the deep learning setting, the model learns internal representations of the input through embeddings and neural architectures that exploit contextual dependencies [11, 12, 8]. This dual perspective makes it possible to analyze how different forms of representation influence POS-tagging performance [2, 4].

The task is especially suitable for this dissertation because it lies at the intersection of linguistic annotation and predictive modeling [1, 2]. It is lin-

guistically meaningful, since it reflects grammatical structure, and methodologically appropriate, since it allows the comparison of classical feature engineering and modern representation learning [2, 8]. This makes Arabic POS tagging an effective case study for exploring annotation-oriented NLP methods [3, 5].

3.2.2 Methodological overview

The overall methodology of the dissertation is organized as a comparative experimental framework [2]. The corpus is first transformed into a computationally usable dataset through formatting, preprocessing, and partitioning [1, 3]. Once the data has been prepared, two families of methods are developed and studied: a classical machine learning approach through Support Vector Machines and a deep learning approach through LSTM, BiLSTM-CRF, and BERT-based architectures [9, 11, 14, 15, 16, 4, 8]. Figure 3.1 illustrates this overall workflow.

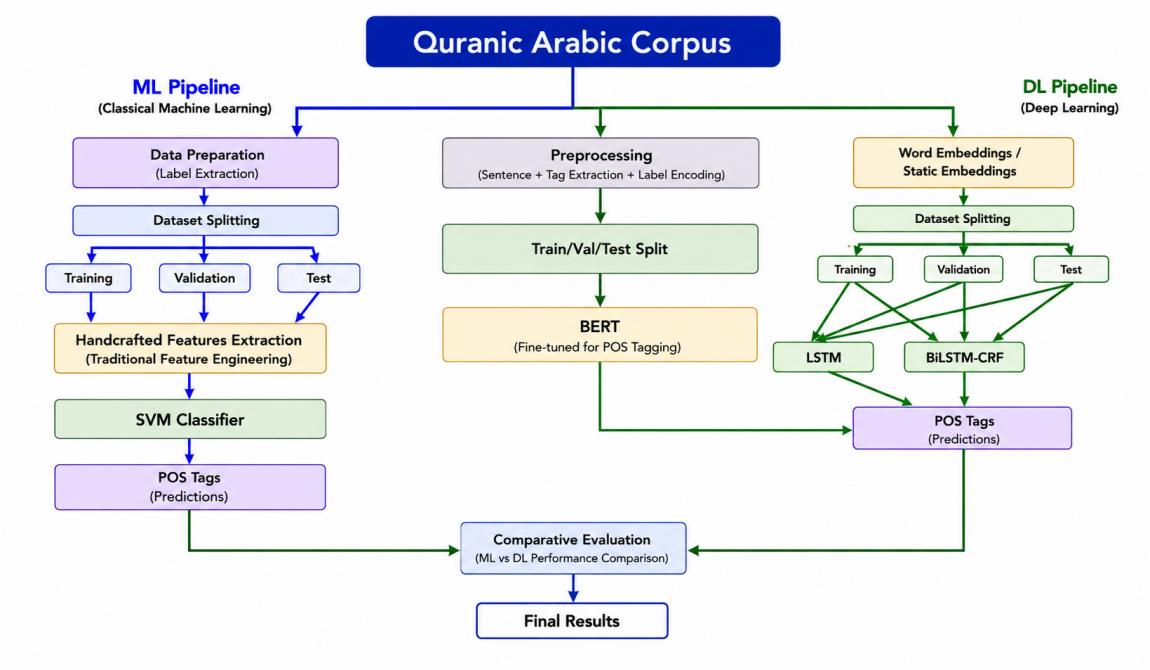


Figure 3.1: Overall methodological workflow of the Arabic POS-tagging system

For the machine learning approach, the study relies on handcrafted linguistic features extracted from each token, which is consistent with the

classical feature-based paradigm in POS tagging . In this part, four dataset variants are considered: non-morphological and morphological feature sets, each evaluated without and with the preceding-tag sequence context feature . This makes it possible to examine the contribution of morphology to the performance of an SVM-based Arabic POS tagger .

For the deep learning approach, the study uses embedding-based representations of Arabic text . Instead of manually constructing all predictive features, the models receive distributed vector representations of tokens and learn contextual patterns automatically through neural architectures . This methodological choice reflects the broader evolution of NLP from explicit feature engineering toward data-driven representation learning .

The methodological comparison is therefore not limited to algorithms alone. It also concerns the nature of the input representation, the role of explicit linguistic knowledge, and the way contextual information is incorporated into the tagging process . In this sense, the chapter establishes the conceptual and procedural basis for the experimental analysis presented later in the dissertation .

3.3 Experimental Dataset Construction

3.3.1 Corpus formatting

Before the corpus can be used in machine learning and deep learning experiments, it must be transformed into a structured computational format [1]. Annotated linguistic material is not always directly suitable for model training, so the first methodological step consists of converting the corpus into a standardized format in which each textual unit is aligned with its corresponding part-of-speech label [1, 2]. This alignment is essential for supervised learning because the model must learn a mapping between input instances and target labels [1, 2].

At this stage, the representation of the data must reflect the annotation scheme adopted in the corpus . Since the corpus is already segmented and annotated, each example preserves the relation between the token and its

POS tag while maintaining consistency across the dataset . In practical terms, this means organizing the data into sequences or token-label pairs that can later be processed by classical classifiers or neural sequence models . The formatting step is therefore a necessary foundation for all later experiments .

3.3.2 Preprocessing steps

After formatting, the corpus undergoes preprocessing in order to improve consistency and prepare the data for computational learning . Preprocessing includes the operations required to transform the annotated corpus into a model-ready dataset while preserving the linguistic information needed by the tagging task. In Arabic POS tagging, this stage is especially important because the language presents orthographic variation and morphological richness .

The preprocessing pipeline may involve operations such as cleaning the text, organizing sentence boundaries, extracting the relevant labels, and ensuring that the selected token representation remains compatible with the annotation scheme [1, 2]. Since the corpus is already segmented, segmentation is treated as part of the corpus representation rather than as a step performed in this work [3, 5]. The aim of preprocessing is therefore not to segment the corpus from scratch, but to make the data coherent and computationally usable [3].

3.4 Dataset Preparation

The Quranic Arabic Corpus is prepared following the pipeline described in Chapter 2, which includes label extraction, and sequence construction. The resulting dataset is then partitioned into training, validation, and test sets. For the deep learning models, the split is 80% training, 10% validation, and 10% testing. For the SVM experiments, an additional partition of 70% training, 15% validation, and 15% testing is used to accommodate cross-validation for hyperparameter selection.

3.5 Machine Learning Method

3.5.1 Support Vector Machine

The machine learning component of this dissertation is based on Support Vector Machines [9]. SVM is a supervised classification method that learns a decision boundary capable of separating instances belonging to different classes [9]. In the context of POS tagging, the classifier receives a representation of each token and predicts the corresponding grammatical category [9]. This makes SVM an appropriate representative of the classical feature-based approach to NLP tagging [9, 3].

SVM is particularly suitable for the comparative design of this dissertation because it contrasts clearly with deep learning methods [9]. Unlike neural architectures, which learn internal representations automatically, SVM depends on explicit feature engineering [9]. This makes it a useful methodological baseline and allows the study to evaluate the role of handcrafted linguistic information in Arabic POS tagging [3, 4]. In this way, the SVM model is not only a classifier, but also a means of studying the usefulness of selected linguistic descriptors .

3.5.2 Non-morphological feature set

The first SVM dataset variant is based on non-morphological features [9]. In this setting, each token is represented through general descriptive properties that do not explicitly encode internal morphological structure [9]. Such features may include surface-level information such as token length, orthographic form, or other basic descriptors that help distinguish grammatical classes [9]. This variant therefore establishes a simpler feature-based representation for POS classification [9].

The motivation behind this dataset version is methodological clarity. By excluding explicit morphological descriptors, the study can observe how far an SVM classifier can perform POS tagging using only basic token-level information [9]. This provides a reference point against which the effect of richer linguistic features can later be measured [3, 4]. It is therefore useful

as a baseline inside the machine learning part of the study [9].

3.5.3 Morphological feature set

The second SVM dataset variant extends the previous representation by incorporating morphological information, which is especially important in Arabic [3, 5]. Arabic morphology carries a large part of the grammatical information needed for POS disambiguation, since words may include prefixes, suffixes, clitics, and inflectional patterns that are strongly related to grammatical category [3, 5]. Thus, adding morphological features is expected to improve the discriminative capacity of the classifier [3, 4].

