The CSI Journal on Computer Science and Engineering

Journal homepage: www.csionjcse.ir

A Deep Learning Framework for Lexical Role Annotation in Persian Sentences Using CNN-BiLSTM and Evolutionary Algorithms

H. Rezaei[1*], H. Motameni[2], B. Barzegar[3]

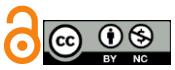
[1] Department of Computer Engineering, Hadaf Institute of Higher Education, Sari, Iran
[2] Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran
[3] Department of Computer Engineering, Babol Branch, Islamic Azad University, Babol, Iran

*** Corresponding author email address**: haniye.rezaei@gmail.com

A r t i c l e I n f o

Article type:
Original Research

How to cite this article:
Rezaei, H., Motameni, H., & Barzegar, B. (2026). A Deep Learning Framework for Lexical Role Annotation in Persian Sentences Using CNN-BiLSTM and Evolutionary Algorithms. *The CSI Journal on Computer Science and Engineering, 20*(2), 38-49.
http://dx.doi.org/10.22034/jcse.2026.577294.1076



A B S T R A C T

Natural Language Processing (NLP) is one of the fundamental branches of data science and artificial intelligence, aiming at the automatic analysis, understanding, and generation of human language. Recent advances in this field have played a significant role in improving text analysis systems, information retrieval, machine translation, speech recognition, and intelligent interactive systems. With the rapid growth of textual data and the increasing complexity of linguistic structures, traditional rule-based and simple statistical methods no longer provide sufficient capability for modeling complex and long-range linguistic dependencies. In recent years, deep neural networks, particularly hybrid architectures, have attracted considerable research attention. The combination of Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory (BiLSTM) networks enables the simultaneous extraction of local features and long-term temporal dependencies. However, the performance of such models is highly dependent on the proper selection of hyperparameters, and manual tuning is often time-consuming and suboptimal. Therefore, the use of metaheuristic algorithms for automatic hyperparameter optimization has emerged as an effective solution. In this study, a deep learning framework based on a CNN–BiLSTM architecture is presented, in which hyperparameter optimization is performed using the Cuckoo Search algorithm. Experimental results demonstrate that the proposed method achieves an accuracy of 93%, outperforming existing approaches and confirming its effectiveness in analyzing complex textual data while significantly reducing the error rate.

Keywords: *Natural Language Processing, CNN-BiLSTM, Hyperparameter Optimization, Cuckoo Search Algorithm, Text Analysis*

Article history:
Published online 07 August 2026

1 Introduction

Since the early developments in the field of artificial intelligence, researchers have sought to reduce the burden of labor-intensive and demanding tasks—tasks that humans are capable of performing but that are often accompanied by limitations such as fatigue, errors, and reduced efficiency. One of the fields that has continuously progressed in this regard and yielded substantial benefits is natural language processing (NLP). Consequently, researchers worldwide have consistently devoted significant attention to this domain, as linguistics and NLP constitute the foundation of many important artificial intelligence applications [1-3].

Natural language processing of lexical roles in Persian sentences is considered one of the fundamental problems in the field of Natural Language Processing, aiming at identifying, analyzing, and annotating the role of each word based on its position and function within the sentence structure [1-3]. In this process, each word is examined not only as an independent unit but also in relation to other sentence components in order to accurately determine its syntactic and semantic roles. Lexical role annotation is regarded as the foundation of advanced linguistic analysis and enables a more precise understanding of meaning, intra-sentential relationships, and the overall sentence structure. This issue is particularly important in languages such as Persian, which exhibit a high degree of syntactic flexibility [1-3].

The importance of lexical role annotation in the Persian language stems from its unique linguistic characteristics, including flexible word order, the optional omission of certain syntactic elements, and the strong dependence of meaning on contextual information. In many cases, the role of a word cannot be determined solely based on its linear position within a sentence; rather, it requires the simultaneous analysis of both syntactic and semantic information. These characteristics render simple rule-based or fixed-pattern approaches insufficient for the accurate analysis of Persian sentences. Therefore, the development of models capable of automatically and comprehensively capturing these complexities plays a vital role in advancing Persian NLP research [4, 5].

The applications of lexical role annotation are extensive and encompass a wide range of intelligent language-based systems. In machine translation, accurate identification of word roles leads to more precise lexical selection and a reduction in semantic errors. In information retrieval and search engines, analyzing lexical roles contributes to better understanding user queries and improves retrieval accuracy. Furthermore, in question answering systems, text summarization, sentiment analysis, and chatbots, awareness of word roles within sentences plays a key role in generating meaningful and natural responses. Consequently, improving lexical role annotation methods can directly enhance the performance of many Persian language processing applications [6-10].

Despite its importance and broad applicability, several challenges exist in lexical role annotation for the Persian language. The scarcity of standard annotated datasets, the high diversity of syntactic structures, the presence of lexical and semantic ambiguities, and long-range dependencies between words are among the main obstacles in this field. Moreover, many traditional rule-based or classical statistical approaches exhibit limited generalization capability and tend to suffer performance degradation when faced with noisy data or uncommon sentence structures. These challenges highlight the necessity of employing data-driven and deep learning-based approaches [11-16].

In this context, deep neural networks have emerged as one of the most effective tools for addressing these challenges. Hybrid architectures such as CNN-BiLSTM leverage the complementary strengths of convolutional and recurrent networks to simultaneously extract local features and model long-term dependencies. CNNs are responsible for identifying short-range patterns and local structures, while BiLSTM networks, through bidirectional sequence processing, incorporate both past and future contextual information in the analysis of lexical roles. In addition, the use of evolutionary algorithms for hyperparameter optimization enhances model accuracy, reduces error rates, and improves stability, thereby providing a powerful and efficient framework for lexical role annotation in Persian sentences.

