An Open-Domain Search Quiz Engine Based on Transformer

Xiaoling Niu*, Ge Guo
Department of Computer Science and Software Engineering,
Pingdingshan Institute of Industry Technology, Pingdingshan 467000, China

***Abstract*—As the volume of information on the Internet continues to grow exponentially, efficient retrieval of relevant data has become a significant challenge. Traditional keyword matching techniques, while useful, often fall short in addressing the complex and varied queries users present. This paper introduces a novel approach to automated question and answer systems by integrating deep learning and natural language processing (NLP) technologies. Specifically, it combines the Transformer model with the HowNet knowledge base to enhance semantic understanding and contextual relevance of responses. The proposed system architecture includes layers for word embedding, Transformer encoding, attention mechanisms, and Bi-directional Long Short-Term Memory (Bi-LSTM) processing, enabling sophisticated semantic matching and implication recognition. Using the BQ Corpus dataset in the banking and finance domain, the system demonstrated substantial improvements in accuracy and F1-score over existing models. The primary contributions of this research are threefold: (1) the introduction of a semantic fusion approach using HowNet for enhanced contextual understanding, (2) the optimization of Transformer-based deep learning techniques for Q&A systems, and (3) a comprehensive evaluation using the BQ Corpus dataset, demonstrating significant improvements in accuracy and F1-score over baseline models. These contributions have important implications for improving the handling of complex and synonym-rich queries in automated Q&A systems. The experimental results highlight that the integrated approach significantly enhances the performance of automated Q&A systems, offering a more efficient and accurate means of information retrieval. This advancement is particularly crucial in the era of big data and Web 3.0, where the ability to quickly and accurately access relevant information is essential for both users and organizations.**

Keywords—Natural language processing; deep learning; transformer; Bi-LSTM; semantic understanding

I. INTRODUCTION

As intelligence, networking, and data become integral to modern society, information is growing exponentially, with vast amounts of data continuously uploaded to the Internet. In the early stages of Internet development, keyword matching technology was used in search engines to retrieve relevant data. However, with the advent of the "big data" and web 3.0 eras [1-2], keyword matching has become inadequate. The sheer volume of information—such as the 50.94 million Internet domain names registered in China alone [3]—makes it challenging for users to find accurate and relevant data.

Automated question-and-answer (Q&A) systems are a form of artificial intelligence designed to efficiently answer users' queries. With the rise of mobile internet and a growing need for fast, accurate information retrieval, the development of advanced Q&A systems has become essential. These systems utilize natural language processing (NLP) to allow machines to interpret human language and generate responses [4-5]. From early rule-based models such as ELIZA [6] and BASEBALL [7], to modern advancements like IBM's Watson [8] and Google's NLP-based systems [9], Q&A models have significantly evolved in their complexity and functionality.

Despite these advances, traditional models still struggle with contextual consistency, especially in complex, sentiment-driven user queries. Deep learning models, particularly Recurrent Neural Networks (RNNs) [10] and Long Short-Term Memory (LSTM) networks [11], have been instrumental in advancing Q&A systems, but they face limitations such as gradient vanishing and high computational costs. The introduction of sentiment analysis into these models is one possible solution to enhance their performance, particularly in generating contextually relevant and emotionally consistent answers. This paper presents a novel approach to automated question and answer (Q&A) systems that combines the Transformer model with the HowNet knowledge base to improve semantic understanding and contextual relevance. The main contributions of this paper are as follows:

1) By integrating HowNet, a structured lexical knowledge base, the system enhances its ability to understand synonyms, near-synonyms, and more complex semantic relationships. This significantly improves the Q&A system's performance in dealing with complex and nuanced queries.

2) While Transformer models have proven highly effective in capturing long-range dependencies in sequential data, we incorporate a Bi-directional Long Short-Term Memory (Bi-LSTM) to complement the Transformer's attention mechanism. Bi-LSTM excels at learning temporal patterns and capturing both forward and backward dependencies in sequences, which is particularly useful in tasks requiring a deeper understanding of contextual information.

3) The system is rigorously evaluated using the BQ Corpus dataset, focusing on queries related to the banking and finance sector. The results demonstrate a substantial improvement in accuracy and F1-score, indicating the system's effectiveness in practical scenarios.

Enhanced semantic understanding via HowNet offers potential for improving Q&A systems across a variety of domains where nuanced language processing is critical. The

*Corresponding Author.

optimized Transformer architecture provides a scalable solution that can be applied to other NLP tasks requiring deep contextual understanding. Finally, the improved performance metrics highlight the system's real-world applicability, particularly in sectors like finance, healthcare, and customer service.

II. Related Work

Early Q&A systems like ELIZA [6] and BASEBALL [7] employed rule-based approaches that relied on manual keyword matching to automate question-answering, and 'LUNAR' was able to respond to a query about baseball games [12]. While these models were groundbreaking in their ability to interact with users through structured queries, they were limited by their inability to fully comprehend natural language nuances. In the 1990s, systems like START [13] introduced keyword-based web search engines that returned multiple results, marking the beginning of more advanced Q&A systems. By the beginning of the 21st century [14], search engine technology had become more sophisticated, allowing search engines to search for answers to a wide range of questions.

As the field evolved, systems such as IBM Watson [8], powered by DeepQA technology, began using massive corpora to enhance the accuracy of responses. Apple's SIRI [15] also paved the way for voice-activated Q&A models based on natural language processing. Despite these advancements, traditional Q&A systems still lacked the ability to handle nuanced and complex queries, particularly those requiring emotional understanding.

Deep learning-based automated quizzing usually involves training a large amount of text data to learn a mapping that can directly generate a corresponding sequence based on a given sequence [16]. Recurrent Neural Networks (RNNs) are extensively utilized as a robust benchmark approach for feature extraction in Seq2seq models [10]. These models enabled systems to better process sequential data, making them more capable of handling dynamic queries. However, RNNs face significant challenges, such as gradient vanishing and their limited ability to capture long-distance dependencies [16].

