International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V73I9P126

Volume 73 Issue 9, 309-317, September 2025

Original Article

Toward Accurate Contextual Arabic Lemmatization Using a Deep Learning Approach

Driss Namly[1], Hakima Khamar[2], Karim Bouzoubaa[3], Fakhreldin Saeed[4]

[1]*Department of Computer Science, Mohammed V University in Rabat, Morocco.*
[2]*Faculty of Letters and Human Sciences, Mohammed V University in Rabat, Morocco.*
[3]*Mohammadia School of Engineers, Mohammed V University in Rabat, Morocco.*
[4]*Computing and Digital Design, University of Roehampton, SW15 5PJ London, U.K.*

[1]*Corresponding Author : d.namly@um5r.ac.ma*

Received: 17 June 2025 Revised: 19 September 2025 Accepted: 25 September 2025 Published: 30 September 2025

Abstract - *In the age of artificial intelligence, effective processing of unstructured textual data is critical, especially for languages with rich morphology such as Arabic. Lemmatization, the process of reducing words to their base or dictionary form, is important in various Natural Language Processing (NLP) applications. Arabic exhibits specific challenges due to its rich morphology, lexical ambiguity, and the absence of diacritics in most texts. Existing Arabic lemmatizers often struggle with context-aware disambiguation, rely heavily on proprietary datasets, or produce overwhelming morphological outputs unsuitable for non-experts. This study introduces SafarLemmatizer2, an advanced Arabic lemmatizer designed to address these limitations. Built upon the original SafarLemmatizer, the new tool integrates BiLSTM and BERT deep learning architectures to enhance contextual lemma selection while maintaining the accuracy of SafarLemmatizer's context-free lemmatization. The study determines the optimal architecture for contextual disambiguation through rigorous evaluation and provides a scalable lemmatization tool suitable for diverse NLP tasks. SafarLemmatizer2 thus represents a significant step forward in Arabic NLP, bridging the gap between traditional morphological analysis and modern deep learning-based approaches.*

Keywords - *Arabic NLP, Arabic contextual lemmatization, Deep learning, BERT, BiLSTM.*

1. Introduction

In the era of artificial intelligence, the necessity to swiftly and precisely retrieve pertinent information from extensive volumes of unstructured data has become paramount. This necessitates the development of sophisticated tools [1] for analyzing and understanding natural language. The current trend in NLP tools revolves around advanced language models and generative AI, which utilize deep learning architectures such as transformers trained on extensive datasets to produce text or answer questions. In their operation, a Large Language Model (LLM) implicitly incorporates morphological mechanisms during both training and generation to create grammatically consistent forms [2]. However, without explicit morphological knowledge, their ability to generalize to rare forms or languages with complex morphology is restricted. Specifically, from the Arabic morphology perspective, canonical units such as root, stem, and lemma play crucial roles in understanding the language's morphology [3]. The root provides the core meaning of a word, the stem offers grammatical context, and the lemma preserves both the grammatical category and the word's meaning. For example, the root "ك-ت-ب" (k-t-b) relates to writing, the stem "كاتِبانِ" (kAtibAni) indicates the dual masculine, and the lemma "كَتَبَ" (kataba) represents the base form of the verb "to write". This study focuses on Arabic lemmatization since it is used in several NLP applications, including text-to-speech systems [4], document clustering [5], text summarization [6], and machine translation [7], where it enhances the accuracy and efficiency of these systems. Lemmatization involves reducing inflected forms of words to their base or dictionary form, known as the lemma [8]. It represents the canonical form of the word without clitics. For nouns, the lemma is typically in the nominative, singular, masculine form, such as "طالِبٌ" (student) for "لِطُلَّابِهِم" (For their students). For verbs, the lemma is often the perfective, indicative, third person, masculine, singular form, such as "شَكَرَ" (to thank) for "وَتَشْكُرانِكَ" (And you thank him). Particles, being non-inflected, generally remain unchanged. Despite advancements in Arabic lemmatization technology [9-17], several critical challenges remain unresolved. Although existing lemmatizers often report high accuracy, these figures are typically derived from evaluations on proprietary datasets, limiting their generalizability. When tested on diverse corpora, the performance of these tools notably declines, revealing a gap between claimed and actual effectiveness. Furthermore, many lemmatizers suffer from lexical ambiguity due to the lack of

diacritics essential for Arabic word disambiguation. Another prevalent shortcoming is the tendency of some systems to generate context-independent lemmas by listing all possible variants for a given word, without leveraging the surrounding sentence context to select the most appropriate form.

Additionally, approaches relying heavily on morphological analysis tend to complicate the lemmatization task for non-expert users, as these tools produce extensive morphological features alongside lemmas, thereby increasing processing time and user effort. This comparison highlights the need for more robust, context-aware, and user-friendly Arabic lemmatization solutions beyond current state-of-the-art systems.

This study introduces SafarLemmatizer2, a novel Arabic lemmatization tool that significantly advances the contextual disambiguation capabilities of the original SafarLemmatizer. Unlike the earlier version, which relied on a traditional HMM-based model, SafarLemmatizer2 leverages state-of-the-art deep learning techniques [18] to enhance lemma selection within context. Specifically, this work investigates and benchmarks the effectiveness of BiLSTM and BERT-based architectures in accurately determining the correct lemma based on surrounding text, while simultaneously maintaining the highly reliable context-free lemmatization performance inherited from the original system.

This integration of modern neural models with proven baseline components represents a key innovation poised to improve both accuracy and usability in Arabic lemmatization tasks. The subsequent sections of the paper review related works in Section 2, while Section 3 explains the proposed approach. Section 4 presents the evaluation and compares the details and results of existing studies. Finally, Section 5 discusses the conclusions and outlines directions for future work.

