Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.034827
Article

Dart Games Optimizer with Deep Learning-Based Computational Linguistics Named Entity Recognition

Mesfer Al Duhayyim[1,*], Hala J. Alshahrani[2], Khaled Tarmissi[3], Heyam H. Al-Baity[4], Abdullah Mohamed[5], Ishfaq Yaseen[6], Amgad Atta Abdelmageed[6] and Mohamed I. Eldesouki[7]

[1]Department of Computer Science, College of Sciences and Humanities-Aflaj, Prince Sattam bin Abdulaziz University, Al Aflaj, 16828, Saudi Arabia
[2]Department of Applied Linguistics, College of Languages, Princess Nourah Bint Abdulrahman University, P. O. Box 84428, Riyadh, 11671, Saudi Arabia
[3]Department of Computer Sciences, College of Computing and Information System, Umm Al-Qura University, Makkah, 24211, Saudi Arabia
[4]Department of Information Technology, College of Computer and Information Sciences, King Saud University, P. O. Box 145111, Riyadh, 4545, Saudi Arabia
[5]Research Centre, Future University in Egypt, New Cairo, 11845, Egypt
[6]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia
[7]Department of Information System, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia

*Corresponding Author: Mesfer Al Duhayyim. Email: m.alduhayyim@psau.edu.sa
Received: 28 July 2022; Accepted: 26 October 2022; Published: 11 September 2023

Abstract: Computational linguistics is an engineering-based scientific discipline. It deals with understanding written and spoken language from a computational viewpoint. Further, the domain also helps construct the artefacts that are useful in processing and producing a language either in bulk or in a dialogue setting. Named Entity Recognition (NER) is a fundamental task in the data extraction process. It concentrates on identifying and labelling the atomic components from several texts grouped under different entities, such as organizations, people, places, and times. Further, the NER mechanism identifies and removes more types of entities as per the requirements. The significance of the NER mechanism has been well-established in Natural Language Processing (NLP) tasks, and various research investigations have been conducted to develop novel NER methods. The conventional ways of managing the tasks range from rule-related and hand-crafted feature-related Machine Learning (ML) techniques to Deep Learning (DL) techniques. In this aspect, the current study introduces a novel Dart Games Optimizer with Hybrid Deep Learning-Driven Computational Linguistics (DGOHDL-CL) model for NER. The presented DGOHDL-CL technique aims to determine and label the atomic components from several texts as a collection of the named entities. In the presented DGOHDL-CL technique, the word embedding process is executed at the initial stage with the help of the word2vec model. For the NER mechanism, the Convolutional Gated Recurrent Unit (CGRU) model is employed in this work. At last, the DGO technique is used as a hyperparameter tuning strategy for the CGRU algorithm to boost the NER's outcomes. No earlier studies integrated the DGO mechanism with the CGRU model for NER. To exhibit the superiority of the proposed DGOHDL-CL technique, a widespread simulation analysis was executed on two datasets, CoNLL-2003 and OntoNotes 5.0. The experimental outcomes establish the promising performance of the DGOHDL-CL technique over other models.

Keywords: Named entity recognition; deep learning; natural language processing; computational linguistics; dart games optimizer

1 Introduction

The demand for Natural Language Processing (NLP) and information extraction mechanisms has been ever-increasing in recent years, thanks to digitalization and the generation of huge volumes of data (primarily unstructured text data) daily [1]. Named-Entity Recognition (NER) is one of the crucial steps in data processing methods. It is a task in which the named entities present in an unstructured text are found and categorized as specific semantic classes like person, organization, location and disease. It is the main step in various processes such as topic detection, question-answering, text summarization and so on [2]. This task has been accomplished in several research articles concerning different languages, especially the English language. However, studies concerning other languages are limited, and substantial developments are occurring in recent years [3]. NER refers to the task of identifying the named entities like an individual, biological protein, organization, time, location, clinical procedure, drug, etc., from the text. The NER mechanisms are often employed as an initial step in processes such as co-reference resolution, question answering, topic modelling, information retrieval, etc., [4]. Therefore, it is important to highlight the recent advancements in recognising the named entities, particularly the recent neural NER structures that achieved excellent performance with minimum feature engineering [5].

When constituting the NER mechanisms, the researchers rely on three predominant techniques: hybrid-based, linguistic-rule-related, and Machine Learning (ML)-based [6]. Amongst these, the linguistic approach uses rule-related methods that the linguists physically write. In this technique, a group of patterns or rules is described to differentiate the Named Entities (NEs) in a text. The ML-related approaches utilize a huge volume of annotated trained data to gain high-level language knowledge [7]. The ML techniques are constructed either as unsupervised methods or supervised methods. The unsupervised NER methods do not need any trained data. The key ideology of this method is to make the probable annotations from the dataset [8]. This method is unfamiliar with the ML techniques since there is no accuracy found in the absence of the supervised techniques.

Conversely, the supervised approaches need a massive volume of annotated data to develop a well-trained mechanism [9]. Some of the instances in which the ML approaches are utilized for NER algorithms include the Support Vector Machine (SVM), Decision Trees (DT), Maximum Entropy Model (MaxEnt), Hidden Markov Model (HMM), Artificial Neural Network (ANN) and many more. Amongst these, the Deep Learning (DL) technique is a subdomain of ML and integrates different numbers of the processed layers. These layers can learn the data representation from many abstracting stages [10]. In general, two fundamental structures are utilized in the extraction of textual representation (word-level or character-level), such as the Recurrent Neural Network (RNN) and the Convolutional Neural Network (CNN)-based method.

The current study introduces a novel Dart Games Optimizer with Hybrid Deep Learning-Driven Computational Linguistics (DGOHDL-CL) model for NER. The presented DGOHDL-CL technique aims to determine and label the atomic components from several texts as a collection of the named entities. In the presented DGOHDL-CL technique, the word embedding process is executed initially using the word2vec model. For the NER mechanism, the Convolutional Gated Recurrent Unit (CGRU) model is employed in this work. Lastly, the DGO technique is applied as a hyperparameter tuning strategy for the CGRU algorithm to boost the NER outcomes. A widespread simulation analysis was carried out to exhibit the superior performance of the proposed DGOHDL-CL technique.

2 Related Works

Fan et al. [11] presented a Deep Learning (DL)-based NER approach, i.e., a Deep, multi-branch BiGRU-CRF approach that integrated the multi-branch Bidirectional Gated Recurrent Unit (BiGRU) and the Conditional Random Field (CRF) approach. In this end-to-end supervised procedure, the presented approach automatically learnt and transformed the features through the multi-branch BiGRU layer and improved the results with the CRF layer. Chenaghlu et al. [12] examined two new DL algorithms by employing a multimodal DL and a transformer. These techniques were combined with utilizing the image features in a short social-media post to achieve optimal outcomes on NER tasks. In this primary method, the extraction of the image features was performed using the InceptionV3 approach. Based on this hybrid model, both textual and image features were retrieved. This technique further proposed a reliable NER in which the image is compared with an entity offered by the users. In the secondary method, the image features were integrated with the text and a Bidirectional Encoder Representation from Transformers (BERT)-like transformer was provided. Khan et al. [13] presented several Deep Recurrent Neural Network (DRNN) learning approaches with the word embedding process. These DRRN approaches estimated forward and the bidirectional extensions of Long Short-Term Memory (LSTM)-Backpropagation (BP) with time systems.

