**IAES International Journal of Artificial Intelligence (IJ-AI)**
Vol. 15, No. 2, April 2026, pp. 1876∼1890
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i2.pp1876-1890 

# A comparative study of Arabic morphological analyzers

**Omar Saadiyeh[1], Alaaeddine Ramadan[2], Chamseddine Zaki[3], Mohamad Hajjar[4], Gilles Bernard[1]**
[1]Paragraphe Research Lab, University of Paris VIII, Paris, France
[2]College of Engineering and Computing, American University of Bahrain, Riffa, Bahrain
[3]College of Engineering and Technology, American University of the Middle East, Egaila, Kuwait
[4]Faculty of Technology, Lebanese University, Saida, Lebanon

**Article Info**

***Article history:***

Received Jun 8, 2025
Revised Jan 9, 2026
Accepted Jan 25, 2026

***Keywords:***

Arabic dialects processing
Arabic linguistics
Arabic natural language processing
Language learning
Morphological analyzer

**ABSTRACT**

The field of Arabic natural language processing (NLP) has witnessed significant advancements, driven by the development of various morphological analyzers. This paper compares several major Arabic morphological analyzers and examines their ability to handle word ambiguities, process dialects, operate efficiently, and support downstream NLP tasks. By reviewing previous studies, we identify key gaps, including the limited resources for dialects, the shortage of annotated corpora, and challenges related to system scalability. The study also highlights future directions, such as building larger and more diverse corpora, adapting neural models for dialects, and developing analyzers that are more interpretable and trustworthy. Overall, this comparative overview aims to provide a clearer understanding of the current state of Arabic morphological analyzers, synthesize existing research, and offer practical recommendations for future work in this area.

*This is an open access article under the CC BY-SA license.*

***Corresponding Author:***

Alaaeddine Ramadan
College of Engineering and Computing, American University of Bahrain
Riffa, Bahrain
Email: alaaeddine.ramadan@aubh.edu.bh

## 1 INTRODUCTION

With more than 400 million speakers worldwide, Arabic is the official language in 22 countries. It is ranked as the fourth commonly used language on the internet [1]. Research conducted by several researchers [2], [3] has identified three variations within Arabic: i) classical Arabic (CA), known for its use in literary works and the Quran, ii) modern standard Arabic (MSA) is commonly used in formal contexts, and iii) dialectal Arabic (DA) utilized in informal conversations and everyday interactions [4]. DA further branches out into six groups, including Egyptian, Levantine, Gulf, Iraqi, Maghrebi, and other regional dialects [2], [5], [6]. Similar, to semitic languages Arabic features a morphological structure characterized by root letters, prefixes, suffixes and diverse grammatical patterns. Morphology involves studying how words are structured from units known as morphemes. Morphemes are the units of meaning, in a language. It's crucial to understand how they are arranged within words for language processing tasks like part of speech tagging parsing and machine translation [7]. In Arabic core words have inflected forms. For instance Arabic verbs boast 5400 forms compared to 6 in English as shown in Table 1.

Table 1. English verb paradigm

| VB | VBD | VBG | VBN | VBP | VBZ |
|---|---|---|---|---|---|
| go | went | going | gone | go | goes |

In grammar verbs can change forms to convey tenses and grammatical aspects. The base form is VB past tense is denoted by VBD, the gerund or present participle form is VBG, past participle form is VBN, non 3rd person singular form is VBP, and the 3rd person singular form is VBZ. Arabic verbs can take on forms based on gender (2), number (3), person (3), aspect (3), particle (2), mood (3), voice (2), pronominal clitic (12), and conjunction clitic (3) combinations as illustrated in Figure 1.

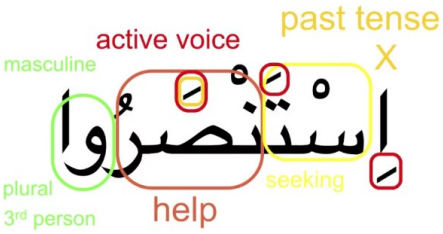


Figure 1. Arabic morphology example

Arabic follows a system based on roots, where words are typically created from a three letter root. This root system enables the formation of words, with meanings through different patterns and affixes, resulting in a diverse range of lexical forms. In Arabic, the process of inflection involves altering prefixes, suffixes and infixes to express functions like tense, mood, voice, number, gender, and case. For example the root "k t b" كتب can give rise to words such as "kataba" كتب (he wrote) "yaktubu" يكتب (he writes) "kitab" كتاب (book), and "maktab" مكتب (office). This morphological complexity allows for versatility and richness in language expression. Also introduces challenges in processing Arabic for natural language applications.

The rest of the document is structured as follows: section 2 provides an overview of advancements in syntactic and semantic analysis specifically tailored for Arabic. Section 3 explores approaches to Arabic morphological analysis, highlighting its key aspects and the available analyzers. Lastly section 4 offers an examination of existing techniques used in morphological analysis, within Arabic linguistics. In section 5 the challenges and future prospects of natural language processing (NLP) are explored. This article concludes with a summary of the findings.

## 2 SYNTACTIC AND SEMANTIC ANALYSIS

The two basic methods for comprehending natural language are syntactic and semantic analysis.

– Syntactic analysis (parsing) examines sentence structure according to grammatical rules. In Arabic, this is challenging due to rich morphology, flexible word order, and diacritics. Words often consist of roots, prefixes, suffixes, and infixes, making morphological analysis a prerequisite. Morphological ambiguity and variable word order notably affect parsing performance [8]. For example, the root ك-ت-ب (k-t-b) produces كاتب (kātib, writer), كتاب (kitāb, book), and مكتوب (maktūb, written). Although Arabic typically follows a verb-subject-object (VSO) order, as in أكل الرجل التفاحة (akal al-rajul al-tuffāḥah, the man ate the apple), it can also use subject-verb-object (SVO) and others, adding complexity. Diacritics, which mark short vowels, are often omitted, leading to ambiguity كتب (ktb) can mean كَتَبَ (kataba, he wrote), كُتِبَ (kutiba, it was written), or كُتُب (kutub, books). Accurate analysis relies on rules governing agreement, conjugation, and particle use.
– Semantic analysis focuses on meaning at word, phrase, and sentence levels. In Arabic, it is complicated by polysemy, synonymy, and context dependence. Word sense disambiguation (WSD) is vital; for example, عين (’ayn) may mean “eye,” “spring,” or “spy.” Named entity recognition (NER) identifies entities such as محمد (Muḥammad) or القاهرة (al-Qāhirah, Cairo). Semantic role labeling defines relationships, as in أعطى محمد الكتاب إلى علي (aʿṭā Muḥammad al-kitāb ilā ʿAlī), where محمد is the giver, الكتاب the object, and علي the recipient. Lexical semantics explores relations like synonyms (سعيد and مسرور), antonyms, and hierarchies. Contextual analysis resolves ambiguities, as in ذهب إلى المدرسة (dhahaba ilā al-madrasah), meaning “He went to school,” where the subject is implied. Ambiguity remains the main challenge in Arabic syntactic and semantic analysis, stemming from omitted diacritics and flexible word order. For instance, كتب الكتاب (kataba al-kitāb) can mean “he wrote the book” or “the book was

written." Dialectal variation further complicates processing; "house" is بيت (bayt) in MSA but دار (dār) or الحوش (al-ḥawsh) in dialects. The scarcity of annotated corpora and linguistic resources also limits progress. Despite these challenges, syntactic and semantic analysis are essential for advancing Arabic NLP tasks such as translation, information retrieval, and sentiment analysis.

## 3 ARABIC MORPHOLOGICAL ANALYSIS APPROACHES AND AVAILABLE ANALYZERS

### 3.1 Arabic morphological analysis approaches

This section explores various approaches to linguistic analysis based on lexicons, which systematically store linguistic rules. The lexicon comprises two main sections: the first contains the word roots, patterns, and stems, and the other displays related information in the analysis outcomes. The key approaches discussed are:

- Root-pattern morphology: focuses on the relationship between meaning and form, using nonconcatenative methods to derive stems from root–pattern combinations, as described by McCarthy. Prominent systems include the Buckwalter Arabic morphological analyzer (BAMA) and standard Arabic morphological analysis (SAMA) (Table 2).
- Stem-based morphology: expands beyond surface forms to provide linguistic and semantic data for each lexical item. It integrates root–pattern structures with syntactic information, offering a more intuitive framework for lexicon expansion.
- Lexeme-based morphology: recognizes that a single lexeme can produce multiple word forms, focusing on stem-level representations rather than individual root or pattern constituents.
- Syllable-based morphology: although effective in some European languages, syllable-based approaches remain largely unexplored in semitic languages like Arabic.