2. State of the Art

The literature review demonstrates that lemmatization methods can broadly be categorized into out-of-context and in-context lemmatization. Both types have specific tools designed for different tasks, with varying accuracy, speed, and complexity levels.

2.1. Out-of-Context Lemmatization

Lemmatization tools that operate out of context identify the base form of a word without taking into account the context in which it appears. These tools frequently produce several possible lemmas for each word, which proves beneficial for managing diverse word forms. Among the available tools, the following are highlighted:

- AlKhalil2 Analyzer [9] includes an extensive collection of lexicons and morphological rules for examining Arabic words. It generates multiple potential analyses, each accompanied by a range of tags, including the lemma tag. This analyzer successfully processed 99.31% of the words from an extensive corpus comprising over 72 million diacritized words.
- Calimastar [10] is an Arabic morphological analyzer and generator that attempts to find the lemma of input text. It operates based on six lexicons, which include prefixes, suffixes, and stems, as well as three compatibility lexicons that encompass prefix-suffix, prefix-stem, and stem-suffix combinations. The authors demonstrate in their evaluation that CALIMAStar performed the highest with 90% accuracy in terms of lemma detection.
- CAMeL Tools [11] is a set of ANLP tools designed for various tasks, including pre-processing, morphological modeling, dialect identification, named entity recognition, and sentiment analysis. Its morphological analysis tool offers several features, including lemmas, part-of-speech tags, gender, number, and case distinctions. The authors' evaluation of the system reveals an accuracy rate of 95.4% in predicting the lemma. Nonetheless, the CAMeL Tools face some challenges, particularly regarding their processing speed, and the omission of diacritics in the input text leads to heightened ambiguity. For instance, if the tool is provided with the input "حَوْلَ" (Hawola), it produces the output "حَوْلَ، حَوْلٌ، حَوْلٍ، حَوْلُ..." (Hawola, Hawol, HawolN, Hawoli, HawolK, Hawolu, ...). Thus, even though the input contains vowels, the output includes solutions with vowels that differ from those in the input.
- Ibn_Ginni [12] is a hybrid analyzer combining the strengths of both BAMA and AlKhalil analyzers. However, it does not account for diacritics, leading to inaccuracies and being unsuitable for diacritic-sensitive contexts.

2.2. In-Context Lemmatization

In-context lemmatization tools, which consider the surrounding context to determine the correct lemma for a given word, generally achieve higher accuracy than out-of-context ones. A review of the literature indicates that the most widely used, well-known, and readily available tools are:

- MADAMIRA [13] is an Arabic morphological analyzer that combines rule-based and statistical approaches. It uses an n-gram language model to predict the correct lemma based on context. In their evaluation, the authors indicate that MADAMIRA provides the correct lemma for 96% of the words examined. Despite that, MADAMIRA is known for being slow due to its complex statistical models and n-gram language processing. This can be a limitation when processing large datasets.
- Farasa [14] is a rapid and effective tool that employs the Support Vector Machine (SVM) machine learning technique for the lemmatization of Arabic words. By leveraging an extensive corpus, it identifies the most

likely lemma, which contributes to its high efficiency. However, Farasa has a limitation regarding its processing of undiacritized text. Farasa achieved a segmentation accuracy of 98.94% during evaluation by its developers. Although it operates quickly, the absence of diacritic support can result in ambiguous outputs, potentially diminishing its reliability.

- ALP [15] is a lemmatization tool that integrates the results from both dictionary-based and machine-learning-based lemmatizers to produce a unified output addressing context-sensitive lemmatization. It is crafted to ensure both efficiency and accuracy across diverse types of Arabic texts. In the evaluation done by owners, the accuracy metrics calculated for the entire lemmatization pipeline indicated an accuracy of 98.4%. Nonetheless, the ALP faces significant challenges, particularly regarding its handling of rare or out-of-vocabulary terms. This issue is exacerbated by the dictionary and learning corpus containing only 29,397 lemmas and 20,407 named entities. Additionally, the system's inadequate support for diacritics results in ambiguous outputs.
- AlKhalil Lemmatizer [16] is a tool that utilizes the AlKhalil analyzer [9] and integrates contextual comprehension to enhance the precision of its lemmatization process. It employs a blend of syntactic rules and morphological analysis to generate potential lemmas, subsequently determining the appropriate lemma by considering the surrounding context through a Hidden Markov Model. The lemmatizer accurately provides the correct lemma for over 94.4% of the words in the test done by the authors.
- ALMA [17] is a lemmatizer designed to handle contextual information. Analyzing words' syntactic and semantic connections seeks to generate more precise lemmas. The underlying algorithm operates by identifying the most commonly occurring solution from a dictionary containing 298,000 lemmas. The evaluation of Alma by its developers on the Salma corpus achieved an F1 score of 90%.

2.3. Summary

Arabic lemmatization encompasses a variety of tools, each presenting distinct trade-offs regarding speed, accuracy, and complexity. The examined tools claim an accuracy rate surpassing 90% when utilizing their respective datasets.

All out-of-context lemmatizers primarily function as morphological analyzers, producing multiple tags for the input text, which is resource-intensive and time-consuming.

Furthermore, certain tools, particularly the in-context lemmatizers FARASA and ALP, yield ambiguous outputs without diacritics. This issue is exacerbated in out-of-context lemmatizers, which generate multiple undiacritized solutions that necessitate contextual interpretation.

