SRTTU

Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

JECEI

Research paper

Deep Reinforcement Learning for Efficient Multilingual Dialogue Management

Mohammad Javad Nasri-Lowshani, Javad Salimi-Sartakhti, Hossein Ebrahimpour-Komle*

Department of Artificial Intelligence, Faculty of Electrical and Computer Engineering, University of Kashan, Kashan, Iran.

Article Info

Article History:
Received 17 January 2025
Reviewed 27 February 2025
Revised 26 March 2025
Accepted 08 April 2025

Keywords:
Task-oriented dialogue systems
Deep reinforcement learning
Multilingual dialogue management
State representation learning
Reward function optimization

*Corresponding Author's Email Address: *salimi@kashanu.ac.ir*

Abstract

Background and Objectives: Developing efficient task-oriented dialogue systems capable of handling multilingual interactions is a growing area of research in natural language processing (NLP). In this paper, we propose SenSimpleDS, a deep reinforcement learning-based joint task-oriented dialogue system, designed for multilingual conversations.

Methods: The system utilizes a deep Q-network and the SBERT model to represent the dialogue environment. We introduce two variants, SenSimpleDS+ and SenSimpleDS-NSP, which incorporate modifications in the ε-greedy method and leverage next sequence prediction (NSP) using BERT to refine the reward function. These methods are evaluated on datasets in English, Persian, Spanish, and German, and compared with baseline methods such as SimpleDS and SCGSimpleDS.

Results: Our experimental results demonstrate that the proposed methods outperform the baselines in terms of average collected rewards, requiring fewer learning steps to achieve optimal dialogue policies. Notably, the incorporation of NSP significantly improves performance by optimizing reward collection. The multilingual SenSimpleDS further showcases the system's ability to function across languages using a random forest classifier for language detection and MPNet for environment construction. In addition to system evaluations, we introduce a new Persian dataset for task-oriented dialogue in the restaurant domain, expanding the resources available for developing dialogue systems in low-resource languages.

Conclusion: SenSimpleDS, a deep reinforcement learning-based joint task-oriented dialogue system, demonstrates superior performance over baseline methods by leveraging deep Q-networks, SBERT. The integration of next sequence prediction (NSP) significantly enhances reward optimization, enabling faster convergence to optimal dialogue policies. This work establishes a foundation for future research in multilingual dialogue systems, with potential applications across diverse service domains.

How to cite this paper:
M. J. Nasri-Lowshani, J. Salimi-Sartakhti, H. Ebrahimpour-Komle, "Deep reinforcement learning for efficient multilingual dialogue management," J. Electr. Comput. Eng. Innovations, 14(1): 11-22, 2026.

DOI: 10.22061/jecei.2025.11348.814

URL: https://jecei.sru.ac.ir/article_2320.html

Introduction

With recent advancements in natural language processing, dialogue systems have emerged as a means of conversing with humans. These systems can be categorized into two types: task-oriented and non-task-oriented dialogue systems. Task-oriented dialogue systems assist humans in achieving specific goals through dialogue. They have gained popularity in customer service chatbots, as they simplify access to information and services, such as restaurant bookings or flight ticket reservations [1]. On the other hand, non-task-oriented systems focus on engaging in open-ended conversations with humans, serving as chatbots.

Dialogue systems are typically consist of three components: natural language understanding, dialogue management, and natural language generation. In the natural language understanding component, incoming human messages (e.g., “I am looking for Iranian food”) are received and transformed into an internal state representation (as a vector). Based on the dialogue state and defined policies, the dialogue management component selects an appropriate action (such as confirming the type of food). The natural language generation component generates responses in natural language form (e.g., “Did you say Iranian food?”) and presents them to the human user.

Traditionally, dialogue systems have been built by designing, training, and evaluating each component separately. However, this approach can be costly, time-consuming, and less generalizable to other domains [2]. It also introduces error propagation, as the performance of each component relies on the output of the preceding component [3].

To address these challenges, jointly learnable models have been proposed, where multiple components are combined to perform multiple tasks.

Building efficient models for dialogue systems requires large corpora. For example, the dialogue system was built using the GPT2 model, which utilized a dataset of 10,438 labeled dialogue samples [4]. However, reinforcement learning methods can start learning from small corpora and improve over time through interaction with the environment. One prominent reinforcement learning method is Q-learning, which utilizes the $Q(s,a)$ evaluation function to estimate the utility of taking action a from state s. These values are typically stored in a Q-table, enabling the selection of desired actions in different situations. Deep Q-learning, on the other hand, employs neural networks to learn policies instead of Q-tables. The input to the neural network is the environment, and the output is the Q-value. Deep Q-learning is particularly useful when the state space is infinite, unlike traditional Q-learning.

Language models are utilized to predict the probability of word occurrences in a sentence based on preceding words, enabling models to learn the language rules. These models enhance the performance of deep learning methods across various applications, like Information Retrieval (IR) [5]. Among the popular language models, the BERT language model stands out [6]. It adopts a transformer architecture, specifically utilizing the encoder block. BERT can be employed to construct representations for words. The conventional approach for constructing sentence representations involves utilizing word representation vectors and calculating their means. While BERT can also be used for this purpose, its performance in this approach is weaker compared to GloVe [7] and Additionally, the BERT model does not offer separate sentence representations [8]. To address these limitations, the SBERT model [8] builds upon BERT and employs siamese neural networks to generate sentence representations that consider the semantic relationship between sentences. This model maps sentences to numeric vectors, representing them in a continuous space.

In this paper, a jointly task-oriented system is designed using deep reinforcement learning, language models, and the SBERT representation model. This system enables text-based conversation between chatbots and humans, with the human achieving their goal at the end of the conversation. Furthermore, a method for a multilingual dialogue system is proposed. Despite advancements in natural language processing methods for Persian, there is no available dataset for restaurant conversations. Therefore, this work provides a dataset by collecting restaurant conversations in Persian.