Table 2. Examples of root-pattern morphology

| Root | Pattern | In Arabic | Meaning |
|---|---|---|---|
| د ر س (drs) | فَعل (CaCaCa) | دَرَسَ (darasa) | study |
| | فَاعِل (CACiC) | دَارِس (dAris) | student |
| | فَعَلَ (CaC aCa) | دَرَسَ (dar asa) | he teaches |
| | فَاعِل (CACiC) | دَارِسُونَ (dAriswn) | group of students |

### 3.2 Available morphological analyzers

Standard Arabic language morphological analysis (SALMA) [9] was evaluated using the SALMA Gold Standard corpus, with a focus on the prediction accuracy of 22 morphological features at the morpheme level. The evaluation included two distinct Arabic text samples: the Qur'an [10] and the CCA [11]. Exact match accuracy reached 71.21% for the CCA corpus and 53.50% for the Qur'an, with many of the discrepancies being minor (e.g., symbol substitutions). The system showed particularly strong performance in 15 morphological categories, including part-of-speech (POS), verb and particle subcategories, definiteness, voice, and root-related features achieving accuracies of 98.53% for CCA and 90.11% for Qur'an. The remaining 7 categories, such as gender, number, and case, showed slightly lower accuracy, ranging from 81.35%–97.51% for CCA and 74.25%–89.03% for the Qur'an. These results demonstrate the SALMA tagger's effectiveness in delivering fine-grained morphological analysis across various Arabic text genres, leveraging traditional Arabic grammar rules within a knowledge-based framework.

In terms of methodology, the SALMA tagger is a rule-based, knowledge-driven analyzer, built on traditional Arabic grammar and the SALMA-ABCLexicon, a massive lexical resource compiled from 23 classical dictionaries (14M tokens; 2.7M vowelized pairs). Its modular design integrates tokenization, lemmatization, root extraction, vowelization, and pattern generation, allowing for highly detailed morpheme-level tagging across 22 features. Its main strength is the high accuracy in features like POS, verb type, and root-related categories, making it a strong choice for detailed corpus annotation. Its weaknesses appear in categories like gender, case, and number, particularly in classical Arabic, where the performance drops compared to MSA. Error analysis shows that many failures are minor (e.g., symbol substitution or misassigned diacritics), though some errors reflect the complexity of handling ambiguous morphosyntactic features. While per-feature accuracy is reported, statistical significance testing and confidence intervals are absent, leaving robustness across corpora less certain.

SAMA [12] follows a rule-based lexical approach rather than statistical or neural methods. It builds on the Buckwalter analyzer by expanding root and pattern coverage through an enriched lexicon and refined affixation rules. The system outputs all possible morphological parses for a given surface form, which provides wide coverage but leaves the task of contextual disambiguation to external modules. This design reflects both a strength comprehensiveness of analysis and a weakness, since in practice the raw outputs are often too ambiguous to use without further processing. The analyzer was primarily developed and distributed by the Linguistic Data Consortium (LDC), and while it does not train on a specific corpus in the way statistical models do, its lexicons are informed by extensive lexical resources curated over years of Arabic linguistic research. In terms of evaluation, SAMA is documented as a linguistic resource rather than a benchmarked system, so no formal evaluation numbers (e.g., accuracy for POS tagging, stemming, or lemmatization) are typically reported, and no confidence intervals or statistical significance testing are provided.

BAMA [13] is a rule-based, lexicon-driven tool for Arabic morphological analysis, designed for MSA by Tim Buckwalter. It uses an ASCII-based representation and includes modules for tokenization, transliteration, lexicon lookup, and morphological analysis, producing detailed output with features like person, number, gender, aspect, and voice. Initially implemented in Perl and later in Java, BAMA supports only Arabic and offers multiple analyses per token. It is widely used in linguistic research, NLP applications, and Arabic language technologies. Resources are curated internally, with no reported training corpus or evaluation against gold standards in the original release. Performance metrics appear only in later comparative studies, and no statistical significance testing is available for BAMA alone.

Farasa analyzer [14] is an advanced Arabic NLP tool developed by the Qatar Computing Research Institute (QCRI). It is grounded in a statistical learning approach, specifically an support vector machine (SVM)-rank classifier with linear kernels, which leverages a wide set of linguistic and probabilistic features such as prefix/suffix likelihoods, stem templates, and lexicon lookups. Unlike purely rule-based analyzers, it combines statistical ranking with curated lexicons, striking a balance between efficiency and accuracy. It provides comprehensive NLP capabilities via a RESTful Web API and is available as standalone Java jars. Farasa supports the Arabic language and includes components such as segmentation, spell checking, POS tagging, lemmatization, diacritization, dependency parsing, constituency parsing, and NER. The accuracy of Farasa (up to 98.94%) matches or slightly surpasses state-of-the-art systems. Error analysis reveals weaknesses in handling foreign named entities and overly long words with multiple valid segmentations. In these cases, the model often generates the correct segmentation but misranks it, suggesting room for improvement through richer gazetteers or feature expansion. The analyzer was trained on parts of the Penn Arabic Treebank (ATB) [15] and a large Aljazeera corpus (94M words, 2000–2011), and tested both on ATB subsets and an independent WikiNews set of 18,271 words. For downstream evaluation, Farasa was benchmarked on machine translation using IWSLT TED talks (183K sentences) and the NEWS corpus (202K sentences), and on information retrieval (IR) using the TREC 2001/2002 Arabic newswire collection (59.6M words, 75 topics).

AlKhalil analyzer has two versions: i) the first version, developed in 2010 [16], provides all possible vowelized forms for a given word. Each vowelized form is accompanied by detailed morphological information, including clitics, stem, root, and POS tag, and ii) the second version, developed in 2017 [17], it adopts a rule-based morpho-syntactic approach implemented in Java. It relies on an extensive, carefully structured lexicon of derived and non-derived words, clitic lists, and root–pattern files, enriched with lemmas and patterns. Its workflow includes normalization, segmentation into proclitics/stems/enclitics, and parallel analysis of stems as exceptional, non-derived, derived nouns, or verbs. Validation steps check compatibility between clitics, stems, and diacritics before producing the set of possible analyses. A major strength of this system is its broad lexical coverage (over 4.1M vowelized stems), high accuracy, and speed, which together make it robust and efficient for downstream tasks. However, like many out-of-context analyzers, it produces multiple candidate analyses for ambiguous words, which can overwhelm applications without a disambiguation module. For example, the non-vowelized form "علم" can yield outputs like عَلْم (science), عَلَم (flag), or عُلِم (was known), underscoring its reliance on external disambiguation for context-sensitive interpretation.

The system was evaluated on more than 72 million diacritized words from the Tashkeela corpus (63M) [18], Nemlar (0.5M), and RDI (8.5M). Results showed coverage of 99.31%, with an average of 4.71 lemmas, 5.08 stems, and 8.05 vowelized forms per word, reflecting its rich lexical resources. On Nemlar, it achieved 97.16% lemma match, 96.76% stem match, and 97.21% diacritization accuracy, with full-feature match at 96.56%. Its throughput reached 632 words/second, balancing speed with coverage. The authors do

not report statistical significance testing or confidence intervals.

Arabic Stanford Segmenter: the Arabic Stanford Segmenter [19] is a widely recognized tool for morphological segmentation and tokenization of Arabic text. Developed as part of the Stanford NLP Group's toolkit, it is based on a conditional random fields (CRF) model trained on annotated Arabic corpora. The tool is particularly effective in addressing the challenges of Arabic morphology, which include affixation, clitics, and the absence of clear word boundaries in written form. The Stanford Segmenter attempts to segment the clitics correctly using a statistical model that learns from linguistic patterns in annotated data, primarily drawing on the Penn Arabic Treebank (PATB) [15]. Unlike rule-based systems that may require extensive linguistic input and manual tuning, the Arabic Stanford Segmenter leverages machine learning techniques, which allow it to generalize well across different domains. It outputs both segmented tokens and their corresponding morphological analyses, making it a comprehensive preprocessing solution for modern Arabic NLP pipelines.

