JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

ROLE OF DEEP REINFORCEMENT LEARNING IN NATURAL LANGUAGE PROCESSING

Dr. Ikvinderpal Singh
Assistant Professor, PG Department of Computer Science & Applications,
Trai Shatabdi Guru Gobind Singh Khalsa College, Amritsar (Punjab)

Abstract
Deep Reinforcement Learning (DRL) has shown significant promise in various fields of artificial intelligence, especially in decision-making tasks where agents need to interact with environments. In recent years, DRL has made considerable strides in Natural Language Processing (NLP), where its role is becoming increasingly pivotal. This paper explores the integration of DRL in NLP, focusing on its ability to address challenges such as sequential decision-making, dialogue management, machine translation, text generation, and semantic understanding. We review various DRL-based approaches applied to NLP tasks, discuss their strengths and limitations, and highlight key developments, including reinforcement learning-based reward shaping, policy learning, and exploration strategies. Additionally, we examine how DRL has enabled advancements in conversational AI, question answering systems, and sentiment analysis, among others. Finally, we outline the challenges facing DRL in NLP, such as sample efficiency, interpretability, and computational costs, and propose future research directions for enhancing its effectiveness in NLP applications.

Keywords: Deep Reinforcement Learning (DRL); Natural Language Processing (NLP); Hierarchical Reinforcement Learning (HRL);

1. Introduction

Natural Language Processing (NLP) has long been a central area of research within artificial intelligence (AI), focusing on enabling machines to understand, generate, and manipulate human language. NLP plays a crucial role in applications ranging from text classification and sentiment analysis to machine translation and dialogue systems [1]. Traditional approaches to NLP have primarily relied on supervised learning models that require large, labeled datasets for training. While these methods have led to significant progress in tasks like sentiment analysis, named entity recognition, and machine translation, they often face several inherent challenges that limit their effectiveness. [2] These include the difficulty in handling sequential dependencies in language, the reliance on vast amounts of annotated data, and the lack of flexibility in addressing tasks that require dynamic decision-making or long-term optimization.

To address some of these challenges, Deep Reinforcement Learning (DRL) has emerged as a promising approach in NLP. DRL combines the principles of reinforcement learning (RL) with deep learning techniques, enabling machines to learn optimal strategies through interaction with their environment rather than relying on large labeled datasets. This ability to learn from trial and error allows DRL to solve tasks that require sequential decision-making, where the optimal action is not always clear at the outset, and the long-term consequences of actions must be considered.

In DRL, an agent learns by taking actions in an environment and receiving feedback in the form of rewards or penalties, which guide the agent toward achieving specific goals. This makes DRL particularly suited for tasks in NLP where long-term dependencies between words or sentences need to be captured, such as in dialogue management, machine translation, text summarization, and question answering. Unlike traditional supervised learning approaches, DRL models do not require extensive labeled datasets. Instead, they can learn from the interaction itself, allowing them to generalize better in real-world settings where labeled data may be scarce or expensive to obtain.

The potential of DRL in NLP is immense, as it enables machines to generate more natural, contextually appropriate language in a wide variety of tasks. DRL can be applied to tasks that require the generation of long, coherent text sequences, such as in language generation, creative writing, and conversational AI, where traditional models may struggle. Furthermore, DRL's ability to optimize over long-term rewards makes it an ideal choice for building systems that require continuous learning and adaptation, such as in real-time dialogue systems and dynamic question-answering systems.

However, the integration of DRL into NLP is not without its challenges. DRL models, while powerful, are often data-hungry, requiring substantial amounts of interaction data to effectively learn useful policies. Moreover, defining appropriate reward functions that encourage good performance in complex NLP tasks can be difficult, as the rewards must capture not only linguistic correctness but also context relevance, coherence, and user engagement. Furthermore, DRL's inherent complexity can make it difficult to interpret or explain its decision-making processes, which is a significant concern in applications where model transparency is critical.

This paper explores the role of DRL in enhancing various NLP tasks, providing an overview of how DRL techniques have been integrated into different NLP domains. By examining recent advancements, we aim to highlight the strengths and potential of DRL in improving the flexibility, efficiency, and performance of NLP models. We hope to demonstrate that DRL is a powerful tool that has the capacity to significantly advance the field of NLP by addressing the shortcomings of traditional methods and offering new, more adaptive approaches to language understanding, generation, and decision-making.