To overcome these challenges, Long Short-Term Memory (LSTM) networks were introduced. LSTMs improved upon RNNs by maintaining longer-term dependencies between input sequences, but they came with higher computational costs and more complex training processes [17-18]. Gated Recurrent Unit (GRU) networks offered a more simplified solution, but they still struggled to capture intricate relationships between different stages of a query [19].

One of the major challenges in current Q&A systems is their inability to capture and integrate sentiment information from user queries. Sentiment analysis allows models to better understand the emotional tone behind a query, which is crucial for generating more contextually appropriate responses. Several researchers have explored integrating sentiment into Q&A models to address this issue. Bowman et al. [20] combined variational autoencoders with an LSTM-based encoder-decoder framework, enabling the model to generate more meaningful responses by accounting for sentiment information. However, existing systems still struggle with achieving a consistent emotional tone in responses, limiting their effectiveness in real-world applications.

Despite the significant advances in NLP and deep learning, current Q&A systems face several challenges, such as generating monotonous, contextually inconsistent responses. Moreover, deep learning models suffer from network degradation and gradient vanishing as the network depth increases. Sentiment-aware models are still in their infancy, and the integration of emotional context into answers remains a challenge [21].

In conclusion, the development and integration of automated question and answer systems have become increasingly vital in the era of big data and web 3.0 [22]. Our work aims to bridge these gaps by integrating sentiment analysis directly into a deep learning-based Q&A system. By doing so, we can improve both the contextual consistency and the emotional relevance of the responses, thereby enhancing the overall user experience.

III. A Textual Semantic Matching Approach Based on Knowledge Fusion of Transformer and HowNet

In the field of Chinese text semantic matching, vocabulary is the smallest unit that can express the correct meaning of Chinese. The existence of a large number of synonyms and near-synonyms in Chinese makes it more difficult to match the semantics of Chinese text, which makes it difficult for deep learning models applied in English to get better results on Chinese datasets. If the computational scheme of the English model is adopted, the word information is completely discarded and the research is based on words, which leads to the loss of a large amount of semantic information in Chinese. To enhance the resolution of synonyms and near-synonyms in Chinese, this study introduces the HowNet knowledge base. It presents a text implication recognition method that leverages the conceptual relationships within HowNet and the superior performance of the Transformer model in handling extended texts.

The model firstly encodes and data-drives the internal structural semantic information of Chinese utterances at multiple levels by Transformer, and introduces HowNet, an external knowledge base, for knowledge-driven modeling of knowledge associations between words in terms of justification, and then utilizes Soft-Attention for interactive attention computation and knowledge fusion with justification matrices, and finally further encodes textual conceptual-level semantic information and conceptual relations by Bi-LSTM. Additionally, Bi-LSTM is employed to further encode contextual information at the conceptual level, facilitating the reasoning and identification of semantic consistency and implication relationships.

The model is divided into six layers, namely, word embedding layer, Transformer layer, Attention layer, Bi-LSTM layer, average pooling and maximum pooling layer, and fully connected layer. Bi-LSTM layers are well-suited for sequential data because they process the input from both directions (forward and backward), allowing the model to learn context from past and future elements in the sequence simultaneously. It contains the process of processing the semantic original information, and the specific structure is shown in Fig. 1.

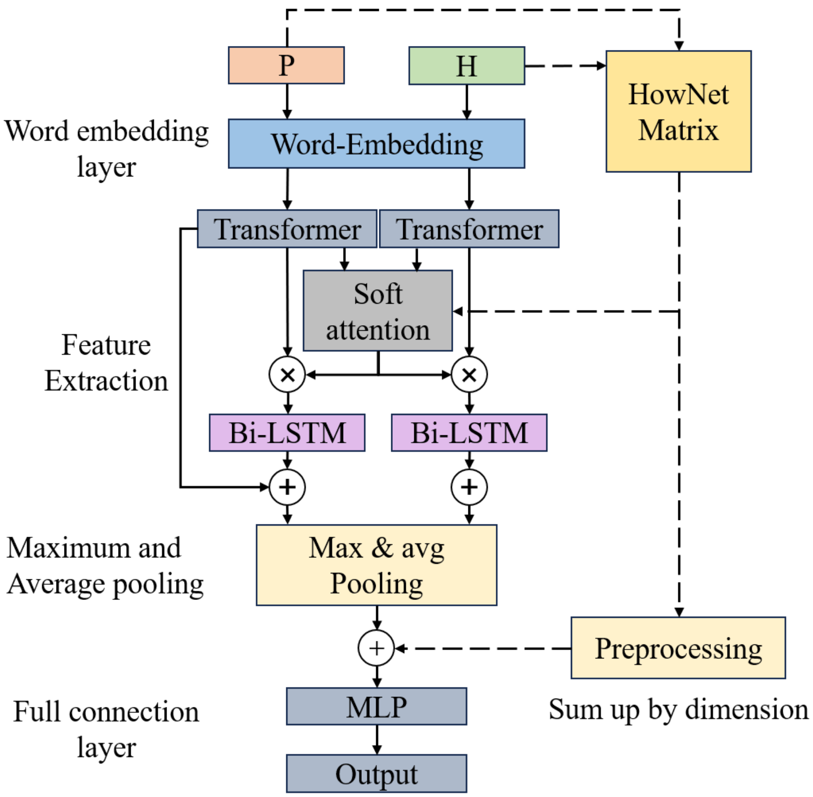

Fig. 1. Structure of the model.

The role of the transformer layer is to make the vectorized text pass through the neural network to obtain deep semantic information. Commonly used neural networks are convolutional neural network, long and short-term memory network, etc. The model uses the Transformer architecture as the encoding layer of the text to process the sentence vector. This model uses the Transformer architecture as the coding layer of the text to process the sentence vectors.