Reported results show strong performance with an F1 of 92.09% on Egyptian Arabic and statistically significant gains ($p < 0.001$) over prior baselines, plus a 7 decoding speedup compared to MADA and MADA-ARZ. Error analysis highlights three issues: i) inconsistencies in gold data, ii) overly local segmentation features, and iii) context-sensitive ambiguities (e.g., *wla* meaning "and not" or "or," and *-na* as pronoun vs. verb suffix). Strengths include dialect-agnostic design, tested improvements, and efficiency; weaknesses lie in handling context-sensitive segmentation and data inconsistencies.

MADAMIRA [20] is a morphological analyzer that assigns morphological tags to each word in a sentence by considering the word's context. It integrates two morphological analysis systems: MADA [21] and AMIRA [22]. Initially, the system analyzes the words of a sentence out of context using the SAMA analyzer [12]. To choose a single solution from the multiple options generated in this first phase, a disambiguation step based on the use of SVM and the language models is performed. It adopts a machine learning approach that relies on linear SVM classifiers and n-gram language models for morphological feature prediction, combined with ranking modules for disambiguation. Unlike its Perl-based predecessors, it is implemented in Java, which contributes to its robustness, portability, and remarkable efficiency, achieving speed improvements of up to 20.

The analyzer supports both MSA and Egyptian Arabic (EGY), using the PATB (parts 1–3) and Egyptian Arabic Treebanks (parts 1–6) as training data, respectively. The test sets included around 25K words for MSA and 20K for EGY. Evaluation shows high accuracy: for MSA, 95.9% POS accuracy, 96.0% lemma accuracy, and 86.3% diacritization; for EGY, 92.4% POS, 87.8% lemma, and 83.2% diacritization. Tokenization reached 98.9% perfect accuracy in MSA and 96.6% in EGY. MADAMIRA's strengths lie in its broad functionality (morphological disambiguation, diacritization, POS tagging, tokenization, glossing, and stemming), speed, and extensibility. It also allows flexible tokenization schemes and provides both XML and HTTP interfaces, making it user-friendly. Weaknesses include a slight drop in accuracy compared to MADA for some metrics (up to 0.6% lower in EGY full morphological accuracy) and heavy memory requirements (up to 2.5 GB heap space). Overall, evaluation results are reported with clear accuracy percentages but without statistical significance testing or confidence intervals, leaving robustness comparisons open for further analysis.

CAMEL MORPH MSA [23] is a comprehensive and publicly available morphological analyzer and generator for MSA. Featuring over 100,000 lemmas and support for rare morphological features inherited from classical Arabic, it significantly expands the analytical capabilities of Arabic NLP tools. The system generates approximately 1.45 billion analyses and 535 million distinct diacritizations. CAMEL MORPH MSA integrates seamlessly with the camel tools Python suite [24], ensuring ease of use. Evaluation across large datasets, including MSA-CB, CA-CB, and PATB-Train, shows robust accuracy and significantly improved coverage. In terms of strengths, CAMEL MORPH MSA dramatically improves lexical coverage and reduces out-of-vocabulary (OOV) rates by 36% compared to SAMA/CALIMA across massive corpora like MSA-CB (9.9B tokens, 11.4M types) and CA-CB (0.7B tokens, 2.4M types). Evaluation on PATB-Train showed a 95.9% recall, with manual inspection attributing about 90% of mismatches to annotation errors rather than the system itself, highlighting its reliability. Error analyses revealed challenges in handling spelling inconsistencies, lemma–stem mismatches, and ambiguous paradigms. Its main weakness lies in speed, running 2.4–2.9 times slower than SAMA, though offering richer analyses per word. Importantly, the results were reported with dataset-scale evaluations and manual error breakdowns, but without explicit statistical significance testing or confidence intervals.

Alma [25] is an open-source tool for Arabic language processing that integrates lemmatization, POS tagging, and root extraction. Its approach is primarily frequency-based and lexicon-driven, leveraging a large pre-computed memory built from the Qabas lexicographic database [26], the Shamela corpus, and digitized

lexicons. This design shifts computational complexity from runtime analysis to memory construction, enabling Alma to achieve very high processing speeds lemmatizing around 34,000 tokens per second. For OOV cases, Alma integrates a fine-tuned bidirectional encoder representations from transformers (BERT) model to improve POS tagging, which achieved F1-scores above 98% on the Arabic Treebank (ATB) for POS classification. Its coverage extends across 40 POS tags and includes the first fully functional root tagger grounded in Qabas.

Evaluation results highlight Alma's competitive performance: on the LDC Arabic Treebank (339k tokens) it reached 87.8% in true lemmatization and 92.7% in POS tagging, while on the SALMA corpus (34k tokens) it achieved 90.5% and 93.8% respectively. These scores were further improved when combined with BERT for OOV handling. Speed comparisons showed Alma vastly outperformed MADAMIRA (1710 seconds vs. 10 seconds on ATB). Error analysis revealed most failures were due to ambiguous lemmatization (61% of errors), where Alma favored the most frequent lemma even if contextually less accurate, and to general POS confusions, such as mistaking adjectives for nouns.

Ibn-Ginni is a hybrid Arabic morphological analyzer that combines the speed and precision of Buckwalter Arabic morphological analyzer (BAMA) with the broader classical Arabic coverage of the Alkhalil analyzer. To improve coverage, morphological data for 3 million unique Arabic words was generated using Alkhalil, refined, and added to BAMA's database. The resulting system analyzed 600,000 more words than BAMA alone, with an average analysis time of 0.3 milliseconds per word. In benchmark testing, Ibn-Ginni provided full morphological solutions for 72.72% of words and partial solutions for 24.24%, demonstrating improved performance and efficiency [27].

SinaTools [28] is an open-source toolkit developed at Birzeit University. It adopts a hybrid methodology that integrates rule-based resources with modern machine learning, particularly fine-tuned BERT models. Its morphological analysis module, Alma, relies on a frequency-based lexicon where lemmatization, POS tagging, and root tagging are handled through dictionary lookups, while a BERT-based model supports OOV handling. Other modules, including NER and word sense disambiguation (WSD), are also powered by transformer models such as AraBERT v2 [29]. This design not only ensures speed and accuracy but also provides flexibility through various integration interfaces, including CLI, API, and SDK. Its modularity and extensibility allow developers to plug in additional NLP tasks with minimal effort, which highlights its strength as a research and applied tool. However, the reliance on pre-computed lexicons limits its adaptability in unseen or domain-shifted contexts, as illustrated by consistent verb-tagging for ambiguous words regardless of context. The toolkit is trained and evaluated on several corpora.

Morphological evaluation was conducted on the Arabic TreeBank (ATB, 339k tokens) and the SALMA dataset (34k tokens), while NER was tested on the Wojood datasets [30], including WojoodGaza (50k tokens from news texts) and a Politics dataset (12k tokens). WSD was benchmarked using the SALMA sense-annotated corpus (34k tokens), and semantic relatedness was assessed through SemEval-2024 with 595 sentence pairs. In terms of performance, SinaTools achieved lemmatization accuracy of 90.5% and POS tagging at 97.5%. Its NER module reached an F1-score of 87.3%, the WSD module recorded 82.6% overall accuracy, and semantic relatedness scored 0.49 Spearman correlation. These evaluations, though impressive, underline that SinaTools' strength lies in high-speed lexicon-backed morphology with hybrid neural extensions.