2. Background

Natural Language Processing (NLP) has been one of the most active areas of research within the field of artificial intelligence (AI), aiming to enable computers to understand, generate, and manipulate human language. Over the past few decades, NLP has seen significant progress due to advances in machine learning, particularly in supervised learning techniques. However, traditional NLP approaches have faced limitations, especially in dealing with dynamic, sequential, and contextual decision-making problems where the outcome depends on long-term dependencies between words or actions. Reinforcement Learning (RL) and, more recently, Deep Reinforcement Learning (DRL) have emerged as promising solutions to address these limitations, enabling more flexible and adaptive NLP models.

2.1 Reinforcement Learning (RL)

Reinforcement Learning (RL) is a type of machine learning in which an agent learns by interacting with an environment and receiving feedback in the form of rewards or penalties. The goal of RL is to maximize the cumulative reward through a series of decisions made over time. Unlike supervised learning, where the model is trained on labeled data, RL relies on the agent's exploration of the environment and its ability to learn from experience [3].

In the context of NLP, RL is particularly useful in scenarios where the agent must make a sequence of decisions over time, such as generating a sentence or interacting in a conversation. RL models can optimize these sequential decision-making tasks by reinforcing actions that lead to better outcomes and penalizing actions that result in poor performance.

2.2 Deep Reinforcement Learning (DRL)

[4] Deep Reinforcement Learning (DRL) is a subfield of RL that integrates deep learning with reinforcement learning. The key innovation of DRL lies in its ability to use deep neural networks (DNNs) to approximate complex functions, such as the value function or policy, which were traditionally computed using simpler models. DRL is particularly well-suited to high-dimensional environments, where the state and action spaces are large and unstructured, such as in image processing, game playing, and natural language understanding.

In DRL, a deep neural network is used to represent the Q-function (in value-based methods) or the policy (in policy-based methods). These neural networks are trained through interactions with the environment, using algorithms like Q-learning, Policy Gradient methods, and Actor-Critic methods. DRL has been successfully applied to a variety of domains, including gaming (e.g., AlphaGo, OpenAI's Dota 2 agent), robotics, and autonomous systems, where it has demonstrated impressive performance in tasks requiring complex decision-making.

In NLP, DRL offers a powerful framework for tasks that require long-term reasoning and optimization of sequential actions. DRL-based approaches enable agents to generate language sequences by learning from interaction feedback, improving language generation quality by considering both immediate rewards and long-term goals. Unlike traditional NLP models, which typically rely on supervised learning with labeled datasets, DRL models learn from the environment, adapting their behavior over time based on real-world interactions or simulated environments.

2.3 Deep Reinforcement Learning in Natural Language Processing

While the integration of RL into NLP has been explored for some time, the application of DRL has gained particular traction in recent years. DRL models are well-suited to NLP tasks that require the generation of sequences of words or sentences, where the optimal output may not be immediately clear. For example, in machine translation, the translation of a sentence involves multiple decisions about which words to choose and how to structure the sentence. In dialogue systems, the agent must decide not only what to say next but also how to engage in a dynamic and contextually appropriate manner over multiple turns [5].

Key challenges in NLP tasks, such as language generation, dialogue management, and machine translation, can be framed as sequential decision-making problems. For instance, generating a sentence or paragraph can be viewed as a series of decisions, where each word or phrase chosen affects the next word or phrase in the sequence. DRL offers the ability to optimize these tasks by considering the long-term consequences of actions (i.e., words or sentences) and ensuring that the overall output maximizes the desired reward, whether it be fluency, relevance, or contextually appropriate responses.

In DRL-based NLP models, the agent typically interacts with a text corpus or a conversational environment. The agent receives rewards based on the quality of its output—such as grammatical accuracy, coherence, or relevance—and adapts its policy to generate better results over time. This allows DRL models to tackle tasks where traditional supervised learning approaches struggle, particularly in scenarios where large labeled datasets are not available or where the task requires dynamic adaptation based on context or user interaction.

2.4 Evolution of Natural Language Processing Models with DRL

Before the rise of DRL, most NLP models relied on traditional machine learning methods, such as Hidden Markov Models (HMMs), Conditional Random Fields (CRFs), and shallow neural networks. These models required extensive feature engineering and were limited in their ability to handle complex, dynamic sequences. The introduction of deep learning, particularly recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, revolutionized NLP by allowing models to automatically learn hierarchical features from raw data.