In literature [14], the RNN-based methods were considered for various activation functions. The functions optimized with NER tools were employed in the extraction of the termed entities, namely, person, organization, and place, from the tweets. Then, the pre-labelled data was trained using the GloVe word-embedded approach and Recurrent Neural Network (RNN) methods. This study projected the outperforming approaches amongst the RNN variations to predict the named entities. In the literature [15], the authors tried to recognize the educational institutions' names and the courses offered in the resume of the education section. In general, the important count of the annotated data is needed for Neural Network (NN)-based NER approaches. A semi-supervised method was utilized in this study to overcome the absence of huge annotated data. The authors trained a Deep Neural Network (DNN) approach using a primary (seed) set of the resume education sections. This approach was utilized to predict unlabelled education sections and rectified using an alteration part.

Sharma et al. [16] presented a DNN structure for NER in the Hindi language, a resource-scarce language. In the presented method, the authors primarily utilized the GloVe and skip-gram word2vec methods for word representation from the semantic vectors heavily utilized in distinct DNN-based structures. Further, the authors utilized the character- and word-level embedding processes to signify the text containing fine-grained data. In the study conducted earlier [17], a new multitask bi-directional RNN approach integrated with Deep Transfer Learning (DTL) was presented as a potential solution for data transmission and data augmentation to enhance the efficiency of the NER using restricted information.

3 The Proposed Model

In this study, a new DL-based DGOHDL-CL technique has been developed for the NER mechanism. The presented DGOHDL-CL aims to determine and label the atomic components from several texts as a collection of the named entities. Fig. 1 showcases the overall processes of the DGOHDL-CL algorithm.

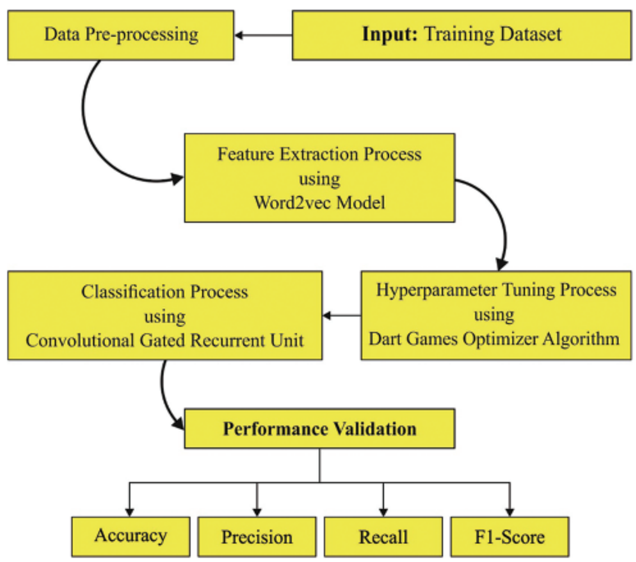

Figure 1: Overall processes of the DGOHDL-CL algorithm

3.1 Word Embedding

In the presented DGOHDL-CL technique, the word embedding process is executed initially using the word2vec model. The word embedding process functions based on the hypothesis, i.e., 'words that take place in the same context tend to have the same meaning' [18]. The main concept of the proposed method is to identify the local context (such as the phrase and sub-phrase of a missing word via the average or the concatenation of the initial word vector. It is computed as follows:

$$p\left(\frac{w_t}{w_{t-k}}, \ldots, w_{t+k}, P\right) = \frac{e^{y_{wt}}}{\sum_i e^{y_i}} \tag{1}$$

The log-probability for every output word is calculated using Eq. (2):

$$y = b + Uh\left(w_{t-k}, \ldots, w_{t+k}, W\right) \tag{2}$$

Here U and b denote the *softmax* parameters, and *h* is created using an average or concatenation of the word vector extracted from a matrix, W.

At every step *t*, the target word vector and the matrix *W* are upgraded to bring a set of similar words closer to the vector space. The words appearing in the corpora's common context are approximately connected in the vector space. This ability to capture the relationships and the semantics of the words among them is a major inspiration for several investigation workers in the domain of NLP.

A word2vec process is a conventional approach to learning the embedded words with the help of NN, and it was proposed by Tomas Mikolov at Google in the year, 2013.

The word2vec approach has two modules for learning word representation: the Skip-gram model and the Continuous bag-of- words (CBOW) model.

3.2 NER Using CGRU Model

For an effectual NER, the CGRU model is employed in this work. Recently, the RNN approach has been proven effective in the speech recognition processes [19]. The present activation h_t is defined using the existing input tex_t and the preceding activation h_{t-1}. The RNN approach can learn the long-term patterns in a better way compared to that of the feedforward DNN. This is because the input context feature should be independent every time in the feedforward DNN approach, as demonstrated herewith.

$$h_t = \varphi \left(W^h x_t + R^h h_{t-1} + b^h \right) \tag{3}$$

But, the RNN approach with a recurrent connection on the hidden state is challenging to train, owing to the important exploding or vanishing gradient problems. The LSTM architecture was actually developed to resolve these problems by presenting the cell state, input, forget and output gates to control the data flow. The primary concept of the LSTM model is its memory cells that maintain their state over time. Fig. 2 illustrates the infrastructure of the GRU technique.

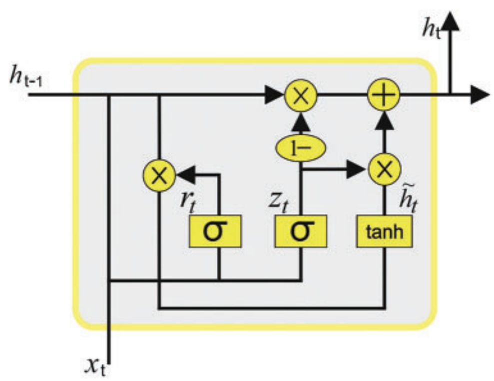

Figure 2: Architecture of the GRU technique

GRU is an alternate framework for the LSTM approach. The GRU technique is superior to the LSTM model in a certain set of tasks, as expressed below:

$$r_t = \delta \left(W^r x_t + R^r h_{t-1} + b^r \right) \tag{4}$$

$$z_t = \delta \left(W^z x_t + R^z h_{t-1} + b^z \right) \tag{5}$$

$$\hat{h}_t = \varphi \left(W^h x_t + r_t \odot \left(R^h h_{t-1} \right) + b^h \right) \tag{6}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \hat{h}_t \tag{7}$$

Now, h_t, r_t and z_t denote the hidden activation, reset and the update gate values at a frame t, correspondingly. The weight employed in the input and the recurrent hidden unit is correspondingly signified by W^* and R^*. The bias is characterized by b^*. $\delta(.)$ and $\varphi(.)$ denote the sigmoid and the tangent activation functions correspondingly. The reset gate r_t decides whether to forget the preceding activation or not.