Camelira [31] is a multi-DA morphological disambiguator that integrates statistical and neural approaches for analysis. Its backbone relies on CAMeL Tools' morphological disambiguation system. The tool covers four Arabic varieties: MSA, Egyptian, Gulf, and Levantine, and is accessible through a user-friendly web interface. Distinguishing itself from prior analyzers, Camelira not only outputs disambiguated readings in context but also presents alternative out-of-context analyses along with probability scores. A key strength is its integration of dialect identification, which automatically selects the appropriate disambiguator, making it valuable for learners or researchers who may not know the input dialect. However, its coverage is limited to specific dialects, and the system struggles with unseen genres or underrepresented varieties, producing occasional errors when processing texts outside its training distribution. Sample outputs in the interface demonstrate diacritized text, tokenized forms, lemmas, and full morphological features, but Gulf Arabic lacks diacritization due to unavailable annotated resources. In terms of resources, Camelira relies on the datasets used in the CAMeL Tools pipeline and the multi-Arabic dialect applications and resources (MADAR) shared task for dialect identification. Evaluation reported for morphological disambiguation, the model achieves accuracy across dialects as follows: MSA (95.9% for all tags, 98.7% POS), Egyptian (90.5%, 94.0%), Gulf (93.8%, 96.6%), and Levantine (85.5%, 92.7%).

According to Zalmout and Habash's [32], bidirectional long short-term memory (Bi-LSTM) morphological disambiguation system is a neural morphological disambiguation model for Arabic that combines Bi-LSTM architectures with morphological analyzers. Unlike earlier rule-based or statistical approaches, the system leverages word and character level embeddings enriched with subword and morphological features (such as affixes or dictionary-based tags). Its strength lies in using the outputs of a traditional morphological analyzer not as a replacement but as a guide, ranking possible analyses with learned probabilities. This hybrid design captures long-distance dependencies better than fixed-window methods and significantly boosts disambiguation for morphologically rich features like case and mood. Weaknesses remain in areas such as case assignment and rare categories (e.g., second-person verbs, passive voice), where ambiguity and data sparsity still limit performance.

The authors provide detailed error analysis, showing, for instance, that while their system doubles the cases where it outperforms MADAMIRA, some errors persist, especially for morphosyntactic cues heavily reliant on syntax. For evaluation, the authors use the PATB parts 1–3 as the main dataset (503K training words, 63K words each for development and test), complemented with pre-trained embeddings from the 2.15 billion-word Arabic gigaword corpus [33]. Results demonstrate full morphological analysis accuracy equal to 90.0%, and 76.9% for OOV words. Across specific features, POS tagging reached 97.9%, case tagging improved by 3.7 points, and diacritization accuracy was equal to 91.7%. These results are statistically significant across metrics, supported by comparative error analyses and confidence-based scoring. Overall, the system illustrates the enduring value of combining deep neural architectures with traditional analyzers, showing measurable improvements while highlighting remaining gaps in modeling fine-grained Arabic morphology.

Neural-based Arabic morphological analyzer [34] employs a neural-based approach, specifically a recurrent neural network (RNN), to perform Arabic morphological analysis. Unlike earlier rule-based systems, this model leverages sub-word information (prefixes, infixes, roots, and suffixes) and converts them into vectors for sequence modeling. The analyzer aims to overcome two main gaps in prior work, particularly in the Jabalin system: the inability to identify nouns and the heavy reliance on dictionaries for verb form classification. By combining pattern extraction, sub-word vectorization, and RNN-based classification, the system is able to automatically identify morphosyntactic descriptions (MSDs) for both verbs and nouns. This design highlights a strength in its ability to handle dictionary dependency problems and generalize to nouns derived from verbal roots, something previous analyzers struggled with.

However, one noted limitation is reduced accuracy for certain rare verb forms "Iii", where performance dropped to 73%, indicating challenges in modeling less frequent patterns. For its dataset, the system relies on the Qur'anic Arabic Corpus [10] which already includes morphological labels. with preprocessing, reducing the initial 1,778 unique words into an expanded dataset of over 30,936 labeled words using linguistic pattern tables. After splitting, 24,748 words were used for training and 6,188 for testing. The evaluation reported 99% overall accuracy, 99% precision, 96% recall, and 97% F1-score, with results broken down by POS, aspect, gender, number, and verb form. Statistical comparisons with the Jabalin system showed a marked improvement (99% vs. 39% overall accuracy), especially in noun recognition (99% vs. 0%). While the study did not report formal significance testing or confidence intervals, the detailed per-feature results (POS, tense, gender, number, and verb form) demonstrate robust evaluation across morphological categories.

Morphosyntactic tagging with pre-trained transformer models (CAMeLBERT) [35] adopts a neural approach, fine-tuning pre-trained transformer models (CAMeLBERT-MSA for MSA and CAMeLBERT-Mix for dialects). Each morphosyntactic feature is modeled with an independent classifier, and in some setups, predictions are refined using external morphological analyzers (SAMA for MSA, CALIMA for Egyptian, and automatically induced analyzers for Gulf and Levantine). This hybrid design shows clear strengths, it achieves state-of-the-art results across all varieties studied, with absolute improvements. Its weaknesses, however, stem from reliance on analyzer quality and dialectal orthographic inconsistency; while manually crafted analyzers improve tagging accuracy, automatically generated ones can sometimes hurt performance as data grows. Error analysis highlights difficulties with enclitics and nominal distinctions, particularly in dialects, with POS misclassifications and annotation inconsistencies contributing to common failures. The model was trained and tested on four corpora: PATB (629k tokens, MSA), Gumar (202k, Gulf), ARZTB (175k, Egyptian), and Curras (57k, Levantine). Evaluation includes POS tagging and full morphosyntactic feature prediction. For POS tagging, accuracy reached 98.9% (MSA), 96.9% (Egyptian), 97.9% (Gulf), and 94.6% (Levantine). For full morphosyntactic tagging (ALL TAGS), accuracy was 96.3% (MSA), 91.0% (Egyptian),

95.7% (Gulf), and 87.6% (Levantine). Results are statistically significant (McNemar's test, $p < 0.05$). The system leverages pre-trained transformers, external analyzers, and cross-dialectal transfer, while highlighting resource and annotation limitations.

Zalmout *et al.* [36] neural disambiguator (Egyptian Arabic) adopts a neural, Bi-LSTM–based approach for morphological tagging and disambiguation of Egyptian Arabic. It integrates word and character embeddings (tested with both convolutional neural network (CNN) and long short-term memory (LSTM) variants), alongside embedding space mapping to handle noisy, user-generated dialectal text. Noise normalization is applied at the vector level, avoiding raw text alterations. The system leverages morphological analyzers derived from SAMA, CALIMA, and ADAM resources to generate candidate analyses, which are then resolved using neural models. A large in-house Egyptian Arabic corpus (410M words) was used for pre-training embeddings, while the annotated ARZ corpus (160K tokens; split into 134K train, 20K dev, and 21K blind test) was used for supervised training and evaluation.

In terms of performance, the best configuration achieved POS accuracy of 93.6%, lemma accuracy of 88.1%, diacritization accuracy of 83.8%, and full morphological analysis accuracy of 78.4%, yielding significant error reductions over the MADAMIRA baseline (e.g., 21.9% relative improvement in POS). Error analysis revealed strengths in handling noisy orthography and clitic segmentation, but also weaknesses such as frequent confusion among nominal categories (74% of POS errors) and issues with Hamza spelling, diacritization propagation, and MSA–EGY cognate mismatches. Interestingly, when trained on CODA-normalized orthography, results nearly matched the best noise-robust setup, suggesting the model closely approaches the performance ceiling for such data. Statistical reporting includes accuracy metrics with relative error reduction; however, no explicit confidence intervals or significance tests were provided.

Stanza (StanfordNLP) for Arabic UD [37] is a fully neural, language-agnostic NLP toolkit developed by Stanford, designed to process raw text through a complete pipeline including tokenization, multi-word token expansion, lemmatization, POS and morphological tagging, dependency parsing, and NER. For Arabic, it relies on the PADT treebank within the Universal Dependencies (UD v2.5) framework. Its models use Bi-LSTM architectures with biaffine scoring for syntactic analysis, and seq2seq ensembles for lemmatization and token expansion, allowing the system to generalize effectively across diverse languages. A key strength lies in its broad multilingual coverage (66 languages) and ability to handle text end-to-end from raw input, producing competitive or state-of-the-art performance. Its weaknesses, however, include slower runtime compared to lightweight systems such as spaCy, and occasional errors in sentence segmentation and multi-word token expansion in morphologically rich languages.