3. Proposed Approach

In the realm of Arabic lemmatization, existing tools employ rule-based approaches [9], machine learning techniques [10-15], or hybrid methods [16, 18] that combine both rules and machine learning. While traditional machine learning models often require extensive feature engineering, the advent of deep learning techniques has transformed this paradigm by enabling models to automatically learn relevant representations from data. Specifically, deep learning techniques that are used to process and capture relationships within text sequences include Recurrent Neural Networks (RNNs) [19] and Transformers [20].

The Bidirectional Long Short-Term Memory (BiLSTM) [21] is an enhanced variant of RNNs that manages long-term dependencies within sequences. It is particularly noted for its ability to handle sequential data by capturing both past and future context, which is often critical for tasks such as contextual lemmatization.

Similarly, attention-based architectures, such as transformer models, especially Bidirectional Encoder Representations from Transformers (BERT), have proven to be more effective in text classification tasks [22]. BERT models are pre-trained on vast amounts of text to learn rich linguistic representations that enhance sentence comprehension. They can subsequently be fine-tuned for specific tasks like lemmatization.

This study presents SafarLemmatizer2, an enhanced version of the original SafarLemmatizer. The original system is dictionary-based and performs lemmatization in two main stages. The first context-free stage generates all possible lemmas using a clitics lexicon and a comprehensive stem/lemma lexicon, achieving an accuracy of over 99%.

In the second stage, contextual lemmatization is applied using the traditional Hidden Markov Model (HMM) machine learning approach to select the most suitable lemma based on the surrounding context, achieving an accuracy of about 95%. While effective, HMMs are limited in capturing long-distance dependencies and bidirectional context. To address these limitations, SafarLemmatizer2 retains the same context-free lemmatization approach but replaces the HMM with a deep learning-based disambiguation pipeline. As illustrated in Figure 1, the system evaluates two neural architectures-a BiLSTM model and a BERT-based model-using a preprocessed dataset to determine the most effective pipeline.

Table 1. Datasets statistics

	Nemlar	SALMA	10k Quran
Sentences	18,435	710	504
Tokens	500,000	34,253	10,000
Unique tokens	90,572	8,715	3,621
Unique lemmas	18,127	3,875	1,426

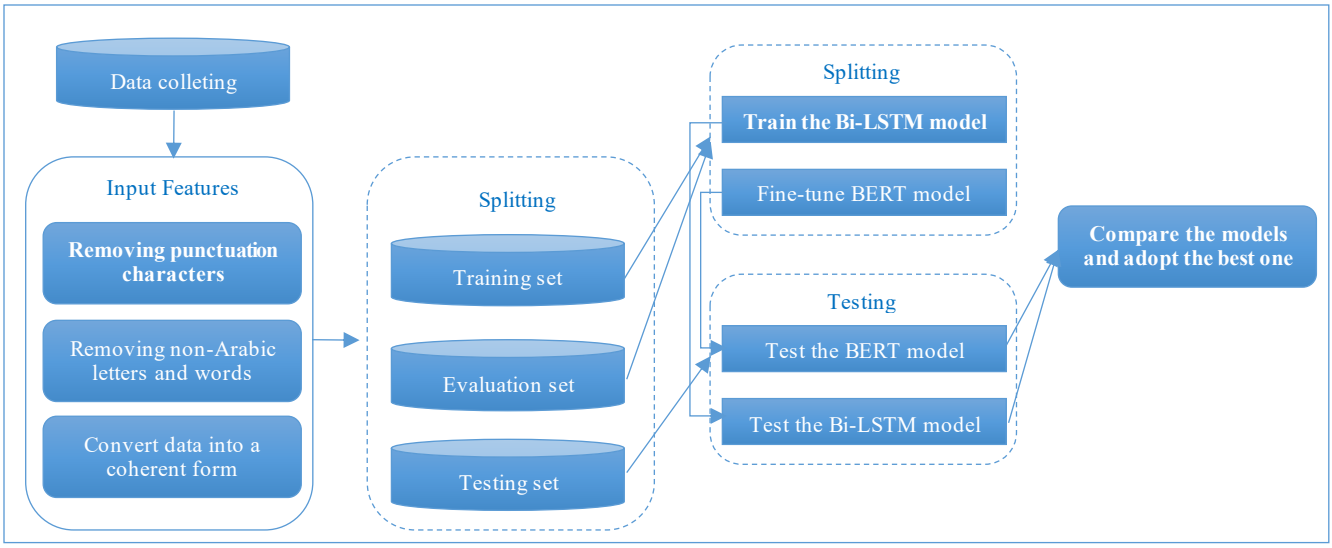

Fig. 1 Adopted architecture

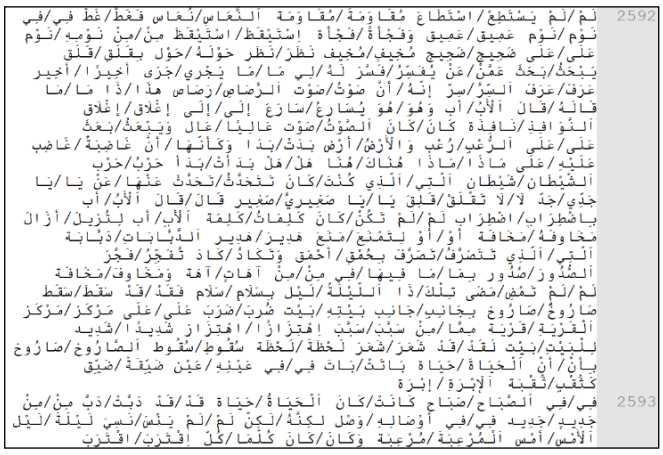

Fig. 2 Dataset extract

3.1. Used Datasets

To avoid facing the same challenges as other tools that depend on proprietary datasets, leading to advantageous results. However, alternative datasets tend to produce inferior outcomes; reliable datasets that accurately represent the Arabic language have been employed. These datasets are manually annotated and supplemented with additional linguistic information to support the development of the lemmatizer. The primary datasets utilized include the NEMLAR corpus, SALMA, and 10,000 tokens extracted from the Quranic text.

