International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

HQA Bot: Hybrid AI Recommender Based Question Answering Chatbot

Dr. J. Felicia Lilian*[1], Dr. Divya Vetriveeran[2], Dr. A. Malini[3], Dr. S. Kayalvizhi[4]

Submitted: 09/05/2023 **Revised**: 17/07/2023 **Accepted**: 08/08/2023

Abstract: The COVID pandemic has presented a number of challenges for education, particularly when it comes to reaching and engaging students. As a result, online education has become increasingly important, and artificial intelligence (AI) has played a crucial role in supporting this shift. The proposed tutor assistance question-answering system uses AI to automatically generate responses to student questions. This system includes a feedback mechanism, known as a satisfaction index that measures the efficiency of the generated responses and suggest relevant follow-up questions. The proposed Hybrid Recommender-based Dijkstra's algorithm (HRD) improves the system's accuracy. This algorithm uses a combination of techniques to group relevant questions based on context, which improves the accuracy of identifying the next relevant question. In our customized dataset, this approach achieved an accuracy of 96% and an average accuracy of 82% across benchmarked datasets. With this system, we aim to bridge the gap between students and education by providing a more engaging and personalized learning experience.

Keywords: *Artificial Intelligence, Chatbot, Deep Learning, Question Answering System, Recommender System, Reinforcement Learning*

1. Introduction

The use of E-learning systems has evolved significantly in recent years, particularly with the integration of chatbots as conversational agents. These human-like computer programs can take the form of video, audio, or text, and utilize artificial intelligence to enhance customer engagement. Chatbots such as Facebook Messenger, Telegram, Siri, Alexa, Google Assistant, chatGPT is widely used in various applications, industries, including communication, entertainment, and business. Recently, chatbots have also made their way into the field of education as virtual tutors and teaching assistants. These conversation agents can be classified as either generative or retrieval-based chatbot[1]. In generative-based chatbots, the bot generates responses to user questions based on the conversation, while in retrieval-based chatbots, the answers are predetermined and fed into the system. These chatbots are evaluated based on their domain, such as open or closed. Open-domain chatbots are trained on a large corpus and can handle any category of conversation, while closed-domain chatbots are limited to predefined questions and responses stored in a database. The open-domain chatbot is considered the most challenging as it must be able to converse coherently with humans.

[1]Thiagarajar college of Engineering, Madurai – 625015, INDIA
ORCID ID : 0000-0002-1221-2460
[2] CHRIST University, Bangalore – 560074, INDIA
ORCID ID : 0000-0003-2249-7591
[3] Thiagarajar college of Engineering, Madurai – 625015, INDIA
ORCID ID : 0000-0002-3324-5317
[4] SRM Institute of Science and Technology, Chengalpattu–603203, INDIA
ORCID ID : 0000-0002-3385-848X
* Corresponding Author Email: jflcse@tce.edu

David L. Waltz, in 1988, [2] accurately predicted the current advancements in merging open-domain intelligence into machines, stating,

"We are approaching an important milestone in the history of Earth's life, as we are developing machines with the potential to exhibit intelligence comparable to ours."

Artificial Intelligence is driving the development of intelligence advancements in various areas of human life. The increases in scalability, computation power, and efficient performance has driven the focus of AI and NLP greatly. One key area of advancement is Reinforcement Learning (RL), which allows machines to study their environment and take actions based on rewards received is a major breakthrough in recommendation system. The paper proposes to create a text-based question-answer chatbot for an academic subject. The traditional chatbot will only provide answers to user queries, without suggesting any follow-up questions. When developing an automated system, the main challenges includes providing relevant responses to queries, understanding the semantics and context of the query, and identifying connections between questions. To overcome these issues, the proposed recommender-based ranking algorithm, in conjunction with Dijkstra's algorithm, is intended to improve the Chabot's performance.

The key objectives of the proposed work are:

- The use of reinforcement learning algorithm to map user queries
- A novel Hybrid Recommender based on Dijkstra

Algorithm (HRD) is proposed to analyze dependencies between questions and recommend the next possible query using a similarity function

- Comparison of the results obtained using Dijkstra's and Floyd-Warshall algorithms.