In practical terms, the morphological feature set may include descriptors such as proclitics, prefixes, suffixes, enclitics, and other indicators related to the internal structure of the token [3, 5]. These features provide the classifier with information that is more directly connected to Arabic grammatical composition than the simpler non-morphological representation [3]. The comparison between the two SVM datasets is therefore methodologically important because it isolates the effect of morphology on a classical classifier [3, 4]. This makes the machine learning component of the dissertation more informative and better adapted to the linguistic properties of Arabic [3].

3.5.4 Experimental configurations

In addition to the morphological and non-morphological feature sets, two additional experimental factors are introduced. First, each feature set is evaluated both with and without the part-of-speech label of the preceding token, which serves as a sequence context feature. Second, each configuration is evaluated with and without the token-length descriptor. This yields four SVM configurations: non-morphological features without and with context, and morphological features without and with context. Such a design makes it possible to isolate the contribution of morphology, sequence information, and token-level properties within a unified comparative framework.

3.5.5 Feature encoding

Once the relevant features have been selected, they must be transformed into numerical representations suitable for SVM classification [9]. Since SVM operates on vectors, each token must be encoded as a fixed-size feature vector that combines the selected descriptors into a machine-readable form [9]. This stage converts linguistic or orthographic observations into structured input for the classifier [9, 3]. Feature encoding therefore functions as the link between feature engineering and statistical learning [9].

Feature encoding is especially important in the machine learning approach because the quality of the representation directly affects the classifier's decision boundary [9]. A weak encoding may hide useful distinctions or introduce unnecessary sparsity, whereas a well-designed encoding scheme can highlight the contrasts that matter for grammatical classification [9]. In this dissertation, feature encoding is therefore treated as a central methodological step in the SVM pipeline [9, 3].

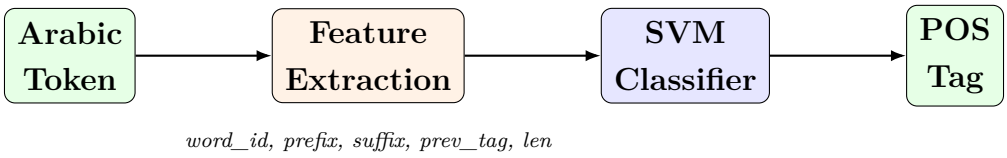


Figure 3.2: SVM pipeline for Arabic POS tagging with feature extraction.

3.6 Deep Learning Methods

3.6.1 Word embeddings

The deep learning component of the dissertation relies on embedding-based representations of Arabic text [12, 8]. Embeddings are dense vector representations of linguistic units in a continuous space, where related words or subword patterns tend to receive closer representations than in sparse symbolic schemes [12]. Unlike manually engineered features, embeddings allow the model to capture useful regularities automatically and to exploit patterns that may not be explicitly encoded by the researcher [12, 8]. This makes them particularly suitable for modern sequence-labeling models [12].

In the present study, embeddings constitute the input representation used by the neural models [12]. Instead of constructing a handcrafted feature vector for each token, the system represents words through learned numerical vectors that are fed directly into LSTM, BiLSTM-CRF, and BERT-based architectures [11, 14, 15, 16, 8]. This shift from manual feature engineering to learned representation is one of the major methodological differences between classical machine learning and deep learning [2]. It is also especially relevant for Arabic, whose morphological richness and ambiguity make dense contextual representation highly useful [3, 5, 8].

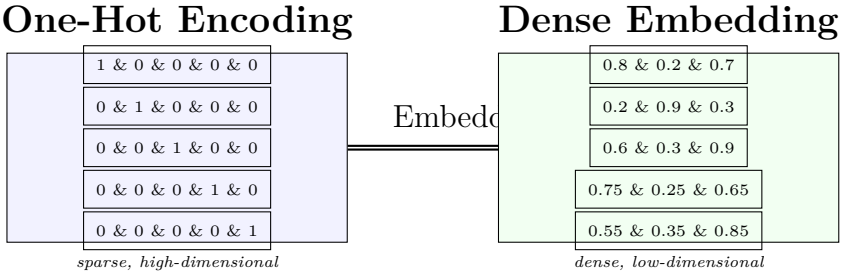


Figure 3.3: Comparison between sparse one-hot encoding and dense word embeddings.

3.6.2 LSTM architecture

The first neural architecture considered in the study is the Long Short-Term Memory network [11]. LSTM is a recurrent neural model designed to process sequential data while preserving relevant information over longer contexts than standard recurrent networks [11]. This makes it well suited to POS tagging, where the interpretation of a token often depends on surrounding words [11]. The model therefore serves as a natural neural baseline for sequence labeling [11].

In this dissertation, the LSTM model receives embedding-based token representations as input and processes them sequentially in order to predict POS labels [11]. Its main advantage lies in its ability to model contextual dependencies without depending entirely on manually engineered features [11]. This is especially useful in Arabic, where many tokens are ambiguous in isolation and require contextual interpretation [3, 5]. The LSTM

architecture thus provides the first deep learning reference point in the comparative framework [11].

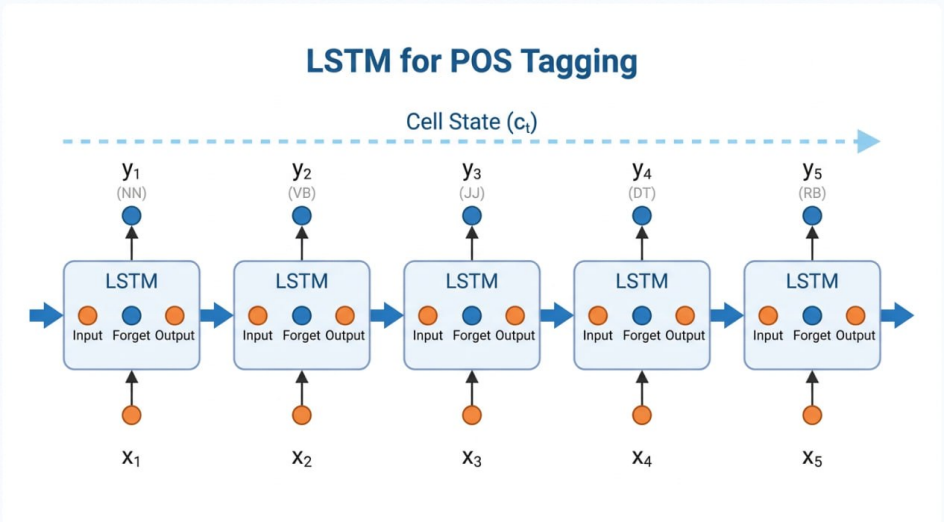


Figure 3.4: LSTM architecture for Arabic POS tagging

3.6.3 BiLSTM-CRF architecture

The second neural architecture adopted in the dissertation is the BiLSTM-CRF model [14]. This architecture combines a bidirectional LSTM, which captures context from both left and right directions, with a Conditional Random Field layer that models dependencies between adjacent output tags [10, 14]. Such a combination is particularly useful for sequence labeling because grammatical labels are not independent from one another [14]. The model therefore supports both contextual representation and structured output prediction [14].

BiLSTM-CRF is especially relevant to POS tagging because certain tag sequences are more plausible than others [14]. A model that predicts tags independently may produce less coherent output, whereas the CRF layer encourages globally consistent label sequences [10, 14]. For Arabic POS tagging, where context and morphology both matter, this architecture is particularly attractive [4, 5]. It therefore provides a stronger neural alternative to simple recurrent tagging models [14, 4].

3.6.4 BERT-based architecture

The third deep learning method considered in the dissertation is a BERT-based architecture [15, 16, 8]. BERT belongs to the family of transformer-based language models and relies on self-attention rather than strictly recurrent computation [15, 16, 8]. Its main strength lies in its ability to generate contextualized representations in which the vector associated with a token depends dynamically on the surrounding sentence [16, 8]. This makes it especially appropriate for ambiguous and context-sensitive NLP tasks [16, 8].

In this study, the BERT-based model is used as an advanced contextual representation approach for POS tagging [16, 8]. Unlike static embeddings, contextualized representations allow the same surface form to receive different vectors depending on grammatical and lexical context [16, 8]. This property is highly relevant for Arabic, where many forms are ambiguous and require context-sensitive interpretation [3, 8]. The inclusion of BERT therefore extends the comparison toward contemporary transformer-based methods and reflects the current state of research in Arabic sequence labeling [15, 16, 8].