However, even with the power of deep learning, these models still struggled with tasks involving long-term dependencies or dynamic decision-making. For example, sequence-to-sequence models (such as those used in machine translation) often fail to generate accurate or contextually appropriate translations in cases where long-range dependencies need to be captured. DRL models, on the other hand, excel at tasks that require sequential decision-making, making them a natural fit for many complex NLP tasks.

In recent years, hybrid models that combine DRL with deep learning architectures, such as Transformers (e.g., BERT, GPT), have further enhanced the potential of DRL in NLP. These models are able to take advantage of the strengths of deep learning (handling high-dimensional data) and the sequential decision-making abilities of DRL, opening up new possibilities for applications in conversational AI, machine translation, and text generation.

3. Deep Reinforcement Learning Algorithms in NLP

Deep Reinforcement Learning (DRL) algorithms combine the principles of reinforcement learning with deep learning techniques, enabling agents to make sequential decisions in complex environments like natural language processing (NLP).[6][7] These algorithms provide the flexibility to handle tasks in NLP that require long-term decision-making, such as language generation, machine translation, and dialogue management. DRL algorithms can be broadly categorized into value-based, policy-based, and actor-critic methods, each offering unique benefits for different NLP tasks. Below is a detailed overview of the main DRL algorithms and their application in NLP:

3.1 Value-Based Methods: Q-Learning and Deep Q-Networks (DQN)

Value-based reinforcement learning algorithms focus on learning a value function, which estimates the expected future reward of taking a particular action in a given state. The agent then chooses actions that maximize this value function.

- **Q-learning**: In classical Q-learning, the agent learns a Q-value function Q(s,a)Q(s, a)Q(s,a), which represents the expected reward of taking action aaa in state sss. The goal is to learn the optimal policy by selecting actions that maximize the Q-value. In NLP, Q-learning has been used for tasks like sentence generation and dialogue systems, where the agent learns the best sequence of actions (e.g., words or phrases) to maximize a long-term reward, such as generating fluent and relevant sentences.
- **Deep Q-Network (DQN)**: The classic Q-learning algorithm struggles with high-dimensional state spaces, such as those found in NLP tasks, where the state space (e.g., the set of all possible sentences or sequences of words) can be vast. Deep Q-Networks (DQN) address this limitation by using deep neural networks to approximate the Q-value function. DQNs can learn complex patterns in large state-action spaces, making them particularly useful for tasks like text summarization, where the model must evaluate numerous potential sentence structures and word choices.

3.2 Policy-Based Methods: REINFORCE and Policy Gradient Methods

Policy-based methods in reinforcement learning focus directly on learning a policy π(a∣s)\pi(a | s)π(a∣s), which represents the probability of taking action aaa in state sss. The policy is typically parameterized by a neural network, which is trained to maximize the expected reward through gradient-based optimization.

- **REINFORCE Algorithm**: The REINFORCE algorithm is a Monte Carlo-based policy gradient method. It estimates the gradient of the expected cumulative reward with respect to the policy’s parameters. The REINFORCE algorithm works by first generating a sequence of actions based on the current policy and then updating the policy parameters using the reward signal after completing the entire sequence. This method is particularly useful for NLP tasks where the entire sequence (e.g., a sentence or paragraph) contributes to the final reward.

- **Policy Gradient Methods**: Policy gradient methods extend the idea of REINFORCE by offering more sophisticated techniques for improving the policy. Instead of using Monte Carlo estimates, these methods use the advantage function to reduce variance in the gradient estimates. Popular policy gradient algorithms, such as the Proximal Policy Optimization (PPO) and Trust Region Policy Optimization (TRPO), have been employed in NLP tasks.

3.3 Actor-Critic Methods: Combining Value and Policy

Actor-critic methods combine the benefits of both value-based and policy-based approaches by having two components: the actor, which selects actions based on the current policy, and the critic, which evaluates the actions by estimating the value function. The actor updates the policy based on feedback from the critic, and the critic evaluates the chosen actions to help refine the policy.

- **Asynchronous Advantage Actor-Critic (A3C)**: A3C is a well-known actor-critic algorithm that uses multiple agents working in parallel on different instances of the environment. Each agent learns its own policy and value function, and their gradients are aggregated to update the shared model. This parallel approach helps reduce the time needed to explore the environment and speeds up the learning process.
- **Deep Deterministic Policy Gradient (DDPG)**: DDPG is an actor-critic algorithm designed for environments with continuous action spaces. In NLP, where action spaces (e.g., the set of all possible words) can be large, DDPG can be effective for tasks like interactive text generation or machine translation, where actions are continuously chosen over time.
- **Twin Delayed Deep Deterministic Policy Gradient (TD3)**: TD3 is an improved version of DDPG that addresses the instability in training caused by function approximation. It uses techniques like target policy smoothing, delayed updates, and a clipped target value to enhance the learning process.