The authors also acknowledge computational cost as a limiting factor for scalability and efficiency. For datasets, Stanza was trained on 112 corpora, with Arabic specifically using the PADT UD treebank (non-copyrighted portion), plus additional NER data such as AQMAR for NER. The Arabic PADT evaluation shows very high tokenization accuracy (99.98) and strong performance in POS tagging (UPOS 94.89, XPOS 91.75, UFeats 91.86), lemmatization (93.27), and dependency parsing (UAS 83.27, LAS 79.33). For Arabic NER, Stanza achieved an F1 score of 74.3 on AQMAR, comparable to FLAIR but outperforming spaCy where available. Results were benchmarked against UDPipe and spaCy using the official UD evaluation script, but no statistical significance tests or confidence intervals were reported. Stanza demonstrates robustness and breadth, though efficiency and handling of genre/domain variability remain areas for improvement.

UDPipe 2 (Neural UD Pipeline for Arabic) for Arabic (PADT treebank), UDPipe 2.0 [38] showed strong but not flawless performance. It achieved very high segmentation scores (tokens: 99.98, words: 93.71, sentences: 80.89 F1), indicating reliable basic preprocessing. In tagging, it reached UPOS 90.64, XPOS 87.81, and UFeats 88.05, while lemmatization stood at 87.38. Parsing results were competitive but lower: UAS 88.94, LAS 72.34, MLAS 63.77, and BLEX 65.66. These numbers highlight that while UDPipe is robust at segmentation and POS tagging, parsing complex Arabic syntax remains challenging. Strengths lie in its end-to-end neural joint model that handles multiple tasks consistently without language-specific parameter tuning.

However, weaknesses emerge with Arabic morphology and syntax, where error analysis indicates struggles with rich inflection, clitic segmentation, and long-distance dependencies. For example, the model often produces incorrect lemma forms when diacritics or clitics alter the base word, and dependency arcs occasionally mislabel subordinate clauses or prepositional phrases, reducing LAS. While these shortcomings are typical in morphologically rich languages, the consistency of UDPipe's results across treebanks suggests its architecture generalizes well, even if Arabic parsing lags behind segmentation accuracy.

UDify (mBERT Multi-task Morphology for UD Arabic) [39] is a multilingual, multi-task neural analyzer built on pretrained mBERT embeddings. It jointly predicts POS tags, morphological features, lemmas, and dependency parses using a self-attention architecture. Trained on 124 UD treebanks (75 languages), including Arabic PADT (around 6.1k sentences), UDify achieves strong syntactic accuracy (UPOS 96.58%, UAS 87.72%, LAS 82.88%) but performs poorly in lemmatization (73.55%) due to lack of character-level embeddings—a key limitation for morphologically rich languages. While multilingual training boosts parsing for Arabic, weaknesses remain in morphology-sensitive tasks like lemmas and UFeats. No statistical significance testing or confidence intervals were reported.

Gulf Arabic neural morphology [40] combines rule-based morphological analyzers with a neural disambiguation model. Specifically, the Gulf Arabic analyzer was created automatically through paradigm completion based on annotated training data, while high-quality manual analyzers were used for MSA (SAMA) and Egyptian Arabic (CALIMA). For disambiguation, the authors employed a neural joint model (sequence-to-sequence with shared encoders for lexical and morphological features) alongside a baseline maximum likelihood estimation (MLE) system.

They tested different setups: no analyzer, Gulf-only analyzer, and combinations with MSA and EGY analyzers, embedding or ranking the candidates. The strengths lie in the system's ability to handle Gulf Arabic morphology for the first time and its adaptability to data size. However, weaknesses appear when analyzers constrain the neural model, especially in lemmatization: ranking candidates often reduces accuracy, showing the analyzer's limited coverage could restrict performance rather than improve it. The system was trained and evaluated on the annotated Gumar Corpus [41], Emirati Arabic novels totaling about 202K tokens across train/dev/test splits (with train 162K tokens). Additional embeddings were drawn from the larger 100M-token Gumar corpus. On test, the best configuration reached full 89.2%, TAGS 92.9%, LEX 93.1%, POS 96.7%, SEG 97.3%. Results were reported with detailed breakdowns but without statistical significance testing or confidence intervals. Error analysis highlighted that lemmatization remained the weakest link, often suffering when analyzers' lexicon failed to match the diversity of Gulf lemmas.

## 4 CLASSIFICATION OF ARABIC MORPHOLOGICAL ANALYSIS TECHNIQUES

The results of all twenty analyzers, along with their performance metrics, are presented in Table 3. These metrics: accuracy, precision, recall, and F1-score, are compiled from reported sources to enable direct comparison. When grouped by methodological approach, the results reflect both strengths and trade-offs. Rule-based analyzers such as AlKhalil (2017) and SALMA (2013) show high reliability on large, curated resources, often exceeding 95% in tasks like lemma accuracy or diacritization. However, their performance drops considerably when applied to dialects or fine-grained categories (e.g., patterns, stems), suggesting limited adaptability. Hybrid systems like Madamira (2014), Ibn Ghini (2024), and SinaTools (2024) extend coverage by combining rules with statistical or neural components, producing robust segmentation and morphological disambiguation. Nonetheless, their performance is not uniform: Madamira achieves over 98% in segmentation but lower scores (77–86%) in diacritization and full solutions, while SinaTools performs well in lemma and POS tagging yet shows moderate outcomes in WSD and semantic tasks.

Corpus size and diversity have a clear impact across systems. Models trained on large, balanced resources such as the PATB (>1.3M words) or Gumar (202K annotated) consistently produce higher accuracy and generalization. For instance, Stanford's segmenter (2020) and Farasa (2016) achieve segmentation accuracy near or above 98%, reflecting the advantage of large-scale training. In contrast, analyzers built on smaller or domain-specific corpora, such as the Qur'anic Arabic Corpus (31K words), show very strong performance within that domain (99% accuracy) but with limited applicability beyond it. Dialectal extensions, as in Camelira (2022) or Zalmout–Habash (2018), demonstrate competitive results (POS 90–94%), though accuracy remains below that of MSA systems, highlighting the challenges of resource scarcity and linguistic variability. Neural architectures dominate recent benchmarks in Arabic morphosyntactic tagging and parsing. Models like CAMeLBERT (2022) and UDify (2019) exceed 95% in UPOS and full-tagging tasks, confirming the strength of contextualized embeddings and multitask learning. Hybrid systems remain relevant: Ibn Ghini (2024) offers notable efficiency (0.3 ms/word), and Madamira provides broad functional coverage, making them practical where speed and explainability matter more than state-of-the-art accuracy. The field is shifting: rule-based systems excel in controlled settings, hybrid approaches offer balance, and neural architectures deliver top accuracy when large, diverse corpora are available.