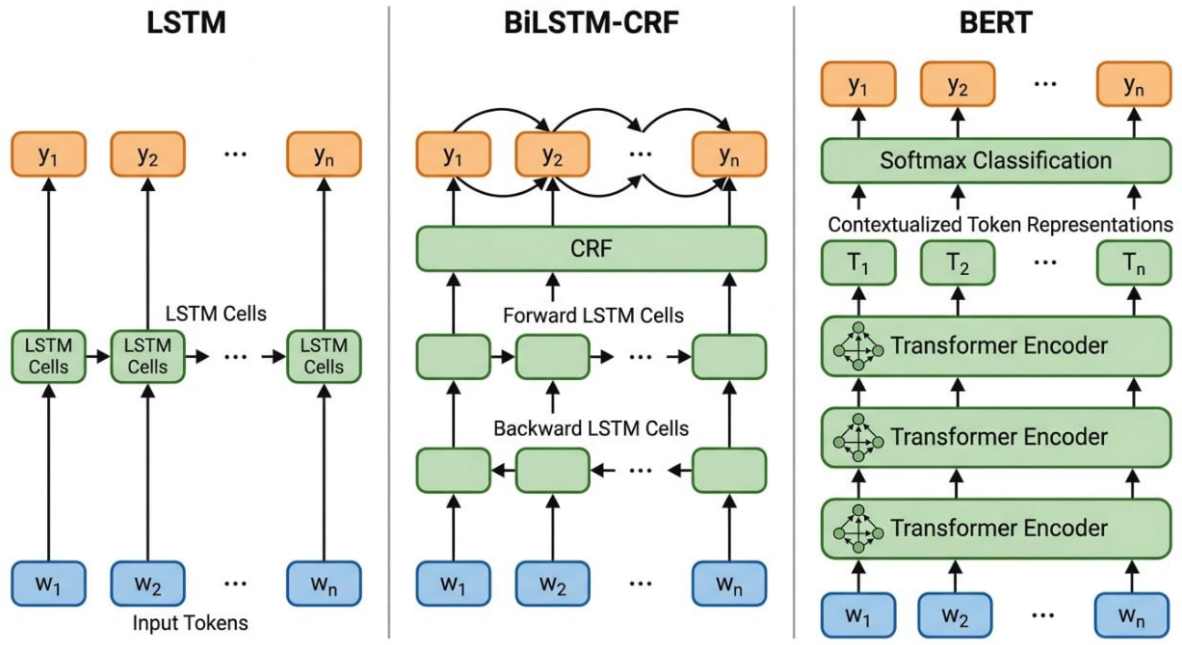


Figure 3.5: Comparison of neural architectures: LSTM, BiLSTM-CRF, and BERT.

3.7 Conclusion

This chapter presented the methodological and experimental design adopted in the dissertation [2]. It first defined Arabic POS tagging as a supervised learning problem and described the comparative framework used in the study [1, 2]. It then explained dataset preparation through formatting, preprocessing, and partitioning, before introducing the two major methodological branches of the work: a classical machine learning approach based on SVM and a deep learning approach based on embeddings and neural architectures [9, 11, 14, 15, 16, 3, 8].

Within the machine learning branch, the chapter explained the use of two dataset variants, one without morphological features and one with morphological features, in order to study the contribution of Arabic morphology to SVM-based tagging [3, 5, 4]. Within the deep learning branch, it presented embedding-based input representations together with LSTM, BiLSTM-CRF, and BERT-based architectures [11, 14, 15, 16, 8]. In this way, the chapter established the methodological basis needed for the following chapter, which will present the experimental configuration, system implementation, and comparative analysis of results [2, 8].

All models are evaluated using three metrics: accuracy, macro F1-score, and weighted F1-score. Accuracy is defined as the ratio of correctly predicted tags to the total: Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$. The F1-score for each class is the harmonic mean of precision and recall: $F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$. Macro F1 averages these scores across all classes without weighting, while weighted F1 averages them weighted by the number of true instances for each class:

$$\text{Macro } F1 = \frac{1}{n} \sum_{i=1}^{n} F1_i$$

$$\text{Weighted } F1 = \frac{\sum_{i=1}^{n} w_i \times F1_i}{\sum_{i=1}^{n} w_i}$$

where w_i is the number of true instances for class i.

Chapter 4

Experimental Results and Comparative Analysis

4.1 Introduction

After presenting the methodological framework in the previous chapter, this chapter reports the experimental results obtained from the SVM, LSTM, BiLSTM-CRF, and BERT-based models applied to Arabic part-of-speech tagging. The purpose of the chapter is to present a detailed and structured comparison of the four approaches using the same evaluation data derived from the Quranic Arabic Corpus. The chapter first describes the experimental setup and the evaluation metrics, then reports the results of each model family separately, and finally provides a comparative analysis to identify the strengths and limitations of each approach. In this way, the chapter serves as the empirical core of the dissertation and provides the evidence needed to compare machine learning and deep learning paradigms in the context of Arabic text annotation.

The chapter is organized as follows. Section 4.2 describes the experimental dataset and the evaluation protocol. Section 4.3 presents the SVM results with and without sequence features. Section 4.4 discusses the LSTM results across different embedding dimensions. Section 4.5 reports the BiLSTM-CRF results. Section 4.6 presents the BERT-based model performance. Section 4.7 provides a comparative analysis across all models.

Section 4.8 examines rare-tag performance, and Section 4.9 summarizes the chapter.

4.2 Experimental Setup

4.2.1 Dataset and preprocessing

The experiments are based on the Quranic Arabic Corpus, a linguistically annotated resource in which each token is associated with a part-of-speech label and additional morphological information. The corpus was preprocessed according to the pipeline described in Chapter 3, which includes tokenization, segmentation, label extraction, normalization, and sequence construction. The final dataset consists of token-label pairs organized into sentences, where each sentence is represented as a sequence of tokens with their corresponding POS tags.

The dataset split follows a stratified partition with 80% of the data used for training, 10% for validation, and 10% for testing. The stratification preserves the distribution of POS labels across the three subsets. For the SVM experiments, an additional partition was used (70% training, 15% validation, 15% testing) to accommodate cross-validation for hyperparameter selection.

4.2.2 Evaluation metrics

Four main evaluation metrics are used to assess model performance. Accuracy measures the proportion of correctly predicted tags over the total number of tokens. Macro F1 computes the unweighted average of F1 scores across all classes, which provides a balanced view of performance across both frequent and rare tags. Weighted F1 computes the average of F1 scores weighted by the number of true instances for each class, which reflects overall performance while accounting for class imbalance. Precision and recall are also reported for each class in the per-tag analysis. These metrics are standard in POS tagging evaluation and allow for consistent

comparison across models.

4.2.3 Models and naming conventions

Four model families are compared in this chapter. The SVM model is evaluated in four configurations: non-morphological features without and with sequence context, and morphological features without and with sequence context, each also tested with and without the token-length descriptor. The LSTM model is evaluated with pre-trained Word2Vec embeddings at four dimensions: 50, 150, 525, and 1000. The BiLSTM-CRF model is evaluated with the same four embedding dimensions. The BERT model uses a DistilBERT-multilingual-cased architecture fine-tuned for token classification.

4.3 Support Vector Machine Results

4.3.1 Hyperparameter selection

The SVM classifier was trained with a linear kernel using the LinearSVC implementation. Hyperparameter tuning was performed through grid search over the regularization parameter C using values 0.01, 0.1, 0.5, 1.0, 2.0, 5.0, and 10.0, with 3-fold stratified cross-validation. The scoring metric for selection was macro F1. Table 4.1 shows the best C values and corresponding cross-validation macro F1 scores for each dataset variant.

Dataset variant	**Group**	**Best C**	**CV Macro F1**
dataset_no_sequence_6	Fair baseline	5.0	0.7756
dataset_no_sequence_token_len_6	Fair baseline	5.0	0.7757
dataset_with_sequence_6	Context-enhanced	0.5	0.7851
dataset_with_sequence_token_len_6	Context-enhanced	0.5	0.7846

Table 4.1: Best C values and cross-validation macro F1 for SVM dataset variants

The results show that sequence-enhanced datasets favor a lower regularization value (C = 0.5), while non-sequence datasets require stronger regularization (C = 5.0). The token length feature has negligible impact on the cross-validation score.

4.3.2 Test set performance

After hyperparameter tuning, the best model for each dataset variant was refit on the combined training and validation sets and evaluated on the held-out test set. Table 4.2 reports the final test results.