In this CGRU model, the Convolutional Neural Network (CNN) is considered a feature extractor and a short window in the elementary feature. Later, the extracted features are provided as GRU-RNN so as to learn the long-term audio pattern. Various background noise and acoustic events occur randomly and repeatedly alongside the entire chunk. This occurs even without the knowledge of the frame location. The CNN approach may assist in extracting the robust features against the background noise using a max pooling function, particularly for the raw waveform. The GRU-RNN method selects the relevant data from a long-term context of all the audio events. Also, the Bi-directional GRU-RNN is applied to exploit the open data. This architecture is flexible enough to be employed in all the forms of the features, particularly in the case of raw waveforms. A raw waveform has many values and produces high-dimension problems.

The Binary cross-entropy is applied as a loss function in the current study because it was proven earlier superior to the MSE for labels with 0 or 1 values as follows:

$$E = -\sum_{n=1}^{N} \left\| T_n log\hat{T}_n + (1 - T_n)\ \log\ \left(1 - \hat{T}_n\right) \right\| \tag{8}$$

$$\hat{T}_n = (1 +\ \exp\ (-O))^{-1} \tag{9}$$

Here, E indicates the binary cross-entropy, T_n and T_n denote the evaluated and the reference tag vector at an instance index n, correspondingly. The DNN linear outcome is described as O before the application of the sigmoid function. Here, the Adam optimizer is applied as the stochastic optimization technique.

3.3 Hyperparameter Tuning Using DGO Algorithm

In this final stage, the DGO algorithm is used as a hyperparameter tuning strategy for the CGRU model to boost the NER's outcomes. The perspective of the Darts game is employed in this stage to design the DGO algorithm [20]. The searcher agents in the DGO denote the players, and their objective is to acquire a maximum score (optimum answer) in this section.

The population of the players is modelled with a matrix in which each row represents a player, and every column signifies the distinct features of every player. In this matrix, the count of the columns is similar to the count of the problem parameters and the value recommended for this variable is shown below:

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix} \begin{bmatrix} X_1^1 & \cdots & X_d^1 & \cdots & X_1^m \\ \vdots & \ddots & \vdots & & \vdots \\ X_i^1 & \cdots & X_i^d & \cdots & X_i^m \\ \vdots & & \vdots & \ddots & \vdots \\ X_N^1 & \cdots & X_N^d & \cdots & X_N^m \end{bmatrix} \tag{10}$$

Now, X represents the player's matrix, x_i^d indicates the *d-th* e dimension of the *i-th* player, m shows the variable count, and N indicates the player's count.

By placing X_i in the fitness function, valuable data is attained, as shown below:

$$F_{best} = \min\ (fit)_{N\times 1} \tag{11}$$

$$X_{best} = X\ \ (location\ of\ min\ (fit),\ 1 : m) \tag{12}$$

$$F_{worst} = \max\ (fit)_{N\times 1} \tag{13}$$

$$X_{worst} = X\ \ (location\ of\ max\ (fit),\ 1:m) \tag{14}$$

$$F^n = \frac{fit - F_{worst}}{\sum_{j=1}^{N}\left(fit_j - F_{worst}\right)} \tag{15}$$

$$P_i = \frac{F_i^n}{\max\ (F^n)} \tag{16}$$

In this expression, F_{best} characterizes an optimum fitness function, X_{best} implies an optimum variable, F_{worst} shows the worst fitness function, X_{worst} indicates the worst variable, F^n denotes the normalized value of the fitness function, and P_i indicates the probability function of the *i-th* player.

For each Dartboard, the following standard dimensions are followed:

- Inner size of the double and treble ring: 8 mm
- An inner diameter of bull: 12.7 mm
- The inner diameter of the outer bull: 31.8 mm
- The centre bull to the inner edge of the treble wire: 107 mm
- The centre bull to the outer edge bull wire: 170 mm
- The outer edge of the double wire to the outer edge of the double wire: 340 mm
- Global dartboard diameter: 451 mm

In the abovementioned dimensions, the Dartboard has a total of 82 regions with distinct scores. All the players can have three darts during every iteration. The dart's position on the Dartboard depends on the following aspects: the player's skill and their chance.

There are six sectors available with distinct regions on the Dartboard. Hence, the throwing score is calculated and modelled for all the players.

$$C_i = round\,(82 \times (1 - P_i)) \tag{17}$$

$$SC_i = \begin{cases} S(1:C),\ rand < P_i \\ S(C+1:82),\ else \end{cases} \tag{18}$$

$$s_i = SC_i\,(k)\ \ \&\ 1 \le k \le 82 \tag{19}$$

$$s_i^n = \frac{\sum_{throws=1}^{3} s_i^{throws}}{180} \tag{20}$$

Here, SC_i indicates the scored candidate for the *i-th* player, S shows the score matrix that is arranged from a high score to a low score, s_i implies the score for all the throws of the *i-th* player, and s_i^n denotes the normalized score of the *i-th* player.

At last, the novel status of all the players and the values of the problem variable is upgraded as follows:

$$X_i = X_i + rand\,(1,\ m) \times \left(X_{best} - 3s_i^n X_i\right) \tag{21}$$

The DGO has a few essential parameters that should be defined. The member count in the population stands at 50, and a thousand repetitions are considered as a stopping criterion for the repetition count. The early population of the players is randomly generated.

All the players actually fall under a m-member vector that characterizes a recommended solution to the problem. The member of these vectors illustrates the problem variable estimated by assigning them under the objective function.

The following steps are applied in DGO:

Start DGO

Step1: Create the primary population of players.

Step2: Calculate the fitness function.

Step3: Update F_{best}, X_{best}, F_{worst}, and X_{worst} by (11) to (14).

Step4: Update F^n and P_i by (15) and (16).

Step5: Calculate s_i^n by (17) to (20).

Step6: Update X_i by (21).

Step7: Check the stopping criteria.

Step8: Print the outcome.

End DGO.