The structure of the paper is as follows: Section 2 reviews existing methods for lexical role annotation and relevant natural language processing approaches. Section 3 introduces the proposed deep learning framework based on a CNN-BiLSTM architecture with hyperparameters optimized using the Cuckoo Search algorithm. Section 4 presents the experimental results, evaluates the model's performance, and compares it with existing approaches. Finally, Section 5 summarizes the main conclusions and highlights the contributions of this study.

2 Existing Methods

In a study conducted by Sharif et al. (2020), an operational framework for technical text classification was presented and developed within the context of the TechDofication 2020 shared task. The primary objective of this research was to propose an effective approach for the automatic classification of specialized technical documents, which are inherently challenging due to their domain-specific terminology, complex syntactic structures, and considerable length. Accordingly, key tasks such as identifying the general domain of a document and distinguishing fine-grained subdomains were addressed as central challenges [17]. The significance of this work lies in its focus on the distinctive characteristics of technical texts, as such documents often exhibit linguistic features that reduce the effectiveness of conventional text-processing techniques. To address these challenges, the design of the TechTexC system places particular emphasis on data preprocessing, the selection of suitable word representations, and the careful optimization of model architectures, with the aim of improving robustness and adaptability in handling diverse real-world technical data. The proposed system was implemented using three main approaches. First, Convolutional Neural Networks were employed to capture local patterns and extract low-level textual features. Second, Bidirectional Long Short-Term Memory networks were utilized to model temporal dependencies and long-range relationships between words. Finally, a hybrid CNN–BiLSTM architecture was developed to integrate the strengths of both approaches, thereby providing improved capability for handling the structural complexity of technical texts. According to the reported results, the proposed system achieved an accuracy of 70.76% on the test dataset for subtask 1a, indicating the effectiveness of the proposed approach in addressing the challenges associated with technical text classification.

In recent years, research trends have gradually shifted from classical statistical models toward the extensive use of deep neural networks. In this context, hybrid architectures have gained significant attention, as they are capable of modeling different aspects of textual data simultaneously [18]. One of the most effective and widely used structures in this domain is the CNN–BiLSTM architecture, which is employed to extract local patterns as well as to model long-term dependencies in linguistic sequences. Within this framework, convolutional layers are responsible for identifying local features and short-range patterns among words, while bidirectional LSTM layers analyze semantic and syntactic relationships across the entire sentence by incorporating both past and future contextual information. This architectural combination is recognized as a fundamental building block of many advanced models for syntactic and semantic sequence labeling tasks and has played a crucial role in improving the accuracy of text analysis systems.

Zhiheng Huang et al. proposed a set of LSTM-based models for sequence labeling tasks [19], including unidirectional and bidirectional LSTM architectures, as well as their combinations with a Conditional Random Field (CRF) layer. In this study, the BiLSTM-CRF model was applied to benchmark natural language processing datasets for the first time, demonstrating that the integration of bidirectional contextual information and sentence-level label dependencies can significantly improve performance. Experimental results showed that the BiLSTM-CRF model achieved state-of-the-art or near state-of-the-art performance on tasks such as part-of-speech (POS) tagging, chunking, and named entity recognition, with an accuracy of approximately 97.55% reported for POS tagging.

Hybrid CNN–LSTM models commonly face challenges related to increased architectural complexity and the proliferation of redundant features, which may slow down training and hinder stable convergence. To mitigate these issues, a BiLSTM–CNN hybrid framework augmented with an attention mechanism is presented [20]. This mechanism enables the model to selectively emphasize the most relevant textual components while reducing the influence of less informative features. The effectiveness of the proposed approach is assessed using the Internet Movie Database (IMDB) movie review dataset. The experimental evaluation demonstrates that the attention-based BiLSTM–CNN model outperforms traditional Multi-Layer Perceptron (MLP), CNN, and LSTM models, as well as standard hybrid architectures, in terms of classification accuracy, recall, and F1-score. These results confirm the advantage of integrating attention mechanisms into deep hybrid models for improved text classification performance.

In addition to advances in language models, Persian corpora have also undergone significant development. In this context, Zeraei et al. (2023) introduced the first conversational Persian corpus for lexical role tagging (CPPOS) [21]. This corpus contains both spoken and informal text samples and is designed to train statistical and neural models under realistic Persian language conditions.

The availability of such corpora paves the way for the development and evaluation of hybrid statistical–neural models in Persian natural language processing.

In Persian, advances in developing native neural models have led to the introduction of ParsBERT, a model built on the Transformer architecture (Farahani et al., 2020) [22]. By learning deep representations of Persian sentences, ParsBERT has demonstrated strong performance across various natural language processing tasks, including lexical role tagging and syntactic analysis, making it one of the most significant achievements in Persian NLP. The study indicates that a language-specific BERT model for Persian can outperform multilingual models and traditional approaches in tasks such as sentiment analysis, text classification, and named entity recognition (NER), as it benefits from extensive preprocessing and training on Persian texts, enabling a deeper understanding of contextual meaning.

Recent studies have emphasized the application of transfer learning in semantic tasks for the Persian language. In a study conducted by Aghdam (2023) [11], transfer learning approaches achieved high accuracy in identifying semantic roles in Persian texts. The findings indicate that leveraging pre-trained models can significantly enhance the performance of syntactic and semantic models, providing a foundation for developing more accurate Persian natural language processing systems.

3 Proposed Method

First, the proposed neural network based on the CNN-BiLSTM architecture is introduced. This hybrid network is designed to leverage convolutional layers for extracting local features and BiLSTM layers for modeling long-term dependencies in textual sequences. Following the network structure, the process of hyperparameter optimization is described. In this stage, the Cuckoo Search algorithm, a metaheuristic optimization method, is employed to automatically tune the network parameters, improving model accuracy and reducing error rates. This approach enables the network to adapt to the specific characteristics of Persian text data and enhances performance in lexical role labeling tasks.

