Artificial Intelligence & Robotics Development Journal
Volume 6, Issue 4, pp 466-478, November 2026, https://doi.org/10.52098/airdj.20266496
ISSN: 2788-9696 Received: 25/11/2026 Revised: 27/11/2026 Accepted: 27/11/2026

A Hybrid Fuzzy Deep Learning Framework for Arabic Part-of-Speech Tagging Using BiLSTM and AraBERT

Jabar H. Yousif [1,*]
[1] Faculty of Computing and Information Technology, Sohar University, Oman, PO Box 44, Sohar, PCI 311,
* Corresponding author: Jabar H. Yousif, jyousif@su.edu.om

Abstract

Tagging of POS in Arabic faces difficulties due to the high morphological richness of the language, absence of diacritics, and dependency on the context of grammatical analysis. In this research, two hybrid approaches, Fuzzy-BiLSTM and Fuzzy-AraBERT, are introduced, which combine soft linguistic features with context-based neural representation. The performance of four different architectures was tested using Arabic-PADT datasets under three different random seeds. Fuzzy-BiLSTM obtained a mean accuracy of 93.44% and F1 score of 84.73%, while BiLSTM obtained 93.03% and 84.58%. Fuzzy-AraBERT obtained the highest mean accuracy (97.03%), F1 (88.80%), and OOV accuracy (86.59%). The addition of fuzzy integration to the systems led to improvements of 2.18 and 0.28 percentage points in terms of OOV accuracies for BiLSTM and AraBERT, respectively; however, fuzzy integration did not improve the average accuracies for the ambiguous words. Fuzzy integration involved adding 289 neural parameters to each of the two systems. These results reveal the complementary advantages of linguistic feature integration and a significant efficiency-performance trade-off between RNNs and transformers.

Keywords*:* Arabic natural language processing; part-of-speech tagging; fuzzy logic; BiLSTM; AraBERT; neuro-fuzzy systems.

1. Introduction

Student Part-of-speech (POS) tagging refers to the process of identifying the grammatical classes of words and is utilized for many NLP tasks (Yousif & Sembok, 2006a). The unique challenge presented by the Arabic language is that of clitic attachment, morphological analysis, and lack of short vowels, all of which result in a possibility of ambiguity for the same word (Darwish et al., 2020; Inoue et al., 2022). The difficulties arising from different

Author(s) and ACAA permit unrestricted use, distribution, and reproduction in any medium, provided the original work with proper citation. This work is licensed under Creative Commons Attribution International License (CC BY 4.0).

annotation schemes, corpora, and vocabularies make comparative analysis difficult. The classical Arabic POS taggers were based on linguistics and dictionaries (Khoja, 2001), which later developed into statistical and hybrid methods (Hadni et al., 2013; Tnaji et al., 2021; Yousif, 2019). The reported performance is based on the grammatical features and test corpora used. The early attempts with neural networks and support vector machines showed the feasibility of acquiring the POS categories from annotated texts (Yousif & Sembok, 2006a, 2006b, 2008).

Bidirectional LSTM (biLSTM) models learn from context on both sides (Hochreiter & Schmidhuber, 1997; Plank et al., 2016). BERT (Devlin et al., 2019) and dialect Arabic AraBERT (Antoun et al., 2020) models build on contextual information using pretraining. Research on Arabic and its dialects shows the necessity of having context-aware representations as well as domain awareness (Bouzahir et al., 2022; Darwish et al., 2020; Inoue et al., 2022).

Soft Computing is the application of algorithms for modeling the imprecision of data and inference approximation (Dai et al., 2011; Yousif, 2013). Fuzzy Sets provide degrees of membership rather than binary sets (Zadeh, 1965). Moreover, there have been attempts to utilize evolutionary and swarm-based algorithms for tagging in the Arabic language (Al-Taani & ALkhazaaleh, 2022; Alhasan & Al-Taani, 2018; Ben Ali & Jarray, 2013); however, the integration of these linguistic fuzzy features along with contextual neural networks has not been sufficiently addressed in the reviewed literature.

The proposed models Fuzzy-BiLSTM and Fuzzy-AraBERT have been tested in comparison to their baseline models on the Universal Dependencies Arabic Prague Dependency Treebank (Arabic-PADT) (Smrž et al., 2008; Universal Dependencies, n.d.) through four different criteria: predictive performance, lexical generalization, classification of ambiguous words, and computational cost.

To fill in the research gaps highlighted above, this research proposes developing a linguistic fuzzy inference system for encoding morphological features and grammatical ambiguity of the Arabic language and developing two hybrid neural network systems named Fuzzy-BiLSTM and Fuzzy-AraBERT by merging fuzzy linguistic reasoning with a bidirectional contextual learning model and pre-trained transformers, respectively. The proposed models have been evaluated on the Arabic-PADT dataset to analyze the role of fuzzy reasoning in classification accuracy, grammatical disambiguation, out-of-vocabulary word classification, and efficiency.

2. Soft Computing Background

Response The field of soft computing (SC) is a part of computational intelligence that offers malleable methodologies for solving complex problems that are associated with uncertainties, imprecision, vagueness, and incompleteness. Contrary to classical computing techniques, which mainly depend on mathematically exact and deterministic solutions, soft computing stresses flexibility, learning, approximate reasoning, and optimization. The basic methodologies of soft computing include artificial neural network (ANN), fuzzy logic (FL), evolutionary computation (EC), and probabilistic reasoning. These methodologies can function as individual methodologies or even be combined together within hybrid intelligent systems.