The attention layer is a widely used and essential element in text semantic matching models. It offers benefits such as high speed, effectiveness, and a low number of parameters. Various types of attention mechanisms exist, including soft attention, hard attention, self-attention, as well as focused and localized attention techniques. In this chapter, the commonly used Soft-attention mechanism (Soft-attention) is used, but the semantic matrix information generated based on How Net is added to it, which as shown in Fig. 2, and the trainable weights HN_{col}.

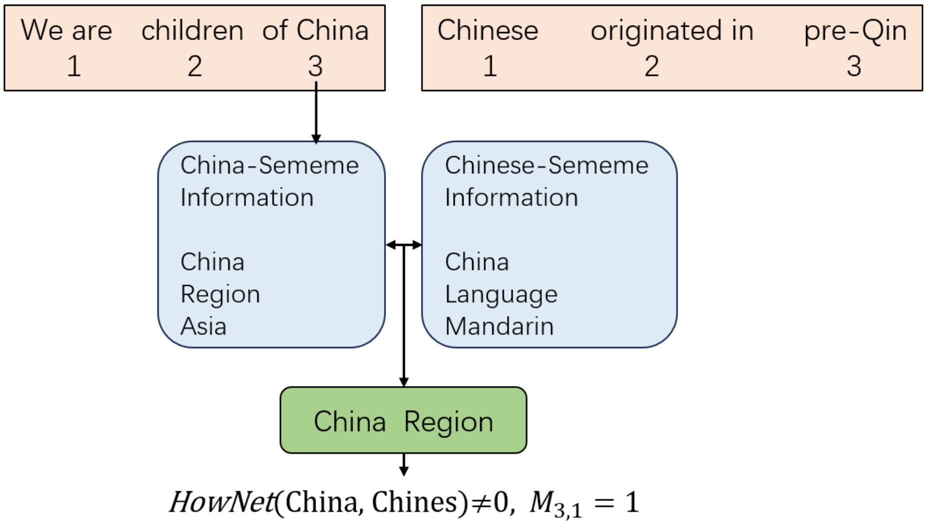

Fig. 2. Semantic information computation.

The top box displays the outcomes of sentence disambiguation, while the middle box shows the various etymological details associated with the current word. The bottom box illustrates the intersection of the etymological information of the two words. As can be seen from the above diagram, “China” has the meanings of China, related to a specific country, place, Asia, etc., and “Huaxia” has the meanings of borrowing, finance, retaining, China, country, etc. The intersection of the semantic principles of the two words is China, country, place, so at this point in time, “China, Huaxia” has the meanings of China, country, and hence $HowNet(\text{China}, \text{Huxia}) \neq 0$.

$$M_{i,j} = \begin{cases} 1 & HowNet(P_i, H_j) \neq 0 \\ 0 & HowNet(P_i, H_j) = 0 \end{cases} \tag{1}$$

$$M = \begin{bmatrix} 0 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 0 \end{bmatrix} \tag{2}$$

The formula indicates that a query for the i-th word in sentence P and the j-th word in sentence H is performed for the meaning of the original, and if the query result is valid, then the value of the corresponding (i,j) position in the M matrix is one.

Attention matrix e is generated as Eq:

$$e = PH^T + \gamma \cdot M \tag{3}$$

In this context, γ is a training factor. The attention matrix e not only combines textual information between sentences but also captures the semantic details of inter-sentence word pairs. As depicted in Fig. 3, the heat map of the matrix changes. Upon incorporating the original justification information, the weight of specific positions increases, indicating that these positions acquire information from the original justification matrix. In addition to this Attention matrix generation method, it is also possible to transform the various justification information of the current word into a continuous vector expression, which can be fused and embedded into the corresponding word vectors according to their weights, and there are various methods for this embedding method, such as embedding it into the Transformer structure, which makes the semantic information and word vector expression fuse.

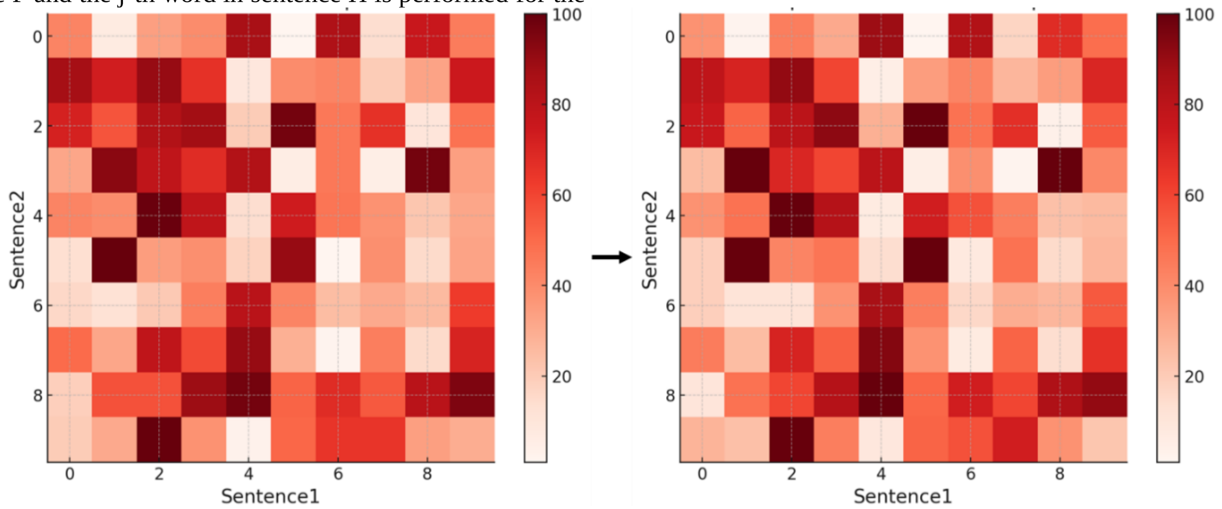

Fig. 3. The heat map of matrix change.