The paper is structured as follows: The second section reviews previous research conducted in the field of joint task-oriented systems. The third section provides detailed explanations of the proposed method for constructing a jointly task-oriented system. The fourth section compares the performance of the proposed method with other approaches. Finally, the fifth section discusses the proposed method and outlines future work.

Related Work

Researchers have proposed various methods for designing jointly task-oriented dialogue systems. In the natural language understanding component, a method based on the gated recurrent unit network has been proposed in [9].

This method performs joint domain identification and intent identification. Another method, called CTRAN, utilizes the BERT language model and convolutional neural networks for domain identification and intent identification [10]. Additionally, in [11], a method that combines n-gram feature extraction and a gated recurrent unit network is proposed for joint domain identification, intent identification, and slot identification.

Moving on to the dialogue management component, a reinforcement learning-based method has been proposed in [12] to jointly perform state tracking and action selection. This method incorporates a recommender system that suggests products to users during the conversation. In the natural language generation component, a method leveraging long short-term memory networks has been proposed in [13]. This method controls the semantic frames of speech and generates appropriate sentences using the output from the dialogue management component. Furthermore, in [14], a method for word and dialogue generation is proposed, utilizing long short-term memory, an attention mechanism at the encoder block, and a linear classifier in the decoder block.

Much research has been conducted on the components of natural language understanding and dialogue management. In [15], a method utilizing reinforcement learning and a user simulator is proposed for a movie recommendation system. The method involves a conversation between a user and a chatbot connected to a knowledge base. It employs n-gram representation and gated recurrent unit for belief tracking. In [16], a task-oriented dialogue system is developed in the restaurant domain using deep Q-learning. This method, called SimpleDS, constructs a state vector by concatenating the representations of human and robot messages using Glove word embeddings. The state vector is then fed into a deep Q-network for learning. The allowed actions are determined by a Naive Bayes classifier trained on pre-conversation data. The ε-greedy approach with linear function is used to reduce the value of epsilon (ε). Additionally, a user simulator is employed to evaluate the performance of deep Q-learning in the dialogue system.

The SimpleDS method was initially developed for single-domain applications, but it was also extended to cater to multi-domain scenarios using networks of deep reinforcement learning [17]. In this method, three deep Q-networks are utilized, each corresponding to a specific domain. A pretrained support vector machine is employed to determine the domain of the input state vector. The state vector is fed into all three deep Q-networks, and the network associated with the domain of the state specifies the next action. In [18], the performance of the multi-domain SimpleDS is enhanced by using a series of parallel dialogues instead of individual dialogues at each step during a pre-training phase, resulting in a 1.4 times faster training speed.

Another method to improve the SimpleDS approach is the SCGSimpleDS method, which is designed for a single-domain application, specifically the restaurant domain [19]. In this method, the state of the environment is encoded into a vector of length n-6 using an auto-encoder. The remaining 6 entries are filled using sentiment analysis and clustering techniques to create better representations of the environment. This method also incorporates adversarial generating networks to train the Naive Bayes classifier with a larger dataset. Additionally, human evaluation is conducted alongside user simulator evaluation to measure the performance of the chatbot.

In [20], a method based on transfer learning and the use of a user simulator is proposed. This method transfers the weights of a deep Q-network from one domain to another. Two approaches for transfer learning are presented. The first approach trains the deep Q-network on one domain, which covers a subset of actions in both domains and then transfers the weights to another domain. The second approach involves training the deep Q-network on the smaller domain and then expanding it to the larger domain, where one domain is a subset of the other.

Several methods utilizing deep Q-learning have been introduced across various applications and domains, primarily employing simulators. However, a lingering question remains regarding the disparity between simulators and real user interactions. In [21], a dialogue system is proposed, wherein a deep Q-network is initially developed using a limited dataset and a simulator in a software environment, specifically in the context of tic-tac-toe. Subsequently, the system is implemented in a real environment with a chatbot. The chatbot, equipped with a camera, observes the game screen and utilizes a convolutional neural network and a support vector machine to identify game movements. Additionally, both its own speech and human speech are fed as inputs to the deep Q-network. Results indicate that after 3000 rounds of gameplay, the deep Q-learning method achieves a success rate of 82.58%.

Another study exploring the use of deep reinforcement learning methods focuses on examining deep Q-networks [22]. The findings demonstrate that deep Q-networks can be effective and converge when the problem space has a limited number of actions. However, as the number of actions increases, the performance of the Q-network deteriorates. Overall, the efficacy of deep Q-networks and other reinforcement learning methods heavily relies on the design of their reward functions.

In the realm of natural language understanding, dialogue management, and natural language generation, collaborative research has been conducted. In [23], multiple deep neural networks are employed for different components of the dialogue system. In this approach, dialogues are structured as sequential sequences, and at each step of the conversation, the dialogue system takes a sequence of user input tokens and transforms them into two representation vectors using the domain network and state tracking network. Another approach focused on the restaurant domain [24], utilizes three deep networks in conjunction with a simulator.

In this method, a long short-term memory network is utilized for natural language understanding, performing tasks such as domain identification and intent identification as a classifier, as well as resolving concepts by filling in certain slots. Additionally, a deep Q-network is employed for dialogue management, while a long short-term memory network is used to generate natural language responses. A different method is presented in [3], where reinforcement learning is employed to learn policies. In this approach, the natural language understanding component consists of a bidirectional long short-term memory network, with the user's message as input. The output vector of this network is concatenated with the previous output and fed into another long short-term memory network responsible for state tracking and action selection using the policy network. In [25], a method is proposed that utilizes a chatbot developed by Amazon for responding to user legal questions. The system incorporates two smaller chatbots, with one handling routine questions and the other collecting user information to address specialized questions.

Another method involves utilizing the GPT-2 language model to develop a task-oriented dialogue system [4]. This approach involves fine-tuning a pre-trained language model using a large amount of labeled data from various domains.