3.4 Other DRL Techniques and Variants in NLP

- **Hierarchical Reinforcement Learning (HRL)**: In HRL, the learning process is decomposed into a hierarchy of sub-goals or tasks. This allows the agent to learn at multiple levels of abstraction, making it more efficient in complex environments. In NLP, HRL can be used for tasks like dialogue management, where the system learns high-level strategies (e.g., initiating a conversation) and low-level actions (e.g., choosing specific responses based on user input).
- **Imitation Learning**: Imitation learning (or learning from demonstration) is an approach where the agent learns by observing human actions and mimicking those actions. In NLP, this can be particularly useful for tasks like conversational AI, where the model can learn dialogue patterns by observing interactions between humans. Combining imitation learning with DRL helps the agent to bootstrap learning from expert demonstrations while still adapting through interaction with the environment.
- **Generative Adversarial Networks (GANs) in DRL**: GANs are used in some DRL-based NLP tasks, where the model learns to generate text by competing against a discriminator that evaluates the quality of the generated text. This setup encourages the agent to improve its text generation capabilities by learning to fool the discriminator into believing that its outputs are real, resulting in more natural and fluent language generation.

3.5 Challenges of DRL Algorithms in NLP

Despite their potential, DRL algorithms face several challenges when applied to NLP:

- **Sample Efficiency**: DRL models often require a large number of interactions with the environment to learn effectively. In NLP, where interactions can be costly (e.g., generating text or conducting dialogues), this can be a significant hurdle. Techniques like imitation learning, HRL, or transfer learning can help alleviate this issue by reducing the number of required samples.
- **Reward Design**: Defining appropriate reward functions in NLP tasks can be difficult, as language generation tasks often involve subjective measures such as fluency, relevance, and coherence. Designing rewards that accurately capture these subtleties is critical for effective training.
- **Exploration vs. Exploitation**: DRL algorithms need to balance exploration (trying new actions) and exploitation (choosing known actions that yield high rewards). In NLP, exploration can lead to incoherent or irrelevant text, making the design of exploration strategies crucial.
- **Computational Complexity**: DRL models, particularly those using deep neural networks, are computationally intensive and require significant resources for training. Optimizing these models efficiently in NLP tasks is still an area of active research.

4. Applications of DRL in NLP

Deep Reinforcement Learning (DRL) has proven to be a valuable tool for advancing various natural language processing (NLP) tasks, particularly those that require sequential decision-making, contextual understanding, and long-term optimization [8]. Unlike traditional supervised learning models, DRL enables agents to dynamically learn from interactions with the environment and adjust their behavior based on rewards received for their actions. This capacity makes DRL highly effective for tasks that involve generating or selecting sequences of words or phrases, such as dialogue systems, text generation, machine translation, summarization, and language modeling.
Below is a detailed overview of how DRL is applied in various NLP domains:

4.1 Dialogue Systems and Conversational AI

One of the most impactful applications of DRL in NLP is in dialogue systems or conversational AI. Traditional rule-based or supervised learning approaches for dialogue generation often fall short in handling open-domain conversations or multi-turn dialogues. DRL, by contrast, can enable more natural and contextually appropriate conversations by continuously learning from user interactions.

- **Dialogue Management**: In dialogue systems, DRL is used for managing interactions between the user and the system. The agent learns the best responses not just based on the immediate user input but also considering the broader conversation context, past interactions, and long-term goals (e.g., customer satisfaction, information retrieval). The system receives rewards for producing relevant and coherent responses, while irrelevant or confusing answers are penalized. This helps the agent improve over time and adapt to different conversation styles.
- **Goal-Oriented Dialogue Systems**: For goal-oriented dialogues (e.g., booking a flight or ordering food), DRL can optimize the flow of conversation to achieve specific goals. The agent learns to ask clarifying questions, select the correct items, and confirm details, all while ensuring the conversation remains natural and user-friendly.

Example: The **Google Duplex** system, which uses DRL to autonomously schedule appointments or make reservations, is a prime example of this application. The system is able to understand the user's needs, manage the conversation, and make decisions based on the current dialogue context.