4 Results and Discussion

The proposed model was simulated in Python 3.6.5 tool. The NER performance of the proposed DGOHDL-CL model was validated utilizing two datasets, namely, CoNLL-2003 and OntoNotes 5.0. Table 1 and Fig. 3 show the overall NER outcomes of the proposed DGOHDL-CL model on the CoNLL-2003 dataset. The obtained values infer that the proposed DGOHDL-CL model achieved enhanced results under all the epochs. For the sample, on 500 epochs, the DGOHDL-CL approach obtained $prec_n$, $reca_l$, $accu_y$ and $F1_{score}$ values such as 96.61%, 97.25%, 97.85%, and 96.21% respectively. Meanwhile, on 1,000 epochs, the proposed DGOHDL-CL approach reached $prec_n$, $reca_l$, $accu_y$ and $F1_{score}$ values such as 96.32%, 96.70%, 96.75% and 96.17%, correspondingly. Eventually, on 1,500 epochs, the DGOHDL-CL algorithm reached $prec_n$, $reca_l$, $accu_y$ and $F1_{score}$ values such as 97.58%, 96.58%, 97.94%, and 97.05%, correspondingly. Moreover, on 2,000 epochs, the DGOHDL-CL methodology gained $prec_n$, $reca_l$, $accu_y$, and $F1_{score}$ values such as 97.59%, 97.92%, 96.46% and 97.26%, correspondingly.

Table 1: Analytical results of the DGOHDL-CL approach on the CoNLL-2003 dataset under distinct epochs

CoNLL-2003				
No. of epochs	Precision	Recall	Accuracy	F1-score
500	96.61	97.25	97.85	96.21
1000	96.32	96.70	96.75	96.17
1500	97.58	96.58	97.94	97.05
2000	97.59	97.92	96.46	97.26
2500	96.76	97.91	96.50	97.31

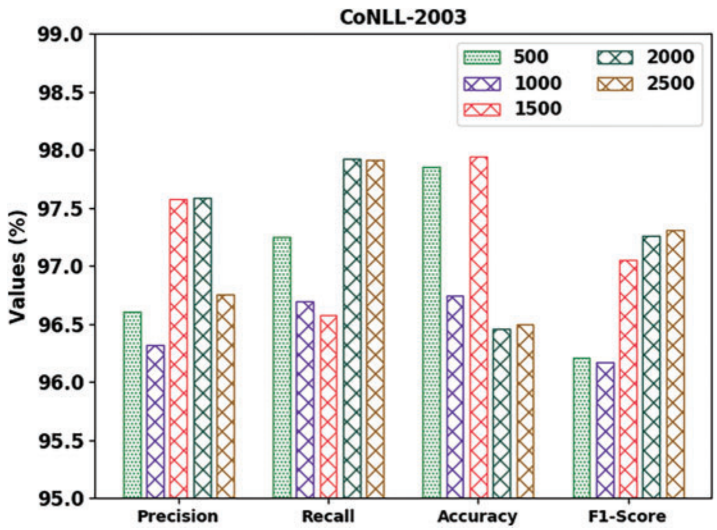

Figure 3: Analytical results of the DGOHDL-CL approach on the CoNLL-2003 dataset

Both Training Accuracy (TRA) and Validation Accuracy (VLA) values acquired by the proposed DGOHDL-CL system under the CoNLL-2003 dataset are depicted in Fig. 4. The experimental results infer that the proposed DGOHDL-CL approach achieved the maximal TRA and VLA values whereas the VLA value was superior to the TRA values.

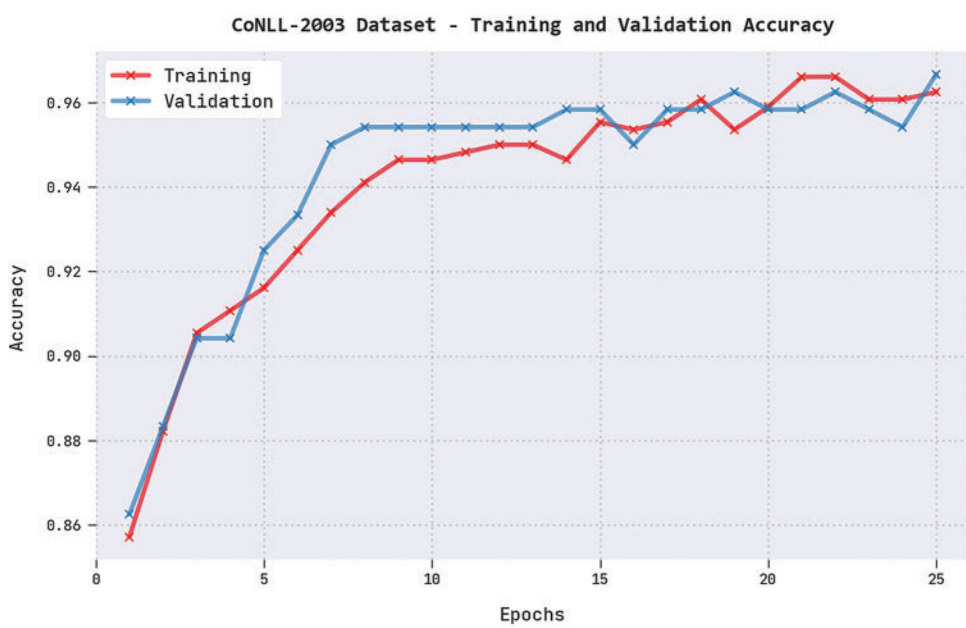

Figure 4: TRA and VLA analyses results of the DGOHDL-CL approach on the CoNLL-2003 dataset

Both Training Loss (TRL) and Validation Loss (VLL) values, realized by the proposed DGOHDL-CL method under CoNLL-2003 dataset, are represented in Fig. 5. The experimental results expose that the proposed DGOHDL-CL system achieved the least TRL and VLL values whereas the VLL values were lesser than the TRL values.

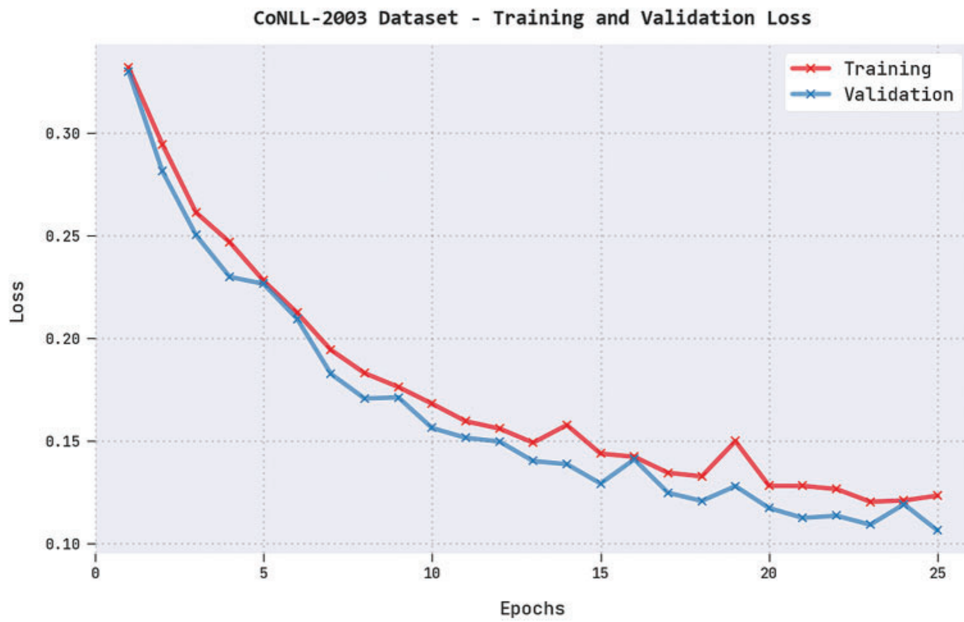

Figure 5: TRL and VLL analyses results of the DGOHDL-CL approach on the CoNLL-2003 dataset