- The NEMLAR corpus [23] is a written resource for the Arabic language, created as part of the NEMLAR project [24]. Its design incorporates a sampling strategy to ensure a representative distribution across various text genres and domains. The corpus is annotated with lexical and morphological analyses (including the lemma tag) to facilitate a wide range of linguistic and natural language processing tasks. It comprises approximately 500,000 words of Standard Arabic text gathered from 13 distinct domains.
- The Salma dataset [25], collected from Modern Standard Arabic (MSA) media sources between 2021 and 2023, consists of tokenized and sense-annotated texts enriched with lemma information. It includes 34,253 tokens across more than 710 sentences, featuring 19,030 nouns, 2,763 verbs, and 12,460 particles, corresponding to 3,875 unique lemmas (2,904 nouns, 677 verbs, and 294 particles). Each token is sense-annotated using both the Modern and Ghani lexicons.
- The third dataset consists of 10,000 tokens extracted from the Quranic text, meticulously manually annotated with lemma tags to provide precise linguistic information. The corpus is organized as a sequence of sentences, where each sentence corresponds exactly to a single verse from the Quran. This structure preserves the original contextual and semantic boundaries.

Table 1 presents key statistics from the three selected datasets, while Figure 2 offers a detailed excerpt of the dataset used. Each dataset consists of multiple lines, with each line comprising a series of word/lemma pairs, forming the basic data structure. Before using the datasets, a pre-processing step is undertaken. The pre-processing of the dataset is critical to ensure the quality and consistency of the input data fed into the neural architecture pipeline.

Initially, all punctuation characters are removed to eliminate extraneous symbols that do not contribute to the linguistic analysis and could introduce noise. Subsequently, the dataset is filtered to exclude non-Arabic letters and words, focusing the model exclusively on Arabic textual content and preventing interference from foreign characters or irrelevant tokens. Finally, the cleaned data is transformed into a structured and coherent tabular format, which standardizes the

representation of each data instance with clearly defined fields (e.g., tokens, lemmas, morphological tags). This tabular organization facilitates efficient data handling, streamlined input processing, and compatibility with subsequent stages of the neural pipeline. In the splitting sub-process, the NEMLAR corpus serves as the training dataset for the Bi-LSTM model and is utilized to fine-tune the BERT model. It is divided into a training set and a validation set, with an allocation of 80% for training purposes and 20% for evaluation. The SALMA corpus and the ten thousand tokens dataset extracted from the Quranic text are added to the test set. The modeling process is detailed in the upcoming sections.

