Research on Construction Site Safety Q&A System Based on BERT

Ang Li
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: la630670659@163.com

Jianguo Wang*
Research Institute of Artificial Intelligence and Data Science
Xi'an Technological University
Xi'an, China
E-mail: wjg_xit@126.com

***Abstract*—This paper aims to utilize the pre-trained language model BERT from deep learning to construct A question and answer system specifically targeting safety knowledge in construction sites, thereby enhancing safety management on-site and increasing workers' awareness of safety issues. Through extensive reading of literature related to construction site safety and the integration of practical case studies, this research compares various pre-trained language models such as word2vec, Pre-trained RNN, GPT, and BERT, analyzing their respective advantages and disadvantages. Despite the fact that word embedding methods such as word2vec have improved the effectiveness of natural language processing to some extent, their ability to understand context is limited. Pre-trained RNNs, although capable of handling sequential data, suffer from the problem of gradient disappearance when dealing with long-range dependencies. In contrast, the GPT model performs well in generative tasks; however, due to its reliance on a unidirectional language model, it falls short in understanding bidirectional contexts. Ultimately, it was determined that a method based on BERT would be most suitable for improving the model to meet the safety needs of construction sites. The system can accurately understand and respond to safety-related questions posed by workers, thereby preventing accidents and ensuring the safety of construction site personnel. This study not only explores the optimization and adjustment of the BERT model but also evaluates its performance in practical application scenarios, providing new technological means for safety education and management within the construction industry.**

Keywords-component; BERT; Question and Answer System; Construction Industry; LLM

I. INTRODUCTION

With the rapid advancement of computer science and artificial intelligence technologies, the field of Natural Language Processing (NLP) has made groundbreaking progress, particularly in the development of intelligent question answering systems. These systems aim to mimic human communication by understanding questions posed by users in natural language and then providing precise answers, significantly enhancing the efficiency and quality of information retrieval. Moreover, this technology demonstrates extensive application potential across various fields, especially in the realm of construction safety. Faced with complex and dispersed safety knowledge and business data, traditional management methods are no longer sufficient to meet the demands for efficient and accurate queries.

Currently, safety management on construction sites faces severe challenges. The site environment is dynamic, with numerous safety hazards present. Additionally, safety standards and operational guidelines are scattered across various sources, such as paper documents, websites, and internal databases. This not only increases the difficulty of information retrieval but also significantly affects the efficiency of safety education, training, and daily management. Therefore, the development of an intelligent question answering system that can integrate and rapidly parse these unstructured data sources has become particularly urgent.

Alongside the development of computer technology, question answering systems have seen extensive development across various vertical domains. The first chatbot, Eliza [1], was invented by foreign computer scientists in 1966 and was

used in psychological counseling, utilizing matching rules similar to decision trees to analyze input sentences. Although question answering systems based on pattern matching and manually written rules could provide reasonable responses, building a large number of dialogue templates to satisfy the rich expressiveness of language was required, making this approach applicable only in a few specialized fields. In recent years, with the advancement of deep learning, the use of neural networks for lexical analysis has become a focal point of research. Currently, in the industry, deep learning-based text matching mainly includes three categories: vector similarity-based matching, deep neural network-based matching, and matching based on pre-trained models [2]. The NNLM [3] proposed a method of computing text vector similarity primarily to address the synonym problem in traditional statistical methods; however, the training resource consumption is significant, requiring several weeks of training using 40 CPUs for a dataset with millions of entries [4]. With the expansion of natural language processing applications, researchers have combined deep learning models for NLP tasks [5], mainly to solve semantic representation at the sentence level and asymmetry problems in text matching. Microsoft introduced DSSM [6] in 2013, which was the earliest deep semantic matching model. The DSSM model maps Queries and Docs to a low-dimensional semantic space and measures the relevance between Query and Doc through Cosine similarity. Although interaction models can capture deeper semantic information with more complex neural network structures, they overlook syntactic and global information across sentences. In 2018, Google introduced the pre-training model BERT (Bidirectional Encoder Representations from Transformers), which ranked first in various NLP task leaderboards [7]. BERT provides a new approach for the rapid acquisition of safety knowledge on construction sites. Leveraging its strong contextual understanding and generation capabilities, BERT has achieved leading positions in multiple NLP tasks. The model, through unsupervised pre-training on large-scale corpora, has learned rich linguistic features and complex semantic relationships, demonstrating excellent performance during the fine-tuning phase for specific tasks.

In summary, the work presented in this paper will focus on utilizing the latest deep learning technologies, particularly BERT and subsequent pre-trained models, in conjunction with domain knowledge and the requirements of real-world applications, to research and develop an efficient and accurate question answering system for construction site safety knowledge. This endeavor aims to improve the current efficiency of safety management and enhance the overall safety levels of construction sites. Through literature analysis and technical exploration, this study hopes to provide strong technological support for the informatization and intelligent transformation of the construction industry.