As for NLP, soft computing provides helpful tools for addressing problems of language vagueness, morphology, and context dependence (Dai et al., 2011; Yousif, 2013). The theory of fuzzy logic was initially introduced by Zadeh (1965) to represent the processes of uncertain reasoning and modeling through mathematical tools. Contrary to classical logic, which permits only two possible states of membership of an object in a set, fuzzy logic allows assigning

an object to more than one category based on its partial membership values from zero to one. A traditional fuzzy inference process includes fuzzification, knowledge base, inference engine, and defuzzification. Fuzzification is responsible for mapping the inputs into fuzzy membership values, while knowledge base contains the information about the membership functions and linguistic IF–THEN rules. The inference engine processes this information in order to produce fuzzy conclusions that can be transformed either into crisp outputs or kept in their fuzzy form for further classification.

Another basic element in soft computing is that of artificial neural networks, which provide adaptive learning capabilities for recognizing non-linear dependencies between input attributes and output categories. Recurrent Neural Networks (RNNs) are an extension of traditional neural network models, where there are dependencies in sequence (Yousif J. & Yousif, 2026). LSTM or Long Short-Term Memory, developed by Hochreiter and Schmidhuber in 1997, overcomes some limitations of long-term dependency learning using memory units.

Bidirectional Long Short-Term Memory (BiLSTM) expands the concept of sequential learning by processing input sequences in both directions, which is significant for POS tagging because the context of a word may be dependent on previous and subsequent words. In the contextual learning, recent advancements have brought forth architectures based on transformers that incorporate attention techniques to learn relationships between words within input sequences. The bidirectional encoder representations from transformers architecture (BERT) by Devlin et al. (2019) produces context-based linguistic representation via large-scale pre-training of language models.

In this regard, AraBERT, designed by Antoun et al. (2020), represents this architecture for Arabic language processing. Even though AraBERT is generally not considered a form of soft computing technique, the combination of this technique with fuzzy reasoning can provide ways to implement hybrid intelligence systems that incorporate context learning along with linguistic knowledge. A hybrid soft computing system integrates complementary techniques of computation in order to address the shortcomings of the techniques separately. In this case, a neuro-fuzzy system represents a combination of approximate reasoning from fuzzy logic and adaptive learning from neural networks.

POS Tagging in the Arabic language constitutes one of the most pertinent applications of hybrid soft computing, where each Arabic word can have several possible grammatical senses depending on its morphology, diacritization, and context. Hence, this study suggests the use of two hybrid computational models, namely, Fuzzy-BiLSTM and Fuzzy-AraBERT, which combine fuzzy linguistic feature representation with context-based neural learning to see whether they improve classification, solve ambiguity, recognize out-of-vocabulary terms, and are computationally efficient.

3. Soft Computing Background

This section reviews Arabic POS tagging, emphasizing contextual learning and linguistic knowledge integration.

3.1. Hybrid and Soft Computing Approaches

The hybrid soft computing methods use linguistic knowledge, probability theory, learning, and optimization inspired by nature to handle the morphology and grammar of Arabic text. The evolutionary algorithms and swarm intelligence can offer adaptive means of investigating other possible grammatical meanings and optimizing classification parameters. Al-Taani and ALkhazaaleh (2022) suggested a hybrid method for part-of-speech tagging in Arabic by combining PSO with genetic crossover and mutation. HMM was used to test the suitability of the solutions

generated. An accuracy of 90.5% was obtained when using the KALIMAT corpus, which has roughly 18 million words with 45 parts of speech. Nevertheless, the comparison with other approaches using much smaller tag sets undermines the validity of the findings.

Ben Ali & Jarray (2013) explored the utilization of genetic algorithms in Arabic Part-of-Speech tagging, where sentences were encoded as chromosomes made up of grammatical tag strings. Solutions were scored based on their probabilities within context. While the technique was shown to be flexible, it had high computational demands as opposed to the conventional probabilistic techniques. Alhasan & Al-Taani (2018) employed the bee colony optimization algorithm in Arabic Part-of-Speech tagging utilizing the KALIMAT corpus. The achieved accuracy was 98.2%. However, disparities in tagging schemes and experimental conditions make it impossible to compare it with other evolutionary algorithms.

RlyMore recently, Bouzahir et al. (2026) have developed Jawhar, which is an amalgamation of a Python-based framework for part-of-speech tagging in Arabic that combines morphological analysis with CAMeL-BERT contextual reranking. Using PADT for the evaluation, the results showed 96.4% classification accuracy, 92.1% macro F1-score, and 84.0% out-of-vocabulary accuracy. However, full morphological analysis could be done for only 65.9% of all the tokens tested. This demonstrates the problems in integrating morphological validity and contextual classification.

The above studies focus on optimizing tag sequences or merging morphological and contextual information. In contrast, this research focuses on linguistic feature fusion with recurrent and transformer encoders.

3.2. Neural Contextual Learning Approaches

The Early Arabic POS-tagging studies explored MLP, RNN and SVM architectures (Yousif & Sembok, 2006a, 2006b, 2008). Differences in experimental protocols limit direct comparisons with contemporary models. In another study, Darwish et al. (2020) analyzed CRF and deep learning models for multilingual Arabic POS tagging. The joint multidialectal model had an accuracy of 92.4% based on 1,400 tweets in Arabic. However, due to the relatively small sample size and context of social media, generalizability was constrained.