- **Multi-turn Conversations**: DRL is highly beneficial for multi-turn dialogue systems where the agent must remember previous context to produce a relevant response. These systems need to balance exploration (trying new topics or questions) with exploitation (sticking to the successful patterns learned from prior conversations), making DRL an ideal solution for continuous learning in real-world interactions.

4.2 Text Generation

Text generation tasks require the model to generate coherent, fluent, and contextually appropriate sequences of words. DRL is particularly valuable in optimizing the generation of high-quality text that aligns with specific goals, such as writing style, content relevance, and grammatical accuracy.

- **Story or Narrative Generation**: In creative applications, DRL has been used to generate coherent narratives or stories. By framing the generation process as a sequential decision-making problem, the agent learns to choose words and sentence structures that maximize the overall quality of the story. DRL helps improve text flow, character development, and overall narrative coherence, making it a promising tool for tasks like scriptwriting or automated content creation.
- **Poetry Generation**: DRL can be used for creative text generation, such as generating poetry or other artistic forms of writing. The challenge here is not just fluency but also creativity and adherence to poetic structures. The reward function may include factors like rhyme, meter, and emotional tone, encouraging the model to generate text that aligns with these creative goals.
- **Open-Domain Text Generation**: Open-domain text generation, like the task faced by chatbots and virtual assistants, is another area where DRL has been applied. These systems need to generate responses that are not only syntactically correct but also contextually appropriate and engaging, making it a perfect fit for DRL algorithms. The agent learns over time to select words and phrases that best align with user preferences and conversation history.

4.3 Machine Translation

Machine translation, the task of automatically translating text from one language to another, can be framed as a sequence-to-sequence problem, where the goal is to generate a translation sequence based on the input sequence. DRL provides a powerful way to improve the translation quality by optimizing long-term objectives, such as fluency and contextual accuracy.

- **Optimizing Translation Quality**: Traditional machine translation models often rely on supervised learning, where the model is trained on pairs of source and target sentences. While effective, these models sometimes fail to capture long-range dependencies or complex linguistic structures. DRL can be used to enhance translation quality by considering not only the immediate translation output but also the overall coherence and fluency of the entire translation. The reward function can be designed to promote translations that are both syntactically and semantically accurate.

- **Sequence-to-Sequence Learning with DRL**: Sequence-to-sequence models, which are widely used in translation tasks, can benefit from DRL by learning to optimize the entire translation process, rather than just individual word-level predictions. By framing translation as a sequential decision-making process, DRL helps the system produce more fluent and contextually relevant translations.

Example: A DRL-based approach may involve fine-tuning an existing translation model (e.g., Transformer-based models) using a reward function that accounts for the quality of the translation, including fluency, grammar, and adherence to domain-specific language use (e.g., medical or legal terms).

4.4 Text Summarization

Text summarization aims to condense long documents into shorter, more concise summaries while retaining the most important information. This task can be challenging, particularly for extractive summarization, where the model needs to identify key sentences, and abstractive summarization, where the model must generate novel sentences that summarize the content.

- **Extractive Summarization**: In extractive summarization, the model selects the most important sentences or phrases from the input text. DRL can be used to optimize the selection process by rewarding the model for choosing sentences that maximize summary quality, such as informativeness, coherence, and readability.
- **Abstractive Summarization**: Abstractive summarization involves generating new sentences that capture the essence of the original content. DRL enables the model to learn how to generate summaries that are not only grammatically correct but also coherent and contextually relevant. The reward function in DRL-based summarization models can include factors such as fluency, information coverage, and precision.

Example: In DRL-based abstractive summarization, the agent receives rewards based on the quality of the summary it generates, including how well it paraphrases the original text, the brevity of the summary, and its ability to retain the original meaning.

4.5 Question Answering Systems

Question answering (QA) systems involve providing correct and relevant answers to questions posed in natural language. DRL can enhance the performance of QA systems, especially in open-domain question answering, where the model must generate answers based on a broad context.

- **Interactive QA Systems**: In interactive QA systems, DRL allows the model to refine its answers over multiple rounds of questioning. The system can ask follow-up questions to gather more context and adjust its answer accordingly. By using DRL, the QA agent learns to improve its answers by considering past interactions and the cumulative reward associated with providing the best response.
- **Textual Entailment and Inference**: DRL can be used to optimize QA systems that require reasoning over a body of text, such as in textual entailment or natural language inference tasks. The model learns to identify the most relevant sections of text that can help answer a given question, thereby improving its reasoning and accuracy.