Dataset variant	**Test Accuracy**	**Macro F1**	**Weighted F1**	**Training Time (min)**
no_sequence_6	0.9215	0.7766	0.9254	13.83
no_sequence_token_len_6	0.9219	0.7767	0.9259	12.14
with_sequence_6	**0.9398**	**0.8071**	**0.9415**	3.79
with_sequence_token_len_6	0.9390	0.8069	0.9407	3.56

Table 4.2: SVM test set results for all dataset variants

The results indicate that the addition of sequence context features (previous token POS tags) leads to a substantial improvement in performance. The context-enhanced variant achieves a macro F1 of 0.8071 compared to 0.7766 for the non-sequence baseline, representing a relative improvement of approximately 3.9%. The token length feature does not provide meaningful additional benefit, as the two sequence variants show nearly identical results.

4.3.3 Overfitting analysis

The gap between training macro F1 and validation macro F1 was examined to assess overfitting. For the non-sequence dataset, the gap is 0.0563 (train: 0.8157, val: 0.7594). For the sequence-enhanced dataset, the gap is 0.0622 (train: 0.8511, val: 0.7889). Both values indicate mild overfitting that is acceptable for the purposes of the comparison. The sequence-enhanced

model achieves a higher absolute validation score despite the slightly larger gap.

4.3.4 Fairness note on sequence features

It is important to note that the sequence features used in the context-enhanced SVM setting are gold-standard POS labels from the preceding tokens. While this configuration demonstrates the theoretical upper bound of SVM performance when contextual information is available, it does not reflect a fully realistic deployment scenario in which previous tags would need to be predicted rather than retrieved from annotation. The non-sequence baseline is therefore the fairest representation of SVM as an independent classifier. This distinction is acknowledged in the comparative analysis.

4.4 Word Embedding Analysis

4.4.1 Pre-trained Word2Vec embeddings

The deep learning models in this study rely on pre-trained Word2Vec embeddings learned from the Quranic Arabic Corpus. The embeddings were trained using the skip-gram architecture with a window size of 5 and a minimum word count of 1. Four embedding dimensions were produced: 50, 150, 525, and 1000. These embeddings were used as the input representation for both the LSTM and BiLSTM-CRF models, with the embedding layer set as trainable to allow fine-tuning during training.

4.4.2 Qualitative inspection

A qualitative inspection of the learned embeddings reveals that the Word2Vec representations capture relevant lexical and morphological regularities. Words that share similar grammatical functions or morphological patterns tend to cluster together in the embedding space. For example, pronouns, demonstratives, and verbal forms show distinguishable distributional patterns

across all embedding dimensions. This confirms that the embeddings provide a useful representation for the POS tagging task.

4.5 LSTM Results

4.5.1 Model architecture and training

The LSTM model consists of an embedding layer initialized with pre-trained Word2Vec weights, a single LSTM layer with 128 hidden units and a dropout rate of 0.2, and a time-distributed dense layer with softmax activation over the tag set. The model was trained with the Adam optimizer, categorical cross-entropy loss, and class weights computed to address label imbalance. Early stopping with a patience of 3 epochs was applied based on validation loss.

4.5.2 Effect of embedding dimension

Table 4.3 presents the test set results for the LSTM model across the four embedding dimensions.

Dim	**Test Acc**	**Macro F1**	**Weighted F1**	**Best Epoch**	**Time (min)**
50	0.9017	0.6929	0.9126	10	4.26
150	0.9179	0.7138	0.9248	7	3.71
525	0.9081	**0.7247**	0.9168	4	4.65
1000	0.9112	0.7069	0.9195	3	8.10

Table 4.3: LSTM test set results across embedding dimensions

The LSTM model achieves its highest macro F1 (0.7247) with embedding dimension 525, which is therefore selected as the best configuration within the LSTM family. The model with dimension 150 achieves the highest accuracy (0.9179) and weighted F1 (0.9248), but the macro F1 criterion prioritizes performance across all classes including rare tags. Higher embedding dimensions converge in fewer epochs, but do not necessarily yield better final performance. Dimension 1000 requires the longest training time

with the weakest macro F1, suggesting that excessively large embeddings introduce noise for this dataset.

4.5.3 Training dynamics

The learning curves show consistent reduction in training loss across all dimensions, while validation loss reaches a minimum within 3 to 10 epochs and then begins to increase, triggering early stopping. The overfitting gap is smallest for dimension 1000 (0.0080) and largest for dimension 525 (0.0191), although all values remain within acceptable bounds. Dimension 50 requires the most epochs (10) to reach its best validation performance, while dimensions 525 and 1000 converge rapidly (4 and 3 epochs respectively).

4.6 BiLSTM-CRF Results

4.6.1 Model architecture and training

The BiLSTM-CRF model consists of an embedding layer initialized with pre-trained Word2Vec weights, a bidirectional LSTM layer with 128 hidden units in each direction (256 combined), a dropout rate of 0.2, a linear classifier mapping the BiLSTM output to tag scores, and a CRF layer for structured prediction. The model was trained with the Adam optimizer at a learning rate of 1e-3, using the negative log-likelihood from the CRF as the loss function. Class weights were applied to address label imbalance. Early stopping with patience 3 based on validation loss was used for model selection.

4.6.2 Effect of embedding dimension

Table 4.4 presents the test set results for the BiLSTM-CRF model across the four embedding dimensions.

The BiLSTM-CRF model achieves its best performance with embedding dimension 150, which yields the highest test accuracy (0.9585), macro

Dim	Test Acc	Macro F1	Weighted F1	Best Epoch	Time (min)
50	0.9562	0.7927	0.9555	9	5.35
150	**0.9585**	**0.8090**	**0.9577**	7	5.29
525	0.9565	0.8008	0.9557	5	7.33
1000	0.9538	0.7476	0.9522	4	11.32

Table 4.4: BiLSTM-CRF test set results across embedding dimensions

F1 (0.8090), and weighted F1 (0.9577). This configuration is selected as the best within the BiLSTM-CRF family. The results show that larger embedding dimensions (525 and 1000) do not improve performance and, in the case of dimension 1000, lead to a notable degradation in macro F1 to 0.7476. This suggests that the Quranic Arabic dataset benefits from compact embedding representations, and that larger dimensions may introduce sparsity or overfitting that negatively impacts rare-tag performance.

4.6.3 Training dynamics

The BiLSTM-CRF models converge quickly, with best epochs ranging from 4 (dimension 1000) to 9 (dimension 50). The training loss decreases steadily across epochs for all dimensions, while validation loss reaches a minimum early and then begins to rise, triggering early stopping. The overfitting gaps are small across all dimensions (0.0246 to 0.0286), indicating that the models generalize reasonably well. Dimension 150 achieves the best balance between convergence speed (7 epochs), training time (317 seconds), and final performance.

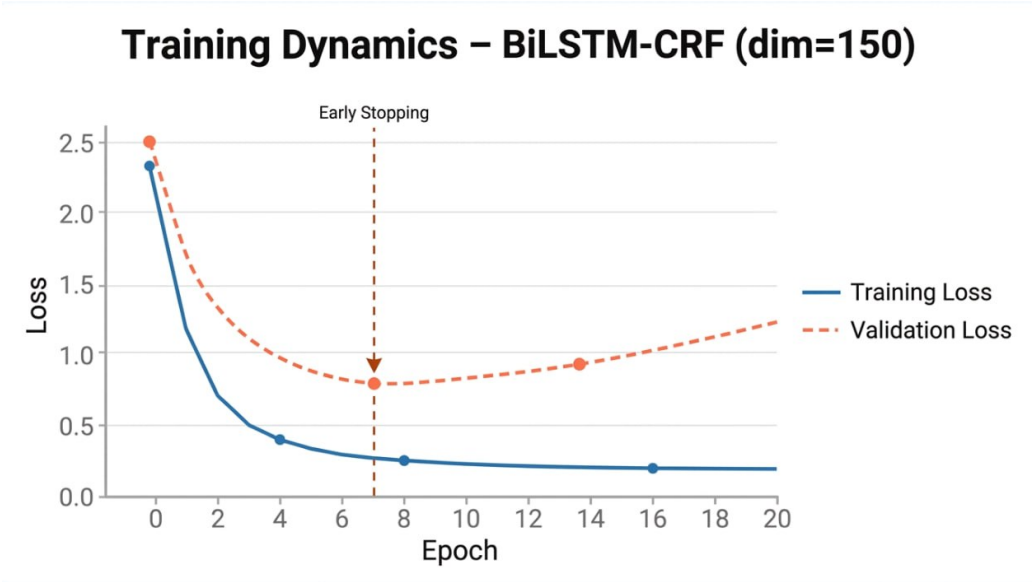


Figure 4.1: Training and validation loss curves for LSTM (dim=525)

4.7 BERT Model Results

4.7.1 Model architecture and fine-tuning

The BERT-based model uses the DistilBERT-multilingual-cased architecture, a distilled version of multilingual BERT with 6 transformer layers, 768 hidden dimensions, and 12 attention heads. The model was fine-tuned for token classification with a learning rate of 2e-5, a batch size of 16, and the AdamW optimizer. Training was performed for up to 15 epochs with early stopping based on macro F1 on the validation set.