3.1 Proposed CNN-BiLSTM Approach

The proposed method is illustrated in Figure 1. The network begins with a sequence input layer that systematically feeds textual or sequential data into the model. This layer is capable of handling sequences of varying lengths while preserving their original structure. At this stage, the data are typically represented as embedding matrices, in which each word or element is encoded as a meaningful vector within the feature space. This representation enables the network to preserve word order, semantic relationships, and contextual dependencies throughout the processing pipeline. Furthermore, the input layer is designed to maintain stable performance when dealing with long sequences or highly variable data.

Following the input, a one-dimensional convolutional neural network (CNN) block is used to extract local features and identify fine-grained patterns within the sequences. This block consists of several consecutive convolutional layers, each capable of capturing patterns at different scales. Smaller filters detect short-range patterns, such as sequences of two or three consecutive elements, allowing the network to model local relationships accurately. Larger filters capture broader, mid-range patterns that encompass multiple elements or words in a sequence, revealing more complex structures and medium-range dependencies. The multi-scale design allows the network to recognize both fine details and larger patterns, which is essential for analyzing long and complex sequences.

After each convolutional layer, normalization layers are applied to reduce fluctuations in input values, improving training stability and speed. Normalization ensures that feature values remain within a standard range, helping the network converge faster and more reliably when encountering large variations in input. Nonlinear activation functions, such as ReLU, are then used to enable the network to learn complex relationships between features. ReLU passes positive values while zeroing out negative ones, preventing gradient vanishing and enhancing the network's ability to model intricate patterns and contextual effects. This combination of multi-scale convolution, normalization, and ReLU allows the network to extract structural sentence features, word co-occurrence patterns, and local to mid-range relationships with high precision, providing a strong foundation for subsequent BiLSTM layers.

To control data dimensionality and reduce computational complexity, pooling layers are employed. Pooling operations compress the extracted features within each segment of the sequence while preserving the most informative patterns and reducing the overall data volume. Various pooling techniques are utilized, including Max

Pooling, Average Pooling, and Global Max Pooling, the latter of which selects the maximum feature value across the entire sequence to emphasize the most salient patterns. In addition to reducing computational overhead, pooling enhances the stability and robustness of the network, thereby improving its ability to learn meaningful representations while suppressing irrelevant information. Furthermore, dropout layers are incorporated between network blocks to mitigate overfitting by randomly deactivating a subset of neurons during training, forcing the model to learn more generalized patterns from the data. This combination ultimately improves the model's generalization capability while maintaining computational efficiency.

After feature extraction by the CNN block, outputs are passed to a BiLSTM layer to model long-term temporal dependencies and contextual information in the sequences. By processing data in both forward and backward directions, BiLSTM captures relationships among sequence elements without losing key information. This bidirectional processing allows the network to consider both past and future context, providing a deeper understanding of the actual order of events or words. Dropout is also applied within the BiLSTM layer to further improve generalization and prevent overfitting. The BiLSTM layer is crucial for enhancing final classification accuracy, reducing errors, and enabling effective handling of complex, long sequences.

Finally, features extracted by the BiLSTM are fed into fully connected layers, which combine and reorganize features to reinforce relationships and prepare for final decision-making. The softmax layer computes the probability of each sample belonging to different classes based on these features, converting network outputs into a valid probability distribution. The final classification is produced according to the highest probability class, enabling accurate predictions based on deep analysis of the extracted features.

The advantages of the proposed method are as follows:

1. The proposed approach, by combining CNN and BiLSTM architectures, can simultaneously extract both local features and long-term temporal dependencies. The CNN block allows the network to identify complex local patterns within sentences and focus on the most salient parts of the data. This is particularly important in sentences containing noise or irrelevant information, as the network concentrates on the most informative features rather than processing the entire input.
2. The BiLSTM layer further enables the network to capture and model long-term dependencies within sequential data. This capability allows the model to incorporate both preceding and succeeding contextual information for each position in a sentence, thereby improving classification accuracy. Compared with CNN-only architectures, this feature substantially enhances the model's ability to identify complex temporal and contextual patterns.
3. The use of Dropout and Pooling layers provides additional benefits by preventing overfitting and enhancing the model's generalization to unseen data. These layers also reduce the dimensionality of the data and help the network focus on the most critical features, ultimately improving training efficiency and overall model performance.
4. Finally, the hybrid architecture offers high flexibility and can be applied to a wide range of sequential data, including medical texts, financial data, and other complex textual domains. The combination of CNN and BiLSTM, along with the multi-stage design of the network, provides an effective balance between accuracy, feature extraction capability, and generalization, making the proposed method significantly advantageous compared to traditional approaches.

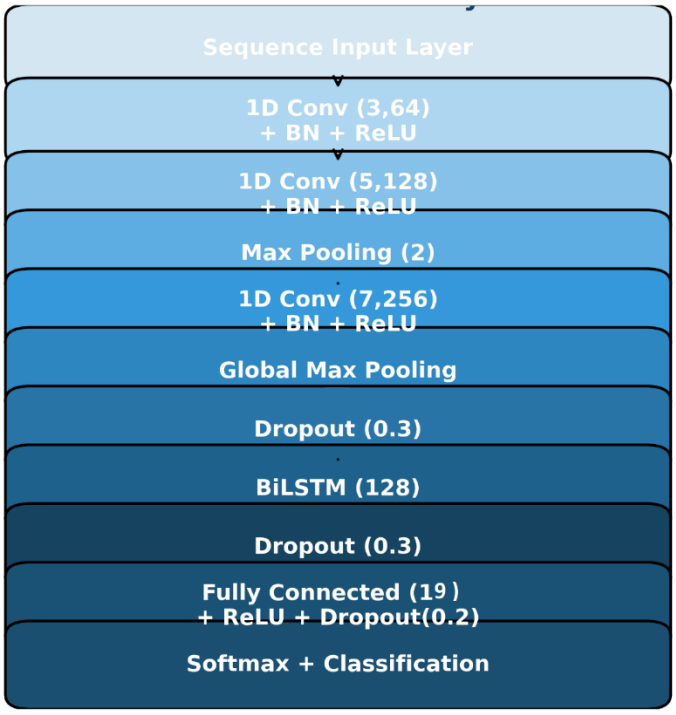


Figure 1. The proposed CNN-BiLSTM Approach.