2. Literature Survey

The survey begins by analyzing the process of embedding words to enable machines to understand human language. It then focuses on understanding the context between sentences and implementing a feedback mechanism by assigning rewards based on user satisfaction.

2.1. Embedding of Words

The initial step in processing natural language data is the pre-processing stage. In 2003, Bengio et al. [3] train the neural network by using embedding methods and adapted the Monte-Carlo method to estimate gradients, which reduced training time. In 2005, Morin and Bengio [4] introduced an efficient approach using a binary-tree concept that replaced the softmax prediction, resulting in faster training and testing of the system. In 2007, Mnih and Hinton [5] introduced a log-bilinear model with a single hidden layer, which, though simple in structure, outperformed previous models. It was further extended with a hierarchical softmax approach that was 200 times faster than the previous model. In 2013, Mikolov [6] introduced two new models, the Continuous BoW (W2V-CBoW) and Skip-Gram model (W2V-SG), which were based on the log-bilinear approach but had a two-step training process using the hierarchical softmax. These models were further extended by Mikolov in 2013 with the concept of Negative Sampling along with the Skip Gram (NS-SG) approach, which used subsampling of frequently used words. These were prediction-based approaches. In [7] describes the introduction of a new smart learning bot that pre-processes the input document. The work uses the concept of Optical Character Recognition (OCR) for data pre-processing, but it was unsuccessful as it could not recognize data that were not in proper format and alignment. Conneau et al. in 2017 [8] proposed a new embedding for sentences known as Infersent Embedding. It attempted to learn semantics through unsupervised sentences but did not produce satisfactory results.

The overall comparison of various embedding methods revealed that the word2vec using the Skip-Gram model and the GloVe model outperformed other embedding models and produced promising results.

2.2 Contextual Understanding of Data

In 2016, Chen et al. [9] introduced the use of RNN (Recurrent Neural Networks) with GRU (Gated Recurrent Units) to improve understanding of semantics. Tan et al. [10] proposed a hierarchical GRU for building context-aware models. The gates were used to match patterns at the word and sentence levels for context dependency. In 2016, Seo et. al., [11] introduced a hierarchical attention mechanism to understand context at different heights of granularity in a QA dataset resulting in 67% and 77% EM and F1 scores respectively. This methodology was promising and improved results in learning based on context. In 2017, Zhang and Ma [12] proposed a co-attention mechanism for answer selection that holds on the interactions between the question and answers to generate a different set of question. Tan et al. [13] and Xiang et al. [14] proposed an attention-based LSTM model to perform encoding and decoding process.

The inclusion of attention in the layers of deep learning model enhances the understanding of the context between the content and query.

2.3 Need for feedback Mechanism

To fully understand a user's query, it is recommended to suggest related queries by assigning rewards for each query. The system will hold some related features between the queries, and thus, we can estimate the reward of one query through the reward of the other query. However, this is difficult to control as the action space changes at each step when using a linear bandit model [15]. A Conceptual bandit model [16] addresses this by considering the action space across steps. Another learning algorithm, Q-learning [17], learns a value based on the action within a state. Ansari et al. [18] proposed the use of a new technique known as the recommender system which enhances the user learning experience [19]. The two filtering techniques such as Collaborative and Content-Based are used to generate recommendations. A hybrid recommender system is used for an interactive and efficient e-learning system, using both content-specific and collaborative methods to analyze users' behaviour and activities, and suggest the most appropriate and possible answers to the users.

3. Proposed Design and Methodology

Based on the survey, the proposed work aims to group context-based queries for faster retrieval and adopts a hybrid recommender-based algorithm to find the shortest path between relevant queries and group them. This helps the recommender system to analyze the query, and is based on rewards or feedback from the learners of the platform. Figure 2 illustrates the flow of the proposed methodology.