4.7.2 Performance and training dynamics

The best checkpoint was obtained at epoch 10 (step 3120), after which validation performance began to degrade. Training was stopped early at epoch 13. Table 4.5 presents the final test set results.

The BERT model achieves a high test accuracy of 0.9541 and a weighted F1 of 0.9522, which are competitive with the best results from the BiLSTM-CRF model. However, its macro F1 of 0.7496 is substantially lower than the BiLSTM-CRF macro F1 of 0.8090. This gap indicates that the BERT model performs less effectively on less frequent and rare POS tags, despite

Metric	Value
Test Accuracy	0.9541
Macro F1	0.7496
Weighted F1	0.9522

Table 4.5: BERT (DistilBERT-mBERT) test set results

strong overall accuracy on frequent tags. The per-tag classification report shows that several rare tags (CAUS, EXP, SUR) receive zero F1 scores, which contributes to the lower macro average.

4.8 Comparative Analysis

4.8.1 Comparison of machine learning models

Within the SVM family, the addition of sequence features provides a clear improvement. The context-enhanced SVM achieves a macro F1 of 0.8071 compared to 0.7766 for the non-sequence baseline, an improvement of 0.0305 points. This demonstrates that contextual information from preceding tags is valuable for SVM-based POS tagging, even when other features remain unchanged. The context-enhanced SVM is therefore the best-performing machine learning model.

4.8.2 Comparison of deep learning models

Among the deep learning models, BiLSTM-CRF with embedding dimension 150 achieves the highest scores across all three metrics. The LSTM model with dimension 525 achieves the second-best macro F1 (0.7247) but falls substantially behind BiLSTM-CRF. The BERT model achieves strong accuracy and weighted F1 but its macro F1 is significantly lower than that of BiLSTM-CRF. This suggests that for the Quranic Arabic dataset, which is relatively small and domain-specific, the BiLSTM-CRF architecture with compact embeddings offers a better balance between overall accuracy and rare-tag coverage than the large-scale transformer model.

4.8.3 Final comparison: best ML versus best DL

Metric	**SVM (context-enhanced)**	**BiLSTM-CRF (dim=150)**
Test Accuracy	0.9398	**0.9585**
Macro F1	0.8071	**0.8090**
Weighted F1	0.9415	**0.9577**

Table 4.6: Comparison between best ML and best DL models

The BiLSTM-CRF model with embedding dimension 150 is the best-performing model overall. It achieves the highest test accuracy (0.9585), the highest macro F1 (0.8090), and the highest weighted F1 (0.9577). The SVM model with sequence features achieves a competitive macro F1 of 0.8071, demonstrating that a well-designed feature-based classifier can perform nearly as well as the best neural model on the macro F1 metric. However, the BiLSTM-CRF model outperforms SVM on accuracy by 1.87 percentage points and on weighted F1 by 1.62 percentage points.

The BERT model, despite its architectural sophistication, does not outperform the BiLSTM-CRF model on this dataset. Its macro F1 of 0.7496 is 5.94 points lower than BiLSTM-CRF, suggesting that the transformer model's strength in capturing contextualized representations is less beneficial for this relatively homogeneous and domain-specific corpus compared to the structured prediction capability of the BiLSTM-CRF architecture.

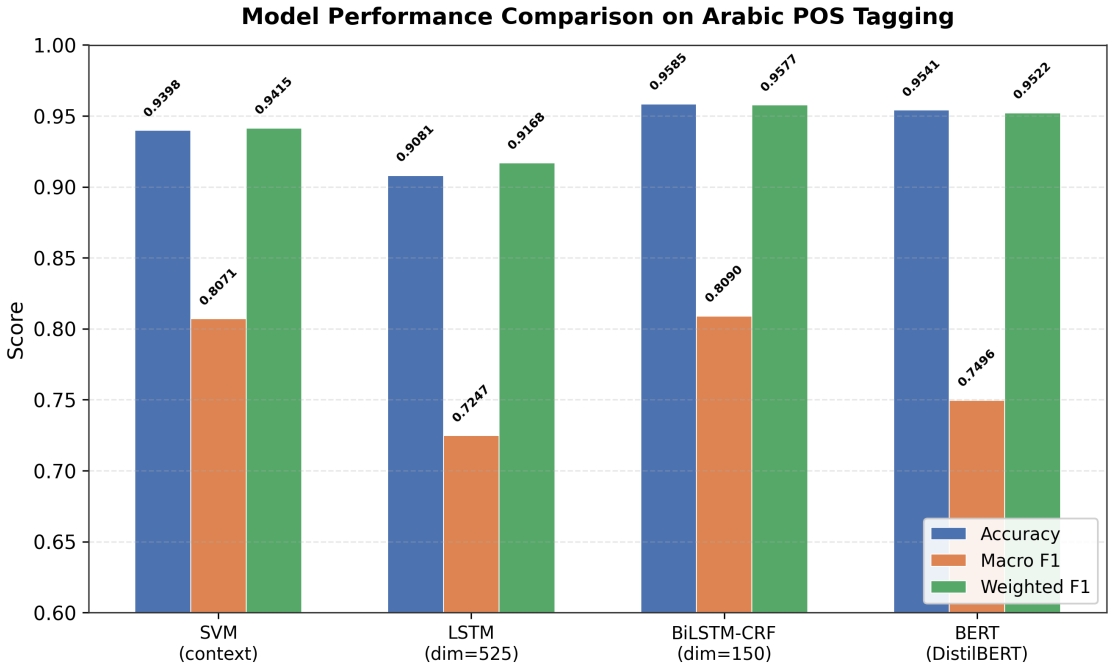


Figure 4.2: Comparative performance of the four POS tagging models across all evaluation metrics.

4.9 Rare-Tag Performance Analysis

4.9.1 Definition of rare tags

Rare tags are defined as POS categories with fewer than 100 training instances. Sixteen tags meet this criterion in the dataset. These tags represent grammatical categories that appear infrequently in the corpus, and their correct prediction is particularly challenging for all models.

4.9.2 Performance across models

Table 4.7 presents the macro F1 scores for selected rare tags across the best configuration of each model family.

Tag	Train count	SVM (seq)	LSTM (525)	BiLSTM-CRF (150)	BERT
RET	98	0.8421	0.9000	0.9000	0.93
INC	72	0.9000	0.6429	0.9000	1.00
EXP	66	0.0000	0.0000	0.0000	0.00
CAUS	65	0.0000	0.1250	0.1538	0.00
IMPV	51	0.9231	0.8571	0.9231	0.89
EXL	50	0.9474	0.9000	0.8889	1.00
AMD	39	0.9412	0.9000	0.9412	1.00
INT	35	0.6667	0.7273	0.8000	0.57
ANS	31	1.0000	1.0000	1.0000	0.67
SUR	28	1.0000	0.5714	1.0000	0.00
AVR	25	1.0000	1.0000	1.0000	1.00
INL	23	0.8889	1.0000	1.0000	1.00