The data used in this method consists of belief states and knowledge base results, which are provided as plain text along with the chatbot and human messages. The GPT-2 language model is then used to generate the next word in the dialogue. In a similar vein, [26] proposes a method that incorporates the entire history of the dialogue as input to the GPT-2 language model. Here, the token representations of the dialogue are fed into the GPT-2 model to predict the next word. Furthermore, [27] introduces a method that builds upon the GPT-2 language model by translating the data into 10 different languages to augment the dataset. This approach randomly selects a sentence from among the 10 languages and trains the GPT-2 model accordingly.

To the best of our knowledge, research on multilingual dialogue systems, particularly those leveraging deep reinforcement learning, remains limited. This is especially true for low-resource languages, such as Persian, where the availability of annotated datasets and prior research is scarce.

In [28], a joint learning-based BERT transformer model was proposed for Persian online shopping dialogue systems, advancing the natural language understanding component but without exploring dialogue management or reinforcement learning techniques. To address the lack of annotated data, they introduced two dataset generation methods: fully-simulated and semi-simulated. Similarly, in [29], PerSHOP, a large Persian dataset for shopping dialogue systems, was developed, and baseline models for intent identification were proposed.

Proposed Method

The proposed method for constructing a joint dialogue system is based on deep reinforcement learning, utilizing the SBERT model to construct the environment state. The architecture of the dialogue system can be seen in Fig. 1. The deep reinforcement learning component, which is built on a deep q network, handles various tasks such as dialogue management component, and domain identification and intent identification from natural language understanding. This component is crucial in the overall functionality of the dialogue system.

The output generated by the deep reinforcement learning component is then passed to the natural language generation component. This component utilizes hand-crafted rules to generate coherent sentences. As shown in Fig. 1, the proposed method is divided into two parts. The lower part pertains to the reinforcement learning process and the interaction with the user, while the upper part represents the procedure before the learning process and its goal is training a classification model.

To facilitate training of the deep reinforcement learning system, we employ a user simulator. This simulator is ingeniously designed to emulate human interactions, thereby generating a plethora of dialogues that contribute to the system's learning and adaptability. A set of sentence templates, along with a set of slots for filling these templates, must be predefined as input to the simulator. The simulator selects a template as output and, based on its predefined goals, populates the slots and generates responses to the chatbot's messages. These goals mirror the objectives a human would have when engaging in a task-oriented conversational system.

Subsequently, different components of the system are explained and discussed.

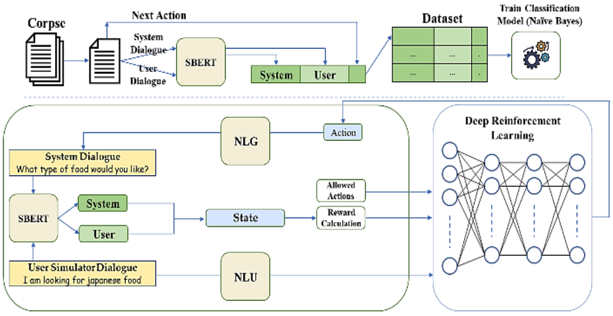


Fig. 1: The proposed architecture, known as SenSimpleDS.

First, the chatbot's action set is extracted from the dataset. Then, dialogues in the corpus are utilized to construct a Dataset. To create the Dataset, each dialogue in the corpus is processed into dialogue steps. For each dialogue step, the SBERT model is employed to generate vectors representing the chatbot's and human's last messages. These vectors are then concatenated to form a new vector, which is associated with the next action of the chatbot.

This process is repeated for all dialogue steps in the dataset, resulting in the generation of vectors for all dialogues. These vectors are used to construct a matrix, with each row representing a vector. This matrix is referred to as the Dataset. We use this Dataset for training a naive Bayes classifier.

This classifier is utilized to calculate rewards and determine allowed actions for the deep reinforcement learning process.

A. Environment State

In this method, the state of the environment is constructed using the SBERT model. The text of the recent message from both the chatbot and the human are provided as input to the SBERT model. The output of the SBERT model is representation vectors for messages. These two vectors are concatenated to create a new vector, which represents the environment state.

B. Allowed Action

In this method, the computation of allowed actions for deep reinforcement learning is performed using a pre-trained naïve Bayes classifier.

The environment state is inputted into the classifier, and actions with probabilities greater than p_{min} are considered as the set of allowed actions for reinforcement learning at that particular dialogue step. However, it is important to manage the allowed actions by removing actions that are not feasible and including possible actions.

For instance, once an action is selected, it should be excluded from the set of allowed actions.

Additionally, certain actions that require more information or context, such as when the chatbot needs specific details to complete a sentence, should be removed from the allowed actions. For example, if the chatbot is expected to ask, "Did you say Indian food?" but does not yet have the value for the "food" slot as "Indian", it cannot choose an action and generate that speech.

C. Reward Calculation

In this method, a reward function similar to the reward function of the SimpleDS method is utilized to calculate the reward for deep reinforcement learning. The reward for performing an action a in state s is derived from (1). The term CR in this equation is obtained using (2). In (2), DR represents the similarity between the current state and vectors in the dataset when action a is taken. This similarity is determined using a pre-trained naive Bayes classifier, which is discussed earlier. DL is a constant value that encourages more efficient dialogues. Specifically, an efficient dialogue is one where the chatbot can fill all three slots in a single step, which is more rewarding compared to filling the three slots in three separate steps. The reward function incorporates two coefficients, α and β, which determine the relative weight given to each part of the reward function.

$$R(s,a) = (\alpha \times CR) + (\beta \times DR) - DL \tag{1}$$

where R represents the value of the reward function for state s by performing action a.

$$CR = \frac{ca}{cb} \tag{2}$$

where ca is the number of filled slots with action a, and cb is the total number of slots.

For training deep reinforcement learning models, an additional reward function is utilized, as described in (3). In this new function, the DL term is modified within the reward function, and the NSP term is introduced. The training methodology that incorporates this modification is referred to as SenSimpleDS-NSP.

$$R(s,a) = (\alpha \times CR) + (\beta \times DR) + (\gamma \times NSP) - DL \tag{3}$$

where γ is a coefficient that regulates the influence of the NSP term.