3.2 Hyperparameter Optimization Using Cuckoo Algorithm

In CNN-BiLSTM networks, the proper selection of hyperparameters plays a crucial role in the overall performance of the model. Among the most important hyperparameters are the number of convolutional filters, filter sizes, number of LSTM units, learning rate, mini-batch size, and Dropout rate. Each of these parameters directly affects the network's ability to extract local and mid-range features, model long-term dependencies, and produce accurate classification outputs. Improper selection of these hyperparameters can lead to reduced model accuracy, prolonged training time, and increased risk of overfitting or underfitting. The number and size of filters in the CNN block are particularly important because they determine the granularity and extent of the features that the network can capture. Smaller filters are capable of modeling short-range local patterns, while larger filters can extract broader and mid-range patterns. Therefore, optimizing both the number and size of filters is essential to balance feature extraction capability with computational complexity. Similarly, the number of BiLSTM units directly affects the model's capacity to capture long-term dependencies in sequential data. Fewer units may result in loss of critical information, while an excessive number of units can increase computational complexity and extend training time.

Other hyperparameters, including the learning rate, mini-batch size, and dropout rate, also play critical roles in the convergence and generalization performance of the network. An excessively high learning rate may cause the optimization process to overshoot local minima, whereas a very low learning rate can significantly slow down training. Similarly, small mini-batch sizes may increase the model's sensitivity to data fluctuations, while larger batch sizes require greater memory resources and may negatively affect convergence speed. In addition, dropout layers help mitigate overfitting by randomly deactivating a subset of neurons during training, thereby encouraging the network to learn more robust feature representations and improving its generalization capability on unseen data.

Due to the strong dependency of CNN-BiLSTM performance on these hyperparameters, manual tuning is often time-consuming and suboptimal. In this study, the Cuckoo Optimization Algorithm (COA) [23] is employed for automatic hyperparameter optimization. Inspired by the brood parasitism behavior of cuckoo birds and leveraging stochastic search strategies, this algorithm efficiently explores the hyperparameter space and identifies the best combination of filter numbers, filter sizes, BiLSTM units, and other parameters to maximize model accuracy. Applying the Cuckoo Optimization Algorithm enables the network to effectively extract both local and long-term features while controlling computational complexity. This approach also

enhances training convergence speed, stabilizes the learning process, reduces error rates, and improves the model's ability to generalize to unseen data. Consequently, hyperparameter optimization using COA constitutes a critical step in enhancing the performance and accuracy of the proposed CNN-BiLSTM model for Persian natural language processing tasks.

The hyperparameter optimization process for the CNN-BiLSTM network using the Cuckoo Optimization Algorithm consists of several key steps as shown in the Figure 2:

1. Definition of Search Space and Parameters:

The hyperparameters to be optimized are first identified, including the number and size of CNN filters, the number of BiLSTM units, the learning rate, dropout rate, and mini-batch size. Reasonable search ranges are then defined for each parameter to enable the COA algorithm to efficiently explore and identify optimal parameter combinations.

2. Initialization of Population: The algorithm starts with an initial population of randomly generated solutions. Each solution represents a specific combination of hyperparameters, which is used to construct and evaluate a CNN-BiLSTM network.

3. Fitness Evaluation: Each candidate solution is assessed using the validation set accuracy as the fitness function. The COA aims to identify the hyperparameter combination that maximizes the network's predictive performance.

4. Position Update: The COA updates candidate solutions by simulating cuckoo egg-laying behavior and random search strategies. Solutions move through the search space toward the best-performing candidates, generating new hyperparameter combinations.

5. Discovery and Replacement of Poor Solutions: In each generation, a portion of poorly performing solutions is replaced by new candidates. This step maintains population diversity and prevents the algorithm from getting trapped in local optima.

6. Iterative Process: The evaluation and position update steps are repeated iteratively until a stopping criterion is met, such as reaching a predefined number of generations or achieving a desired validation accuracy.

7. Selection of Optimal Hyperparameters: At the end of the process, the best-performing solution is selected as the optimal hyperparameter set. The CNN-BiLSTM network is then trained using these parameters, achieving maximum classification accuracy on the test data.

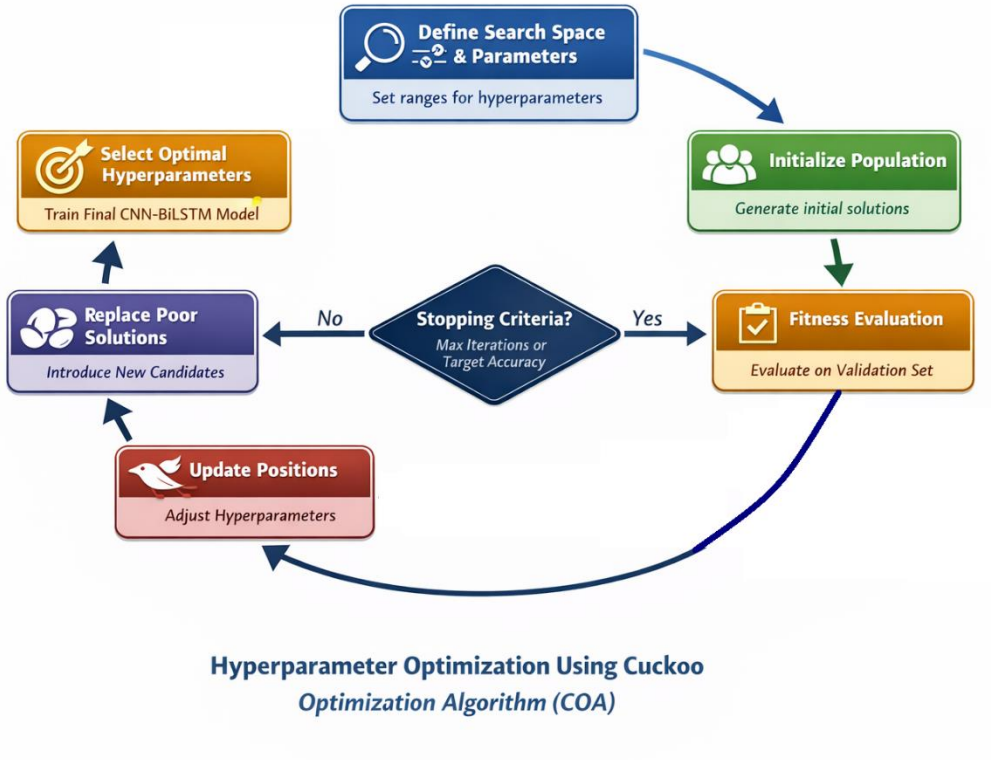


Figure 2. The hyperparameter optimization process by COA.