After obtaining the improved attention matrix e, the soft attention is computed as follows:

$$\hat{P} = \sum_{j=1}^{l_h} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_h} \exp(e_{ik})} P_{tf}, \forall i \in [1,2,\cdots,l_p] \tag{4}$$

$$\hat{H} = \sum_{j=1}^{l_p} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_p} \exp(e_{ik})} H_{tf}, \forall i \in [1,2,\cdots,l_h] \tag{5}$$

Where P_{ff}, H_{ff} are the matrix vectors of sentence pairs after Transformer. l_p, l_h denote the sentence lengths, and $\hat{P}$, $\hat{H}$ denote the outputs after soft attention mechanism.

The Bi LSTM layer is used to process the outputs P ̂and H ̂ of the transformer layer after the soft attention mechanism, and the two-way. The Bi-LSTM layer processes the outputs $\hat{P}$ and $\hat{H}$ from the Transformer layer following the application of the soft attention mechanism. The bidirectional Long Short-Term Memory Network (Bi-LSTM), which combines forward and backward encoding, enhances the acquisition of contextual information.

$$P_{bi-lstm} = BiLSTM(\hat{p}) \tag{6}$$

where $\hat{P}$ is the output vector of sentence P after the soft attention mechanism, and $P_{bi-lstm}$ is the vector after the bi-directional long-short-term neural network. Bi-LSTM is spliced from a combination of forward LSTM and backward LSTM, and compared to the long-short-term memory network LSTM, which is incapable of encoding the information passed from the back of the text to the front, Bi-LSTM can better capture the bi-directional semantic dependencies of the text.

$$P_o = Concat([P_{tf}; P_{bi-lstm}]) \tag{7}$$

$$P_{rep} = [Max(P_o); Mean(P_o)] \tag{8}$$

Here, P_{tf} represents the output vector of sentence P after the soft attention mechanism, while $P_{bi-lstm}$ is the vector resulting from the bidirectional long-short-term neural network, $Max(P_o)$ denotes the maximum pooling result of P_o, and $Mean(P_o)$ denotes the average pooling result of P_o. In most cases, maximum pooling can effectively improve the model performance, which can suppress the panning distortion and noise, etc., and also reduce overfitting. Average pooling can eliminate the effect of local maxima and make the model more stable. The combination of these two pooling methods can make the vector representation contain more information, which can improve model performance significantly.

After obtaining the complete sentence vector expressions P_{rep} and H_{rep} of sentence pairs, in the proposed model, the concatenation process also integrates information from the HowNet matrix. The sums of the two dimensions of the HowNet matrix, HN_{row} and HN_{col}, are computed and concatenated with P_{rep} and H_{rep}. This concatenated result forms the final input H for the feedforward neural network.

$$HN_{row} = sum(M, axis = 0) \tag{9}$$

$$HN_{col} = sum(M, axis = 1) \tag{10}$$

$$H = concat\left(P_{rep}; H_{rep}; P_{rep} - H_{rep}; HN_{col}; HN_{row}\right) \tag{11}$$

Where sum(•) means summing along the axis dimension. Taking HN_{row} as an example, HN_{row} represents the result of summing the HowNet matrix according to the first dimension, which corresponds to the HowNet information for sentence 1., and HN_{col} similarly represents the result of summing the HowNet matrix along the second dimension. Through vector splicing, H obtains the corresponding justification original information of the two sentences, where $P_{rep} - H_{rep}$ contains the discrepancy between the two sentence vectors.

After acquiring the final sentiment vector represents H for the sentence pairs, the model utilizes a two-layer fully connected neural network to determine the final matching result. The loss function employed is the cross-entropy loss function, calculated as follows:

$$Loss = \frac{1}{N}\sum_i -[y_i \times \log(p_i) + (1 - y_i) \times \log(1 - p_i)] \tag{12}$$

Here, y_i represents the label of sample i, with 1 indicating positive samples and 0 indicating negative samples. p_i denotes the predicted probability that sample i is a positive sample. Besides the standard cross-entropy loss function, this study also explores the KL Divergence loss function to evaluate the model's performance. The formula for the CoSent loss function in the context of binary classification is as follows:

$$\mathrm{KL}(P \parallel Q) = \sum_i P(i)\log\frac{P(i)}{Q(i)} \tag{13}$$

"Where " P(i)" is the true probability distribution and " Q(i)" is the predicted probability" " distribution." In the context of this study, using KL Divergence loss helps ensure that the positive sample pairs were more closely alike than the negative sample pairs, effectively maximizing the separation between positive and negative samples in the vector space. Experiments show that for pre-training methods such as BERT, SentenceBERT, etc., the method of directly using the pre-vector model to generate sentence vectors and then connecting them to the fully-connected layer for prediction is effective, and this method makes the pre-training model converge more quickly. For the model proposed in this chapter, it is not as effective as the cross-entropy loss function in the non-pre-training case.

During the training phase, MultiStepLR was utilized to progressively adjust the learning rate. The learning rate was updated with a decay rate of 0.5 at the 20th, 40th, 80th, 120th, and 160th iterations. By adjusting the learning rate dynamically as the number of iterations increases, the model's convergence speed is enhanced. The trend of the learning rate is illustrated in Fig. 4.

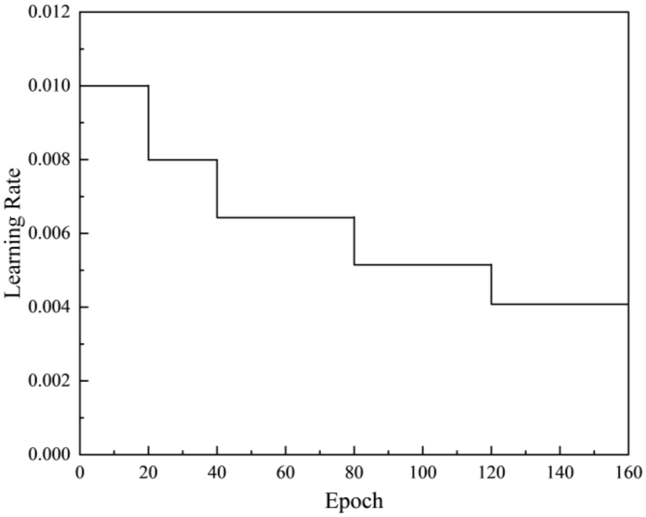

Fig. 4. Trends in learning rates.