The NSP value for a given state s and action a is obtained using the BERT language model. The BERT language model is employed to predict the next sequence. In this context, the dialogue texts of the chatbot and the simulator are considered as the first input sequence, while the current text of the chatbot is regarded as the second input sequence. These two sequences are provided to the BERT language model, which is expected to predict the second sequence following the first sequence. The inclusion of the NSP term in the reward function aims to leverage the next sequence prediction capability of language models to guide action selection. Specifically, if there is a higher (lower) probability of a particular sentence occurring after the current statements, the agent will accumulate additional (fewer) rewards.

D. Reduce Epsilon

In this dialogue system, the ε-greedy approach is employed for deep reinforcement learning training to strike a balance between exploration and exploitation. To decrease the value of ε, two functions are utilized in this method. The first function, represented by (4), employs a linear function to gradually reduce the value of ε. In this context, l denotes the number of learning steps used for training deep reinforcement learning, while b represents the number of steps in which the agent acts randomly in the environment.

$$\varepsilon_{new} = \min(1, \max(\varepsilon_{min}, 1 - \frac{(age - b)}{(l - b)})) \quad (4)$$

where ε represents the new value of ε, ε_{min} denotes the minimum possible value for ε, and age signifies the agent's age (with each learning step equating to a year). The fractional term in (4). Increases the value of ε gradually as the learning progresses, from the start of the learning steps (denoted by l) up until the value of ε_{min} is reached.

In the second function, depicted in (5), another function is utilized to decrease the value of ε. In this case, β represents a decreasing factor. A system trained using this second function is referred to as SenSimpleDS+. Additionally, a system that incorporates NSP in the reward function is known as SenSimpleDS-NSP+.

$$\varepsilon_{new} = \max(\varepsilon_{min}, (\varepsilon \times \beta)) \quad (5)$$

where ε represents the current value of ε.

The steps involved in the dialogue between the chatbot and the simulator in a single dialogue step are outlined in Algorithm 1, which is presented in the form of pseudocode.

Algorithm 1 Steps involved in the dialogue between the chatbot and the simulator

```
Start the conversation
Choose the initial action $a_{start}$ with the help of the chatbot
while (final action $a_{end}$) do
    Build the environment state
    Classify the environment state using a naive Bayes classifier
    Calculate the rewards for all actions
    Calculate the allowed actions
    Provide the environment state, rewards, and allowed actions to the deep Q-network
    Select the next action using the deep Q-network
    Generate the chatbot's speech using a natural language generation component based on the selected action
    Generate the simulator's message based on the chatbot's message
    Identify the speech slots of the simulator using natural language understanding component
End the conversation
```

Experiments

The proposed dialogue system has been implemented using Python and Google Colab. For deep reinforcement learning training, the environment and the agent are entirely separated, and their interaction is based on the text of messages, actions, and rewards.

To assess the performance of the proposed methods and compare them with other approaches, an English dataset publicly available has been utilized. All methods have been trained and evaluated using 3,000 dialogue turns with the user simulator. The input layer of the deep Q-network consists of 1536 nodes, which is equivalent to the length of the state vector. The network includes two hidden layers, each containing 40 nodes, and these hidden layers employ the ReLU activation function. The output layer of the network comprises 36 nodes, matching the number of actions available to the chatbot. This layer utilizes a linear activation function. We use an argmax on the network's output and action with the highest Q-value is selected. The total number of network parameters is 64,596. The loss function used by the network is the mean square error. The discount factor γ in the $Q(s_t, a_t)$ function is set to 0.7. The deep Q-network is trained using mini-batch stochastic gradient descent, with a learning rate of 0.001 and a minibatch size of 32. The weights of the deep Q-network layers are initialized as zeros at the start of the learning process. Additionally, the l_2 method is employed to adjust the weights during training. To train the deep Q-network, Experience Replay is utilized with a size of 10,000. Furthermore, a target network is employed during the deep reinforcement learning process, and it is updated every 2500 learning epochs. The allowed actions are calculated using a naive Bayes classifier with a probability threshold (p_{min}) of 0.001.

A. Provided Dataset

The existing conversations in the SimpleDS dataset are designed to create a recommendation system that suggests restaurants based on user preferences. However, it is possible that when the conversation pertains to the restaurant domain, it is about discussing matters within the restaurant and ordering food. With this in mind, a new dataset has been provided, consisting of conversations simulated within a restaurant environment, which is publicly accessible.

The simulation involves a restaurant operator and five other customers who visit the restaurant three times to purchase food.

The properties of the provided dataset are compared to the SimpleDS dataset shown in Table 1.

The provided dataset includes 25 complete conversations, from which 75 actions are extracted for the chatbot. The simulator also incorporates 95 sentences, which are completed using seven different types of slots.

In this dataset, the human participants pursue a wider range of goals compared to those in the SimpleDS dataset, increasing the complexity and diversity of interactions. Additionally, the dialogue system's natural language generation component is enhanced by providing multiple textual variations for certain actions, which are randomly selected during conversations. This randomization improves the linguistic variety and naturalness of the system's responses. Furthermore, the dataset includes an expanded set of chatbot actions and simulator slots, making it more intricate and better suited for testing advanced dialogue management systems than the SimpleDS dataset.

Table 1: Comparison of details between the provided dataset and the SimpleDS dataset

	Provided Dataset	**SimpleDS Dataset**
Number of Conversations	25	6
Number of chatbot Actions	75	36
Number of Simulator Slots	7	3
Number of Simulator Sentences	95	33
Example chatbot Sentence	What kind of appetizers and drinks would you like?*	What type of food would you like?

*Translated to English

Result and Discussion

In this section, we evaluate the performance of the proposed methods on both the English and Persian datasets.

To measure the efficacy of these methods, we apply a moving average to the rewards accumulated by the deep reinforcement learning agent over 3000 dialogue turns with the user simulator. Specifically, the average reward is computed within a sliding window of 1000 steps, capturing the most recent interactions.