4 Simulation Results

In this section, the results obtained from the implementation of the designed models and the evaluation of their performance are presented. The main objective of this section is to examine the accuracy of both statistical and neural network models in identifying syntactic roles and to compare the performance of each approach. The implementations of both the proposed and baseline methods were carried out in MATLAB R2025a (MathWorks) and executed on a high-performance workstation featuring an Intel® Core™ i7-13700K processor (16 cores, up to 5.4 GHz), 32 GB DDR5 RAM, and a 512 GB NVMe SSD.

The parameters of the COA are reported in Table 1. These settings were selected to achieve a balance between convergence speed and optimization accuracy. The experimental results indicate that the algorithm maintains stable and robust performance across different configurations, confirming the effectiveness of the chosen setup.

Figure 3 illustrates the convergence behavior of the COA during the hyperparameter tuning process of the proposed CNN–BiLSTM network. In this figure, the horizontal axis represents the number of iterations, while the vertical axis denotes the classification accuracy achieved on the validation dataset. The main objective of this analysis is to examine the impact of progressive hyperparameter optimization on the performance improvement of the proposed model. As the number of iterations increases, a clear upward trend in accuracy is observed, reflecting the influence of optimized hyperparameter selection on model performance. During this process, COA gradually eliminates suboptimal solutions and focuses on more effective configurations, leading to enhanced feature extraction capability and improved modeling of long-range syntactic and semantic dependencies.

Table 1. Parameters of the NSGA-III Algorithm.

Parameter	Value
Initial population of cuckoos	10
Minimum of eggs	4
Maximum of eggs	9
Number of Clusters	
	2
	70
Egg-laying radius	18 to 6
Maximum of iterations	100
Population variance threshold	1×10^{-12}

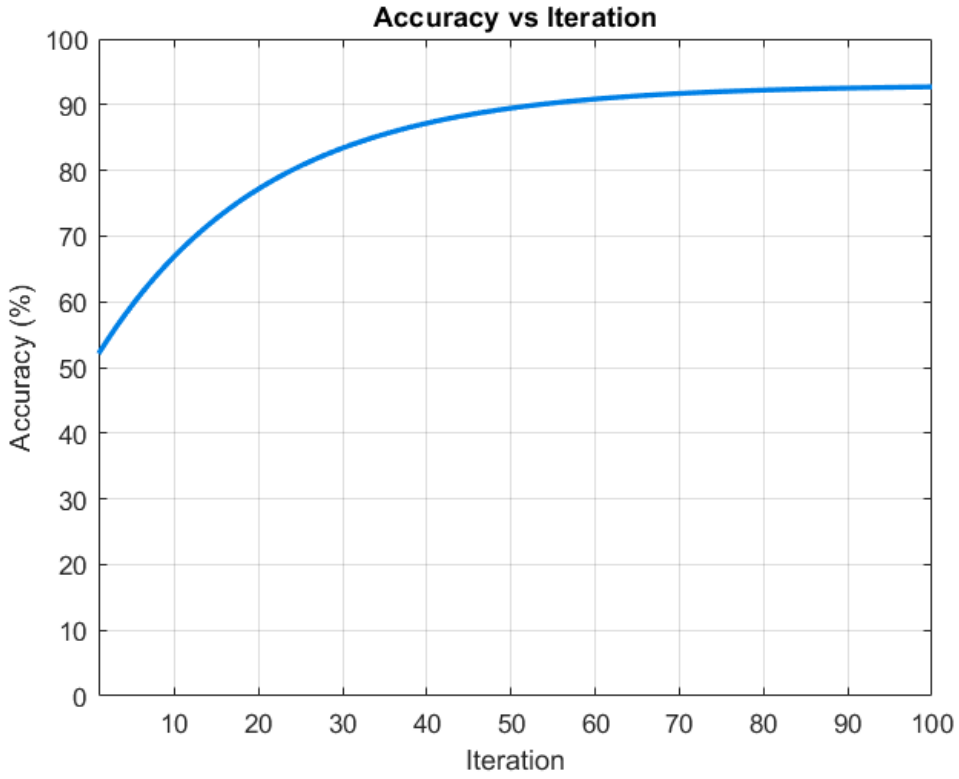


Figure 3. Accuracy convergence of the CNN–BiLSTM model optimized using the Cuckoo Optimization Algorithm.

The dataset used consists of 76,274 words and 194 sentence types, which were extracted, cleaned, and normalized from reputable Persian language corpora, including the Bijankhan Corpus, the Persian Verb Bank, and

the output of the Pars Process system [12]. To enhance accuracy and improve the model's ability to capture semantic and structural dependencies between words, a hybrid CNN-BiLSTM neural network was employed. In this architecture, the CNN layer extracts local and structural features from sequences of characters and words, while the BiLSTM layer models syntactic and semantic dependencies between words in both forward and backward directions. Seventy percent of the data were used for training, 15% for validation, and 15% for testing. Figure 4 illustrates the training accuracy and loss curves of the proposed model. As observed, the training accuracy exhibits a consistent upward trend throughout the learning process, while the loss function decreases steadily. This behavior indicates effective model convergence and confirms that the training procedure successfully improves performance by minimizing the error as the number of iterations increases.