IV. Question and Answer Engine Design

The Q&A system model was executed on a host computer having Intel Core i7-8750H CPU, NVIDIA 1060 GPU with 8 GB RAM, and 16 GB RAM. To implement this model, the Python programming language was used and the TensorFlow 1.9 deep learning framework was used. This system is capable of taking the user questions into a vector-based text retrieval, rank the top ten matching answer candidates, and then using the pre-trained.

Then the pre-trained Transformer model is used to extract the answers accurately. Therefore, this system includes database module, pre-processing technology, vector-based document retrieval, Transformer-based answer extraction and the final user operation platform, and its overall structure design is shown in Fig. 5.

The intelligent Q&A system mainly includes two roles: administrator and user, and it mainly realizes the functions of intelligent Q&A, knowledge map display and standardized text search. Users can quickly ask natural language questions in the domain, and the system automatically parses the semantic information involved in the questions and returns the correct answers through the search of the knowledge base, which mainly includes multiple use cases, such as asking questions, checking the semantic parsing, checking the returned answers, checking the corresponding knowledge base, checking the canonical retrieval text, and editing personal information, etc. Administrators mainly manage and maintain the system's knowledge base, document database, and user privileges. The manager is mainly responsible for managing and maintaining the system's knowledge base, document database, and user privileges. It mainly contains question category information management, knowledge base management, specification storage management, user information management and other major use cases (see Tables I, II, III and IV).

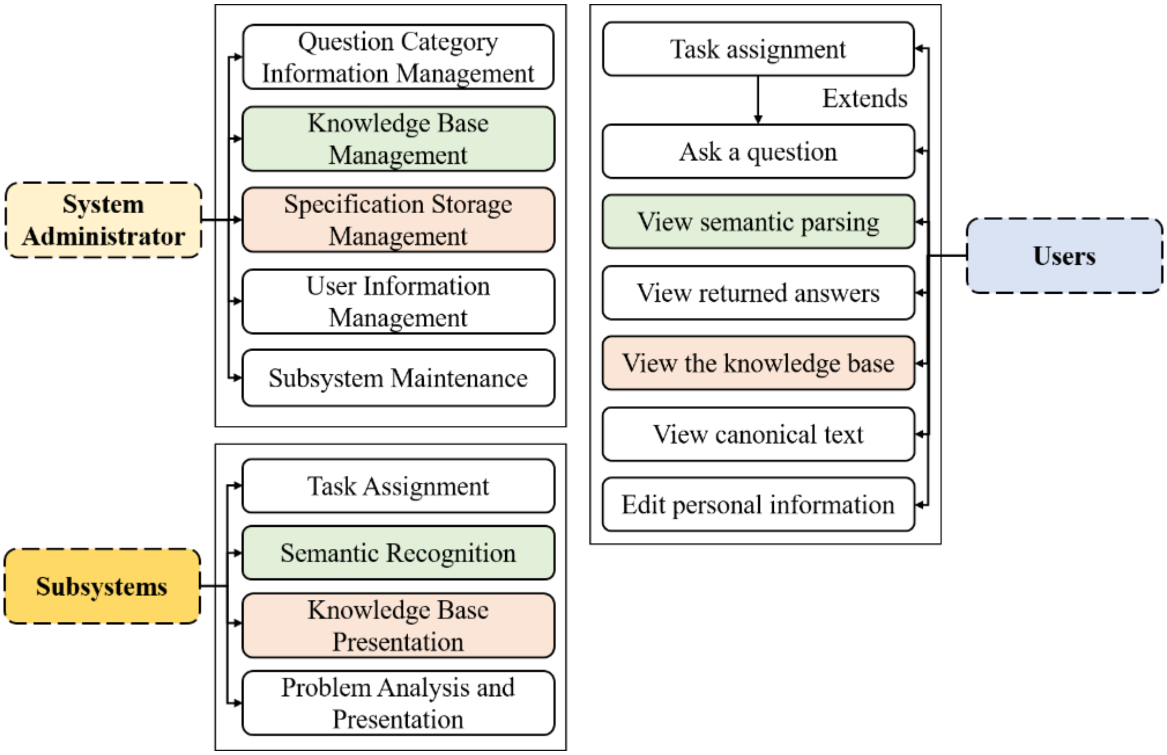

Fig. 5. Question and answer engine architecture design.

TABLE I. USE CASE DESCRIPTION OF THE QUESTION FUNCTION

Use Case Name	Question
Use Case Number	CIVILQASYS001
Use Case Description	User fills out natural language questions on the system.
Participants	Registered users of the system, database, quiz model
Pre-conditions	User login to the quiz system
Event flow	1. Registered user logs into the system 2. Click on button enter the Q&A detail information filling page. 3. Fill in the natural language question 4. Click Q&A, and the question is synchronized to the platform.
Postconditions	Can be manually selected to browse the display mode of the proposed questions.
Remark process	Fill in the natural language question format error reminder, re-entry Users abandon posting, whether to save as a draft

TABLE II. VIEW THE SEMANTIC ANALYSIS FUNCTION USE CASE DESCRIPTION

Use Case Name	View Semantic Parsing
Use Case Number	CIVILQASYS002
Use Case Description	Based on the natural language questions filled in by users on the system, the system returns parses of the questions
Participants	The system registers the user, the database, and the Q&A model.
Pre-conditions	User submits natural language question information to the Q&A model.
Event flow	1. Registered users log in and enter the system 2. Click on the Q&A module button in the platform's Q&A module to enter the Q&A details filling page. 3. Fill in the natural language question 4. Click Q&A, the question is synchronized to the platform, and the platform returns the parsing of the question.
Postconditions	Can manually choose to browse the question analysis display mode
Remark process	No

TABLE III. GET ANSWER FUNCTION USE CASE DESCRIPTION

Use Case Name	View Answer
Use Case Number	CIVILQASYS003
Use Case Description	System returns answers for users to view
Participants	System registers users, database, and quiz model
Pre-conditions	User submits natural language question information to the Q&A model.
Event flow	1. Registered users log in and enter the system 2. Click on the Q&A module button in the platform's Q&A module to enter the Q&A details filling page. 3. Fill in the natural language question 4. Click Q&A, the question is synchronized to the platform, and the platform returns the answer and displays it.
Postconditions	Can manually select how to browse the question analysis display
Remark process	The answer is returned as empty or error alert, resubmit to get the answer.