Table 2 and Fig. 6 demonstrate the overall NER outcomes of the proposed DGOHDL-CL system on the OntoNotes5.0 dataset. The attained values represent that the proposed DGOHDL-CL approach exhibited improved results in all the epochs. For sample, on 500 epochs, the proposed DGOHDL-CL algorithm achieved $prec_n$, $reca_l$, $accu_y$ and $F1_{score}$ values such as 91.03%, 91.47%, 91.76% and 93.19% correspondingly. On 1,000 epochs, the proposed DGOHDL-CL system reached $prec_n$, $reca_l$, $accu_y$ and $F1_{score}$ values such as 93.22%, 91%, 91.83% and 93.69% correspondingly. At the same time, on 1,500 epochs, the proposed DGOHDL-CL technique gained $prec_n$, $reca_l$, $accu_y$ and $F1_{score}$ values such as 93.25%, 93.95%, 93.10% and 91.57% correspondingly. At last, on 2,000 epochs, the proposed DGOHDL-CL algorithm reached $prec_n$, $reca_l$, $accu_y$ and $F1_{score}$ values such as 92.90%, 91.67%, 93.42%, and 93.46% correspondingly.

Table 2: Analytical results of the DGOHDL-CL approach on OntoNotes5.0 dataset under distinct epochs

OntoNotes5.0				
No. of epochs	Precision	Recall	Accuracy	F1-score
500	91.03	91.47	91.76	93.19
1000	93.22	91.00	91.83	93.69
1500	93.25	93.95	93.10	91.57
2000	92.90	91.67	93.42	93.46
2500	92.75	91.05	92.53	92.80

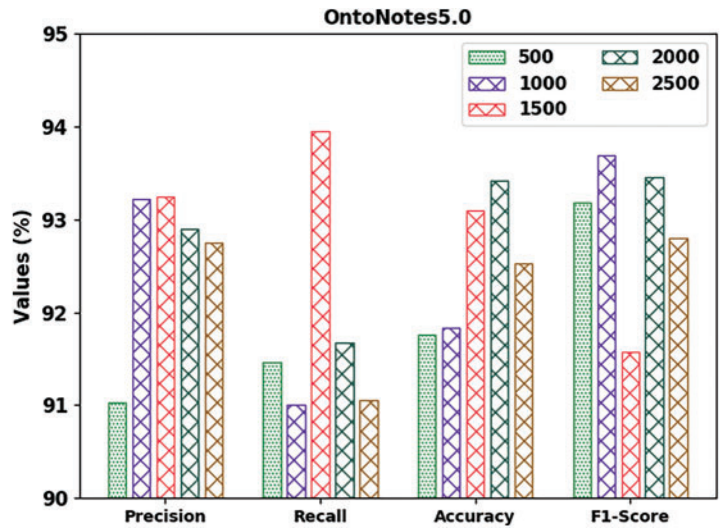

Figure 6: Analytical results of the DGOHDL-CL approach on OntoNotes5.0 dataset

Both TRA and VLA values, achieved by the proposed DGOHDL-CL algorithm on OntoNotes5.0 dataset, are displayed in Fig. 7. The experimental results revealed that the proposed DGOHDL-CL system achieved the maximal TRA and VLA values whereas the VLA values were superior to the TRA values.

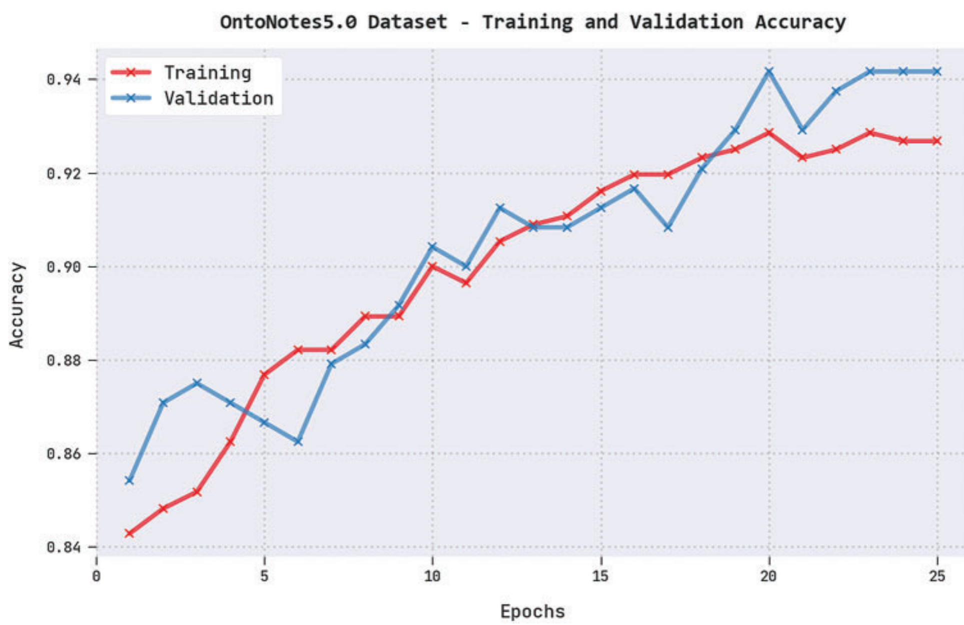

Figure 7: TRA and VLA analyses results of the DGOHDL-CL approach on OntoNotes5.0 dataset

Both TRL and VLL values, accomplished by the DGOHDL-CL system on OntoNotes5.0 dataset, are represented in Fig. 8. The experimental results expose that the proposed DGOHDL-CL system achieved the minimal TRL and VLL values whereas the VLL values were lesser than the TRL values.