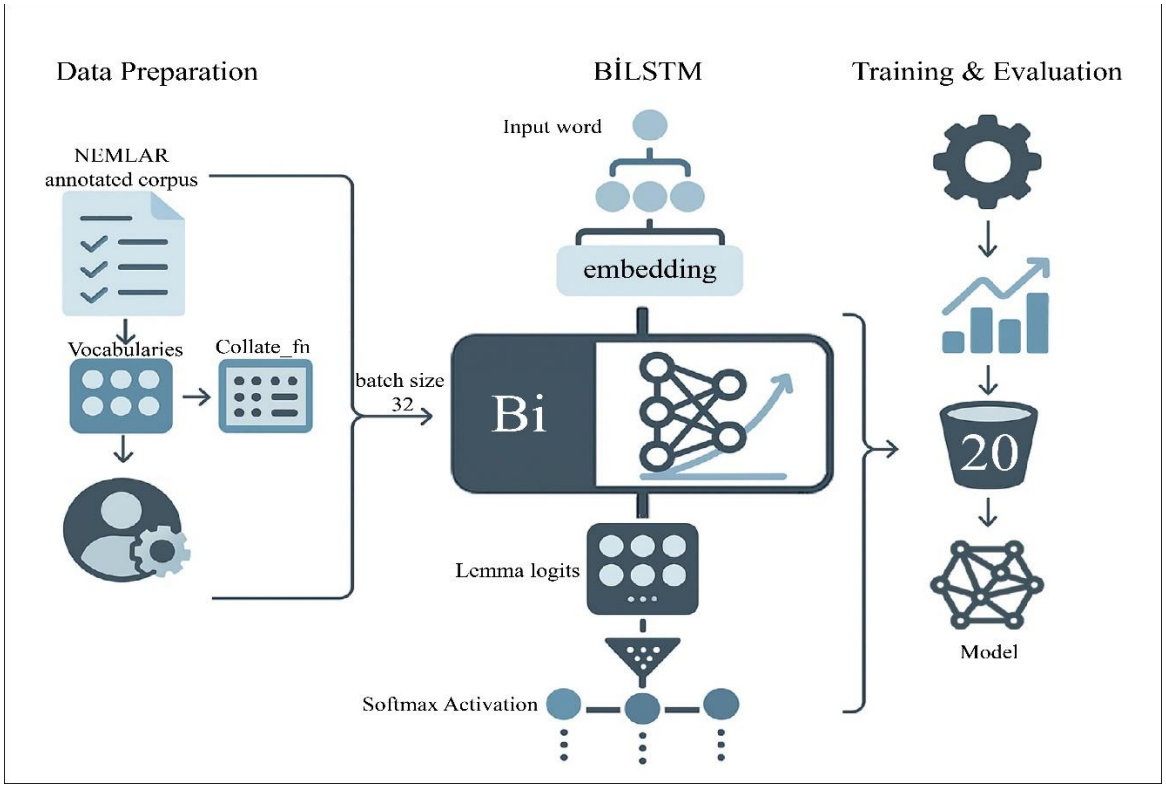

Fig. 3 BiLSTM pipeline

3.2. The BiLSTM Model

As illustrated in Figure 3, the BiLSTM pipeline begins with comprehensive data preparation, where raw sentences and their corresponding annotated versions are read from the NEMLAR corpus. This step includes building vocabularies for both words and lemmas and constructing a morphological lexicon that maps words to their possible lemmas. A custom dataset class is implemented to handle Arabic lemmatization, and a specialized collate function is used in the data loader to pad sequences of words, lemmas, and masks to uniform lengths, ensuring batch processing compatibility.

The core of the pipeline is a BiLSTM-based neural network designed for lemmatization. The architecture consists of an embedding layer that transforms input word indices into dense vector representations, followed by a BiLSTM layer that captures contextual information from both past and future tokens in the sequence. The BiLSTM pipeline leverages a softmax activation at the output layer, trains with a batch size of 32, with a learning rate around 0.001, and typically incorporates dropout and hidden layer sizes tuned to the dataset. The output of the LSTM is passed through a fully connected layer that projects to lemma logits, representing scores for each possible lemma using a softmax activation. A morphological filter is applied by masking out invalid lemma candidates, forcing the model to consider only linguistically plausible lemmas during prediction. Finally, the pipeline includes training and evaluation routines with a custom collate function. The model is trained using the Adam optimizer and cross-entropy loss, ignoring padded tokens to avoid skewing the loss calculation. Training proceeds for 20 epochs, with performance evaluated on a test set after each epoch. This setup allows monitoring of both training loss and test accuracy, ensuring the model learns to accurately predict lemmas from Arabic text tokens.

3.3. The BERT Model

The second model is a transformer model based on BERT, a language representation framework that utilizes transformer architecture to understand context from both directions, thereby improving performance in tasks such as lemma prediction based on sentence context. To accomplish

this, a pretrained BERT model is fine-tuned. During the fine-tuning process, the model parameters are adjusted by incorporating an additional layer into the foundational BERT architecture. This extra layer is specifically designed to train the model for the lemmatization task. The fine-tuning is conducted using the Hugging Face Transformers framework, which offers pretrained BERT models and tools for token classification tasks.

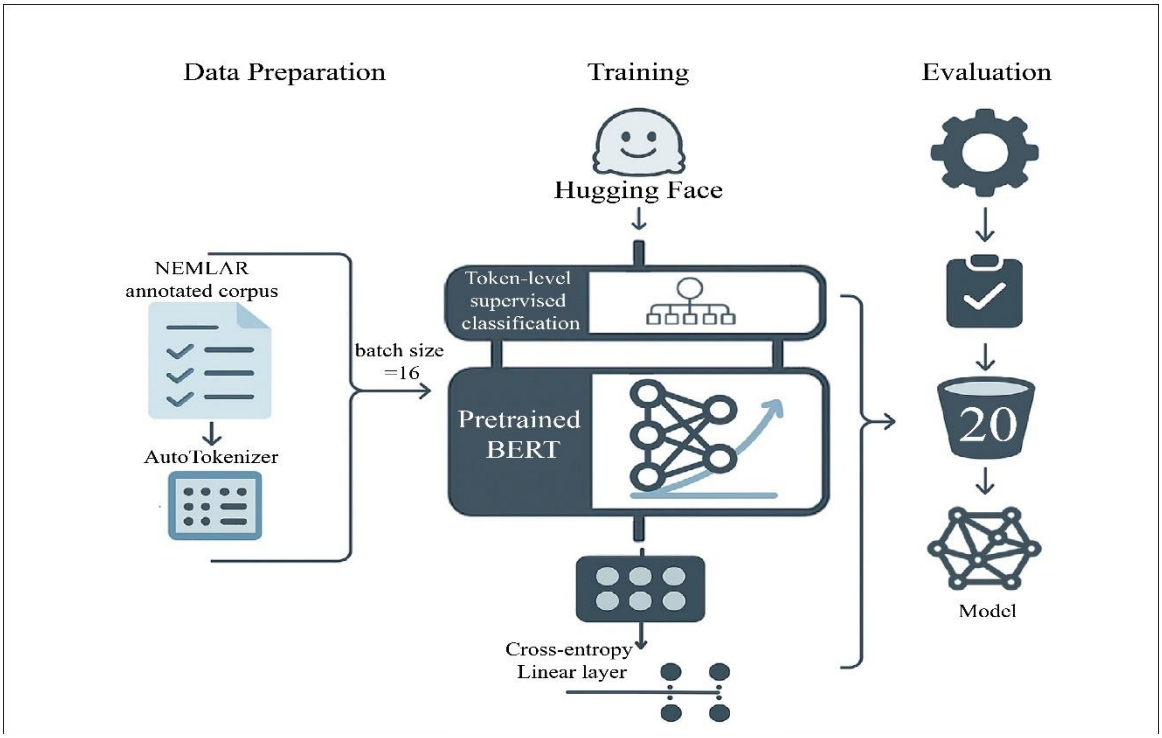

Fig. 4 BERT pipeline

As depicted in Figure 4, the initial step involves organizing the data for fine-tuning purposes. BertTokenizer is used to tokenize the NEMLAR corpus and prepare lemma labels corresponding to the tokens. Once the data preparation is complete, the pre-trained BERT model is fine-tuned. AraBERTv2 [26], which is trained on an extensive Arabic corpus and includes Arabic-specific tokenization and segmentation methods, is well-suited for lemmatization. Fine-tuning involves using AraBERTv2 as a feature extractor to generate contextualized embeddings for each token.