TABLE IV. VIEW THE USE CASE DESCRIPTION OF THE CORRESPONDING KNOWLEDGE BASE FUNCTION

Use Case Name	View Answer
Use Case Number	CIVILQASYS004
Use Case Description	The user fills in natural language questions on the system and gets the knowledge base corresponding to the answers returned by the system
Participants	The system registers the user, the graph database, and the Q&A model.
Pre-conditions	User submits natural language question information to the Q&A model.
Event flow	1. Registered users log in and enter the system 2. Click on the Q&A module button in the platform Q&A module to enter the Q&A detail information filling page. 3. Fill in the natural language question, click Q&A, and the question is synchronized to the platform. 4. The platform returns the corresponding map display
Postconditions	Click and drag the map manually
Remark process	No

V. RESULTS AND DISCUSSION

A. Experimental Data Collection

In order to validate the effectiveness of the model, this paper uses the BQ Corpus dataset (Bank Question Corpus) as a data source for training and testing. The BQ dataset, published by the Intelligent Computing Research Center of Shenzhen Graduate School, Harbin Institute of Technology, is a question and answer dataset in the field of banking and finance containing 120,000 question and answer pairs from online banking customer service logs. Question-answer pairs, of which 100000 are used as the training set, 10000 as the training set, and 10000 as the validation set, which is a binary classification dataset. For example, “How did the microparticle loan disappear” and “The microparticle loan is gone” are labeled as 1, indicating that they are entailment, and “How can I change my card?” and “please change your card faster” are labeled 0, indicating a contradiction. There are a large number of synonyms and near-synonyms in this dataset, which is a good way to test the validity of the model. Table V shows example of BQ dataset.

TABLE V. EXAMPLE OF BQ DATASET

Sentence 1	Sentence 2	Tags
Does Microsoft spending count?	How much is left to pay back?	0 (contradiction)
What are the best products for next week?	What financial products are available in January	1 (implied)
Can you check your bill?	I can check your bill	0 (contradiction)
Can't borrow	qq has particulate generation	0 (contradiction)

B. Evaluation Indicators

In this chapter, Accuracy (Acc) and the F1-score are employed as evaluation metrics. Accuracy is a commonly calculated value used to assess a model's classification performance, representing the percentage of correct predictions on a dataset. The F1-score, a statistical measure of a binary classification model's accuracy, evaluates the performance of a binary text semantic matching model. The F1 score is the harmonic mean of precision and recall, offering a balanced measure that takes both into account. The formulas for both metrics are presented below:

$$Acc = \&\frac{TP+TN}{TP+FN+FP+TN}\&\& \tag{14}$$

$$P = \frac{TP}{TP+FP}\& \tag{15}$$

$$R = \frac{TP}{TP+FN}\& \tag{16}$$

$$F1 - scor\&e = 2 \times \frac{P \times R}{P+R}\& \tag{17}$$

Where TP is the true case, TN is the true negative case, FP is the false positive case and FN is the false negative case.

A 4-card GPU server with model RTX2080ti was used for the experiments in this chapter. The model training parameters as well as the software version are shown in Table VI. The software versions are as follows: Python 3.6.13, PyTorch 1.10.2, OpenHowNet 2.0, Transformer 4.18.0.

C. Test Results

In the pre-training experiments, BERT processes text at the word level. To obtain the vector for each word, this model extracts the word vectors, concatenates them, and then applies average pooling. The resulting vector is used as the representation for the current word. The selected dataset for these experiments is the BQ Corpus. To ensure consistency, all models use the same Jieba word list for this dataset. The metrics used for comparison are accuracy (Acc) and F1 score. Table VII presents the relative enhancement value, which indicates the percentage improvement in effectiveness of the proposed model compared to the baseline model with the highest performance.

From Table VII, it can be seen that the model proposed in this paper has higher accuracy Acc and F1 than the other models on the BQ Corpus dataset, and on the non-pre-trained model, the accuracy Acc improves by 2.21% and the F1 value improves by 1.98% when compared to the best performing DSSM model. While on the pre-trained model, the accuracy is improved by 0.177% and F1 value by 0.464% compared to the native BERT-wwm-ext. This indicates that the semantic information between two sentences plays a more important role in processing the semantic information of banking and finance, and there is a large improvement on the non-pre-trained model, while the improvement on the pre-trained model is more limited, mainly because there are fewer words that can be matched in the sentence pairs of this dataset, which leads to a certain limitation of the experimental effect.

TABLE VI. MODEL PARAMETERS

Parameters	Numerical value
Word Embedding Layer Dimension	500
Number of hidden layers	164
Maximum sequence length	80
Batch size batch_size	128
Transformer encoder layers	12
Model Optimizer	AdamW
Initial learning rate	0.05

TABLE VII. EXPERIMENTAL RESULTS FOR THE BQ DATASET

Model	**Pre-trained or not**	**Acc**	**F1**
DSSM	×	0.7693	0.7594
MwAN	×	0.7421	0.7289
DRCN	×	0.7485	0.7576
Ours	×	**0.7902**	**0.7691**
Relative promotion	×	**+2.21%**	**+1.98%**
BERT-wwm-ext	√	0.8391	0.8402
BERT	√	0.8466	0.8399
Ours-BERT	√	0.8495	0.8433
Relative promotion	√	**+0.177%**	**+0.464%**

Regarding error analysis, since both methods directly use Jieba for text preprocessing, the segmentation errors produced by Jieba have varying degrees of impact on the experimental results. Although there is a word separation error, for the same dataset, all the models use the same word list, in comparison, the model proposed in this study has better results. Table VIII shows practical application test results.