After every 100 learning steps, the rewards in the current window are averaged to track the agent's learning progress.

The performance of the SenSimpleDS, SenSimpleDS+, and SenSimpleDS-NSP methods is compared based on the rewards obtained, as shown in Fig. 2. All three methods consistently outperform the baseline models (SimpleDS and SCGSimpleDS) toward the later stages of training. Notably, the SenSimpleDS+ method reaches a higher cumulative reward by the 5000th learning step, surpassing both SimpleDS and SCGSimpleDS.

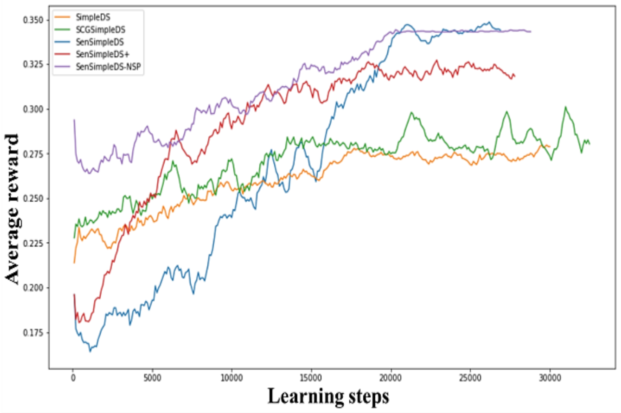


Fig. 2: Average reward collected by the proposed methods compared to other methods in 3000 dialogue turn.

In contrast, the SenSimpleDS method achieves this performance level at the 15,500th step, indicating a slower convergence.

This performance difference is attributed to the utilization of an enhanced ε-reduction function in the SenSimpleDS+ method, which accelerates exploration-exploitation balancing. Initially, the SenSimpleDS and SenSimpleDS+ methods yield lower rewards than SimpleDS and SCGSimpleDS due to their larger input dimensionality.

The increased complexity of the input representations causes the deep Q-network to require more time to converge, as it processes richer and more complex data. Additionally, the SenSimpleDS-NSP method demonstrates stable reward collection earlier in the training, a stability not observed in the other methods, further highlighting its robustness.

As illustrated in Fig. 2, prior to the 19,200th learning step, the SenSimpleDS+ method consistently achieves higher rewards than the SenSimpleDS method.

This performance gain is attributed to the use of the second function for reducing ε, which allows the deep reinforcement agent to more effectively exploit the knowledge acquired during earlier learning phases, leading to higher cumulative rewards. However, beyond the 19,200th learning step, the SenSimpleDS method surpasses SenSimpleDS+ in terms of reward accumulation. This can be explained by the adoption of the linear function for reducing ε, which enables the agent to explore the environment more efficiently in earlier stages, eventually leveraging the knowledge it has gathered to achieve superior performance in later steps.

Before the 20,000th learning step, the SenSimpleDS-NSP method demonstrates the highest reward among all approaches.

This improvement can be attributed to the integration of the Next Sentence Prediction component within its reward function, enhancing the agent's ability to capture longer-term conversational dependencies. Nevertheless, after the 20,000th learning step, the rewards obtained by SenSimpleDS-NSP converge to levels comparable to the SenSimpleDS method, with the added benefit that the SenSimpleDS-NSP method maintains a more consistent reward trajectory, indicating greater stability in performance.

As shown in Fig. 2, the proposed SenSimpleDS, SenSimpleDS+, and SenSimpleDS-NSP methods require significantly fewer learning steps compared to the baseline methods. This suggests that the proposed approaches enable the agent to achieve its objectives in fewer dialogue turns, thereby facilitating more efficient interactions.

The maximum average rewards achieved by each method, along with the total number of dialogue steps required, are summarized in Table 2.

Table 2: Maximum collected rewards and total dialogue steps for each method

Method Name	Maximum Reward	Total Dialogue Steps
SimpleDS	0.2797	30,000
SCGSimpleDS	0.3012	32,500
SenSimpleDS	**0.3489**	**26,900**
SenSimpleDS+	0.3272	27,800
SenSimpleDS-NSP	0.3442	28,800

A. *Human Evaluation*

The methods SenSimpleDS, SenSimpleDS+, SenSimpleDS-NSP, SimpleDS, and SCGSimpleDS, were evaluated through human judgment. A total of 10 dialogues, comprising two conversations from each method, were presented to a panel of referees.

They assessed these dialogues based on four predefined criteria. It is noteworthy that this human evaluation closely follows the methodology outlined in [30].

1. **Quality:** How would you rate the overall quality of the conversation?
2. **Fluency:** How fluent did you find the system's (SYS) responses? (This includes the correct use of grammar and sentence structure.)
3. **Non-repetitiveness:** How diverse were the system's (SYS) responses in terms of avoiding repetition? (This pertains to the variety in sentence construction.)
4. **Mutual Understanding:** How well did the system (SYS) demonstrate mutual understanding in the conversation? (This refers to the emotional and empathetic dimensions of the dialogue.)

The referees were asked to rate their responses on a six-point Likert scale, ranging from "very low" to "very high." The individual ratings for each method were aggregated, and an average score was computed. A score of 0 indicated a "very low" rating, while a score of 5 represented a "very high" rating. The results of the human evaluation, which involved 15 referees, are depicted in Fig. 3, presented in the form of bar charts for comparative analysis.

According to the Quality criterion, the proposed SenSimpleDS method received the highest score, followed closely by the SenSimpleDS-NSP method. The SenSimpleDS+ and SCGSimpleDS methods showed only minor differences in their quality scores. The SimpleDS method was rated the lowest in terms of quality by the referees.

For the Fluency criterion, most methods achieved comparable scores, though the SenSimpleDS method exhibited a slightly higher fluency in the dialogue, indicating a more coherent and grammatically sound interaction.