Bouzahir et al. (2022) provided a literature review of the deep learning architectures used for Arabic POS tagging and recognized LSTM and BiLSTM as significant approaches for context-based learning. The literature review emphasized the significance of sequential information in overcoming the grammatical ambiguity problem. However, variations in experimental settings hindered direct comparison of the models reviewed.

Inoue et al. (2022) examined pretrained transformer models for morphosyntactic tagging of Arabic, including both MSA and regional dialects. The research produced absolute improvements compared to previously developed methods by 2.6%, 2.8%, 1.6%, and 8.3% for MSA, Gulf, Egyptian, and Levantine Arabic, respectively. The study also showed the advantages of using morphological resources and multidialectal training data, especially in low-resource situations.

Akra et al. (2025) developed an active-learning method for POS tagging of multidialectal Arabic. The approach focused on the selection of informative samples to minimize the number of annotations necessary and reached 97.6% accuracy on the Emirati Arabic dataset with only 16,000 annotated tokens. The work shows the possibilities of combining contextual approaches and effective sample selection but is more related to annotation efficiency than network architecture improvements. The performance of AraBERT, MARBERT, and CAMeL Tools has been analyzed by Mahdjoubi and Laskri (2025) on the basis of an Arabic corpus containing 8,147 sentences and 293,199

tokens. The macro-F1 score obtained in this regard has been reported to be around 83.5% for journalistic texts, 66% for synthetic texts, and 55% for literary texts.

3.3. Morphological and Lexical Integration

The difficulty with the morphology of the Arabic language is that one undiacritized word can be related to multiple grammatical classes, roots, and morphologies. Thus, combining the two has become a valuable area of research. Alma (2024) was designed by Jarrar et al. as an open-source Arabic POS tagger, lemmatizer, and root identifier. This system operates on the basis of frequency ranked morphological knowledge acquired from the Qabas lexical database. It uses a rich grammatical tag-set and was tested on the Arabic Treebank and Salma corpora. This system proved to be useful in efficient Arabic grammatical analysis, but context might be required in frequency-based morphological choice cases.

Kallas et al. (2024) have introduced EMAD, an intermediate morphological tag set for facilitating interoperability across annotation systems for Arabic. This framework deals with the problem of inconsistencies in the grammatical representation of annotations that inhibit the effective combination of corpora. While EMAD itself is not a new POS tagger per se, it helps in achieving better datasets and evaluation practices. This body of literature inspires the need to bring together linguistic knowledge and contextual representations under one evaluation protocol.

3.4. Research Gaps

Although there have been great strides made in the field of Arabic POS tagging, some limitations have been observed in the literature reviewed. Some of the methods used to solve the problem include rule-based, probabilistic, recurrent neural networks, and pre-trained transformers. Despite the successful classification results obtained by context-aware models based on deep learning, their approach is largely focused on learned language representation without applying any fuzzy reasoning. It has not been sufficiently explored how to integrate fuzzy membership functions and linguistic rules into context-aware neural architectures, such as BiLSTM and AraBERT, in the literature. The problem lies in the ability of fuzzy logic to assign gradual membership to different categories of grammar, which complements language representation of ambiguous words.

In addition, variations in benchmarking data, tagging schemes, preprocessing methods, and evaluation measures constrain any meaningful comparisons between current Arabic POS tagging systems. Little is known about the role played by fuzzy linguistic reasoning for contextual deep learning models under standard experimental settings based on the Arabic Prague Dependency Treebank (Arabic-PADT). Also, none of the current works consider the issues of grammatical ambiguity disambiguation, OOV classification, and computation time efficiency along with traditional classification measures.

To address these gaps, the paper introduces a novel hybrid soft computing model made up of two computational frameworks, Fuzzy-BiLSTM and Fuzzy-AraBERT. The Fuzzy-BiLSTM framework employs fuzzy linguistic reasoning alongside bidirectional contextual reasoning, while the Fuzzy-AraBERT framework utilizes fuzzy linguistic reasoning in addition to transformer-based pre-training. The two computational frameworks employ fuzzy membership functions and linguistic reasoning to represent morphological properties and different interpretations of grammar. Regular BiLSTM and AraBERT frameworks act as baseline frameworks in order to measure the impact of

fuzzy reasoning. All four frameworks are implemented in Python using the standard partitioning of the Arabic-PADT data.

4. Proposed Methodology

The current study presents a novel hybrid soft computing approach for POS tagging in Arabic using Python programming that uses fuzzy linguistic reasoning along with contextual deep learning. The hybrid soft computing framework presented is focused on solving problems related to Arabic morphology, grammatical ambiguity, context dependency, and OOV words. The framework has three stages, as shown in Figure 1.

Step 1: Preprocessing and Feature Extraction. The Arabic language text is extracted from the Arabic-PADT benchmark dataset and then goes through preprocessing, tokenization, morphological feature extraction, and word representation. This step helps to identify the linguistic features in the form of prefixes, suffixes, lexicon, and context that help fuzzy reasoning.

Stage 2: Fuzzy Linguistic Inference. The extracted linguistic features are then subjected to the fuzzy inference process, which is made up of three components: fuzzification, linguistic IF-THEN rules, and fuzzy feature combination. The three components are employed to represent grammatical ambiguity in terms of different membership values to various possible POS categories.