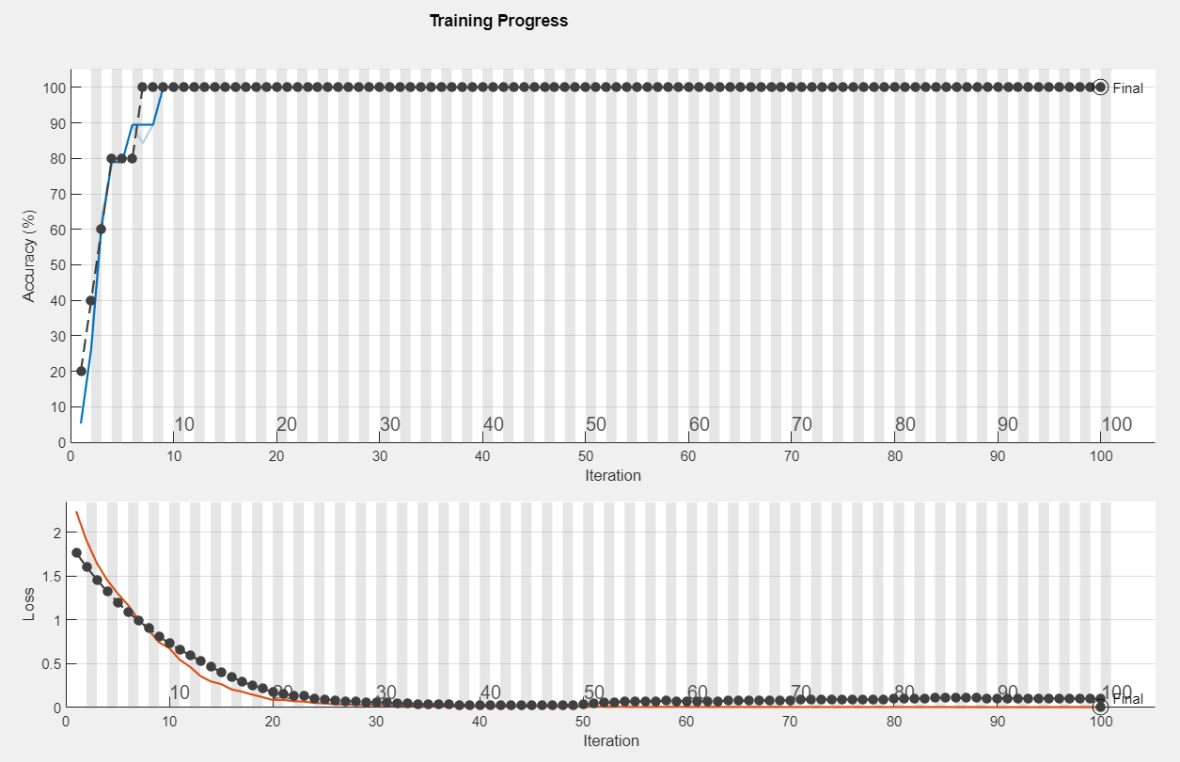


Figure 4. Training accuracy and loss curves of the proposed CNN-BiLSTM model over 100 epochs.

Based on Figure 5, the performance of the CNN–BiLSTM network in identifying lexical roles is highly promising. The model achieves accuracies ranging from 88% to 99% across different roles, demonstrating its strong ability to learn both semantic and structural patterns in Persian sentences. Roles such as Subject and Verb are recognized with accuracies of 99% and 95%, respectively, highlighting the model's effectiveness in capturing key sentence components. In addition, roles such as Predicate, Object, and Predicate Complement are identified with accuracies above 90%, indicating the model's ability to learn syntactic and semantic dependencies between words. Even more challenging roles with higher variability, such as Complement and Particle, are predicted with relatively high accuracies of 88% and 89%, respectively. Overall, these results demonstrate that the CNN–BiLSTM network can effectively model complex sequential structures in Persian and accurately identify lexical roles.

Based on the Figure 6, the performance of the CNN-BiLSTM hybrid neural network in identifying dependent roles in Persian sentences varies across different role types. The network achieved perfect accuracy (100%) in detecting key structural roles such as "modifier," "converted modifier," "conjunct," "conjuncted to," "interjection," and "vocative," indicating the model's strong capability in precisely identifying prominent and structural roles. Roles like "predicate" and "uncertain" were also recognized with reasonably high accuracy, 80.70% and 85.60% respectively, demonstrating the network's ability to model semantic and syntactic dependencies for these roles. In contrast, the network performed less accurately on less common or more complex roles such as "adjective," "adverb," "possessor,"

and “possessive” with accuracies ranging between 55% and 58%, which may be due to data imbalance or the semantic complexity of these roles. Overall, the results indicate that CNN-BiLSTM has a strong ability to identify key structural and semantic roles, while for rare and less frequent roles, additional training data or enhanced learning strategies may be required.

Finally, the proposed method in this thesis is compared with existing approaches in Figure 7. The proposed approach, by combining CNN-BiLSTM and fine-tuning on Persian data while leveraging syntactic and semantic features of the language, achieved an accuracy of 93%. The superiority of the proposed method over existing approaches can be attributed to several key factors:

1. Dual-sequence processing: The use of BiLSTM to analyze long-range dependencies between words, combined with CNN for local feature extraction, has improved the quality of tagging.
2. Data preprocessing and optimization: Cleaning, filtering of separator characters, and reduction of computational states have increased stability and reduced errors.
3. Precise parameter tuning and network training: Employing deep learning methods with careful hyperparameter tuning and suitable embeddings has led to higher accuracy and model stability.