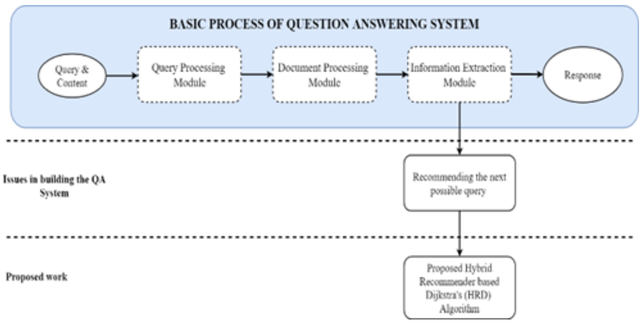

Fig 1. Flow of the proposed methodology

The proposed work is a Hybrid Recommender-based QA system that suggests the next possible query by using a reinforcement-based ranking system. The recommender system finds the shortest path between relevant queries and groups them, which is then given to a Reinforcement Learning (RL) agent. The proposed work compares the use of Dijkstra's and Floyd Warshall's algorithms, whose weights are then taken up by the RL agent to produce the final answer to the query.

3.1 Dataset

The dataset used for our system is a customized set, from various sources consisting of content, questions, and relevant answers. The dataset includes over 10,000 questions and answers, which are segregated into 80% and 20% for training and testing. The performance of the proposed methodology is compared with other benchmarked datasets such as SQuAD1.1, QuAC, CoQA, and NewsQA.

3.2 Word Embedding

The input data, which includes the query (Q) and the document (D), is pre-processed for embedding. Document D is first split into individual sentences D=S1, S2, ..., Sn. The user's input query (Q) is received and tokenized using the Natural Language Toolkit (nltk), and the vector for these tokenized words is obtained by using word embedding. The embedding is accomplished with the GloVe vector, which is used to bring out recommender-based words. The use of a sentence encoding like Bi-directional LSTM (BiLSTM) along the max-pooling layer generates the appropriate vector 'U' for the words based on semantics. The vector for the user's query is represented as Q=u_{q1}, and the documents considered as D = u_{s2}, ..., u_{sn} are the tokenization of words

3.3 Information Retrieval Based on Context and Reward

Context-based information retrieval needs to consider the aspect of semantics and social relevance. The semantics enhances the suggestion framework by exploiting the embedding goals and the learning model. The QeCSO algorithm [20] helps to obtain semantic learning over the content. It optimizes the query (Q) based on semantics and thus generates a reframed Query (Q*). The Q and Q* are for the same content (C). The optimized query will retrieve the response efficiently.

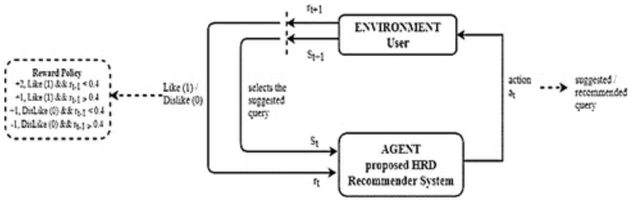

Fig. 2 Learning through reward system

The reward mechanism helps the user to gain information about the content. The proposed algorithm recommends a new query, q, only if the previous query receives a positive response, q+. Figure 2 illustrates the learning system based on reward, where the proposed algorithm suggests a relevant query by using the recommender system.

3.4 Proposed Floyd – Reinforcement Learning Recommender Algorithm (FR^2)

The proposed algorithm utilizes Floyd-Warshall-based Reinforcement Learning to predict the next query a user may ask. The system operates in an environment with states represented as s_n and actions represented as a_t. Users are given positive rewards for successful interactions, and negative rewards for unsuccessful interactions represented as r_t. The system sets a new goal state (s') for each episode, which is a set number of steps based on the previous user interaction with the system (s). The initial state is s_{init} and the initial action is a_{init}. The objective is to identify the most likely sequence of states or queries which an user may ask. The rewards assigned to the edges in the graph, as seen by the RL agent, are represented by equation 1.

$$F_\pi = \sum_{t=0}^{n} ||s_{init} = s, a_{init} = s, s_n = s'|| \quad (1)$$

The policy followed, denoted by π, is stochastic in nature. The Q-function, represented by Q(s, a), is calculated by incorporating both the state probability and the edge function, as outlined in equation 2.