4.6 Language Modeling

Language modeling is the task of predicting the likelihood of a sequence of words based on context. While traditional language models (e.g., n-gram models or recurrent neural networks) can generate sentences based on learned statistical relationships, DRL-based language models can go beyond mere probability estimation and learn to optimize the generation process based on specific goals.

- **Context-Aware Language Models**: DRL can be used to train language models that are more context-aware, optimizing sentence generation based on the broader context and long-term objectives. For example, in text completion tasks, DRL can be used to predict not just the next word but a sequence of words that maximize coherence with the surrounding context.
- **Adaptation and Personalization**: DRL can help personalize language models based on individual user preferences or interaction history. For example, a language model could be trained to adjust its tone or style based on the user's previous interactions, providing more personalized responses over time.

4.7 Other NLP Applications

- **Sentiment Analysis**: DRL can be applied to sentiment analysis tasks where the model learns to classify sentiments (e.g., positive, negative, or neutral) from text, taking into account long-term dependencies and subtle contextual clues.
- **Speech Recognition**: DRL is also being explored in the realm of speech recognition, where sequential decision-making plays a critical role in converting spoken language into text.

- **Information Retrieval**: DRL-based approaches can optimize ranking algorithms for information retrieval, selecting the most relevant documents based on a user's query and context.

5. Conclusion

Deep Reinforcement Learning (DRL) has emerged as a powerful paradigm for addressing complex tasks in Natural Language Processing (NLP), where traditional supervised learning approaches often fall short. The unique capability of DRL to learn from interaction, feedback, and experience, while optimizing long-term rewards, makes it particularly suitable for dynamic and sequential tasks in NLP. This research has provided a comprehensive overview of the integration of DRL into various NLP tasks, highlighting both its potential and challenges.

In conclusion, DRL offers several advantages for NLP, including the ability to model sequential decision-making processes, learn from human feedback, and adapt to complex, changing environments. Applications such as dialogue systems, text generation, machine translation, summarization, and language understanding stand to benefit significantly from the incorporation of DRL techniques.

Future research directions should focus on improving sample efficiency, reward function design, and scalability. Advances in meta-learning, imitation learning, and memory-augmented architectures hold promise for overcoming some of the current limitations of DRL. Additionally, more robust techniques for human-in-the-loop learning and cross-lingual transfer can significantly improve the adaptability and generalization of DRL models. Exploring new methods for model interpretability and controllability will be essential for increasing the transparency and trustworthiness of DRL-based systems, especially in high-stakes applications like healthcare or law.

Despite the obstacles, the integration of DRL into NLP is progressing steadily, and the continuous evolution of both reinforcement learning techniques and large-scale language models paves the way for more intelligent, contextually aware systems. With further advancements in computational efficiency, domain adaptation, and evaluation metrics, DRL can become a cornerstone of next-generation NLP applications.

References

[1] S. Vajjala, B. Majumder, A. Gupta and H. Surana, Practical natural language processing: a comprehensive guide to building real-world NLP systems, O'Reilly Media, 2020.

[2] D. Khurana, A. Koli, K. Khatter and S. Singh, "Natural language processing: State of the art current trends and challenges", *Multimedia Tools and Applications*, pp. 1-32, 2022.

[3] V. Uc-Cetina, N. Navarro-Guerrero, A. Martin-Gonzalez, C. Weber and S. Wermter, "Survey on reinforcement learning for language processing", *Artificial Intelligence Review*, pp. 1-33, 2022.

[4] W. Xu, L. Chen and H. Yang, "A comprehensive discussion on deep reinforcement learning", *2021 International Conference on Communications Information System and Computer Engineering (CISCE)*, pp. 697-702, 2021.

[5] Pierre Dognin, Inkit Padhi, Igor Melnyk, and Payel Das. 2021. ReGen: Reinforcement Learning for Text and Knowledge Base Generation using Pretrained Language Models. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 1084–1099.

[6] Sung-Jun Park, Dong-Kyu Chae, Hong-Kyun Bae, Sumin Park, and Sang-Wook Kim. 2022. Reinforcement learning over sentiment-augmented knowledge graphs towards accurate and explainable recommendation. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining. 784–793.

[7] Xianchao Zhu and William Zhu. 2021. A New Hierarchical Reinforcement Learning Framework for Relation Extraction. In 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI). IEEE, 223–225.

[8] Zahra Dehghanipour and Javad Salimi Sartakhti, "An Improved Deep Reinforcement Learning for Task-oriented Dialogue System", 2022.