This process includes the addition of a token-level supervised classification head, with hyperparameters configured to epochs=5 and batch_size=16. The model employs a cross-entropy linear layer to translate BERT embeddings into the lemma classes. Following the training phase, the model's performance is evaluated using a test dataset to determine its capability in predicting the lemmas of previously unseen words.

4. Experiment

4.1. Experimental Setup

In recognition of the varied outputs generated by contemporary lemmatization tools - some producing non-diacritized lemmas, others offering partially or fully diacritized forms - a rigorous benchmarking framework comprised of three distinct lemmatization experiments was developed. The first experiment (Exp1) quantifies the exact-match accuracy between the tool-generated lemmas and the fully diacritized lemmas provided in the corpus.

This metric strictly requires an exact correspondence character-by-character, including diacritics, enabling a fine-grained assessment of tool precision. In the second experiment (Exp2), this constraint is relaxed by measuring accuracy based on overall lemma correspondence, allowing accepted matches when slight diacritic variations or the omission of case endings (e.g., final short vowels) occur.

This approach reflects real-world usage where certain morphological variations do not impact semantic interpretation and thus can be permissible. Finally, the third experiment (Exp3) entirely disregards diacritics by comparing the non-diacritized form of the output lemmas against the non-diacritized lemmas in the corpus. This experiment captures the underlying lexical accuracy independent of vocalization marks and thus is anticipated to yield the highest accuracy scores. To illustrate, consider the word "الكِتَابُ" (AlkitaAbu) in the corpus with the lemma "كِتَابُ" (kitaAbu). Under Exp1's exact-match criterion, a lemmatizer outputting "كِتَابُ" receives

full credit, whereas "كِتَّاب" (kitaAb) is counted as incorrect due to the missing final vowel diacritic. Exp2, however, accepts both "كِتَّاب" and "كِتاب" (with minor vowel differences) as correct since the dropped final vowel and vowel preceding the ت can vary without changing semantic meaning. In the most permissive Exp3, "كتاب" (ktAb), the non-diacritized lemma, is considered correct regardless of any vocalization variations. Empirical results confirm this gradient: aggregated accuracy scores averaged across evaluated tools were 74.3% (Exp1), 81.7% (Exp2), and 89.9% (Exp3), demonstrating the expected increase in matching rates with the relaxation of diacritic enforcement.

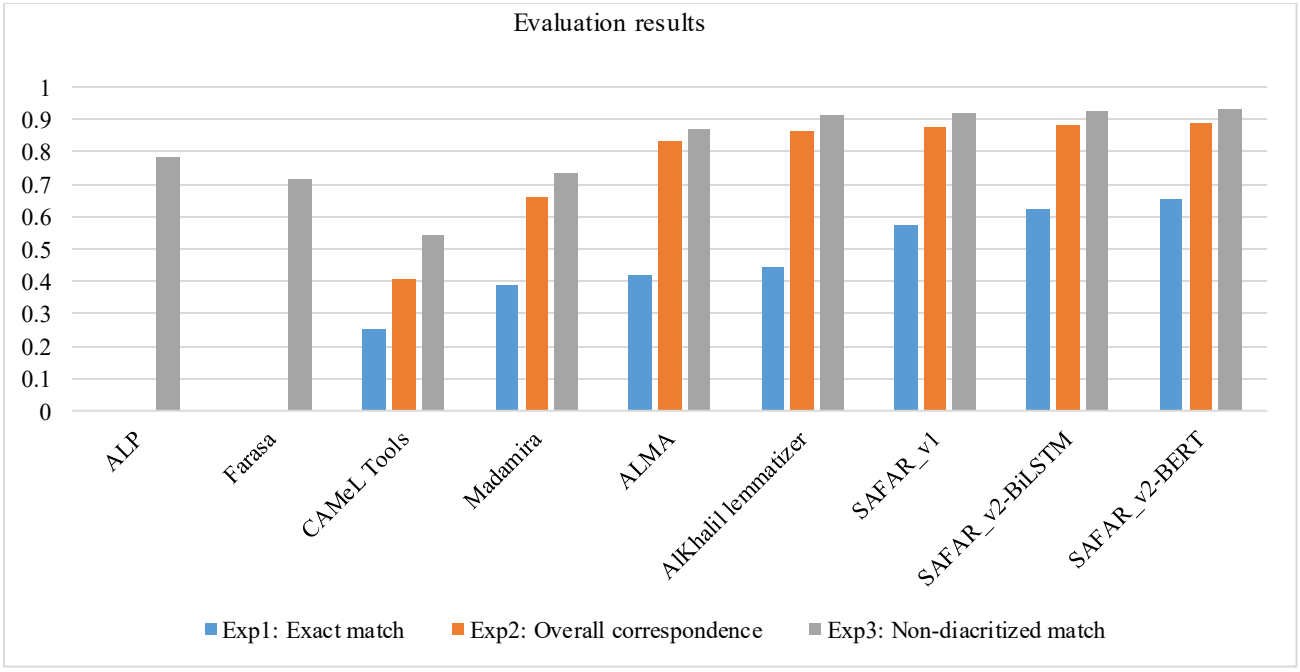

Fig. 5 Lemmatizers benchmark

Building on the state-of-the-art contextual lemmatizers reviewed, namely ALP, Farasa, CAMeL Tools, Madamira, ALMA, and AlKhalil, the benchmark aims to comprehensively compare their lemmatization precision versus the novel tools. For a robust and unbiased evaluation, a corpus amalgamating heterogeneous domain data is used to mitigate overfitting and corpus overlap biases. This included a 20% stratified subset of the NEMLAR dataset - previously adopted in evaluating SAFAR_v2-BERT and SAFAR_v2-BiLSTM models - augmented with an additional 10,000-token Quranic text dataset and the SALMA corpus. The combined evaluation set totals approximately 150,000 tokens spanning Modern Standard Arabic and Classical Arabic genres, thus ensuring diverse linguistic phenomena coverage.