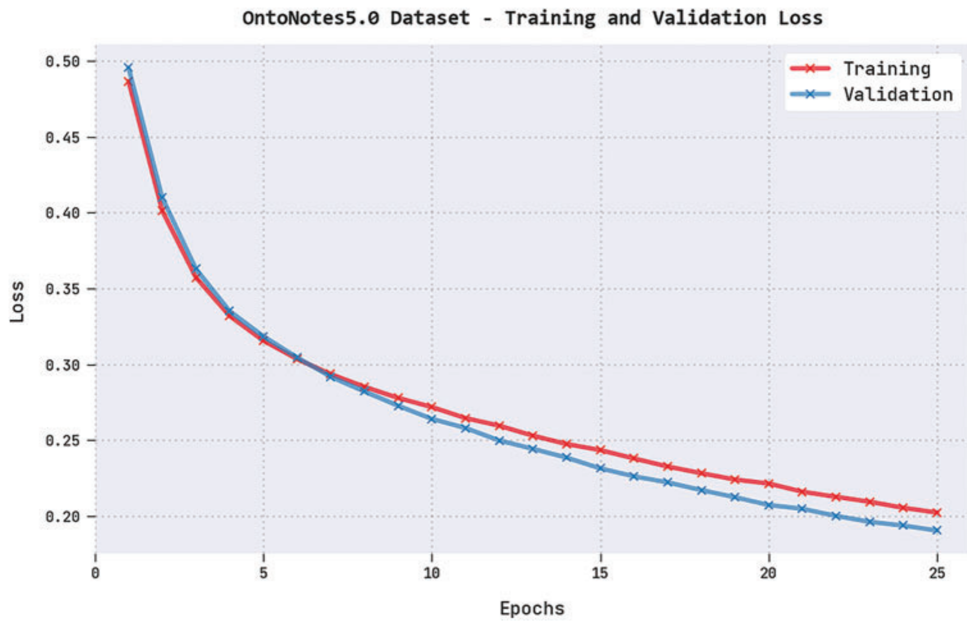

Figure 8: TRL and VLL analyses results of the DGOHDL-CL approach on OntoNotes5.0 dataset

Table 3 offers the detailed comparative study outcomes achieved by the DGOHDL-CL model and other recent models on CoNLL-2003 dataset [21]. Fig. 9 shows the comparison study outcomes of the DGOHDL-CL model and other existing models on CoNLL-2003 dataset. The figure highlights that the word embedding with BiLSTM-CRF (WORD-EMB-BLC) model yielded a poor performance with a minimal $prec_n$ of 85.29%. The BLSTM-CNN model attained a slightly improved $prec_n$ of 91.87% whereas the HSCRF model achieved an increased $prec_n$ of 93.87%. Followed by, the BiLSTM-CRF (BLC) BE-BLC, ELMO-BLC and BERT-BLC models established the closer $prec_n$ values such as 95.93%, 95.48% and 95.11% correspondingly. However, the proposed DGOHDL-CL model produced a maximum $prec_n$ of 97.59%.

Table 3: Comparative analysis results of the DGOHDL-CL approach and other methods on CoNLL-2003 dataset

	CoNLL-2003		
Methods	Precision	Recall	F1-score
DGOHDL-CL	97.59	97.92	97.26
BE-BLC	95.93	95.92	95.39
WORD-EMB-BLC	85.29	82.00	83.05
ELMO-BLC	95.48	95.73	95.65
BERT-BLC	95.11	94.96	94.86
BLSTM-CNN	91.87	92.06	92.11
HSCRF	93.87	91.62	90.02

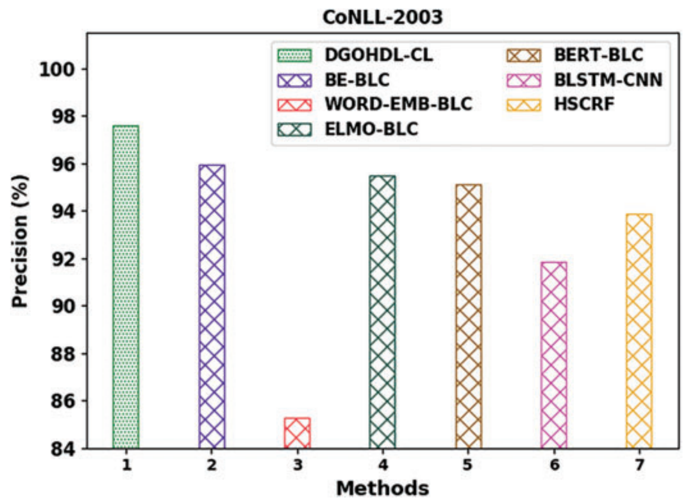

Figure 9: $Prec_n$ analysis results of the DGOHDL-CL approach on CoNLL-2003 dataset

Fig. 10 depicts the comparison examination outcomes achieved by the proposed DGOHDL-CL approach and other existing techniques on CoNLL-2003 dataset. The figure expose that the WORD-EMB-BLC system achieved the least performance with a low $reca_l$ value of 82%. Afterward, the BLSTM-CNN model reached a somewhat enhanced $reca_l$ value of 92.06%. The HSCRF methodology achieved an increased $reca_l$ of 91.62%. Likewise, the BE-BLC, ELMO-BLC and the BERT-BLC techniques exhibited closer $reca_l$ values such as 95.92%, 95.73% and 94.96% correspondingly. But, the proposed DGOHDL-CL approach accomplished a superior $reca_l$ of 97.92%.

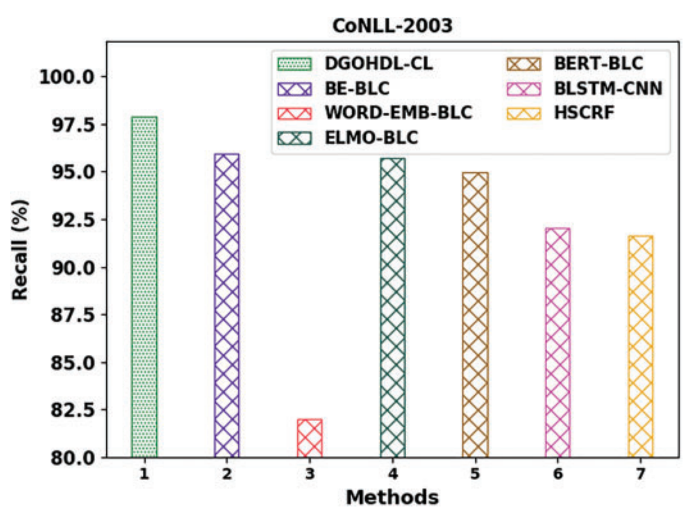

Figure 10: $Reca_l$ analysis results of the DGOHDL-CL approach on CoNLL-2003 dataset

Fig. 11 illustrates the comparative analysis results of the proposed DGOHDL-CL algorithm and other existing approaches on CoNLL-2003 dataset. The figure demonstrates that the WORD-EMB-BLC methodology attained the least performance with a minimal FI_{score} of 83.05%. Similarly, the BLSTM-CNN algorithm attained a slightly higher FI_{score} of 92.11%. The HSCRF technique attained an increased FI_{score} of 90.02%. Next, the BE-BLC, ELMO-BLC and the BERT-BLC methodologies outperformed other methods and achieved closer FI_{score} values such as 95.39%, 95.65% and 94.86% correspondingly. At last, the proposed DGOHDL-CL algorithm attained a maximal FI_{score} of 97.26%.