Stage 3: Hybrid Deep Learning Classification. Two fuzzy-neural architectures, namely Fuzzy-BiLSTM and Fuzzy-AraBERT, use fuzzy linguistic information along with the contextual neural representations. In Fuzzy-BiLSTM, the context is learned using bidirectional sequential learning, while in Fuzzy-AraBERT, context is learned using pre-trained transformer representations. Both architectures concatenate their contexts with the fuzzy linguistic features before doing POS tagging through the Softmax layer.

The proposed framework combines the approximate reasoning capabilities of fuzzy logic with the contextual learning capabilities of modern neural architectures. Conventional BiLSTM and AraBERT models serve as experimental baselines to investigate the contribution of fuzzy reasoning to classification performance, grammatical ambiguity resolution, OOV classification, and computational efficiency.

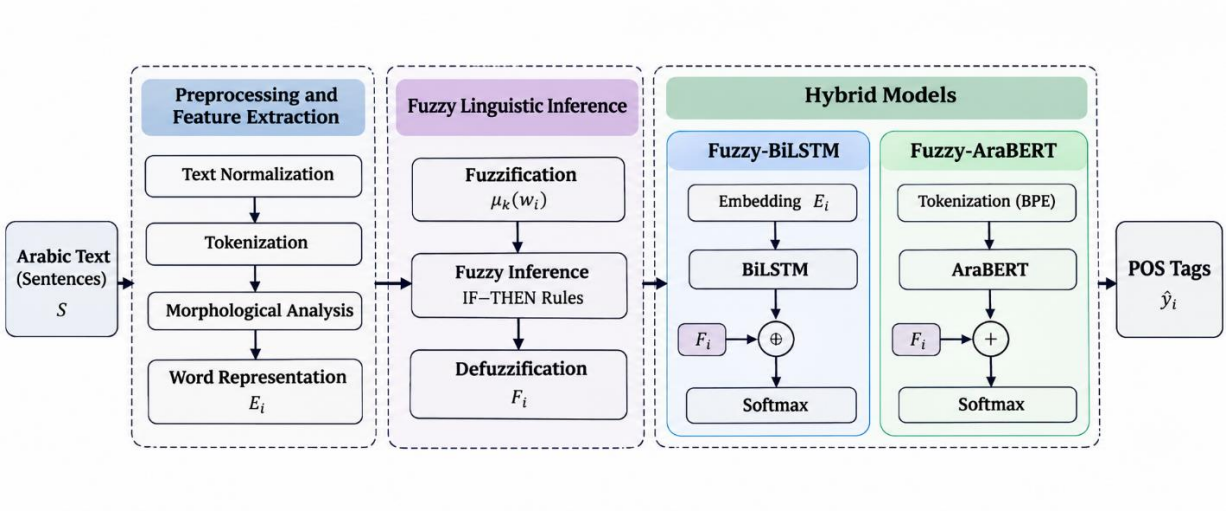


Figure 1. Proposed hybrid fuzzy contextual learning architecture.

4.1. Mathematical Formulation

For each Arabic word w_i, the fuzzy inference system generates a membership vector:

$$F_i = [\mu_1(w_i), \mu_2(w_i), \dots, \mu_K(w_i)] \dots (1)$$

where: w_i represents the input Arabic word and Krepresents the number of POS categories. The factor $\mu_K(w_i)$ represents the membership degree of the word in grammatical category K. The fuzzy vector represents linguistic evidence supporting different grammatical interpretations. The BiLSTM generates a contextual representation by processing the sentence in forward and backward directions.

$$H_i = \left[\overrightarrow{h_i}; \overleftarrow{h_i}\right] \dots (2),$$

The contextual representation is combined with the fuzzy linguistic representation: $Z_i = [H_i; F_i] \dots (3)$

The final POS prediction is calculated using: $\hat{y}_i = \text{Softmax}(WZ_i + b) \dots (4)$

AraBERT generates a contextual representation for each input word: $B_i = \text{AraBERT}(S)_i \dots (5)$

where S represents the input sentence, and subword representations are aligned and aggregated at the word level.

The fuzzy linguistic vector is integrated with the contextual representation: $Z_i = [B_i; F_i] \dots (6)$

The final grammatical classification is calculated as:

$$\hat{y}_i = \text{Softmax}(WZ_i + b) \dots (7)$$

$$\hat{y}_i = \arg\max_k P(y_i = k \mid S) \dots (8)$$

where S represents the input Arabic sentence, W and b denote the trainable classification parameters, and $\hat{y}_i$ represents the predicted POS tag.

Both architectures use a common fuzzy linguistic inference component, enabling a controlled investigation of its effectiveness across recurrent and transformer-based architectures.

4.2. Experimental Setup

The framework was developed in Python using PyTorch version 2.14.1+cu130 and Transformers version 4.57.3. The archived configuration includes Arabic-PADT version 2.17, aubmindlab/bert-base-arabertv02, a limit of eight epochs, batch sizes of 32 (BiLSTM) and 8 (AraBERT), and FP32 training. Both models have been assessed with three different seeds: 42, 43, and 44. Official train, development and test splits have been utilized. In the archived test predictions, there are 28,264 tokens in 680 sentences, consisting of 1,685 OOV tokens and 11,983 ambiguous tokens. The macro evaluation takes the arithmetic mean of all 17 labels, and any potential division by zero is considered 0. INTJ has no test instances. OOV and Ambiguous-word accuracy is evaluated on each one's respective subset. The ambiguity criteria, fuzzy rules' inventory and membership functions must be published along with the implementation.