Table 4.7: Rare-tag F1 scores across best model configurations

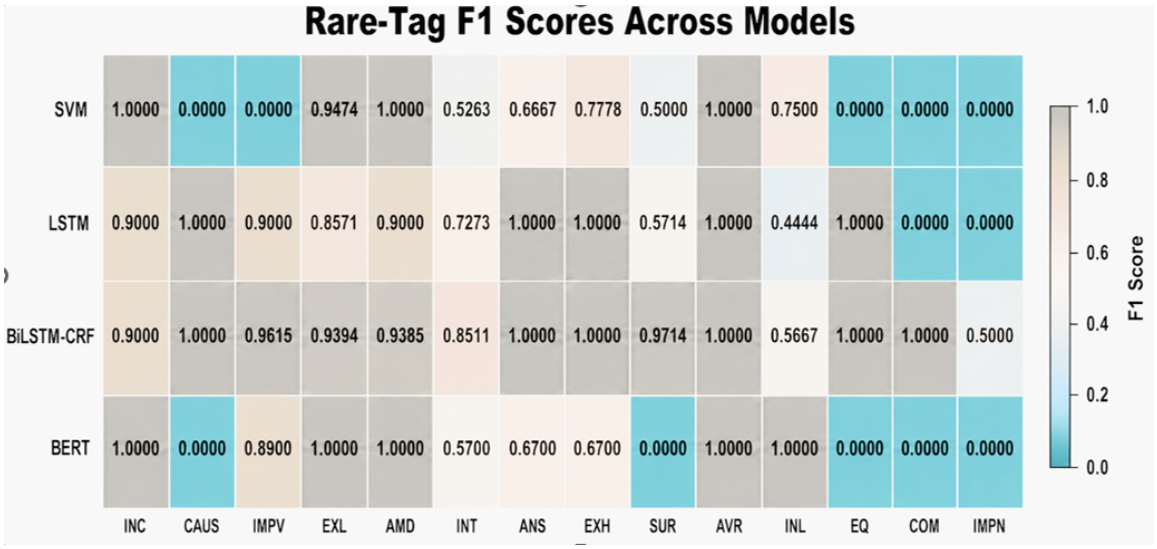


Figure 4.3: Rare-tag F1 scores across all models. Red-bordered cells indicate zero F1.

Several observations emerge from the rare-tag analysis. First, tags with extremely low support (EQ with 5, COM with 3, IMPN with 2) receive zero F1 from all models, indicating that the data is simply insufficient for any approach to learn these categories. Second, the BiLSTM-CRF model generally performs best on rare tags, achieving perfect F1 scores on AVR, ANS, SUR, and INL, and strong scores on IMPV, AMD, and RET. Third, the SVM model with sequence features performs surprisingly well on rare tags, matching BiLSTM-CRF on several categories. Fourth, the BERT

model struggles more noticeably on rare tags, with zero F1 on SUR and CAUS and lower scores on INT and ANS, which explains its weaker macro F1 overall.

4.10 Summary of Findings

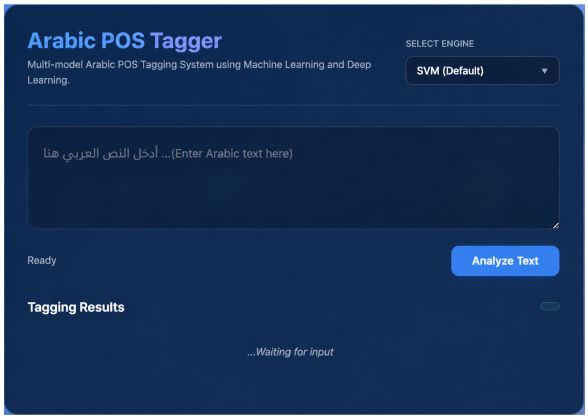
The experimental results presented in this chapter lead to several important findings. First, the BiLSTM-CRF model with embedding dimension 150 is the best-performing model overall, achieving the highest accuracy (0.9585), macro F1 (0.8090), and weighted F1 (0.9577). Second, the SVM model with sequence features achieves a competitive macro F1 of 0.8071, demonstrating that classical machine learning with well-designed features can approach the performance of neural models on this task. Third, the BERT model, despite its sophistication, does not outperform BiLSTM-CRF on this dataset, particularly on rare tags. Fourth, the addition of context features improves SVM performance substantially, while the choice of embedding dimension significantly affects both LSTM and BiLSTM-CRF performance.

These findings confirm that for the Quranic Arabic POS tagging task, the BiLSTM-CRF architecture offers the best trade-off between overall accuracy, rare-tag coverage, and computational efficiency. The results also highlight the importance of careful feature design and representation selection, which can be as impactful as the choice of model architecture.

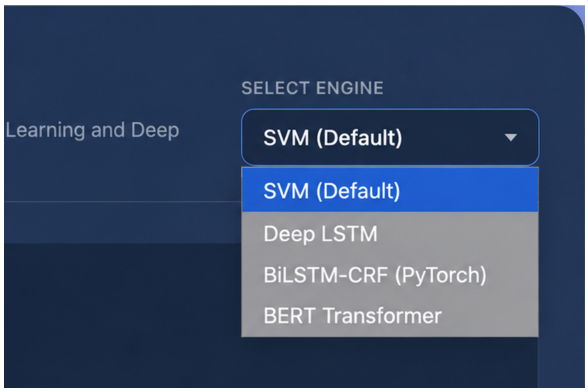
4.11 Conclusion

This chapter presented a comprehensive experimental comparison of four model families applied to Arabic POS tagging using the Quranic Arabic Corpus. The SVM, LSTM, BiLSTM-CRF, and BERT models were evaluated under consistent conditions, and their results were analyzed in terms of overall accuracy, macro F1, weighted F1, and rare-tag performance. The BiLSTM-CRF model with embedding dimension 150 emerged as the best-performing configuration, achieving the highest scores across all evaluation

metrics. The next chapter discusses the broader implications of these findings, examines the limitations of the study, and proposes directions for future research.



(a) Screenshot 1



(b) Screenshot 2

(c) Screenshot 3

(d) Screenshot 4

Figure 4.4: Web application screenshots

Chapter 5

Discussion and Conclusion

5.1 Introduction

The previous chapter presented the experimental results of the SVM, LSTM, BiLSTM-CRF, and BERT-based models for Arabic part-of-speech tagging using the Quranic Arabic Corpus. This chapter discusses the broader implications of those findings, interprets the observed patterns, and reflects on the contributions of the dissertation. The chapter also acknowledges the limitations of the study and proposes directions for future research. In this way, the chapter closes the dissertation by connecting the empirical results to the original objectives and to the wider context of Arabic NLP research.

5.2 Summary of the Study

This dissertation set out to study text annotation through the task of Arabic part-of-speech tagging and to compare machine learning and deep learning approaches within a unified experimental framework. The study was motivated by the observation that Arabic presents unique challenges for annotation due to its morphological richness, contextual ambiguity, clitic attachment, and dialectal diversity, and that different computational

paradigms may respond to these challenges in different ways.

The dissertation was organized into five chapters. Chapter 1 introduced the theoretical background of text annotation and POS tagging, including the formulation of tagging as a sequence labeling problem, the specific challenges of Arabic, and the evolution of annotation approaches from manual and rule-based methods to machine learning and deep learning. Chapter 2 presented the Arabic language, the Quranic Arabic Corpus, and the data preparation pipeline, including tokenization, segmentation, label extraction, normalization, and sequence construction. Chapter 3 described the methodological framework, including the SVM approach with handcrafted features and the deep learning approach with Word2Vec embeddings, LSTM, BiLSTM-CRF, and BERT architectures.Chapter 4 reported the experimental results and comparative analysis across all models. Chapter 5 discusses the findings, presents the contributions, acknowledges the limitations of the study, and proposes directions for future work. The experimental results showed that the BiLSTM-CRF model with embedding dimension 150 achieved the best overall performance, with a test accuracy of 0.9585, a macro F1 of 0.8090, and a weighted F1 of 0.9577. The SVM model with sequence features achieved a competitive macro F1 of 0.8071, while the BERT model, despite its architectural sophistication, achieved a macro F1 of 0.7496 that was notably lower than the best deep learning model.

5.3 Interpretation of Results

5.3.1 Why BiLSTM-CRF outperforms other models

The superior performance of BiLSTM-CRF can be attributed to several factors. First, the bidirectional LSTM component captures contextual information from both preceding and following tokens, which is essential for resolving ambiguity in Arabic POS tagging. Second, the CRF output layer models dependencies between adjacent tags, ensuring that the predicted tag sequences are globally coherent rather than consisting of independent

per-token decisions. This combination of contextual representation and structured prediction is especially valuable for Arabic, where the grammatical category of a word often depends on its position and on the categories of neighboring words.

Another factor is the size and nature of the dataset. The Quranic Arabic Corpus, while linguistically rich, is relatively small compared to the large corpora typically used for training transformer models. The BiLSTM-CRF architecture is well suited to this setting because it can learn effective representations from moderate amounts of data without requiring the massive pretraining that makes BERT successful on large-scale tasks. The compact embedding dimension of 150 further helps by avoiding overfitting and maintaining generalization on infrequent tags.