Table 3. Arabic morphological analyzers performance

| Analyzer | Approach | Corpus/Size | Morpheme | Evaluation |
|---|---|---|---|---|
| CAMEL MORPH 2024 | Rule-based | PATB / 1.5M words | lemma + analysis + diacritization | Recall: 95.9% |
| Alma 2024 | Frequency-based + Lexicon-driven + BERT (OOV) | LDC ATB (1.5M), SALMA (500K) | Morphological analysis | F1: 88% (ATB), 90% (SALMA) |
| Ibn Ghini 2024 | Hybrid | 3M words / 600K analyzed | Full morphological solutions | Accuracy: 72.72% |
| | | Alkhalil + BAMA Extended / 3M words | Partial solutions | Accuracy: 24.24% |
| | | 0.3ms per word | Analysis speed | Time: 0.3 ms/word |
| AlKhalil 2017 | Rule-based | Nemlar (500K) & Teshkeela (75M) | Rate-Lemma | Accuracy: 97.16% |
| | | | Rate-stem | Accuracy: 96.76% |
| | | | Rate-diac | Accuracy: 97.21% |
| | | | Rate-full | Accuracy: 96.56% |
| | | Gold standard (MSA, 546 words) | Root | Acc: 74.96%, Prec: 78.94%, Rec: 74.96%, F1: 76.90% |
| | | | Stem | Acc: 54.43%, Prec: 57.33%, Rec: 54.43%, F1: 55.84% |
| | | | Pattern | Acc: 36.35%, Prec: 38.28%, Rec: 36.35%, F1: 37.29% |
| | | Multi-dialect (EGY, TUN, 10 sentences each) | N/A | Acc: 68%, Prec: 68%, Rec: 66%, F1: 67% |
| Stanford 2020 | Statistical ML (CRF-based) | Penn ATB (>1.3M words) | Segmentation | F1: 98.24% |
| Farasa 2016 | Statistical (SVM-rank + Lexicon) | Noor–Ghateh (223,690 words) | Word segmentation | Acc: 81%, Prec: 81%, F1: 89% |
| | | Penn ATB (>1.3M words) | Segmentation (base) | Acc: 98.76% |
| | | Penn ATB (>1.3M words) | Segmentation (lookup) | Acc: 98.94% |
| Madamira 2014 | Hybrid (Rule-based + ML disambiguation with SVM + LMs) | MSA (25K words) | EVALDiac | Acc: 86.3% |
| | | | EvalLex | Acc: 96% |
| | | | EvalFull | Acc: 84.1% |
| | | | Perfect Tok | Acc: 98.9% |
| | | | Correct segmentation | Acc: 99.2% |
| | | EGY (20K words) | EVALDiac | Acc: 83.2% |
| | | | EvalLex | Acc: 87.8% |
| | | | EvalFull | Acc: 77.3% |
| | | | Perfect Tok | Acc: 96.6% |
| | | | Correct segmentation | Acc: 97.6% |
| | | Penn ATB (>1.3M words) | Segmentation | Acc: 98.76% |
| | | Noor–Ghateh (223,690 words) | Word segmentation | Acc: 80%, Prec: 80%, Rec: 99%, F1: 88% |
| | | Multi-dialect (EGY, TUN, MSA, 10 sentences each) | N/A | Acc: 85%, Prec: 86%, Rec: 88%, F1: 87% |
| SALMA 2013 | Rule-based, Knowledge-driven (Grammar + Lexicon) | CCA (500K), Qur'an (77K) | Morphological features | Acc: 98.53% (CCA), 90.11% (Qur'an) |
| | | CCA (500K), Qur'an (77K) | Remaining categories | Acc: 81.35–97.51% (CCA), 74.25–89.03% (Qur'an) |
| SinaTools 2024 | Hybrid (Rule-based + BERT/Transformers) | ATB (339K), SALMA (34K) | Morphology (Lemma, POS) | Lemma: 90.5%, POS: 97.5% |
| | | Wojood (50K), Politics (12K) | NER | F1: 87.3% |
| | | SALMA Sense (34K) | WSD | Acc: 82.6% |
| | | SemEval-2024 (595 pairs) | Semantic relatedness | Spearman: 0.49 |
| Camelira 2022 | Statistical + Neural (CAMeL tools backbone) | MSA | Morphological disambiguation | All tags: 95.9%, POS: 98.7% |
| | | Egyptian | | All: 90.5%, POS: 94.0% |
| | | Gulf | | All: 93.8%, POS: 96.6% |
| | | Levantine | | All: 85.5%, POS: 92.7% |
| Zalmout and Habash 2017 | Neural (Bi-LSTM + Analyzer guidance) | PATB (503K train, 63K dev/test) | Morphological disambiguation | Full: 90.0%, OOV: 76.9% |
| | | Gigaword (2.15B) | | POS: 97.9%, Diac: 91.7% |
| Neural analyzer (RNN) | Neural (RNN + subword vectors) | Qur'anic Arabic Corpus (31K) | Morphological analysis | Acc: 99%, Prec: 99%, Rec: 96%, F1: 97% |
| CAMeLBERT 2022 | Neural (Transformer-based + Analyzer support) | ATB (629K) | Morphosyntactic tagging | POS: 98.9%, All tags: 96.3% |
| | | ARZTB (175K), Gumar (202K), Curras (57K) | | Dialects: 91–95% |
| Zalmout–Habash 2018 (EGY) | Neural (Bi-LSTM, noise-robust) | ARZ (160K), Gumar (410M pretrain) | Morphological disambiguation | POS: 93.6%, Lemma: 88.1%, Diac: 83.8%, Full: 78.4% |
| Stanza 2020 | Neural (Bi-LSTM + seq2seq) | PADT UD (Arabic) | Full UD morphology | UPOS: 94.9, XPOS: 91.8, UFeats: 91.9, Lemma: 93.3 |
| | | | Dependency parsing | UAS: 83.3, LAS: 79.3 |
| | | AQMAR (NER) | NER | F1: 74.3 |
| UDPipe 2.0 (2018) | Neural (Joint model) | PADT UD (Arabic) | Morphology + Parsing | UPOS: 90.6, XPOS: 87.8, UFeats: 88.1 Lemma: 87.4, UAS: 88.9, LAS: 72.3 |
| UDify 2019 | Neural (mBERT multitask) | UD Arabic PADT (6.1K sents) | POS + Features + Lemma + Parsing | UPOS: 96.6, UAS: 87.7, LAS: 82.9, Lemma: 73.6 |
| Gulf Morph 2020 | Hybrid (Rule-based analyzers + Neural disambiguation) | Gumar annotated corpus (202K), embeddings 100M | Gulf morphology (POS, SEG, LEX) | POS: 96.7%, SEG: 97.3% LEX: 93.1%, Full: 89.2% |

## 5 CHALLENGES AND FUTURE RESEARCH DIRECTIONS IN MORPHOLOGICAL ANALYSIS

Despite significant advancements in Arabic NLP, morphological analysis still faces considerable challenges. These challenges arise primarily from the unique linguistic characteristics of Arabic. These include its complex morphology, flexible word order, diacritics omission and rich semantics.

### 5.1 Challenges in arabic morphological analysis

These challenges stem from the complexity of Arabic—its unique word structures and diverse forms. Over time, several key issues have emerged, including:

- Morphological ambiguity: Arabic's root-and-pattern morphology and rich inflection cause high ambiguity, where a single surface form may yield multiple valid analyses, complicating accurate morphological processing [42].
- Dialectal and orthographic variations: Arabic dialects differ markedly from MSA in phonology, syntax, and lexicon, yet lack standardized orthography. Inconsistent spelling and diacritic usage further hinder analysis. Tachicart *et al*. [43] highlight that dialectal diversity and limited resources pose major challenges for developing robust morphological analyzers.
- Resource scarcity: robust morphological analysis relies on large, annotated corpora, which remain scarce for Arabic, especially for dialects. This lack of resources restricts model training, evaluation, and cross-dialectal generalization [43].
- Impact on higher-level NLP tasks: errors in morphological analysis propagate to downstream tasks such as translation and question answering, degrading performance. Essam *et al*. [44] note that unresolved morphological and syntactic complexities impede accurate query interpretation.
- Templatic (root-and-pattern) morphology: a defining characteristic of Arabic morphology is its templatic, or root-and-pattern, structure. Lexical meaning is encoded in consonantal roots, which interdigitate with vocalic patterns to produce surface forms. For example, the root كتب combines with patterns to yield forms such as كَتَبَ, يَكْتُبُ, and مَكْتَب. This non-linear morphology differs from purely concatenative systems and complicates computational segmentation and generation, as both the root and the template must be identified simultaneously.
- Cliticization and affix stacking: Arabic's agglutinative nature allows multiple prefixes and suffixes to attach to a single stem, encoding person, number, gender, tense, and case within one token. For example, وَسَيَكْتُبُونَهَا (and they will write it), a conjunction (وَ), a future tense prefix (سَ), a verb stem (يَكْتُبُ), a plural subject suffix (ونَ), and an object pronoun (هَا) are all combined. Such affix stacking increases ambiguity and challenges both rule-based and neural analyzers [45].
- Clitic segmentation: Arabic orthography permits clitics such as conjunctions, prepositions, and pronouns, to attach directly to host words, forming long orthographic tokens. For example, وَبِكِتَابِهِمْ (and with their book) combines وَ, بِ, كِتَابِ, and هِمْ. Accurate segmentation of these clitics is essential for POS tagging, parsing, and semantic interpretation, as segmentation errors often cascade through downstream tasks [46].