TABLE VIII. PRACTICAL APPLICATION TEST RESULTS

Sentence pairs	Model name	Predicted results	True label
A: Unbind Chants B: Cancel the opening of chanting	ESIM	×	√
	MwAN	×	
	Our model	√	

To understand the relative importance and effectiveness of various components of the model, an ablation study was performed on the different structures of the proposed model. This study also aimed to examine the extent to which the granularity of disambiguation affects the model's performance, experiments were conducted to evaluate the effect of using three different disambiguation tools, Jieba, PKUseg and HanLP, on the results of the experiments, which were conducted using the BQ Corpus dataset.

TABLE IX. EXPERIMENTAL RESULTS OF DIFFERENT SEGMENTATION TOOLS

Segmentation tool used	Whether using HowNet	Acc	F1
Jieba	√	**0.7892**	**0.7668**
	×	0.7778	0.7627
PKUseg	√	**0.7876**	**0.7667**
	×	0.7783	0.7622
HanLP	√	**0.7861**	**0.7608**
	×	0.7742	0.7521

From the experimental results Table IX, it can be obtained that the use of HowNet can improve the performance of the model to a certain extent, and the accuracy is improved under various word segmentation tools compared to not using HowNet. The different accuracies of different segmentation tools are in line with the expected estimation because the segmentation tools have subtle differences for very few words. When a data sample contains words with extensive original information and complex relationships with other words, incorporating an external knowledge base can significantly enhance the model's sensitivity to polysemous and near-synonymous words. This integration improves the model's comprehension of synonym information and substantially boosts its overall performance.

In order to evaluate the influence of the number of layers of Transformer on the experimental effect, multi-layer Transformer experiments were carried out, and the experimental results are shown in Fig. 6.

From the data presented in Fig. 6, it is evident that the model's performance improves with an increased number of Transformer layers, a trend also observed in the BERT model. While batch stacking Transformer encoding layers can enhance model performance to some extent, it also leads to a significant increase in the model's parameters, training time, and a slower convergence speed.

In the context of non-pre-trained models, the one with the highest F1 score, featuring six Transformer encoding layers, is considered optimal and has 16 million parameters. In comparison, the DRCN model achieves the success results in non-pre-trained settings, has 19 million parameters. This indicates that the model shown in this paper has fewer cost than the DRCN model, suggesting better suitability for lightweight deployment on the BQ dataset while achieving superior experimental results.

During training, the trainable parameter γ varies with the number of iterations. As shown in Fig. 7, observing the change in γ of the attention matrix reveals that as iterations increase, the weight of the original information matrix from HowNet within the attention matrix gradually increases and stabilizes. This indicates that the semantic raw information generated by word pairs in the original text through HowNet positively impacts the model's performance enhancement.

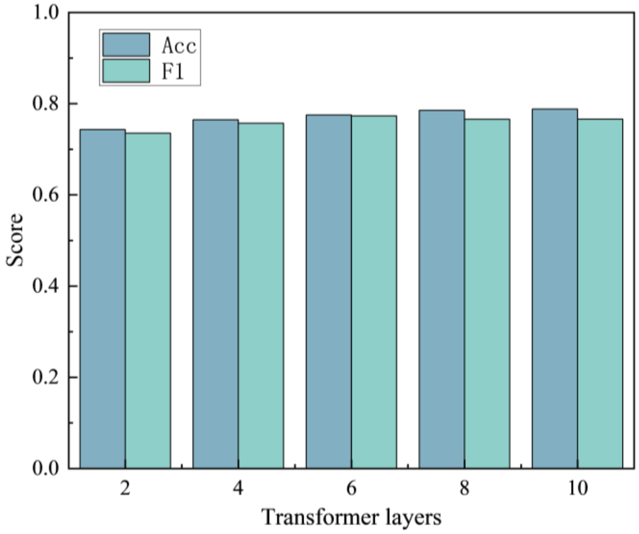

Fig. 6. Experimental results with different number of transformer layers.

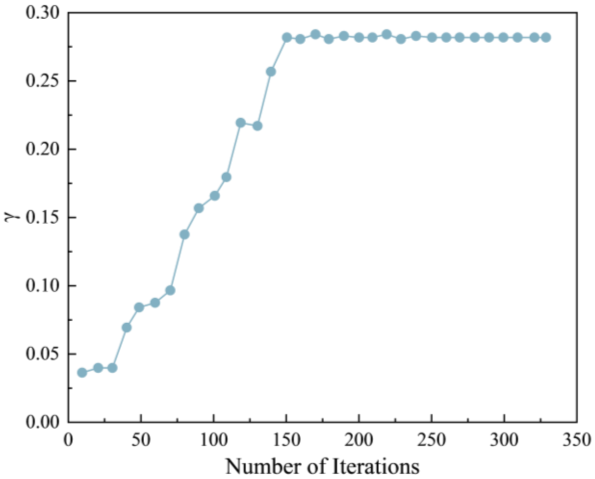

Fig. 7. Plot of changes in trainable parameters.

VI. Conclusion

This study presents a significant advancement in the field of automated question and answer (Q&A) systems by leveraging the integration of deep learning and natural language processing (NLP) technologies. With the ever-growing volume of information on the Internet, traditional keyword matching techniques have become increasingly inadequate for addressing the complexity and diversity of user queries. Our approach, which combines the Transformer model with the HowNet knowledge base, provides a robust solution to enhance semantic understanding and contextual relevance in automated Q&A systems. The architecture of our proposed system is meticulously designed, incorporating multiple layers for word embedding, Transformer encoding, attention mechanisms, and Bi-LSTM processing. This multi-layered approach allows for a comprehensive analysis and processing of natural language queries, facilitating accurate and contextually appropriate responses. The inclusion of the HowNet knowledge base is particularly noteworthy, as it enables the system to handle synonyms and near-synonyms more effectively, a critical aspect when dealing with Chinese text.