5.3.2 Why SVM performs competitively on macro F1

The SVM model with sequence features achieves a macro F1 of 0.8071, which is only 0.0019 points below the BiLSTM-CRF macro F1. This result demonstrates that careful feature engineering, combined with contextual information, can approach the performance of neural sequence models on Arabic POS tagging. The handcrafted features used in the SVM approach capture exactly the morphological and lexical information that is most relevant for grammatical classification, and the addition of previous tag information provides the contextual signal needed to resolve ambiguity.

This finding is consistent with the literature on Arabic POS tagging, which has shown that classical machine learning methods remain effective when supported by well-designed features [3, 5]. It also suggests that for domain-specific or resource-constrained settings, SVM-based tagging may offer a practical alternative to deep learning, particularly when interpretability and computational efficiency are important considerations.

5.3.3 Why BERT underperforms on macro F1

The BERT model achieves a test accuracy of 0.9541 and a weighted F1 of 0.9522, which are competitive with the BiLSTM-CRF model. However, its

macro F1 of 0.7496 is substantially lower. This discrepancy is explained by the BERT model's weaker performance on rare tags. An examination of the per-tag classification report shows that the BERT model assigns zero F1 to several rare categories, including CAUS, EXP, and SUR, which the BiLSTM-CRF model handles more effectively.

There are several possible explanations for this result. First, the BERT model was fine-tuned on a relatively small dataset, and its large number of parameters may lead to overfitting on frequent tags while neglecting rare ones. Second, the WordPiece tokenizer used by BERT may segment Arabic words into subword units in ways that are not optimally aligned with the morphological structure of the language. Third, the multilingual nature of the DistilBERT-mBERT model means that the pretraining data includes many languages, and the representations may not be optimally tuned for the specific linguistic properties of Quranic Arabic. These factors together explain why the BERT model, despite its strong overall accuracy, does not outperform the BiLSTM-CRF model on the macro F1 metric.

5.3.4 The role of embedding dimension

The experimental results show that embedding dimension significantly affects model performance, but not in a monotonic way. For the LSTM model, dimension 525 gives the best macro F1, while for the BiLSTM-CRF model, dimension 150 gives the best results. In both cases, dimension 1000 yields the worst macro F1, suggesting that excessively large embeddings introduce noise that harms performance, particularly on rare tags.

This finding is important because it shows that larger embeddings do not automatically lead to better performance in POS tagging. The optimal dimension depends on the size of the dataset, the complexity of the model architecture, and the nature of the linguistic task. For the Quranic Arabic Corpus, which contains approximately 77,000 tokens, compact embeddings in the range of 150 to 525 dimensions provide the best balance between representational capacity and generalization.

5.4 Contributions of the Dissertation

This dissertation makes several contributions to the study of Arabic text annotation and POS tagging. First, it provides a systematic comparison of four model families under consistent experimental conditions, using the same corpus, preprocessing pipeline, and evaluation metrics. This makes the results directly comparable and provides a clear picture of the relative strengths and weaknesses of each approach.

Second, the study demonstrates the importance of architecture selection for Arabic POS tagging. The finding that BiLSTM-CRF outperforms BERT on this task challenges the assumption that larger and more recent models always produce better results, and highlights the need to consider dataset size, domain specificity, and linguistic properties when choosing a model.

Third, the dissertation provides a detailed analysis of rare-tag performance across all models, which is important for understanding the practical limitations of POS tagging systems. The observation that several rare tags receive zero F1 from all models underscores the need for better data collection, data augmentation, or specialized handling of infrequent categories.

Fourth, the study shows that classical machine learning methods, when supported by well-designed features and context information, can achieve competitive performance on Arabic POS tagging. This finding is relevant for practical applications where computational resources, interpretability, or deployment constraints may favor simpler models.

5.5 Limitations of the Study

Several limitations of this study must be acknowledged. First, the experiments are based exclusively on the Quranic Arabic Corpus, which represents a specific linguistic domain. The results may not generalize directly to other varieties of Arabic, such as Modern Standard Arabic used in news and formal communication, or the numerous regional dialects used in everyday interaction. Models trained on Quranic Arabic may perform differently

when applied to contemporary or informal text.

Second, the study uses static Word2Vec embeddings for the LSTM and BiLSTM-CRF models. While these embeddings capture useful lexical and morphological regularities, they do not provide the contextualized representations that characterize more recent approaches such as BERT and other transformer models. The comparison between the static-embedding models and the BERT model is therefore not only a comparison of architectures but also of input representation types.

Third, the study does not explore ensemble methods, data augmentation techniques, or semi-supervised approaches that could potentially improve performance, particularly on rare tags. The experiments are limited to fully supervised learning with a fixed annotated corpus.

Fourth, the SVM sequence features rely on gold-standard previous tags, which may not be available in a fully automatic deployment scenario. The non-sequence SVM results provide a more realistic baseline for independent classification, but the comparison between the two SVM settings should be interpreted with this caveat in mind.

5.6 Directions for Future Research

Several directions for future research emerge from this study. First, the use of contextualized embeddings within the BiLSTM-CRF framework could combine the structured prediction strength of the CRF layer with the richer representations provided by contextual models. This hybrid approach may improve performance on ambiguous and rare tags.

Second, the application of large Arabic-specific language models such as AraBERT or CamelBERT could be explored for the same task. These models are pretrained on large Arabic corpora and may provide better representations for Arabic POS tagging than the multilingual DistilBERT model used in this study.

Third, the extension of the experimental setting to other Arabic varieties, including dialectal and mixed-register data, would test the generalizability of the findings and help identify which approaches are most robust

to linguistic variation. This is particularly important given the increasing volume of user-generated Arabic content on social media and other platforms.

Fourth, data augmentation techniques specifically designed for Arabic morphology could be explored to improve rare-tag performance. For example, synthetic examples could be generated by applying morphological transformations to existing tokens while preserving their POS labels.

Fifth, the integration of explicit morphological features into the deep learning models, either as additional input signals or through multi-task learning, could potentially improve performance by combining the strengths of handcrafted features and learned representations.

5.7 Conclusion

This dissertation has investigated the task of Arabic part-of-speech tagging as a case study for comparing machine learning and deep learning approaches to text annotation. Through a systematic experimental framework applied to the Quranic Arabic Corpus, the study evaluated SVM, LSTM, BiLSTM-CRF, and BERT-based models under consistent conditions.

The results showed that the BiLSTM-CRF model with embedding dimension 150 achieves the best overall performance, with a test accuracy of 0.9585 and a macro F1 of 0.8090. The SVM model with sequence features demonstrated that classical machine learning can approach neural model performance when supported by well-designed features. The BERT model, while achieving strong accuracy and weighted F1, underperformed on rare tags, resulting in a lower macro F1.

These findings confirm that text annotation remains a challenging task for Arabic, that different model families have different strengths and limitations, and that the choice of architecture must be guided by the properties of the data and the requirements of the application. The study also highlights the importance of evaluating models not only on their overall accuracy but also on their performance across the full range of linguistic

categories, including rare and difficult tags.

In a broader sense, this dissertation illustrates that the evolution from classical machine learning to deep learning and transformer-based models is not a simple linear progression, but rather a diversification of available tools, each with its own advantages and appropriate contexts. Understanding these trade-offs is essential for building effective, efficient, and linguistically informed annotation systems for Arabic and other morphologically rich languages.

Bibliography

[1] Jurafsky, D. and Martin, J. H. (2026). *Speech and Language Processing*. Draft, January 2026. https://web.stanford.edu/~jurafsky/slp3/

[2] Wang, P., Qian, Y., Soong, F. K., He, L., and Zhao, H. (2015). Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network. *arXiv preprint arXiv:1510.06168*. https://arxiv.org/abs/1510.06168

[3] Darwish, K., Mubarak, H., Abdelali, A., Eldesouki, M., Samih, Y., Alharbi, R., Attia, M., Magdy, W., and Kallmeyer, L. (2018). Multi-Dialect Arabic POS Tagging: A CRF Approach. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. https://www.research.ed.ac.uk/en/publications/multi-dialect-arabic-pos-tagging-a-crf-approach/

[4] Alharbi, R., Magdy, W., Darwish, K., and Mubarak, H. (2018). Part-of-Speech Tagging for Arabic Gulf Dialect Using Bi-LSTM. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. https://aclanthology.org/L18-1620/

[5] AlKhwiter, S. and Al-Twairesh, N. (2020). Part-of-speech tagging for Arabic tweets using CRF and Bi-LSTM. *Natural Language Engineering*. http://www.mobt3ath.com/uplode/book/book-99691.pdf?ver=accessable

[6] Darwish, K., Attia, M., Mubarak, H., Samih, Y., Abdelali, A., Marquez, L., Eldesouki, M., and Kallmeyer, L. (2020). Effective Multi-Dialectal Arabic POS Tagging. *Natural Language Engineering*, 26(6), 677–690.