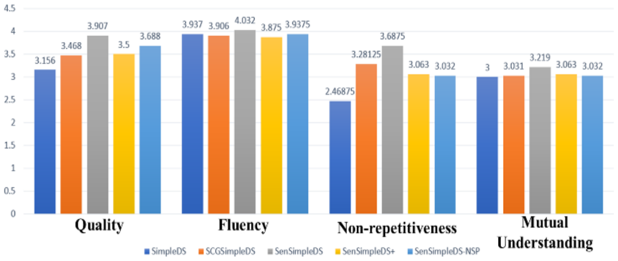


Fig. 3: Human evaluation results of SenSimpleDS, SenSimpleDS+, SenSimpleDS-NSP, SimpleDS, and SCGSimpleDS methods.

In terms of the Non-repetitiveness criterion, the SenSimpleDS method outperformed the others, receiving the highest ratings. This outcome reflects the system's ability to generate more varied responses throughout the learning steps. Following SenSimpleDS, the SCGSimpleDS, SenSimpleDS+, and SenSimpleDS-NSP methods were ranked, respectively. According to the referees, SimpleDS method produced more repetitive conversations compared to the other approaches. For the Mutual Understanding criterion, both the SenSimpleDS and SenSimpleDS+ methods attained higher scores, indicating stronger empathy and contextual understanding in their responses. The referees observed that the SenSimpleDS-NSP, SCGSimpleDS, and SimpleDS methods exhibited similar levels of mutual understanding, with little distinction among them.

B. *Performance Evaluation on the Provided Dataset*

To thoroughly evaluate the performance of the proposed dialogue systems, the methods were trained and tested on the provided Persian dataset over 3000 dialogue turns, as shown in Fig. 4.

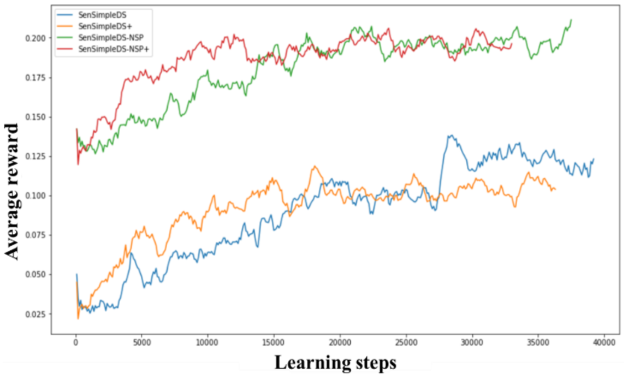


Fig. 4: Average rewards collected by the proposed methods over 3000 dialogue turns on the provided dataset.

The results demonstrate that all methods contributed to the overall average of rewards collected during training, confirming the successful training of the deep Q-networks. Notably, the integration of the Next Sentence Prediction component into the reward function resulted in significantly higher reward levels. Additionally, consistent with previous findings, methods utilizing the second function for ε reduction achieved higher rewards during the early stages of training, continuing up to the midpoint. Furthermore, the results indicate that, compared to methods employing a linear ε-reduction strategy, those using the second function required fewer learning steps to achieve similar outcomes, suggesting that these methods enable humans to reach their conversational goals more efficiently. The total number of learning steps for each method, along with their corresponding maximum reward values, are detailed Table 3. Furthermore, the SimpleDS and SCGSimpleDS methods have not been implemented for the Persian language.

Table 3: Maximum reward and total dialogue steps for the proposed methods on the provided dataset

Method Name	Total Dialogue Steps	Maximum Reward
SenSimpleDS	39,200	0.1314
SenSimpleDS+	36,300	0.1187
SenSimpleDS-NSP	37,500	**0.2114**
SenSimpleDS-NSP+	**33,000**	0.2060

This limitation can be attributed to the constraints of the GloVe word representation in Persian, as both methods rely on GloVe embeddings for constructing the environment state. Notably, GloVe has not been formally trained on large-scale Persian corpora, though unofficial versions have been released. However, due to the lack of support for out-of-vocabulary (OOV) words in these unofficial versions, constructing a robust dialogue system in Persian using GloVe embeddings presents significant challenges.

C. Performance of Multilingual Dialogue System

In this section, we evaluate the performance of the multilingual SenSimpleDS method across four languages: English, Persian, Spanish, and German. This method incorporates a random forest algorithm for language detection, and the environment state is constructed using the multilingual MPNet model [31]. The method's performance is comparable to that of the multilingual simulator, as depicted in Fig. 5.

The reward progression, as tracked by the Q-network during the learning steps, demonstrates the effectiveness of deep reinforcement learning in a multilingual dialogue system.

However, this approach yields slightly lower reward levels compared to single-language methods, likely due to the additional complexity introduced by handling multiple languages. After 20,000 learning steps, the reward stabilizes.

The maximum reward achieved by the multilingual SenSimpleDS method is 0.2544, which is reached after 26,200 learning steps, with the total number of steps taken by the model amounting to 29,800. The multilingual SenSimpleDS method does not differentiate actions based on individual languages. Instead, it integrates a language detection component to manage language-specific tasks.

However, the majority of research in the field of reinforcement learning and deep reinforcement learning has primarily focused on developing single-language dialogue systems [32].

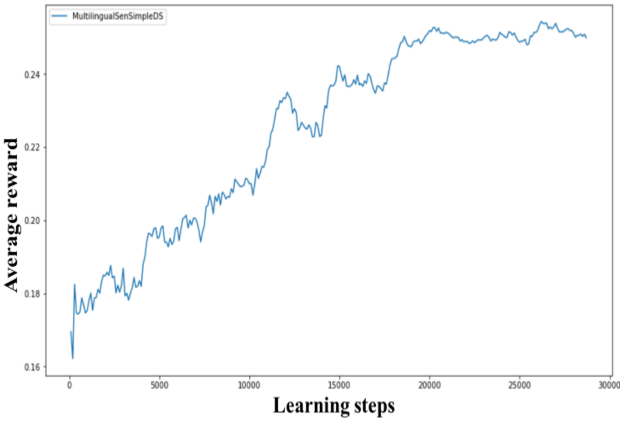


Fig. 5: The average reward collected by the multilingual SenSimpleDS method in 3000 dialogue turns is calculated.