The experimental evaluation using the BQ Corpus dataset, which consists of question-and-answer pairs in the banking and finance domain, underscores the efficacy of our model. The results demonstrate substantial improvements in both accuracy and F1-score compared to existing models. Specifically, our system achieved an accuracy improvement of 2.19% and an F1-score improvement of 1.96% over the best-performing non-pretrained model, DSSM. For pre-trained models, our system showed an accuracy improvement of 0.177% and an F1-score improvement of 0.464% over the native BERT-wwm-ext model. These enhancements validate the effectiveness of integrating external knowledge bases with deep learning models in improving the performance of automated Q&A systems. Furthermore, the incorporation of HowNet significantly enhances the system's ability to process and understand the semantic relationships between words, leading to more accurate and contextually relevant responses. This is particularly important in domains where precise information retrieval is critical, such as banking and finance.

This study's findings underscore the significance of integrating external knowledge bases, such as HowNet, with advanced deep learning models to address the limitations of traditional keyword matching methods. The proposed system enhances both the efficiency and accuracy of information retrieval while providing a more user-friendly approach to accessing precise and meaningful data. This improvement is vital in the era of big data and web 3.0, where rapid and accurate access to relevant information is essential for users and organizations alike.

Future research and development offer several promising directions. Enhancing the knowledge base with more diverse and comprehensive datasets could further augment the system's capabilities. Additionally, exploring alternative deep learning architectures, including reinforcement learning and more advanced attention mechanisms, may yield further performance enhancements. Applying this system across various domains beyond banking and finance, such as healthcare, legal, and customer service, could demonstrate its versatility and broad applicability. In conclusion, the integration of deep learning with external knowledge bases represents a promising avenue for developing automated Q&A systems. This study lays a solid foundation for future advancements, paving the way for more efficient, accurate, and user-friendly information retrieval systems in the big data and web 3.0 era.

References

[1] Fan, Jianqing, Fang Han, and Han Liu. "Challenges of big data analysis." National science review 1.2 (2014): 293-314.

[2] Gan, Wensheng, et al. "Web 3.0: The future of internet." Companion Proceedings of the ACM Web Conference 2023. 2023.

[3] Tzenios, Nikolaos. Corporate Espionage and the Impact of the Chinese Government, Companies, and Individuals in Increasing Corporate Espionage. Apollos University, 2023.

[4] Sejnowski, Terrence J. "Large language models and the reverse turing test." Neural computation 35.3 (2023): 309-342.

[5] Gao, Rujun, et al. "Automatic assessment of text-based responses in post-secondary education: A systematic review." Computers and Education: Artificial Intelligence (2024): 100206.

[6] Ciesla, Robert. The Book of Chatbots: From ELIZA to ChatGPT. Springer Nature, 2024.

[7] Aithal, Shivani G., Abishek B. Rao, and Sanjay Singh. "Automatic question-answer pairs generation and question similarity mechanism in question answering system." Applied Intelligence (2021): 1-14.

[8] Chen J, Zhang R, Mao Y, Wang B, Qiao J. 2019. A Conditional VAE-Based Conversation Model. Communications in Computer and Information Science, 165-174 .

[9] Gu X, Cho K, Ha J, Kim S. 2019. Dialogwae: Multimodal response generation with conditional Wasserstein auto-encoder. The 7th International Conference on Learning Representations, 56-63.

[10] Staudemeyer, Ralf C., and Eric Rothstein Morris. "Understanding LSTM--a tutorial into long short-term memory recurrent neural networks." arXiv preprint arXiv:1909.09586 (2019).

[11] Huang, Zhiheng, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF models for sequence tagging." arXiv preprint arXiv:1508.01991 (2015).

[12] Arsovski S, Cheok A D, Govindarajoo K, Salehuddin N, Vedadi S. 2020. Artificial intelligence snapchat: Visual conversation agent. Applied Intelligence, 50(7):2040-2049

[13] Kolomiyets, Oleksandr, and Marie-Francine Moens. "A survey on question answering technology from an information retrieval perspective." Information Sciences 181.24 (2011): 5412-5434.

[14] Bowman S R, Vilnis L, Vinyals O, Dai A M, Jozefowicz R, Bengio S. 2016. Generating sentences from a continuous space. The 20th SIGNLL

Conference on Computational Natural Language Learning, Berlin, Germany, 10-21

[15] Cai Y, Zuo M, Zhang Q, Xiong H, Li K. 2020. A Bichannel Transformer with Context Encoding for Document-Driven Conversation Generation in Social Media. Complexity, 48(7):1-13.

[16] Yu, Yong, et al. "A review of recurrent neural networks: LSTM cells and network architectures." Neural computation 31.7 (2019): 1235-1270.

[17] Sherstinsky, Alex. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." Physica D: Nonlinear Phenomena 404 (2020): 132306.

[18] Zhou, Chunting, et al. "A C-LSTM neural network for text classification." arXiv preprint arXiv:1511.08630 (2015).

[19] Cahuantzi, Roberto, Xinye Chen, and Stefan Güttel. "A comparison of LSTM and GRU networks for learning symbolic sequences." Science and Information Conference. Cham: Springer Nature Switzerland, 2023.

[20] Xu, Weidong, et al. "Long-short-term-memory-based deep stacked sequence-to-sequence autoencoder for health prediction of industrial workers in closed environments based on wearable devices." Sensors 23.18 (2023): 7874.

[21] Abumohsen, Mobarak, Amani Yousef Owda, and Majdi Owda. "Electrical load forecasting using LSTM, GRU, and RNN algorithms." Energies 16.5 (2023): 2283.

[22] Lu, Guangquan, et al. "Multi-task learning using variational auto-encoder for sentiment classification." Pattern Recognition Letters 132 (2020): 115-122.