Arithmetic mean and standard deviation across seeds are reported. Difference values between hybrids and their corresponding baselines are expressed in percentage points (pp). The archived bootstrap interval for seed 42 of the

paired sentence-level predictions is provided; the value does not account for the variance among different training seeds (Uraibi et al, 2009).

5. Results and Discussion

5.1. Classification Performance

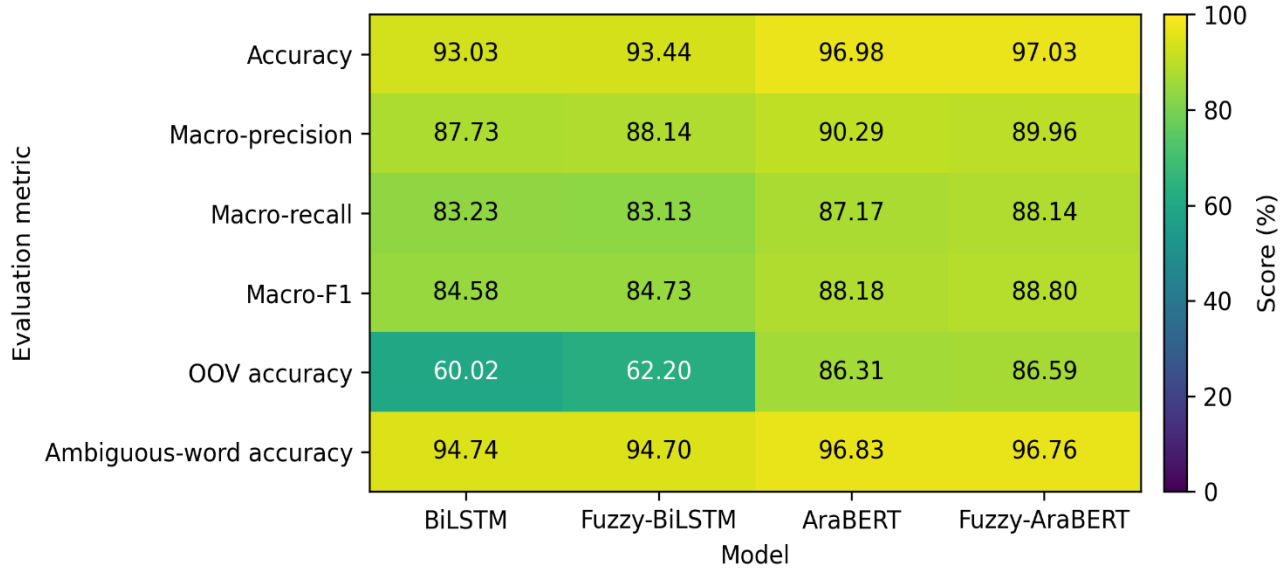
The mean and sample standard deviation for the three runs can be found in Table 1. Transformer models perform better than recurrent models in all six prediction measures. The fuzzy-AraBERT model performs best in mean accuracy, macro-recall, macro-F1 score, and out-of-vocabulary accuracy, while the traditional AraBERT model performs best in macro-precision and ambiguous-word accuracy.

Table 1. Classification performance (mean ± sample SD, %).

Metric	BiLSTM	Fuzzy-BiLSTM	AraBERT	Fuzzy-AraBERT
Accuracy	**93.03 ± 0.05**	**93.44 ± 0.17**	**96.98 ± 0.05**	**97.03 ± 0.01**
Macro-precision	**87.73 ± 0.08**	**88.14 ± 0.21**	**90.29 ± 0.38**	**89.96 ± 0.77**
Macro-recall	**83.23 ± 0.24**	**83.13 ± 0.15**	**87.17 ± 0.34**	**88.14 ± 0.74**
Macro-F1	**84.58 ± 0.15**	**84.73 ± 0.09**	**88.18 ± 0.31**	**88.80 ± 0.48**
OOV accuracy	**60.02 ± 0.22**	**62.20 ± 0.46**	**86.31 ± 0.36**	**86.59 ± 0.16**
Ambiguous-word accuracy	**94.74 ± 0.12**	**94.70 ± 0.14**	**96.83 ± 0.03**	**96.76 ± 0.05**

Figure 2. Mean classification performance across three seeds.

5.2. Incremental Linguistic Features



The Mean hybrid-minus-baseline differences (pp) are listed in Table 2 below. The performance of Fuzzy-BiLSTM increases in terms of accuracy by 0.41 pp, macro-F1 by 0.15 pp, and OOV accuracy by 2.18 pp; however, macro-

recall and ambiguity word accuracy decrease slightly. In contrast, Fuzzy-AraBERT increases macro-recall by 0.97 pp and macro-F1 by 0.62 pp while showing a slight increase in accuracy of 0.05 pp; its macro-precision decreases by 0.33 pp.

Table 2. Mean hybrid-minus-baseline differences (pp).

Metric	Fuzzy-BiLSTM	Fuzzy-AraBERT
Accuracy	+0.412	+0.047
Macro-precision	+0.407	-0.330
Macro-recall	-0.104	+0.969
Macro-F1	+0.150	+0.624
OOV accuracy	+2.176	+0.277
Ambiguous-word accuracy	**-0.039**	**-0.061**

The precision on the ambiguous words decreases by 0.039% and 0.061%, respectively. Therefore, the desired ambiguity resolution benefit is not confirmed by the average performance. This may be due to the high base precision, insufficient use of the rules, and unweighted feature combination.