Addressing these challenges requires developing standardized orthographies, expanding annotated corpora, and designing advanced models capable of capturing the intricacies of Arabic morphology.

### 5.2 Future research directions

Future research in Arabic morphological analysis should adopt hybrid approaches combining rule-based systems, statistical methods, and deep learning. Rule-based models capture linguistic nuances, while machine learning ensures scalability. Transformer-based models like AraBERT and CAMeL-BERT show strong potential; fine-tuning them can address challenges such as morphological ambiguity, flexible word order, and diacritic variation. DA remains a priority, as most tools target MSA. With dialects dominating social networks and informal text, robust models for MSA–dialect switching, dialect-specific parsers, and cross-dialect transfer learning are essential. Improving diacritization and integrating it with context-aware embeddings can reduce ambiguity.

Leveraging transfer learning from multilingual models (e.g., mBERT, XLM-R) helps overcome limited annotated data. Building large, diverse annotated corpora for both MSA and dialects is critical. These should include academic, social media, and spoken data, supported by better labeling tools. Explainable artificial intelligence (XAI) is also vital for trust in domains such as healthcare and finance. Finally, cross-disciplinary collaboration among linguists, computer scientists, and AI experts can drive innovation

in speech recognition, virtual assistants, and multimodal systems, while promoting open datasets and community-driven progress.

## 6 CONCLUSION

Arabic NLP has benefited from powerful morphological analyzers that address core challenges such as ambiguity, inflectional richness, and flexible word order. However, unresolved issues including limited annotated corpora, insufficient support for dialects, and high computational costs continue to restrict progress. Recent advances in hybrid models and transformer-based architectures present promising opportunities, but practical adoption still requires careful tool selection. For tasks in MSA, mature analyzers such as MADAMIRA and SAMA remain reliable, especially in academic or formal text processing. For DA, emerging neural systems (e.g., CAMeL Morph or dialect-specific analyzers) are better suited, though they often require adaptation and domain-specific fine-tuning. For deployment in resource-constrained environments (e.g., mobile or real-time applications), lightweight analyzers like Farasa offer an optimal trade-off between speed and accuracy. For integrative NLP pipelines (machine translation, speech recognition, educational platforms), transformer-based models fine-tuned for Arabic show the most promise, provided sufficient annotated corpora are available.

## FUNDING INFORMATION

Authors state no funding involved.

## AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

| Name of Author | C | M | So | Va | Fo | I | R | D | O | E | Vi | Su | P | Fu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Omar Saadiyeh | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | | ✓ | | | |
| Alaaeddine Ramadan | ✓ | ✓ | | | ✓ | | ✓ | ✓ | | ✓ | | | ✓ | ✓ |
| Chamseddine Zaki | ✓ | ✓ | ✓ | ✓ | | ✓ | | | ✓ | | ✓ | ✓ | | ✓ |
| Mohamad Hajjar | ✓ | ✓ | | | ✓ | | ✓ | | | ✓ | | ✓ | ✓ | |
| Gilles Bernard | ✓ | ✓ | | | ✓ | | ✓ | | | ✓ | | ✓ | ✓ | |

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

## CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

## DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author [AR] upon reasonable request.

## REFERENCES

[1] N. Boudad, R. Faizi, R. O. H. Thami, and R. Chiheb, "Sentiment analysis in Arabic: a review of the literature," *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 2479–2490, Dec. 2018, doi: 10.1016/j.asej.2017.04.007.

[2] N. Y. Habash, *Introduction to Arabic natural language processing*. Cham, Switzerland: Springer International Publishing, 2010, doi: 10.1007/978-3-031-02139-8.

[3] S. Harrat, K. Meftouh, and K. Smaïli, "Maghrebi Arabic dialect processing: an overview," *Journal of International Science and General Applications*, vol. 1, no. 1, Mar. 2018,

[4] S. A. Katat, C. Zaki, H. Hazimeh, I. E. Bitar, R. Angarita, and L. Trojman, "Natural language processing for Arabic sentiment analysis: a systematic literature review," *IEEE Transactions on Big Data*, vol. 10, no. 5, pp. 576–594, Oct. 2024, doi: 10.1109/TBDATA.2024.3366083.

[5] F. Sadat, F. Mallek, M. Boudabous, R. Sellami, and A. Farzindar, "Collaboratively constructed linguistic resources for language variants and their exploitation in NLP application – the case of Tunisian Arabic and the social media," in *Proceedings of Workshop on Lexical and Grammatical Resources for Language Processing*, 2014, pp. 102–110, doi: 10.3115/v1/W14-5813.

[6] O. F. Zaidan and C. C. Burch, "Arabic dialect identification," *Computational Linguistics*, vol. 40, no. 1, pp. 171–202, Mar. 2014, doi: 10.1162/COLI_a_00169.

[7] N. Habash, "Arabic morphological representations for machine translation," in *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Dordrecht, Netherlands: Springer, 2007, pp. 263–285, doi: 10.1007/978-1-4020-6046-5_14.

[8] O. Saadiyeh, A. Ramadan, M. Hajjar, and G. Bernard, "A comparative study of Arabic syntactic analyzers," *Frontiers in Artificial Intelligence*, vol. 8, Aug. 2025, doi: 10.3389/frai.2025.1638743.

[9] M. Sawalha, E. Atwell, and M. A. M. Abushariah, "SALMA: Standard Arabic language morphological analysis," in *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, Feb. 2013, pp. 1–6, doi: 10.1109/ICCSPA.2013.6487311.

[10] K. Dukes and N. Habash, "Morphological annotation of Quranic Arabic," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, May 2010, pp. 2530-2536.

[11] L. Al-Sulaiti and E. S. Atwell, "The design of a corpus of contemporary Arabic," *International Journal of Corpus Linguistics*, vol. 11, no. 2, pp. 135–171, Jul. 2006, doi: 10.1075/ijcl.11.2.02als.

[12] D. Graff, M. Maamouri, B. Bouziri, S. Krouna, S. Kulick, and T. Buckwalter, "Standard Arabic morphological analyzer (SAMA) version 3.1," *Linguistic Data Consortium LDC2009E73*, pp. 53–56, 2009.

[13] T. Buckwalter, "Buckwalter Arabic morphological analyzer Version 2.0," *Philadelphia: Linguistic Data Consortium*, Dec. 15, 2004, doi: 10.35111/050Q-5R95.

[14] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, "Farasa: A fast and furious segmenter for Arabic," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 2016, pp. 11–16, doi: 10.18653/v1/N16-3003.

[15] M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki, "The Penn Arabic Treebank: building a large-scale annotated Arabic corpus," in *NEMLAR conference on Arabic language resources and tools*, Cairo, 2004, pp. 466–467.

[16] A. Boudlal, A. Lakhouaja, A. Mazroui, A. Meziane, M. Bebah, and M. Shoul, "Alkhalil Morpho Sys: a morphosyntactic analysis system for arabic texts," in *International Arab conference on information technology*, 2010, pp. 1–6.

[17] M. Boudchiche, A. Mazroui, M. O. A. Ould Bebah, A. Lakhouaja, and A. Boudlal, "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, Apr. 2017, doi: 10.1016/j.jksuci.2016.05.002.

[18] T. Zerrouki and A. Balla, "Tashkeela: Novel corpus of Arabic vocalized texts, data for auto-diacritization systems," *Data in Brief*, vol. 11, pp. 147–151, Apr. 2017, doi: 10.1016/j.dib.2017.01.011.

[19] W. Monroe, S. Green, and C. D. Manning, "Word segmentation of informal Arabic with domain adaptation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 206–211, doi: 10.3115/v1/P14-2034.

[20] A. Pasha *et al*., "MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2014, pp. 1094–1101.

[21] N. Habash, R. Roth, O. Rambow, R. Eskander, and N. Tomeh, "Morphological analysis and disambiguation for dialectal Arabic," in *North American Chapter of the Association for Computational Linguistics*, 2013.