[7] Mokh, N., Dakota, D., and Kubler, S. (2022). Improving POS Tagging for Arabic Dialects on Out-of-Domain Texts. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP 2022)*. https://aclanthology.org/2022.wanlp-1.22.pdf

[8] Cheragui, M. A., Dahou, A. H., and Abdedaiem, A. (2023). Exploring BERT Models for Part-of-Speech Tagging in the Algerian Dialect: A Comprehensive Study. In *Proceedings of the 6th International Conference on Natural Language and Speech Processing*. https://aclanthology.org/2023.icnlsp-1.14.pdf

[9] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. https://homepages.inf.ed.ac.uk/hshimoda/svm/index.html

[10] Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*. https://www.cs.cornell.edu/courses/cs6784/2010sp/lecture/10-LaffertyEtAl01.pdf

[11] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. https://deeplearning.cs.cmu.edu/S23/document/readings/LSTM.pdf

[12] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR 2013)*.

[13] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost)

from scratch. *Journal of Machine Learning Research*, 12, 2493–2537. https://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf

[14] Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991.*

[15] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS 2017).*

[16] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019).* https://arxiv.org/abs/1810.04805

[17] Quranic Arabic Corpus. (2026). *Data Download*. Available at: https://corpus.quran.com/download/ Accessed: 2026-05-03.

Appendix A

Detailed Tagset

Tag	Category	Abbreviation	Description
N	Noun	Noun	Common noun
V	Verb	Verb	Verb (perfect, imperfect, imperative)
P	Preposition	Preposition	Preposition
PN	Proper noun	ProperNoun	Proper noun
ADJ	Adjective	Adjective	Adjective
PRON	Pronoun	Pronoun	Personal pronoun
DEM	Demonstrative	Demonstrative	Demonstrative pronoun
REL	Relative	Relative	Relative pronoun
T	Time adverb	TimeAdverb	Time adverb
LOC	Location	LocationAdverb	Location adverb
NEG	Negative	Negative	Negative particle
CONJ	Conjunction	Conjunction	Coordinating conjunction
SUB	Subordinator	Subordinator	Subordinating particle
ACC	Accusative	Accusative	Accusative particle
VOC	Vocative	Vocative	Vocative particle
INT	Interrogative	Interrogative	Interrogative particle

CAUS	Causative	Causative	Causative particle
EXP	Exception	Exception	Exception particle
SUR	Surprise	Surprise	Surprise particle
RET	Retraction	Retraction	Retraction particle
INC	Inceptive	Inceptive	Inceptive particle
IMPV	Imperative	Imperative	Imperative verb
EXL	Exclamation	Exclamation	Exclamation particle
AMD	Amendment	Amendment	Amendment particle
ANS	Answer	Answer	Answer particle
AVR	Adverbial	Adverbial	Adverbial qualifier
INL	Initial	Initial	Initial conjunction
EMPH	Emphatic	Emphatic	Emphatic particle
RES	Restriction	Restriction	Restriction particle
VOC	Vocative	Vocative	Vocative particle

Appendix B

SVM Feature Sets

B.1 Non-Morphological Features

The following features were extracted for each token in the non-morphological SVM variant:

- **Word identity**: the raw token form
- **Word length**: number of characters in the token
- **Prefix n-grams**: first 1, 2, 3 characters
- **Suffix n-grams**: last 1, 2, 3 characters
- **Digit presence**: binary indicator of numeric characters
- **Capitalization**: binary indicator of capital letters (for transliterated forms)

B.2 Morphological Features

The morphological variant extends the non-morphological set with:

- **Proclitic**: attached particle at word beginning
- **Prefix**: morphological prefix

- **Stem**: lexical root or stem
- **Suffix**: morphological suffix
- **Enclitic**: attached pronoun at word end
- **Tag of preceding token** (sequence context)

Appendix C

Model Hyperparameters

Model	Parameter	Value
SVM	Kernel	Linear
SVM	C	0.5 (sequence), 5.0 (non-sequence)
SVM	Loss	Squared hinge
SVM	Max iterations	2000
LSTM	Embedding dim	50, 150, 525, 1000
LSTM	Hidden units	128
LSTM	Dropout	0.2
LSTM	Optimizer	Adam
LSTM	Learning rate	0.001
LSTM	Batch size	32
LSTM	Max epochs	50 (early stopping patience: 3)
BiLSTM-CRF	Embedding dim	50, 150, 525, 1000
BiLSTM-CRF	Hidden units	128 per direction
BiLSTM-CRF	Dropout	0.2

BiLSTM-CRF	Optimizer	Adam
BiLSTM-CRF	Learning rate	0.001
BiLSTM-CRF	Batch size	32
BiLSTM-CRF	Max epochs	50 (early stopping patience: 3)
BERT	Model	DistilBERT-multilingual-cased
BERT	Learning rate	2e-5
BERT	Batch size	16
BERT	Optimizer	AdamW
BERT	Max epochs	15 (early stopping patience: 3)

ملخص

« وسم النصوص: مقاربات التعلّم الآلي والتعلّم العميق» هي أطروحة تدرس دور الوسم اللغوي في معالجة اللغة الطبيعية من خلال وسم أجزاء الكلام في العربية بوصفه مهمةً تمثيليةً من مهام وسم السلاسل. وتقارن الدراسة أربع مقاربات حاسوبية: آلات المتجهات الداعمة المعتمدة على سمات مصممة يدويًا، وشبكات الذاكرة طويلة قصيرة المدى، ونماذج BiLSTMثنائية الاتجاه المدمجة معConditional Random Fields ، والنماذج المعتمدة على المحوّلات. وقد أُجريت التجارب علىQuranic Arabic Corpus ، وهو مورد غني بالوسوم يوفّر معلومات صرفية ونحوية صريحة .وتُظهر النتائج أن بنية BiLSTM-CRF حققت أفضل أداء إجمالي، إذ بلغت دقة الاختبار 0.9585 ودرجة Macro F1 مقدارها 0.8090، في حين ظل نموذج SVM المعتمد على السمات السياقية منافسًا بتحقيقه درجة Macro F1 بلغت 0.8071. كما يكشف تحليل الوسوم النادرة عن تحديات مستمرة عبر جميع النماذج، حيث حصلت عدة فئات نحوية قليلة التكرار على درجة F1 تساوي صفرًا. وتشير هذه النتائج إلى أن البنى العصبية ذات قدرات التنبؤ البنيوي تُعدّ مناسبة بشكل خاص لمهمة وسم أجزاء الكلام في العربية، مع إبراز أهمية حجم البيانات، والتعقيد الصرفي، والتوزيع طويل الذيل للفئات اللغوية عند اختيار النموذج.

Abstract

«Text Annotation: Machine Learning and Deep Learning Approaches» studies Arabic part-of-speech tagging as a sequence-labeling task. It compares Support Vector Machines, Long Short-Term Memory networks, BiLSTM-CRF, and transformer-based models on the Quranic Arabic Corpus. The results show that BiLSTM-CRF performs best, with a test accuracy of 0.9585 and a macro F1-score of 0.8090, while the SVM model remains competitive with a macro F1-score of 0.8071. The study shows that structured neural models are well suited to Arabic POS tagging. \keywords{Natural Language Processing, Linguistic Annotation, Arabic POS Tagging, Sequence Labeling, BiLSTM-CRF, Transformer Models, Quranic Arabic Corpus, Morphological Analysis}

Résumé

«Text Annotation: Machine Learning and Deep Learning Approaches» étudie l’étiquetage morpho-syntaxique de l’arabe comme tâche d’étiquetage de séquences. L’étude compare les machines à vecteurs de support, les réseaux Long Short-Term Memory, BiLSTM-CRF et les modèles fondés sur les transformeurs sur le Quranic Arabic Corpus. Les résultats montrent que BiLSTM-CRF obtient les meilleures performances, avec une précision de test de 0,9585 et un macro F1-score de 0,8090, tandis que le modèle SVM reste compétitif avec un macro F1-score de 0,8071. L’étude confirme l’efficacité des modèles neuronaux structurés pour l’étiquetage de l’arabe.