4.2. Comparison Results

The benchmarking results presented in Figure 5 reveal significant variance in lemmatization accuracy among the evaluated tools across the three distinct experimental settings. These metrics progressively relax the stringency of output matching from full diacritized exactness (Exp1) to ignoring diacritics altogether (Exp3), allowing us to assess each tool's performance under increasingly lenient conditions. In Exp1, the highest precision is achieved by SAFAR_v2-BERT with an accuracy of 65.71%, followed by SAFAR_v2-BiLSTM (62.38%) and SAFAR_v1 (57.64%). In contrast, tools such as CAMeL Tools (25.46%), Madamira (39.15%), ALMA (41.92%), and AlKhalil lemmatizer (44.43%) show notably lower exact-match accuracies. This performance gap highlights the challenge these tools face in correctly predicting full diacritics and exact lemma forms, possibly due to limited contextual modeling or dependency on rule sets. Notably, ALP and Farasa did not report results for Exp1 and Exp2, likely reflecting the nature of their output formats, which might not emphasize diacritized lemmas.

When allowing some flexibility in matching (Exp2), the accuracy values rise significantly across tools capable of partial diacritization comparison. SAFAR_v2-BERT again leads with 88.96% accuracy, closely followed by SAFAR_v2-BiLSTM at 88.29% and SAFAR_v1 at 87.86%. This approximately 20-25% absolute improvement from Exp1 suggests that many errors in strict exact match stem from minor diacritic or vowel variations that do not severely impact linguistic correctness. AlKhalil lemmatizer (86.30%) and ALMA (83.12%) perform competitively, demonstrating their capacity to capture relevant lemma correspondences even if some diacritics are incorrect or omitted. Madamira's moderate score (65.95%) and CAMeL Tools's lower accuracy (40.71%) also support the conclusion that these tools likely focus less

on fine-grained vocalization details and more on core lexical forms. Exp3 represents the most lenient evaluation scenario, comparing only non-diacritized lemmas. This removes diacritic variation as a source of error, emphasizing lexical correctness alone. Expectedly, the highest accuracy scores are observed here, confirming improved and more robust lemma identification across all tools. The deep learning-based SAFAR models again outperform others, with SAFAR_v2-BERT reaching 93.10%, SAFAR_v2-BiLSTM 92.76%, and SAFAR_v1 91.92%. These figures highlight their superior generalization ability in capturing lemma stems across diverse Arabic texts regardless of vocalization.

Among the tools, AlKhalil lemmatizer achieves 91.26%, ALMA 86.74%, and Madamira 73.14%, indicating varying success in extracting correct lemmas when diacritics are ignored. ALP and Farasa produce overall correspondence accuracy scores of 78.22% and 71.38% respectively, which suggests their lemmatization outputs are mostly non-diacritized and perform moderately well on lexical correctness alone.

4.3. Discussion

As expected, the scores in Experiment 2 surpass those in Experiment 1, and similarly, Experiment 3 achieves higher scores than Experiment 2, owing to its more permissive evaluation criteria. Overall, the SAFAR tools, particularly the BERT-based and subsequently the BiLSTM-based versions, consistently outperform other lemmatizers across all experiments, demonstrating superior accuracy. In contrast, competing tools struggle, especially with the Quranic corpus, highlighting their limited ability to generalize across texts with diverse linguistic structures and vocabularies. The Classical Arabic style of the Quran, characterized by infrequent Modern Standard Arabic expressions and lengthier, more complex sentences, poses significant challenges for most tools. Notably, SAFAR_v2-BERT, SAFAR_v2-BiLSTM, and the CAMeL Tools (also BERT-based) show remarkable resilience to these challenges, likely due to their deep learning architectures' ability to generalize across varied linguistic domains and extensive training on diverse datasets.

That said, SafarLemmatizer2, soon to be accessible via the Safar web platform - an upcoming in-context lemmatizer built on BERT architectures - may face limitations in certain scenarios. For instance, it might struggle with highly ambiguous words lacking sufficient context, rare or out-of-vocabulary lemmas not well represented in its training data, or extremely domain-specific jargon and idiomatic expressions.

Additionally, since it leverages contextual embeddings, performance may degrade if the input text contains noisy data or errors, such as typos or unconventional orthography, which can impair contextual understanding. These limitations have broader implications for NLP applications reliant on accurate lemmatization. Tasks such as machine translation, information retrieval, or sentiment analysis may inherit errors if lemmatization is inconsistent or inaccurate, especially when processing domain-specific texts.

5. Conclusion

This paper introduced SafarLemmatizer2, an in-context lemmatizer developed using advanced deep learning techniques. Two neural architectures were explored: the first combines a BiLSTM model with a morphological filter that excludes invalid lemma candidates, ensuring only linguistically valid options are considered during prediction; the second involves fine-tuning a pretrained BERT model enhanced with a task-specific layer tailored for lemmatization. Multiple experiments evaluating seven existing lemmatizers alongside the proposed models demonstrated that the BERT-based architecture achieved the highest accuracy.