$$Q(s,a) = \sum Prob(s'||s,a) F_\pi (s'||s,a) \quad (2)$$

The probability of transitioning to state s' from states based on action a, represented by Prob(s'||s, a), is crucial in ensuring the system functions effectively. To achieve this, the expected value of this probability must be maximized through rewards and in accordance with the stochastic policy $F^*_{\pi^* s'}(s'_i||s_k, a_k)$. Where π^* is the optimal policy selected from the set of F_π policies, and s_k, a_k are the optimal states and actions chosen from the set of states s_i and s_j. The equation states that the rewards in the chosen path must be greater than those of any other alternative path, which optimizes the recommendation of queries to the user. The steps involved in this process, utilizing Floyd's methodology, are outlined in Algorithm 1.

Algorithm 1: Floyd RL Recommendation (FR^2)

Input : states, actions and rewards

Initialize : Edge weights to ∞, t=0

Parameter : N – maximum time slot

Threshold – retrieved using cosine similarity between query and sentences,

r_t – reward obtained using utility matrix

1. Procedure BEGIN:
2. for t=1 to N do
3. choose greedy policy π^*
4. keep track of s' and reward r_t
5. if (r_t > threshold) then
6. continue
7. else
8. $s_t = s'$ and estimate r_t
9. end if
10. for each state and action do
11. Maximize reward
12. $F^*_{\pi^*_{s'}}(s'_i||s_k, a_k) \geq F^*_{\pi^*_{s'}}(s'_j||s_k, a_k) + \max F^*_{\pi^*_{s'}}(s'_i||s_{k-1}, a_{k-1})$
13. end for
14. end for
15. Return the best policy of state (s_t) which is the query
16. end procedure

3.5 Proposed Hybrid Recommender-based Dijkstra's Algorithm (HRD)

The FR^2 algorithm, which uses cosine similarity and a utility matrix to determine rankings, may not be sufficient. To improve the ranking system, Dijkstra's algorithm is integrated into the process. The running time of the system is determined by the number of edges (E) and vertices (V) and is calculated as O(‖E‖+‖V logV‖). Dijkstra's algorithm is preferred as it is faster in determining paths between vertices. In the modified shortest path methodology, the algorithm considers the path with the highest weight query to be the most likely recommended suggestion. The proposed hybrid QA recommendation system uses a reward-based approach. The system employs the use of shortest path algorithms, Dijkstra's to assign weights (ratings) to relevant queries and group them according to the user's feedback. The input query (Q*) and the content (C) go through pre-processing, where Glove embedding is used to convert them into numerical vectors. The generated responses are assigned a satisfaction index value, also known as a reward.

In our proposed work, comparison with the vector of the query, vec(query) as v(Q) with the vector value of sentences, vec(individual_content_sentence) as v(S_i) in the document using the Cosine similarity equation 3. This generates a similarity value between the query and each sentence. A threshold value is set to limit the number of similar sentences for the given query, which are then grouped into a candidate set (C_S), as shown in equation 4. Here, u_{q1} represents the threshold value for the query and u_{s1}, u_{s2}, ..., u_{sn} are the cosine distances obtained for every sentence in the content

$$Cos_{dist(S_i)} = \frac{v(Q).v(S_i)}{||Q||||S_i||} \tag{3}$$

The proposed ranking algorithm utilizes a reinforcement-based approach, where users can provide feedback in the form of a "like" or "dislike" to optimize the satisfactory index. This allows the chatbot to learn and identify the most relevant answer through the shortest distance between similar words, creating a candidate set that is ranked based on the satisfactory index. This process ensures that the chatbot can continuously improve its performance through the user's satisfaction.

$$Candidate_{Set(C_S)} = u_{q1} = u_{s1}, u_{s2}, \ldots, u_{sn} \tag{4}$$

The proposed algorithm is applied to a candidate set of answers to the given query. The candidate set is ranked based on a satisfactory index value, which is determined by the user through a like (1) or dislike (0) option. The algorithm assigns a positive reward for the most relevant answer and a negative reward for the less relevant answers. By maximizing the satisfactory index, the system can learn and intelligently identify the best answer on its own.