5.3. Class-level Performance

Figure 3 provides the Mean F1 by universal POS category. The INTJ gets no test coverage. The Fuzzy-BiLSTM model exhibits its maximum class-specific F1 improvements for NUM (+1.60 pp), VERB (+1.45 pp), and X (+0.98 pp). The maximum decreases are observed for SYM (-0.84 pp), AUX (-0.76 pp). Rare classes demand careful consideration since only a few predictions can significantly affect their F1. The Fuzzy-AraBERT model exhibits its maximum class-specific F1 improvements for PROPN (+9.84 pp), AUX (+0.44 pp), and ADJ (+0.31 pp). The maximum decreases are observed for ADV (-0.31 pp), PART (-0.12 pp). Rare classes demand careful consideration since only a few predictions can significantly affect their F1. The score of the INTJ is zero due to no test data. The inclusion of this label in the macro average decreases the absolute value of the macro-F1 reported but makes comparisons fair since all the models use the same label set.

5.4. Statistical Validation

In both intervals, no value of zero is included for seed 42, indicating that macro-F1 values for the fuzzy models are significantly higher than those of the base model, conditioned on that particular training model pair and testing set, as indicated in Table 3.

Nonetheless, with respect to all three seeds, the Fuzzy-BiLSTM macro-F1 value is slightly lower than zero for seed 43, whereas Fuzzy-AraBERT performs better in all three runs. Three seeds are too few to make a claim about training variability. Bootstrap intervals do not indicate significance across seeds.

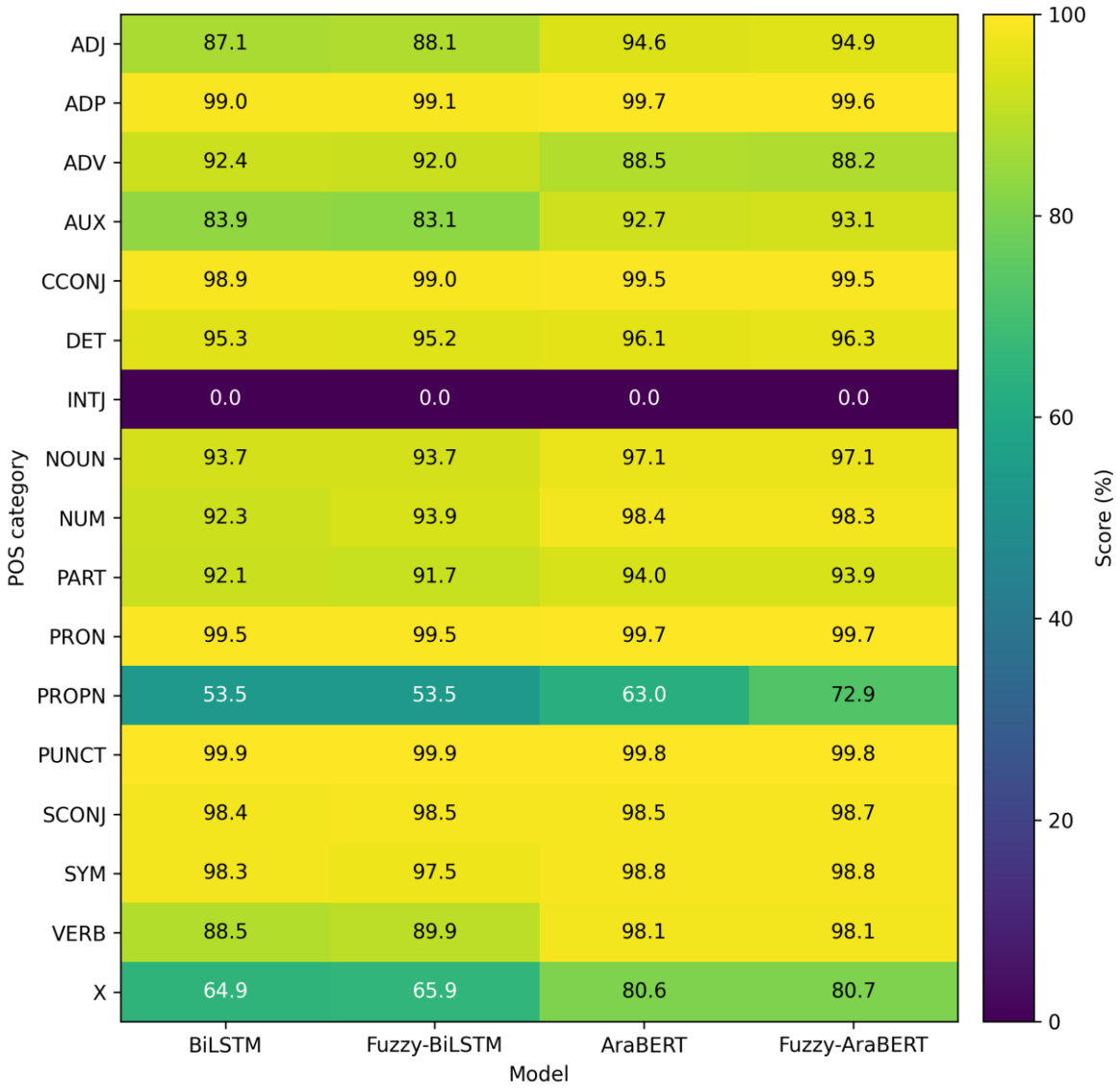


Figure 3. Mean F1 by universal POS category. INTJ has zero test support.