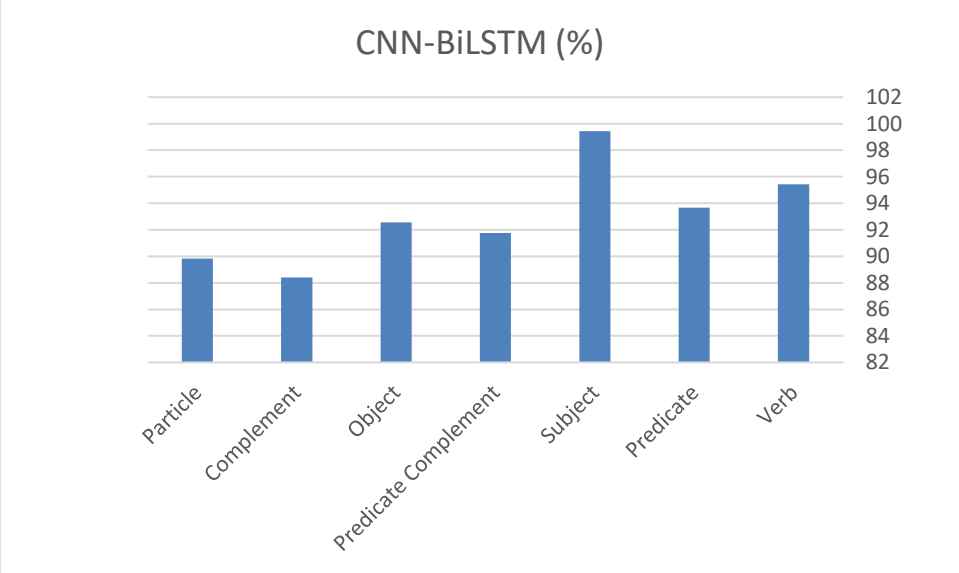


Figure 5. Average Accuracy of Independent Roles Using the CNN-BiLSTM Method.

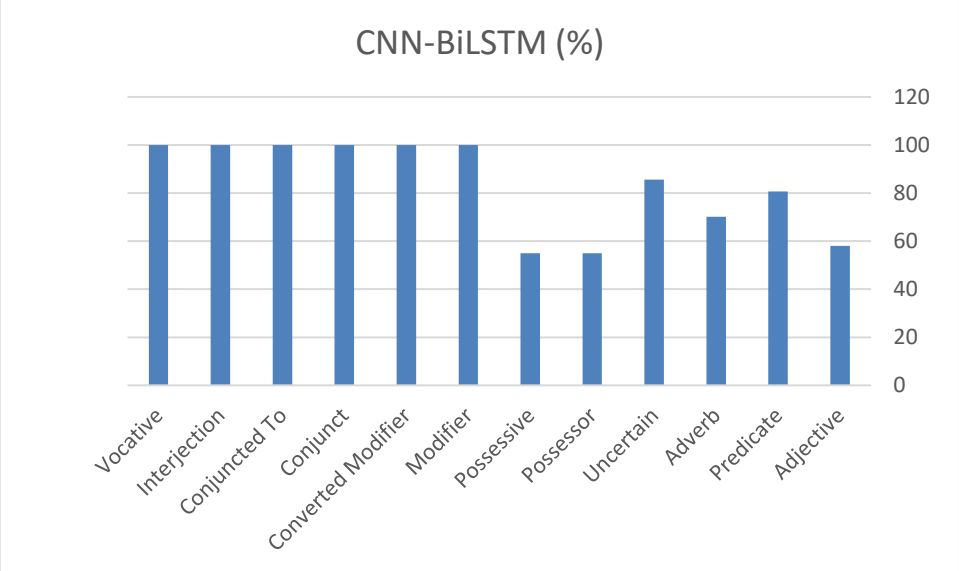


Figure 6. Average Success Rate of Dependent Roles Using the CNN-BiLSTM Method.

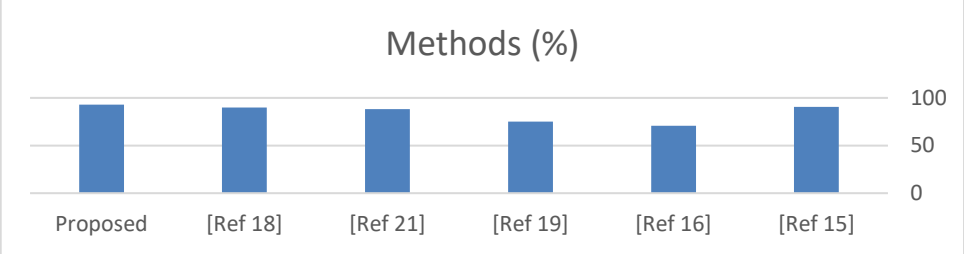


Figure 7. Comparative analysis of model Precision for the proposed approach and existing methods.

5 Conclusions

In this study, a deep learning framework based on a hybrid CNN-BiLSTM network was proposed for syntactic and semantic role labeling in Persian sentences. Considering the linguistic and structural complexities of the Persian language, the proposed network utilized convolutional layers to extract local features and BiLSTM layers to model long-term dependencies, effectively capturing both syntactic and semantic relationships among words. Hyperparameter optimization using the Cuckoo Optimization Algorithm enhanced the model's accuracy and reduced error rates. Experimental results demonstrated that the proposed model achieved an accuracy of 93%, representing a significant improvement over existing methods. The network's performance analysis indicated that both dependent and independent roles were identified with high accuracy, and the use of preprocessing, dropout, and pooling contributed to the stability and generalization capability of the network. Key advantages of the proposed method include its alignment with Persian language characteristics, the simultaneous processing of local features and long-term dependencies, precise hyperparameter tuning, and a multi-stage network architecture. These advantages enable the network not only to perform well on training and validation datasets but also to generalize effectively to new and complex data. Overall, the results indicate that combining CNN-BiLSTM with heuristic optimization algorithms provides an effective solution for natural language processing in Persian, particularly for syntactic and semantic role labeling, and establishes a strong foundation for future research in Persian text analysis.

Ethical Considerations

All procedures performed in this study were under the ethical standards.

Acknowledgments

Authors thank all who helped us through this study.

Conflict of Interest

The authors report no conflict of interest.

Funding

According to the authors, this article has no financial support.