The Hybrid Recommender-based Dijkstra's Algorithm 2 (HRD) updates edge weights based on user rewards. Weights on the path are referred to as rewards, with possible values of +2, +1, or -1. These values are based on user actions (like or dislike) and a threshold value, γ, which is a discount factor reflecting the user's preference for the recommended query. The size of the graph is determined by the number of relevant queries, with similar queries grouped close together to improve retrieval and suggestion accuracy.

In the proposed work, we aim to retrieve similar answers to questions from a database that has been rated with satisfactory index values to aid in the selection of the correct answer. Cosine Similarity finds the similarity score between the query and sentences in the document. A threshold value is set to limit the number of similar sentences for the given query, and these sentences are grouped as a candidate set (CS).

To rank the sentences within the (CS) we have proposed a reinforcement-based ranking algorithm to maximize the satisfaction index for that solution. This is achieved by incorporating Like or Dislike options, allowing the chatbot to learn through the user's satisfaction and identify the best answer. Our ranking algorithm finds the shortest distance between similar words for retrieval and ranks them based on the satisfactory index.

Algorithm 2 outlines the steps for our Dijkstra's-based recommender system. The edge weights are updated according to the rewards provided by the user.

Algorithm 2: Algorithm for Hybrid Recommender based Dijkstra's algorithm (HRD)

Input : $q_{user}, User\ Query$

Output : $q_{rec}, Recommended\ Query$

Parameter : $Q^*, Possible$ queries from the content, C

E, Weighted Edge

R, Reward value for the query

Q, Set of all query nodes

q_j, individual query node in Q

Threshold value, γ = 0.4, based on the user's preference

States(S_T) = q_{user}, q_j

Action(A_T) = user_selection(1-LIKE, 0-DISLIKE)

$$Reward\ (R) = \begin{cases} +2, A_T = 1\ AND\ \gamma < 0.4 \\ +1, A_T = 1\ AND\ \gamma > 0.4 \\ +1, A_T = 0\ AND\ \gamma < 0.4 \\ -1, A_T = 0\ AND\ \gamma > 0.4 \end{cases}$$

Transition Probability (TP) = $(S_{T+1}||S_T, A_T)$

Procedure BEGIN:
    Fun_dij(Graph(Q^*, E), q_{user})
    For each vertex E in Q^* do
        Dist[E] = ∞
        Previous[E] = undefined
    End For
    Dist[q_{user}] = R
    While (Q is not empty) do
        Remove q_{user} from Q
        For each neighbor q_j from q_{user} do
            Alt – Dist[q_{user}] + $dist\_between(q_{user}, q_j)$
            If(Alt > Dist[q_j]) then
                Dist[q_j] = alt
                Previous[q_j] = q_{user}
            End if
            Return Previous[q_j]
        End For
    End While
End Fun_dij()
For (S_T = q_j) do
    If (A_T == 1) then
        R= S_T
        Update S_{T+1}
        R++
    Else
        If(A_T == 0) then
            S_T retrieves next possible query from Q
            R--
        End if
    End if
    $q_{rec} = q_j$
    Return q_{rec}
End For
End Procedure

4 Results and Discussions

When a user requests information on a specific topic, related topics and a set of relevant queries are provided. These queries are determined to be the most likely searched based on the previous query. Upon receiving the suggestions from the chatbot, the feedback provided by the users is used to update the recommendations. When a user selects a query from the suggested options, the system records it as a positive reward. Conversely, if the user does not choose a query, it is recorded as a negative reward.

The weight of the edges is adjusted depending on the user's feedback. When the user selects the "dislike" option for a suggested query, the weight of that query is decreased by 1, indicating that the user is not interested in that search. Similarly, when the user selects the "like" option, the weight is increased. The proposed Dijkstra and Floyd Warshall algorithms were tested using both a custom dataset and benchmark datasets. The results were compared in terms of accuracy, as shown in Figure 3.