Conclusion

In this paper, we introduced SenSimpleDS, a joint task-oriented dialogue system based on deep reinforcement learning, designed to handle multilingual conversations. The system employs a deep Q-network along with the SBERT model to represent the environment. Several variants of the system were proposed, including SenSimpleDS+, which incorporates an alternative function for reducing ε, and SenSimpleDS-NSP, which leverages next sequence prediction using BERT to influence the reward function based on dialogue history.

Through rigorous evaluation on both English and Persian datasets in the restaurant domain, the proposed methods demonstrated superior performance in terms of collected rewards, compared to baseline methods such as SimpleDS and SCGSimpleDS.

The incorporation of representations like SBERT and the use of a simulator allowed for efficient training without human intervention.

The experimental results show that the three proposed methods not only achieved higher rewards but also required fewer learning steps, thereby improving the efficiency of the dialogue systems. In particular, the SenSimpleDS-NSP method benefited from a more sophisticated reward function, leading to improved performance during training.

Additionally, the multilingual version of SenSimpleDS, which integrates a random forest for language detection and the multilingual MPNet for environment construction, demonstrated the feasibility of deep reinforcement learning in developing multilingual dialogue systems.

While the proposed system achieved promising results, several areas remain for further exploration. First, more advanced deep reinforcement learning methods could be applied, such as policy gradient methods or actor-critic architectures, to enhance the learning capabilities of the system.

Additionally, incorporating recursive neural networks into the environment construction could allow for more effective utilization of dialogue history, improving the system’s ability to handle longer and more complex conversations.

In the natural language generation component, machine learning or deep learning techniques could be employed to enhance the system’s ability to generate more diverse and contextually appropriate responses. Furthermore, the integration of speech processing capabilities, such as automatic speech recognition (ASR) and text-to-speech (TTS) systems, would enable the dialogue system to support spoken interactions, expanding its usability in real-world applications.

Expanding the system to other domains, such as ticket booking or customer support, and employing transfer learning techniques to apply knowledge from one domain to another, presents another promising avenue for future research.

Additionally, the newly introduced Persian dataset will facilitate further development and evaluation of dialogue systems in the restaurant domain, providing a foundation for future multilingual and multi-domain dialogue systems.

Author Contributions

M. J. Nasri-Lowshani, J. Salimi-Sartakhti, and H. Ebrahimpour-Komle conceptualized the ideas and proposed the development of the model structure. M. J. Nasri-Lowshani implemented the proposed ideas and collected the dataset. The experiments were conducted by M. J. Nasri-Lowshani under the supervision of J. Salimi-Sartakhti and H. Ebrahimpour-Komle. M. J. Nasri-Lowshani and J. Salimi-Sartakhti wrote the manuscript, presenting the results and experiments.

Acknowledgment

The authors extend their heartfelt thanks to the reviewers and editors of JECEI for their insightful feedback and thoughtful recommendations, which have greatly improved the quality of this manuscript.

Funding

This research received no external funding.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

References

[1] X. Wang, C. Yuan, "Recent advances on human-computer dialogue," CAAI Trans. Intell. Technol., 1(4): 303-312, 2016.

[2] H. Chen, X. Liu, D. Yin, J. Tang, "A survey on dialogue systems: recent advances and new frontiers," SIGKDD Explor. Newsl., 19(2): 25-35, 2017.

[3] B. Liu, G. Tür, D. Hakkani-Tür, P. Shah, L. Heck, "Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems," in Proc. Human Language Technologies: 2060-2069, 2018.

[4] P. Budzianowski, I. Vulić, "Hello, It's GPT-2 - How can i help you? towards the use of pretrained language models for task-oriented dialogue systems," in Proc. 3rd Workshop on Neural Generation and Translation: 15-22, 2019.

[5] F. Almeida, G. B. Xexéo, "Word embeddings: A survey," ArXiv, abs/1901.09069, 2019.

[6] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. Human Language Technologies: 4171-4186, 2019.

[7] J. Pennington, R. Socher, C. Manning, "GloVe: Global vectors for word representation," in Proc. Empirical Methods in Natural Language Processing (EMNLP): 1532-1543, 2014.

[8] N. Reimers, I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-Networks," in Proc. Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): 3982-3992, 2019.

[9] X. Zhang, H. Wang, "A joint model of intent determination and slot filling for spoken language understanding," in Proc. 25th International Joint Conference on Artificial Intelligence: 2993-2999, 2016.

[10] M. Rafiepour, J. S. Sartakhti, "CTRAN: CNN-Transformer-based network for natural language understanding," Eng. Appl. Artif. Intell., 126(PC): 9, 2023.

[11] Y. Shi, K. Yao, H. Chen, Y. C. Pan, M. Y. Hwang, B. Peng, "Contextual spoken language understanding using recurrent neural networks," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP): 5271-5275, 2015.

[12] Z. Yan, N. Duan, P. Chen, M. Zhou, J. Zhou, Z. Li, "Building task-oriented dialogue systems for online shopping," in Proc. AAAI conference on artificial intelligence, 31(1): 4618-4625, 2017.

[13] T. H. Wen, M. Gašić, N. Mrkšić, P. H. Su, D. Vandyke, S. Young, "Semantically conditioned LSTM-based natural language generation for spoken dialogue systems," in Proc. Empirical Methods in Natural Language Processing: 1711-1721, 2015.

[14] O. Dušek, F. Jurčíček, "Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings," in Proc. 54th Annual Meeting of the Association for Computational Linguistics, 2: 45-51, 2016.

[15] Z. Jiang, X. L. Mao, Z. Huang, J. Ma, S. Li, "Towards end-to-end learning for efficient dialogue agent by modeling looking-ahead ability," in Proc. 20th Annual SIGdial Meeting on Discourse and Dialogue: 133-142, 2019.

[16] H. Cuayáhuitl, "SimpleDS: A simple deep reinforcement learning dialogue system," in Proc. International Workshop on Spoken Dialogue Systems Technology, 2016.