[22] M. Diab, K. Hacioglu, and D. Jurafsky, "Automatic processing of modern standard Arabic text," in *Arabic Computational Morphology*. Dordrecht, Netherlands: Springer, 2007, doi: 10.1007/978-1-4020-6046-5_9.

[23] C. Khairallah, S. Khalifa, R. Marzouk, M. Nassar, and N. Habash, "Camel morph MSA: a large-scale open-source morphological analyzer for modern standard Arabic," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, May 2024, pp. 2683–2691.

[24] O. Obeid *et al*., "CAMeL tools: an open source Python toolkit for Arabic natural language processing," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, May 2020, pp. 7022–7032.

[25] M. Jarrar, D. Akra, and T. Hammouda, "Alma: Fast lemmatizer and POS tagger for Arabic," *Procedia Computer Science*, vol. 244, pp. 378–387, 2024, doi: 10.1016/j.procs.2024.10.212.

[26] M. Jarrar and T. H. Hammouda, "Qabas: An open-source Arabic lexicographic database," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, Torino, Italia: ELRA and ICCL, May 2024, pp. 13363–13370.

[27] W. Nazih, A. Fashwan, A. El-Gendy, and Y. Hifny, "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, Feb. 2024, doi: 10.1145/3639050.

[28] T. Hammouda, M. Jarrar, and M. Khalilia, "SinaTools: Open source toolkit for Arabic natural language processing," *Procedia Computer Science*, vol. 244, pp. 388–396, 2024, doi: 10.1016/j.procs.2024.10.213.

[29] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based model for Arabic language understanding," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, May 2020, pp. 9–15.

[30] M. Jarrar, M. Khalilia, and S. Ghanem, "Wojood: Nested Arabic named entity corpus and recognition using BERT," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, Jun. 2022, pp. 3626–3636.

[31] O. Obeid, G. Inoue, and N. Habash, "Camelira: An Arabic multi-dialect morphological disambiguator," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2022, pp. 319–326, doi: 10.18653/v1/2022.emnlp-demos.32.

[32] N. Zalmout and N. Habash, "Don't throw those morphological analyzers away just yet: neural morphological disambiguation for Arabic," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 704–713, doi: 10.18653/v1/D17-1073.
[33] D. Graff, K. Chen, J. Kong, and K. Maeda, *Arabic gigaword second edition*, Philadelphia, United States: Linguistic Data Consortium, 2006, doi: 10.35111/scbm-4q37.
[34] T. I. Ramadhan, M. A. Bijaksana, and A. F. Huda, "Neural-based Arabic morphological analyzer," in *2021 International Conference on Data Science and Its Applications (ICoDSA)*, Bandung, Indonesia, Oct. 2021, pp. 16–21, doi: 10.1109/ICoDSA53588.2021.9617211.
[35] G. Inoue, S. Khalifa, and N. Habash, "Morphosyntactic tagging with pre-trained language models for Arabic and its dialects," in *Findings of the Association for Computational Linguistics: ACL 2022*, 2022, pp. 1708–1719, doi: 10.18653/v1/2022.findings-acl.135.
[36] N. Zalmout, A. Erdmann, and N. Habash, "Noise-robust morphological disambiguation for dialectal Arabic," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 953–964, doi: 10.18653/v1/N18-1087.
[37] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, "Stanza: A Python natural language processing toolkit for many human languages," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020, pp. 101–108, doi: 10.18653/v1/2020.acl-demos.14.
[38] M. Straka, "UDPipe 2.0 prototype at CoNLL 2018 UD shared task," in *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Oct. 2018, pp. 197–207, doi: 10.18653/v1/K18-2020.
[39] D. Kondratyuk and M. Straka, "75 languages, 1 model: parsing universal dependencies universally," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 2779–2795, doi: 10.18653/v1/D19-1279.
[40] S. Khalifa, N. Zalmout, and N. Habash, "Morphological analysis and disambiguation for Gulf Arabic: the interplay between resources and methods," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, Marseille, France: European Language Resources Association, May 2020, pp. 3895–3904.
[41] S. Khalifa, N. Habash, D. Abdulrahim, and S. Hassan, "A large scale corpus of Gulf Arabic," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, May 2016, pp. 4282–4289.
[42] Y. A. Moaiad, M. Alobed, M. Alsakhnini, and A. M. Momani, "Challenges in natural Arabic language processing," *Edelweiss Applied Science and Technology*, vol. 8, no. 6, pp. 4700–4705, Nov. 2024, doi: 10.55214/25768484.v8i6.3018.
[43] R. Tachicart, K. Bouzoubaa, S. Harrat, and K. Smaïli, "Arabic dialects morphological analyzers: a survey," in *Recent Innovations in Artificial Intelligence and Smart Applications*, Cham: Springer International Publishing, 2022, pp. 189–203, doi: 10.1007/978-3-031-14748-7_11.
[44] M. Essam, M. A. Deif, and R. Elgohary, "Deciphering Arabic question: a dedicated survey on Arabic question analysis methods, challenges, limitations and future pathways," *Artificial Intelligence Review*, vol. 57, no. 9, Aug. 2024, doi: 10.1007/s10462-024-10880-6.
[45] M. E. Abdi, B. S. B. Ali, and S. B. Yahia, "A new approach of morphological analysis of Arabic syntagmatic units based on a linguistic ontology," in *Computational Collective Intelligence*, vol. 13501, Cham: Springer International Publishing, 2022, pp. 364–377, doi: 10.1007/978-3-031-16014-1_29.
[46] N. Habash and O. Rambow, "Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*, 2005, pp. 573–580, doi: 10.3115/1219840.1219911.

## BIOGRAPHIES OF AUTHORS

**Omar Saadiyeh** received the B.S. degree in Computer and Communication Engineering and the M.S. degree in Information Systems Engineering from the Faculty of Technology, Lebanese University, Lebanon. He is currently pursuing a Ph.D. degree at Paris 8 University, Saint-Denis, France, where his research focuses on syntactic analysis in NLP, specifically for modern standard Arabic. This resource is intended to support the evaluation and training of both supervised and unsupervised Arabic NLP systems. He can be contacted at email: omar_saadiyeh@hotmail.com.

**Alaaeddine Ramadan** received his B.Sc. in Computer and Communication Network Engineering from Lebanese University, Lebanon, in 2005, and his M.S. and Ph.D. in Electronics and Communication Engineering from the University of Limoges, France, in 2007 and 2010, respectively, with research funded by the French Space Agency and Thales Alenia Space. Currently, he is an associate professor and program lead at the American University of Bahrain, focusing his research on AI, IoT, WSN, and machine learning. He can be contacted at email: aramadan@ieee.org.

**Chamseddine Zaki** currently an assistant professor at the American University of the Middle East, he received his master's degree in Computer Science from École Polytechnique de Nice-Sophia Antipolis and his Ph.D. in Computer Engineering from Centrale Nantes. His research interests include geographic information systems, spatiotemporal data analysis, multi-criteria analysis, conceptual modeling, and travel time estimation. He can be contacted at email: chamseddine.zaki@aum.edu.kw.

**Mohamad Hajjar** received the Ph.D. degree in Computer Science from the University of Nantes, France, in 1997, and the M.S. degree in Applied Computer Science and Control Systems from the same university in 1993. He joined the Lebanese University in 2000, where he became a professor in 2007 and has served as dean of the Faculty of Technology since 2014. He has contributed to academic program development and established numerous international cooperation agreements. His research interests include informatics, computer networks, systems modeling, and data management. He is a member of the GRIT Research Group and serves as a reviewer for international journals and conferences. He can be contacted at email: mohammadhajjar@ul.edu.lb.

**Gilles Bernard** spent his childhood in Russia, India, and France, cultivating an early passion for languages. He holds a B.A. in Chinese, a M.S. in Mathematics, and a M.S. in Linguistics, and completed advanced studies in computer science at the University of Paris–Denis Diderot. He earned his habilitation (HDR) at the University of Paris VIII. His research focuses on modeling the contextual understanding of language through the integration of syntactic parsing and semantic representation. He has developed cognitively inspired approaches to linguistic analysis and explored hybrid methods that combine symbolic and connectionist models for natural language processing. He can be contacted at email: gilles.bernard@iedparis8.net.