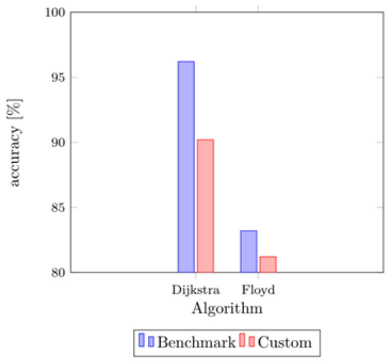

Fig 3. Performance of proposed Hybrid algorithms

The results of the user's feedback on the chatbot application were analysed, and it was found that 47 students were asked to use the application. A summary of their responses is provided in Figure 4.

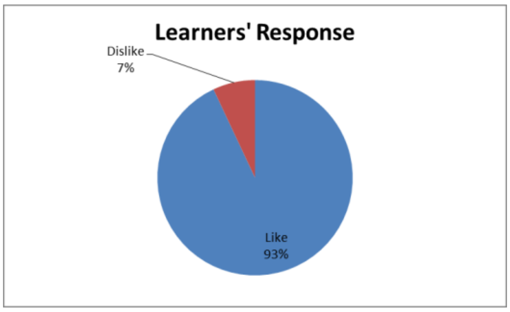

Fig 4. Learner's Response

The performance of our system is compared with the advanced models, such as LSTM, BiLSTM, ATT-BLSTM, ATT-BLSTM-QeCSO, and ATT-BLSTM-HRD, as given in Figure 5. Our proposed model, which utilizes optimized information retrieval, demonstrated superior performance on our customized dataset. As depicted in Figure 6, when tested on benchmarked datasets, our model achieved an average accuracy of 82%.

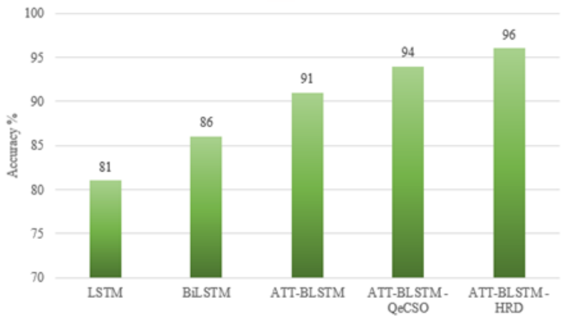

Fig 5. Comparison of QA models for customized dataset

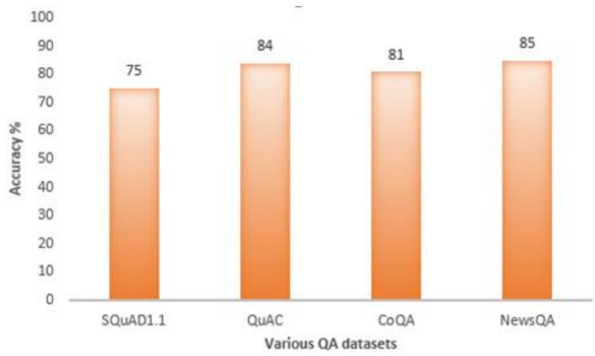

Fig 6. Comparison of the proposed work with various QA Datasets

5 Conclusion and Future Work

The development of educational chatbots is rapidly increasing, providing students with sophisticated ways to enhance their learning experience. The proposed work investigates the creation of such a bot that utilizes a recommendation system through the application of reinforcement learning techniques. The implementation of rewards allows the system to learn quickly and adapt dynamically. The proposed F R2 and HRD algorithms have demonstrated exceptional performance with a customized data accuracy of 96% and an average accuracy of 82% on benchmarked datasets. This work can be expanded for various educational purposes and with the use of additional benchmarked datasets

Conflict of Interest

Authors do not have any conflict of interest based on financial or other personal considerations.

References

[1] Varghese E, & Pillai M R, 2018, April, A standalone generative conversational interface using deep learning, In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 1915-1920), IEEE

[2] Waltz D.L, 1988, The prospects for building truly intelligent machines,Daedalus, 191-212.

[3] Bengio Yoshua, Ducharme Rejean, Vincent Pascal & Janvin Christian, 2003, A neural probabilistic language model, The Journal of Machine Learning Research, vol. 3,pp: 11371155.