[17] H. Cuayáhuitl, S. Yu, A. Williamson, J. Carse, "Deep reinforcement learning for multi-domain dialogue systems," ArXiv, abs/1611.08675, 2016.

[18] H. Cuayáhuitl, S. Yu, A. Williamson, J. Carse, "Scaling up deep reinforcement learning for multi-domain dialogue systems," in Proc. International Joint Conference on Neural Networks (IJCNN): 3339-3346, 2017.

[19] Z. Dehghanipour, J. Salimi, "An improved deep reinforcement learning for task-oriented dialogue system," Preprint, 2022.

[20] V. Ilievski, C. Musat, A. Hossmann, M. Baeriswyl, "Goal-oriented chatbot dialog management bootstrapping with transfer learning," in Proc. 27th International Joint Conference on Artificial Intelligence: 4115-4121, 2018.

[21] H. Cuayáhuitl, "A data-efficient deep learning approach for deployable multimodal social robots," Neurocomputing, 396: 587-598, 2020.

[22] Y. Ma, X. Wang, Z. Dong, H. Chen, "Cascaded LSTMs based deep reinforcement learning for goal-driven dialogue," in Proc. Natural Language Processing and Chinese Computing: 29-41, 2018.

[23] T. H. Wen, D. Vandyke, N. Mrkšić, M. Gašić, L. M. Rojas-Barahona, P. H. Su, S. Ultes, S. Young, "A network-based end-to-end trainable task-oriented dialogue system," in Proc. 15th Conference of the European Chapter of the Association for Computational Linguistics, 1: 438-449, 2017.

[24] X. Li, Y. N. Chen, L. Li, J. Gao, A. Celikyilmaz, "End-to-end task-completion neural dialogue systems," in Proc. 8th International Joint Conference on Natural Language Processing, 1: 733-743, 2017.

[25] M. Sharma, T. Russell-Rose, L. Barakat, A. Matsuo, "Building a legal dialogue system: development process, challenges and opportunities," ArXiv, abs/2109.00381, 2021.

[26] D. Ham, J. G. Lee, Y. Jang, K. E. Kim, "End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2," in Proc. 58th Annual Meeting of the Association for Computational Linguistics: 583-592, 2020.

[27] J. Kulhánek, V. Hudeček, T. Nekvinda, O. Dušek, "AuGPT: Auxiliary tasks and data augmentation for end-to-end dialogue with pre-trained language models," in Proc. 3rd Workshop on Natural Language Processing for Conversational AI: 198-210, 2021.

[28] Z. Borhanifard, H. Basafa, S. Z. Razavi, H. Faili, "Persian language understanding in task-oriented dialogue system for online shopping," in Proc. 11th International Conference on Information and Knowledge Technology (IKT): 79-84, 2020.

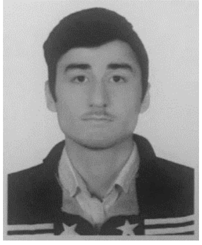
[29] K. Mahmoudi, H. Faili, "PerSHOP--A Persian dataset for shopping dialogue systems modeling," ArXiv, abs/2401.00811, 2024.

[30] A. Ghandeharioun, J. H. Shen, N. Jaques, C. Ferguson, N. Jones, A. Lapedriza, R. Picard, "Approximating interactive human evaluation with self-play for open-domain dialog systems," in Proc. 33rd International Conference on Neural Information Processing Systems: 13665-13676, 2019.

[31] N. Reimers, I. Gurevych, "Making monolingual sentence embeddings multilingual using knowledge distillation," in Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP): 4512-4525, 2020.

[32] E. Razumovskaia, G. Glavas, O. Majewska, E. M. Ponti, A. Korhonen, I. Vulic, "Crossing the conversational chasm: A primer on natural language processing for multilingual task-oriented dialogue systems," J. Artif. Intell. Res., 741351-1402, 2022.

Biographies



Mohammad Javad Nasri-Lowshani was born in Manjil, Iran, on October 18, 1998. He received his B.S. degree in Computer Engineering from Shahreza University in 2020 and M.Sc. degree in Artificial Intelligence from University of Kashan in 2023. His major research interests include Machine Learning, Natural Language Processing, Language Model and Dialogue System.

- Email: mohammad.j.nasri@gmail.com
- ORCID: 0009-0000-2978-9330
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: www.linkedin.com/in/mohammadjavadnasri

Javad Salimi-Sartakhti received his B.S. degree in Computer Engineering (Software) from University of Kashan in 2009, his M.Sc. degree in Computer Engineering (Software) from Tarbiat Modares University in 2012 and his Ph.D. degree in Computer Engineering (Artificial Intelligence) from Isfahan University of Technology in 2016. He is an Associate Professor in Dept. Artificial Intelligence at University of Kashan. His research interests include Game Theory, Machine Learning, Natural Language Processing, and Large Language Model.

- Email: salimi@kashanu.ac.ir
- ORCID: 0000-0003-1183-1232
- Web of Science Researcher ID: HJY-2812-2023
- Scopus Author ID: 51864592100
- Homepage: https://faculty.kashanu.ac.ir/salimi/en

Hossein Ebrahimpour-Komleh received his B.S. degree in Computer Engineering (Hardware) from Isfahan University of Technology in 1994, his M.Sc. degree in Computer Engineering (Artificial Intelligence) from Amirkabir University of Technology in 1997 and his Ph.D. degree in Computer Engineering (Artificial Intelligence) from QUT Queensland University of Technology, Australia in 2004. He is an Assistant Professor in Dept. Artificial Intelligence at University of Kashan. His research interests include Computer Vision, Pattern Recognition and Medical Image Processing.

- Email: ebrahimpour@kashanu.ac.ir
- ORCID: 0000-0002-9935-7821
- Web of Science Researcher ID: ABG-2658-2020
- Scopus Author ID: 36924805200
- Homepage: https://faculty.kashanu.ac.ir/ebrahimpour/en