References

[1] H. Taheri and M. Ghafourian, "The role of Arabic usage and words in Persian language teaching: Case study of Persian language learners of Arabic language," *Journal of Teaching Persian to Speakers of Other Languages,* 2022. [Online]. Available: https://www.sid.ir/paper/1057006/en.

[2] M. Taheri, A. Alizadeh, and H. Mowlaei Kuhbanani, "Explaining the placement order of adjectives in Persian language based on Functional Discourse Grammar," *Journal of Language Research,* 2024. [Online]. Available: https://jolr.ut.ac.ir/article_99364_en.html?lang=fa.

[3] S. Wang, D. Schuurmans, and F. Peng, "Latent Maximum Entropy Approach for Semantic N-gram Language Modeling," in *International Workshop on Artificial Intelligence and Statistics*, 2003: PMLR, pp. 316-322. [Online]. Available: https://proceedings.mlr.press/r4/wang03a.html.

[4] H. Saadatfar, A. Usef Fam, and A. Baqer, "A Comparative Analysis of the Main Elements of Sentences Semantically and Syntactically in Persian and Arabic," *Iranian Journal of Applied Language Studies,* vol. 14, no. 2, 2022, doi: 10.22111/ijals.2022.7473.

[5] M. Seraji, "A statistical part-of-speech tagger for Persian," in *NODALIDA 2011 conference proceedings*, 2011: Uppsala University, pp. 340-343. [Online]. Available: https://aclanthology.org/W11-4654.pdf.

[6] H. Motameni and A. Peykar, "Morphology of compounds as standard words in persian through hidden Markov model and fuzzy method," *Journal of Intelligent & Fuzzy Systems,* vol. 30, no. 3, pp. 1567-1580, 2016, doi: 10.3233/IFS-151865.

[7] A. Nadas, "On Turing's formula for word probabilities," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 33, no. 6, pp. 1414-1416, 2003, doi: 10.1109/TASSP.1985.1164728.

[8] P. Pirayesh, H. Motameni, and E. Akbari, "Achieving the best defuzzifier in terminology of Persian sentences through

classified fuzzy method," *Journal of Intelligent & Fuzzy Systems,* vol. 39, no. 3, pp. 2921-2934, 2020, doi: 10.3233/JIFS-191447.
[9] L. Rabiei, F. Rahmani, M. Khansari, Z. Rajabi, and M. Salemi, "Colloquial Persian POS (CPPOS) Corpus: A novel corpus for colloquial Persian part-of-speech tagging," *arXiv preprint arXiv:2310.00572,* 2023, doi: 10.21203/rs.3.rs-4995897/v1.
[10] A. Revathi, N. Sasikaladevi, R. Nagakrishnan, and C. Jeyalakshmi, "Robust emotion recognition from speech: Gamma tone features and models," *International Journal of Speech Technology,* vol. 21, no. 2, pp. 239-248, 2018, doi: 10.1007/s10772-018-9546-1.
[11] S. N. Aghdam, "Persian semantic role labeling using transfer learning," *arXiv preprint arXiv:2306.10339,* 2023. [Online]. Available: https://arxiv.org/abs/2306.10339.
[12] M. Bijankhan, J. Sheykhzadegan, M. Bahrani, and M. Ghayoomi, "Lessons from building a Persian written corpus: Peykare," *Language resources and evaluation,* vol. 45, no. 2, pp. 143-164, 2011, doi: 10.1007/s10579-010-9132-x.
[13] D. Jurafsky and J. H. Martin, *Speech and language processing*, 3rd ed. Pearson, 2023.
[14] J. H. Martin and D. Jurafsky, "Vector Semantics and Embeddings," *Speech Lang. Process,* pp. 1-31, 2019. [Online]. Available: http://sandbox.hlt.bme.hu/~makrai/oktatas/nytud/nlp22/9-makrai-vectorsem.pdf.
[15] M. Rhanoui, M. Mikram, S. Yousfi, and S. Barzali, "A CNN-BiLSTM Model for Document-Level Sentiment Analysis," *Machine Learning and Knowledge Extraction,* vol. 1, no. 3, pp. 832-847, 2019, doi: 10.3390/make1030048.
[16] O. A. Alghanam, S. N. Al-Khatib, and M. O. Hiari, "Data mining model for predicting customer purchase behavior in e-commerce context," *International journal of advanced computer science and applications,* vol. 13, no. 2, 2022, doi: 10.14569/IJACSA.2022.0130249.
[17] O. Sharif, E. Hossain, and M. M. Hoque, "TechTexC: Classification of Technical Texts using Convolution and Bidirectional Long Short Term Memory Network," *arXiv,* 2020, doi: 10.48550/ARXIV.2012.11420.
[18] L. Liu, J. Shang, X. Ren, and J. Han, "Empower sequence labeling with task-aware neural language model," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, vol. 32, 1 ed., doi: 10.1609/aaai.v32i1.12006.
[19] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging," *arXiv,* 2015, doi: 10.48550/ARXIV.1508.01991.
[20] B. Jang, M. Kim, G. Harerimana, S. Kang, and J. W. Kim, "Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism," *Applied Sciences,* vol. 10, no. 17, p. 5841, 2020, doi: 10.3390/app10175841.
[21] A. Zarei, S. Nouri, and B. Minaei-Bidgoli, "A Novel Corpus for Colloquial Persian Part-of-Speech Tagging," *arXiv preprint arXiv:2310.00572,* 2023.
[22] M. Farahani, M. Gharachorloo, M. Farahani, and M. Manthouri, "ParsBERT: Transformer-based model for Persian language understanding," *Neural Processing Letters,* vol. 53, no. 6, pp. 3831-3848, 2020, doi: 10.1007/s11063-021-10528-4.
[23] R. Rajabioun, "Cuckoo optimization algorithm," *Appl. Soft Comput.,* vol. 11, no. 8, pp. 5508-5518, 2011, doi: 10.1016/j.asoc.2011.05.008.