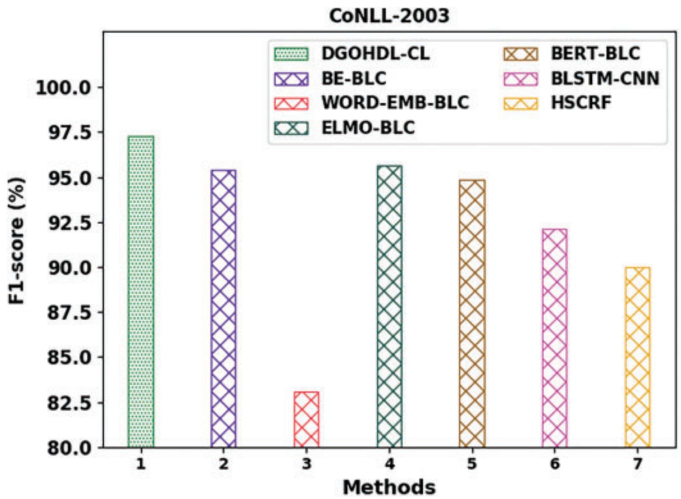

Figure 11: $F1_{score}$ analysis results of the DGOHDL-CL approach on CoNLL-2003 dataset

Table 4 provides the detailed comparative analysis results of the proposed DGOHDL-CL technique and other recent techniques on OntoNotes5.0 dataset. Fig. 12 depicts the comparative investigation outcomes of the proposed DGOHDL-CL methodology and other existing systems on OntoNotes5.0 dataset. The figure states that the WORD-EMB-BLC methodology portrayed the least performance with a low $prec_n$ of 82.14%. Also, the BLSTM-CNN algorithm accomplished somewhat improved $prec_n$ of 86.57% whereas the HSCRF technique attained an increased $prec_n$ of 85.12%. In addition, the BE-BLC, ELMO-BLC and the BERT-BLC approaches revealed closer $prec_n$ values such as 88.29%, 87.92% and 89.36%, respectively. But, the proposed DGOHDL-CL model produced a superior $prec_n$ of 93.25%.

Table 4: Comparative analysis results of the DGOHDL-CL approach and other recent algorithms on OntoNotes5.0 dataset

OntoNotes5.0			
Methods	Precision	Recall	F1-score
DGOHDL-CL	93.25	93.95	91.57
BE-BLC	88.29	89.61	89.43
WORD-EMB-BLC	82.14	77.72	79.27
ELMO-BLC	87.92	89.83	88.92
BERT-BLC	89.36	90.14	89.88
BLSTM-CNN	86.57	86.35	86.67
HSCRF	85.12	86.02	86.15

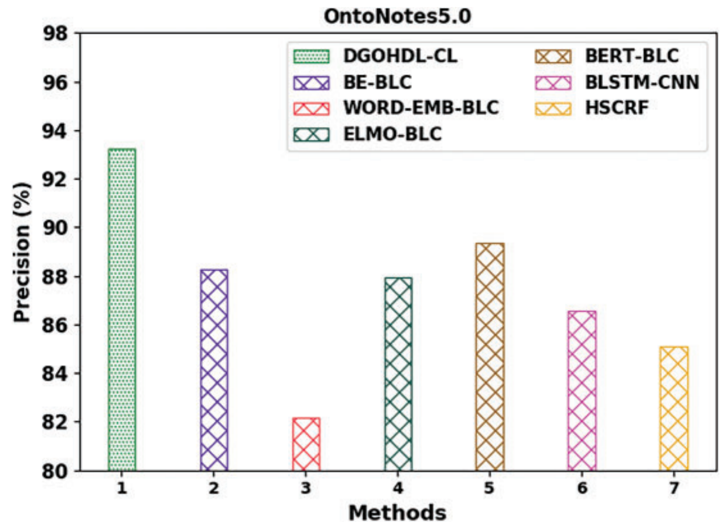

Figure 12: $Prec_n$ analysis results of the DGOHDL-CL approach on OntoNotes5.0 dataset

Fig. 13 exhibits the comparison analysis results of the proposed DGOHDL-CL method and other existing methodologies on OntoNotes5.0 dataset. The figure highlights that the WORD-EMB-BLC model achieved a poor performance with a low $reca_l$ of 77.72%. Besides, the BLSTM-CNN system attained slightly an improved $reca_l$ value of 86.35% whereas the HSCRF model attained an increased $reca_l$ of 86.02%. Furthermore, the BE-BLC, ELMO-BLC and the BERT-BLC systems demonstrated closer $reca_l$ values such as 89.61%, 89.83% and 90.14% respectively. Finally, the proposed DGOHDL-CL algorithm achieved an increased $reca_l$ of 93.95%.

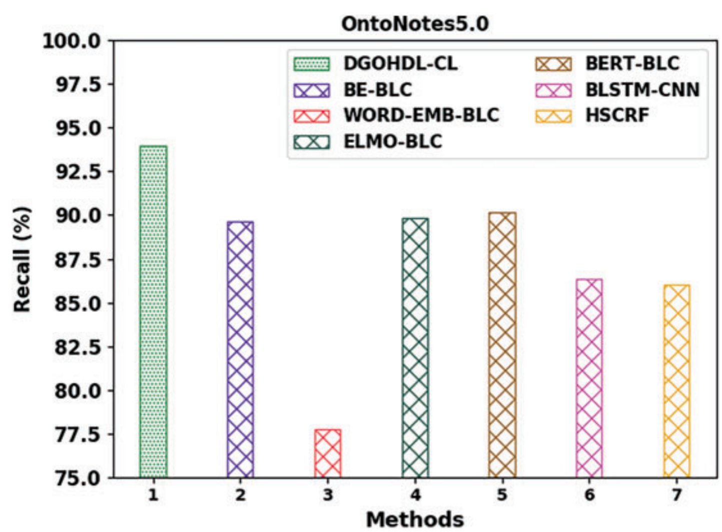

Figure 13: $Reca_l$ analysis results of the DGOHDL-CL approach on OntoNotes5.0 dataset

Fig. 14 portrays the comparative analysis results of the proposed DGOHDL-CL model and other existing models on OntoNotes5.0 dataset. The figure highlights that the WORD-EMB-BLC model achieved the least performance with a minimum FI_{score} of 79.27%. The BLSTM-CNN approach attained an enhanced FI_{score} of 86.67% whereas the HSCRF methodology achieved an increased FI_{score}

of 86.15%. Moreover, the BE-BLC, ELMO-BLC and the BERT-BLC techniques achieved closer FI_{score} values such as 89.43%, 88.92% and 89.88%, respectively. Lastly, the proposed DGOHDL-CL approach produced a maximum FI_{score} of 91.57%. Based on these results and the discussion, it can be inferred that the proposed DGOHDL-CL approach achieved the enhanced NER outcomes than the rest of the models.