Table 3. Archived paired sentence-bootstrap intervals for macro-F1 differences, seed 42.

Hybrid	**Difference (pp)**	**95% interval (pp)**
Fuzzy-BiLSTM	0.354	[0.060, 0.662]
Fuzzy-AraBERT	1.507	[0.451, 2.619]

5.5. Computational Efficiency

The computational costs are depicted in Figure 4, where the metrics are normalized individually, and the original units are shown in the annotations. Every hybrid has 289 parameters. The fuzzy AraBERT is 23.6 times

slower to train than Fuzzy-BiLSTM, and it has 75.9 times more parameters. It has a 20.7% increase in mean training time compared to AraBERT. The runtime evaluation is dependent on early stopping, number of completed epochs, and GPU variance; it does not consider the fuzzy inference overhead.

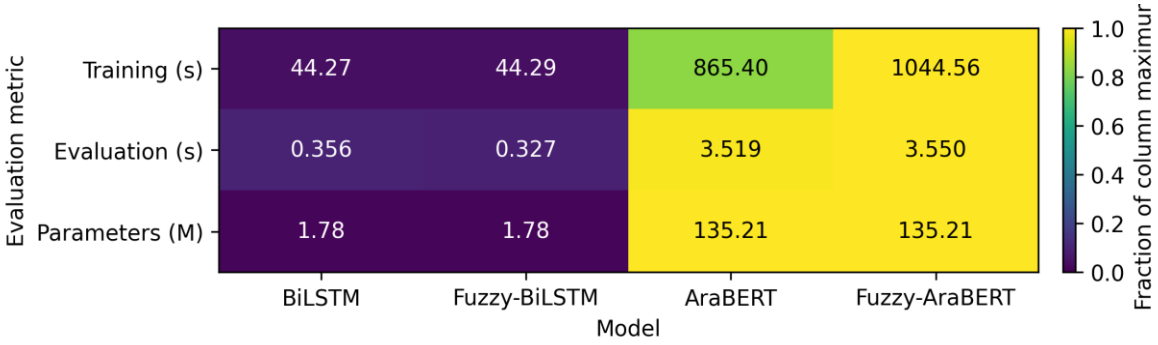


Figure 4. Computational costs, normalized within each metric; annotations show original units.

6. Conclusion and Future Work

The current study explored the use of fuzzy feature integration in conjunction with contextual deep learning for Arabic part-of-speech tagging. For three separate experiments conducted using Arabic-PADT, Fuzzy-BiLSTM produced an accuracy of 93.44% and a macro-F1 score of 84.73%, while Fuzzy-AraBERT yielded an accuracy of 97.03% and a macro-F1 of 88.80%. The recurrent hybrid yielded a greater overall increment in OOV performance (2.18 pp) compared to the transformer hybrid, which produced the best OOV accuracy (86.59%) and macro-F1 increment (0.62 pp) overall. Both of these models were built using an extra 289 neural units. Yet none of them managed to improve average ambiguous-word accuracy in relation to their respective baselines. It is clear, then, that these results highlight the promise of integrating linguistic features into a lightweight model for classification.

This paper makes a contribution through a common linguistic feature fusion framework for two context-based architectures, a benchmarking study using controlled conditions on three random seeds, and the empirical study of linguistic generalization and computation tradeoffs. This work also reveals one negative contribution: graded linguistic features are not automatically better for ambiguity resolution. The lack of a control for non-fuzzy linguistic features means that the improvement can't be definitively credited to fuzzy reasoning. The analysis is limited to one treebank, and the detailed rule definition and preprocessing procedures must be used for replication.

For future research, comparison with other linguistic forms such as crisp and probabilistic, performing ablation studies, performing membership function sensitivity studies, or performing adaptive rule weighting and feature gating may be considered. Performing experiments on independent Arabic genres and dialects, using different random seeds, and performing uncertainty analysis will prove the generalization capability of the model beyond its current benchmark. Performing an error analysis of rare parts-of-speech forms and ambiguous forms, along with measuring the end-to-end latency and memory requirements of each model architecture, will help us assess their utility in practice.

Acknowledgment:
The research leading to these results has received no Research Grant Funding.

Author contribution: All authors have contributed, read, and agreed to the published version of the manuscript results.

Conflict of interest: The authors declare no conflict of interest.