Looking ahead, future work aims to extend these approaches to cover various Arabic dialects, which pose additional challenges due to their lexically and morphologically diverse nature. Moreover, expanding training datasets to include more dialectal and domain-specific texts could significantly improve robustness and generalization. These future directions will guide the responsible advancement of SafarLemmatizer2, helping it achieve technical excellence in natural language processing applications.

References

[1] Xue Jiang et al., "Applications of Natural Language Processing and Large Language Models in Materials Discovery," *Nature Publishing Journal Computational Materials*, vol. 11, no. 1, pp. 1-15, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[2] Mete Ismayilzada et al., "Evaluating Morphological Compositional Generalization in Large Language Models," *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, Albuquerque, New Mexico, pp. 1270-1305, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[3] Khaled Shaalan et al., *Challenges in Arabic Natural Language Processing*, Computational Linguistics, Speech and Image Processing for the Arabic Language, pp. 59-83, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[4] Oumaima Zine, Abdelouafi Meziane, and Mohamed Boudchiche, "Towards a High-Quality Lemma-Based Text to Speech System for the Arabic Language," *International Conference on Arabic Language Processing*, pp. 53-66, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[5] Tahani Almutairi et al., "Preprocessing Techniques for Clustering Arabic Text: Challenges and Future Directions," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 8, pp. 1301-1314, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] Asma Bader Al-Saleh, and Mohamed El Bachir Menai, "Automatic Arabic Text Summarization: A Survey," *Artificial Intelligence Review*, vol. 45, no. 2, pp. 203-234, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[7] Mohamed Seghir Hadj Ameur, Farid Meziane, and Ahmed Guessoum, "Arabic Machine Translation: A Survey of the Latest Trends and Challenges," *Computer Science Review*, vol. 38, 2020.[CrossRef] [Google Scholar] [Publisher Link]

[8] Rawan Al-Matham et al., "KSAA-RD Shared Task: Arabic Reverse Dictionary," *Proceedings of ArabicNLP 2023*, Singapore, pp. 450-460, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Mohamed Boudchiche et al., "AlKhalil Morpho Sys 2: A Robust Arabic Morpho-Syntactic Analyzer," *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 2, pp. 141-146, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[10] Dima Taji et al., "An Arabic Morphological Analyzer and Generator with Copious Features," *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Brussels, Belgium, pp. 140-150, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[11] Ossama Obeid et al., "CAMeL Tools: An Open Source Python Toolkit for Arabic Natural Language Processing," *Proceedings of the Twelfth Language Resources and Evaluation Conference*, Marseille, France, pp. 7022-7032, 2020. [Google Scholar] [Publisher Link]

[12] Waleed Nazih et al., "Ibn-Ginni: An Improved Morphological Analyzer for Arabic," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 23, no. 2, pp. 1-22, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Arfath Pasha et al., "Madamira: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic," *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, pp. 1094-1101, 2014. [Google Scholar] [Publisher Link]

[14] Ahmed Abdelali et al., "Farasa: A Fast and Furious Segmenter for Arabic," *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, San Diego, California, pp. 11-16, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[15] Abed Alhakim Freihat et al., "Towards an Optimal Solution to Lemmatization in Arabic," *Procedia Computer Science*, vol. 142, pp. 132-140, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[16] Mohamed Boudchiche, and Azzeddine Mazroui, "A Hybrid Approach for Arabic Lemmatization," *International Journal of Speech Technology*, vol. 22, no. 3, pp. 563-573, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] Mustafa Jarrar, Diyam Akra, and Tymaa Hammouda, "ALMA: Fast Lemmatizer and POS Tagger for Arabic," *Procedia Computer Science*, vol. 244, pp. 378-387, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Amitha Mathew, P. Amudha, and S. Sivakumari, "Deep Learning Techniques: An Overview," *International Conference on Advanced Machine Learning Technologies and Applications*, Singapore, pp. 599-608, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Chandra Nidhi et al., "Utilizing Gated Recurrent Units to Retain Long Term Dependencies with Recurrent Neural Network in Text Classification," *Journal of Information Systems and Telecommunication*, vol. 9, no. 34, pp. 89-102, 2021. [Google Scholar] [Publisher Link]

[20] Kai Han et al., "Transformer in Transformer," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15908-15919, 2021. [Google Scholar] [Publisher Link]

[21] Gilberto Rivera et al., *Innovative Applications of Artificial Neural Networks to Data Analytics and Signal Processing*, Springer Nature, vol. 1171, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[22] A. Dennis Ananth et al., *Deep Learning & Applications*, Quill Tech Publications, pp. 1-220, 2024. [Google Scholar]

[23] Mohamed Boudchiche, and Azzeddine Mazroui, "Enrichment of the Nemlar Corpus by the Lemma Tag," *Workshop Language Resources of Arabic NLP: Construction, Standardization, Management and Exploitation*, Rabat, Morocco, 2015. [Google Scholar]

[24] Bente Maegaard, "The Nemlar Project on Arabic Language Resources," *Proceedings of the 9th EAMT Workshop: Broadening Horizons of Machine Translation and its Application*, Malta, pp. 124-128, 2004. [Google Scholar] [Publisher Link]

[25] Mustafa Jarrar et al., "Salma: Arabic Sense-Annotated Corpus and WSD Benchmarks," *arXiv preprint*, pp. 1-11, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[26] Wissam Antoun, Fady Baly, and Hazem Hajj, "AraBERT: Transformer-based Model for Arabic Language Understanding," *arXiv preprint*, pp. 1-7, 2020. [CrossRef] [Google Scholar] [Publisher Link]