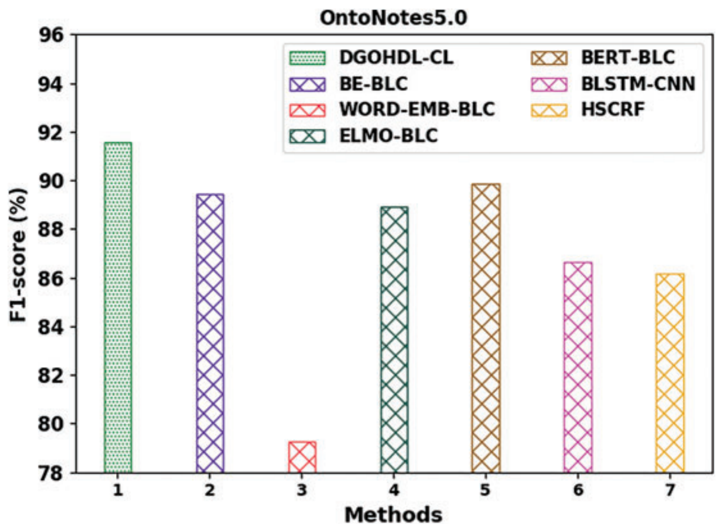

Figure 14: $F1_{score}$ analysis results of the DGOHDL-CL approach on OntoNotes5.0 dataset

5 Conclusion

In this study, a new DL-based DGOHDL-CL technique has been developed for NER mechanism. The aim of the presented DGOHDL-CL technique is to determine and label the atomic components from several texts as a collection of the named entities. In the presented DGOHDL-CL technique, the word embedding process is executed at the initial state using the word2vec model. For NER mechanism, the CGRU model is employed in this work. Lastly, the DGO system is used as a hyperparameter tuning strategy for the CGRU approach to boost the NER's outcomes. To exhibit the superiority of the proposed DGOHDL-CL technique, a widespread simulation analysis was conducted. The experimental outcomes established the promising performance of the proposed DGOHDL-CL technique over other models. In the future, the performance of the DGOHDL-CL technique can be extended to utilize the feature selection methodologies.

Funding Statement: Princess Nourah Bint Abdulrahman University Researchers Supporting Project Number (PNURSP2022R281), Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4331004DSR10).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. Li, A. Sun, J. Han and C. Li, "A survey on deep learning for named entity recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 50–70, 2020.

[2] X. Liu, H. Chen and W. Xia, "Overview of named entity recognition," *Journal of Contemporary Educational Research*, vol. 6, no. 5, pp. 65–68, 2022.

[3] B. Song, F. Li, Y. Liu and X. Zeng, "Deep learning methods for biomedical named entity recognition: A survey and qualitative comparison," *Briefings in Bioinformatics*, vol. 22, no. 6, pp. bbab282, 2021.

[4] S. Kanwal, K. Malik, K. Shahzad, F. Aslam and Z. Nawaz, "Urdu named entity recognition: Corpus generation and deep learning applications," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 19, no. 1, pp. 1–13, 2019.

[5] H. Cho and H. Lee, "Biomedical named entity recognition using deep neural networks with contextual information," *BMC Bioinformatics*, vol. 20, no. 1, pp. 1–11, 2019.

[6] W. Yoon, C. H. So, J. Lee and J. Kang, "Collabonet: Collaboration of deep neural networks for biomedical named entity recognition," *BMC Bioinformatics*, vol. 20, no. 10, pp. 55–65, 2019.

[7] L. Li, J. Zhao, L. Hou, Y. Zhai, J. Shi *et al.,* "An attention-based deep learning model for clinical named entity recognition of Chinese electronic medical records," *BMC Medical Informatics and Decision Making*, vol. 19, no. 5, pp. 1–11, 2019.

[8] Z. Chai, H. Jin, S. Shi, S. Zhan, L. Zhuo *et al.,* "Hierarchical shared transfer learning for biomedical named entity recognition," *BMC Bioinformatics*, vol. 23, no. 1, pp. 1–14, 2022.

[9] C. Helwe and S. Elbassuoni, "Arabic named entity recognition via deep co-learning," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 197–215, 2019.

[10] J. Sun, Y. Liu, J. Cui and H. He, "Deep learning-based methods for natural hazard named entity recognition," *Scientific Reports*, vol. 12, no. 1, pp. 1–15, 2022.

[11] R. Fan, L. Wang, J. Yan, W. Song, Y. Zhu *et al.,* "Deep learning-based named entity recognition and knowledge graph construction for geological hazards," *ISPRS International Journal of Geo-Information*, vol. 9, no. 1, pp. 15, 2019.

[12] M. A. Chenaghlu, M. R. F. Derakhshi, L. Farzinvash, M. A. Balafar and C. Motamed, "CWI: A multimodal deep learning approach for named entity recognition from social media using character, word and image features," *Neural Computing and Applications*, vol. 34, no. 3, pp. 1905–1922, 2022.

[13] W. Khan, A. Daud, F. Alotaibi, N. Aljohani and S. Arafat, "Deep recurrent neural networks with word embeddings for Urdu named entity recognition," *ETRI Journal*, vol. 42, no. 1, pp. 90–100, 2020.

[14] N. Eligüzel, C. Çetinkaya and T. Dereli, "Application of named entity recognition on tweets during earthquake disaster: A deep learning-based approach," *Soft Computing*, vol. 26, no. 1, pp. 395–421, 2022.

[15] B. Gaur, G. S. Saluja, H. B. Sivakumar and S. Singh, "Semi-supervised deep learning-based named entity recognition model to parse education section of resumes," *Neural Computing and Applications*, vol. 33, no. 11, pp. 5705–5718, 2021.

[16] R. Sharma, S. Morwal, B. Agarwal, R. Chandra and M. S. Khan, "A deep neural network-based model for named entity recognition for Hindi language," *Neural Computing and Applications*, vol. 32, no. 20, pp. 16191–16203, 2020.

[17] X. Dong, S. Chowdhury, L. Qian, X. Li, Y. Guan *et al.,* "Deep learning for named entity recognition on Chinese electronic medical records: Combining deep transfer learning with multitask bi-directional LSTM RNN," *PLoS One*, vol. 14, no. 5, pp. e0216046, 2019.

[18] E. Rudkowsky, M. Haselmayer, M. Wastian, M. Jenny, Š. Emrich *et al.,* "More than bags of words: Sentiment analysis with word embeddings," *Communication Methods and Measures*, vol. 12, no. 2, pp. 140–157, 2018.

[19] Z. Shen, S. P. Deng and D. S. Huang, "RNA-Protein binding sites prediction via multi-scale convolutional gated recurrent unit networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 17, no. 5, pp. 1741–1750, 2019.

[20] M. Dehghani, Z. Montazeri, H. Givi, J. M. Guerrero and G. Dhiman, “Darts game optimizer: A new optimization technique based on darts game,” *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 5, pp. 286–294, 2020.
[21] M. Affi and C. Latiri, “BE-BLC: BERT-ELMO-based deep neural network architecture for English named entity recognition task,” *Procedia Computer Science*, vol. 192, pp. 168–181, 2021.