[4] Morin Frederic & Bengio, Yoshua, 2005, Hierarchical probabilistic neural network language model, AISTATS, Vol. 5, pp: 246252.

[5] Mnih Andriy & Hinton Geoffrey, 2007, Three new graphical models for statistical language modelling, Proceedings of the 24th International Conference on Machine learning, pp: 641648.

[6] Mikolov Tomas, Chen Kai, Corrado Greg & Dean Jeffrey, 2013, Efficient estimation of word representations in vector space, ArXiv Preprint, ArXiv:1301.3781

[7] Pichponreay, L., Kim, J. H., Choi, C. H., Lee, K. H., & Cho, W. S. (2016, July). Smart answering chatbot based on OCR and overgenerating transformations and ranking. In Ubiquitous and Future Networks (ICUFN), 2016 Eighth International Conference on (pp.1002-1005). IEEE

[8] Conneau Alexis, Kiela Douwe, Schwenk Holger, Barrault Loic & Bordes Antoine, 2017, Supervised learning of universal sentence representations from natural language inference data, ArXiv Preprint, ArXiv:1705.02364.

[9] Chen S,Wen J & Zhang R, 2016, GRU-RNN based question answering over Knowledge Base, In Springer China Conference on Knowledge Graph and Semantic Computing, pp. 8091.

[10] Tan C,Wei F, Zhou Q, Yang N, Du B, Lv W & Zhou M, 2018, Contextaware answer sentence selection with hierarchical gated recurrent neural networks, IEEE/ACM Transaction Audio, Speech, Language Process, Vol. 26, no. 3, pp. 540549.

[11] Seo Minjoon, Kembhavi Aniruddha, Farhadi Ali & Hajishirzi Hannaneh, 2016, Bidirectional attention flow for machine comprehension, arXiv preprint, arXiv:1611.01603.

[12] Zhang Lei & Ma Longxuan, 2017, Coattention based BiLSTM for answer selection, IEEE International Conference on Information and Automation (ICIA), pp. 10051011.

[13] Tan Ming, Dos Santos Cicero, Xiang Bing & Zhou Bowen, 2016, Improved representation learning for question answer matching, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 464473.

[14] Xiang Yang, Chen Qingcai, Wang Xiaolong & Qin Yang, 2017, Answer selection in community question answering via attentive neural networks, IEEE Signal Processing Letters, Vol.24, no.4, pp.505509.

[15] Mueller Jonas & Thyagarajan Aditya, 2016, Siamese recurrent architectures for learning sentence similarity, Proceedings of the AAAI Conference on Artificial Intelligence,Vol.30,no.1.

[16] Zeng Chunqiu, Wang Qing, Mokhtari Shekoofeh & Li Tao, 2016, Online context-aware recommendation with time varying multi-armed bandit, Proceedings of the 22nd ACMSIGKDD international conference on Knowledge discovery and data mining, pp. 20252034.

[17] Watkins Christopher JCH & Dayan Peter, 1992, Q-learning, Machine learning, Springer, Vol. 8, no. 3-4, pp. 279292.

[18] Ansari Mohammad Hossein, Moradi Mohammad, NikRah Omid & Kambakhsh Keyvan M, 2019, CodERS: A hybrid recommender system for an E-learning system, 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS),

[19] Hermann Karl Moritz, Kocisky Tomas, Grefenstette Edward, Espeholt Lasse, Kay Will, Suleyman Mustafa & Blunsom Phil, 2015, Teaching machines to read and comprehend, Advances in Neural Information Processing Systems, pp. 16931701.

[20] Lilian J F, Sundarakantham K, & Shalinie S M, 2021, QeCSO: Design of hybrid Cuckoo Search based Query expansion model for efficient information retrieval, Sdhan, 46(3), 1-11

[21] Patil, A. ., & Govindaraj, S. K. . (2023). Enhanced Deep Learning Models for Efficient Stroke Detection Using MRI Brain Imagery. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3), 191–198. https://doi.org/10.17762/ijritcc.v11i3.6335

[22] Yulia Sokolova, Deep Learning for Emotion Recognition in Human-Computer Interaction , Machine Learning Applications Conference Proceedings, Vol 3 2023.