References

[1]. Akra, D., et al. (2025). Active learning for Arabic dialect part-of-speech tagging. https://doi.org/10.18653/v1/2025.findings-emnlp.1359
[2]. Alhasan, A., & Al-Taani, A. T. (2018). POS tagging for Arabic text using bee colony algorithm. Procedia Computer Science, 142, 158–165. https://doi.org/10.1016/j.procs.2018.10.471
[3]. Al-Taani, A. T., & ALkhazaaleh, F. A. (2022). Part-of-speech tagging for Arabic text using particle swarm optimization and genetic algorithm. Recent Advances in Computer Science and Communications, 15(5), Article e060422190303. https://doi.org/10.2174/2666255814666210114120558
[4]. Antoun, W., Baly, F., & Hajj, H. (2020). AraBERT: Transformer-based model for Arabic language understanding. In Proceedings of OSACT 2020 (pp. 9–15). https://aclanthology.org/2020.osact-1.2/
[5]. Ben Ali, B., & Jarray, F. (2013). Genetic approach for Arabic part of speech tagging. International Journal of Natural Language Computing, 2(3). https://doi.org/10.5121/ijnlc.2013.2301
[6]. Bouzahir, M., Ait Abdelouahad, A., & Nabil, M. (2022). How far can deep learning improve Arabic part of speech tagging? In Business intelligence (Vol. 449, pp. 206–215). Springer. https://doi.org/10.1007/978-3-031-06458-6_17
[7]. Bouzahir, M., Ait Abdelouahad, A., & Nabil, M. (2026). Jawhar: Optimized morphological analysis and contextual reranking for Arabic part-of-speech tagging. Information, 17(9), Article 874. https://doi.org/10.3390/info17090874
[8]. Dai, Y., Chakraborty, B., & Shi, M. (2011). Kansei engineering and soft computing: Theory and practice. IGI Global. https://doi.org/10.4018/978-1-61692-797-4
[9]. Darwish, K., et al. (2020). Effective multi-dialectal Arabic POS tagging. Natural Language Engineering, 26(6), 677–690. https://doi.org/10.1017/S1351324920000078
[10]. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT 2019 (pp. 4171–4186). https://doi.org/10.18653/v1/N19-1423
[11]. Hadni, M., Alaoui Ouatik, S., Lachkar, A., & Meknassi, M. (2013). Hybrid part-of-speech tagger for non-vocalized Arabic text. International Journal of Natural Language Computing, 2(6). https://doi.org/10.5121/ijnlc.2013.2601
[12]. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
[13]. Inoue, G., Khalifa, S., & Habash, N. (2022). Morphosyntactic tagging with pre-trained language models for Arabic and its dialects. In Findings of the Association for Computational Linguistics: ACL 2022 (pp. 1708–1719). https://doi.org/10.18653/v1/2022.findings-acl.135
[14]. Jarrar, M., Akra, D., & Hammouda, T. (2024). Alma: Fast lemmatizer and POS tagger for Arabic. Procedia Computer Science, 244, 378–387. https://doi.org/10.1016/j.procs.2024.10.212
[15]. Kallas, O., Inoue, G., & Habash, N. (2024). EMAD: A bridge tagset for unifying Arabic POS annotations. In Proceedings of LREC-COLING 2024 (pp. 5637–5643).
[16]. Khoja, S. (2001). APT: Arabic part-of-speech tagger. In Proceedings of the NAACL Student Workshop.
[17]. Mahdjoubi, R., & Laskri, M. T. (2025). Tackling Arabic NLP challenges: POS-tagging with transformer-based models and nuanced evaluation. International Journal of Performability Engineering, 21(8), 411–421. https://doi.org/10.23940/ijpe.25.08.p1.411421
[18]. Plank, B., Søgaard, A., & Goldberg, Y. (2016). Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In Proceedings of ACL 2016 (Short Papers) (pp. 412–418). https://doi.org/10.18653/v1/P16-2067
[19]. Smrž, O., Bielický, V., Kouřilová, I., Kráčmar, J., Hajič, J., & Zemánek, P. (2008). Prague Arabic Dependency Treebank: A word on the million words. In Proceedings of the Workshop on Arabic and Local Languages (LREC 2008) (pp. 16–23).

[20]. Tnaji, K., Bouzoubaa, K., & Aouragh, S. L. (2021). A light Arabic POS tagger using a hybrid approach. In Digital technologies and applications (Vol. 211, pp. 199–208). Springer. https://doi.org/10.1007/978-3-030-73882-2_19
[21]. Universal Dependencies. (n.d.). UD Arabic PADT (Release 2.17) [Data set]. https://universaldependencies.org/treebanks/ar_padt/
[22]. Yousif, J. (2019). Hidden Markov model tagger for applications based Arabic text: A review. Journal of Computation and Applied Sciences IJOCAAS, 7(1).
[23]. Yousif, J. H., & Sembok, T. (2006a). Design and implement an automatic neural tagger based Arabic language for NLP applications. Asian Journal of Information Technology, 5(7), 784–789. https://doi.org/10.3923/ajit.2006.784.789
[24]. Yousif, J. H., & Sembok, T. (2006b). Recurrent neural approach based Arabic part-of-speech tagging. In Proceedings of ICCCE 2006.
[25]. Yousif, J. H., & Sembok, T. (2008). Arabic part-of-speech tagger based support vectors machines. In Proceedings of the International Symposium on Information Technology.
[26]. Yousif, J. H. (2013). Natural language processing based soft computing techniques. International Journal of Computer Applications, 77(8).
[27]. Yousif, J. H., & Yousif, M. J. (2026). Critical Review of Neural Network Generations and Models Design. Preprints. https://doi.org/10.20944/preprints202309.1149.v3
[28]. Zadeh, L. A. (1965). Fuzzy sets. Information and Control, 8(3), 338–353. https://doi.org/10.1016/S0019-9958(65)90241-X
[29]. Uraibi, H. S., Midi, H., Talib, B. A., & Yousif, J. H. (2009). Linear regression model selection based on robust bootstrapping technique. American Journal of Applied Sciences, 6(6), 1191.

Author(s) and ACAA permit unrestricted use, distribution, and reproduction in any medium, provided the original work with proper citation. This work is licensed under Creative Commons Attribution International License (CC BY 4.0).