II. IMPROVED BERT MODEL

A. BERT Model Analysis

In traditional question answering systems, static word vector algorithms such as Word2Vec are often employed. These methods map input words into unique and invariant vectors, resulting in word embeddings that do not incorporate contextual information and cannot resolve the issue of polysemy in texts. In recent years, the emergence of pre-trained language models has ushered in a new era for natural language processing, replacing the original static word vectors and downstream task integration, thereby enhancing performance. BERT is currently one of the most successfully applied language models in the industry. Due to its powerful feature representation capabilities, fine-tuning with BERT requires only a small amount of relevant corpus data and appropriate parameter adjustments to reach the usability threshold of the domain [11]. To meet the needs of generating responses according to specific question-answer styles in a construction site safety knowledge QA system, the BERT model has been introduced. BERT is a deep bidirectional pre-trained language model based on the Transformer architecture [12], capable of capturing the common features of both preceding and following contexts to encode unlabeled text. When used in different natural language

processing scenarios, BERT typically requires the addition of an extra input layer. It is currently the most popular and widely used pre-trained language model, serving as the foundational architecture for enhancing model performance across various downstream tasks.

Prior to BERT, the most successful pre-trained language model was GPT, which utilized a left-to-right autoregressive approach for pre-training. However, in terms of language understanding, it is considered a bidirectional process because information from both the left and right sides of the current content is helpful for comprehension. In other words, GPT represents a unidirectional process, for reasons including the need for a defined direction. Naturally, this approach breaks down long texts into smaller segments and generates them progressively, essentially decomposing the process. Secondly, if all the content were fed into a bidirectional model, it could lead to information leakage. Figures 1 and 2 illustrate the unidirectional and bidirectional processes of GPT.

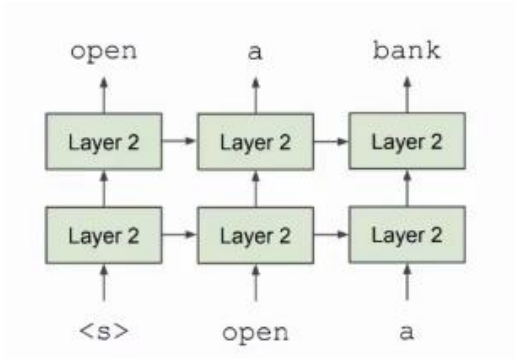

Figure 1. Unidirectional context build representation incrementally.

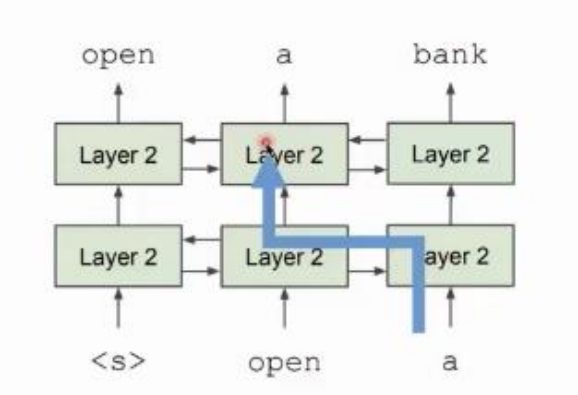

Figure 2. Bidirectional context words can "see themselves".

BERT proposes a solution with the masked language model approach. For example, in the sentence 'the man went to the [MASK] to buy a [MASK] of milk,' two [MASK] tokens cover the original words 'store' and 'gallon.' This masked language model is the core pre-training task of BERT, similar to a cloze test process. It employs a strategy of randomly masking 15% of the tokens. The choice of 15% is a trade-off. There are two main considerations: if the masked proportion is too low, there would be very little supervisory signal; if the proportion is too high, there would be very little usable information left in the text.

Masking solves the problem of information leakage, but it also introduces another issue: masked tokens do not appear during downstream tasks, creating a significant discrepancy between the pre-training and fine-tuning stages, which may degrade the model's performance.

B. Improvement Ideas and Methods

Therefore, to address the aforementioned issues, this paper proposes the following: when masking 15%, it should be divided into several subtypes for processing. Specifically, 80% of the time, the tokens should be replaced with [MASK]. For example, 'went to the store' would be converted to 'went to the [MASK],' meaning that the word 'store' is replaced with [MASK].

Next, 10% of the time, randomly replace the token with another word at a 10% chance. For example, in 'went to the store,' the word 'store' would be replaced with 'running,' and the model would be required to predict 'store' from 'running.' This approach forces the model to pay attention to words that do not appear to be masked, thus maintaining a better representation.

However, there is still another issue: the model might always assume that the randomly inserted word in the real scenario is incorrect. To address this, an additional strategy is implemented. An additional 10% of the time, the word order is kept normal. For example, 'went to the store' remains 'went to the store,' meaning that 'store' is used to predict 'store.' By combining these three masking strategies—masking, random replacement, and

keeping the original word—the issues brought about by masking are mitigated.

The structure of BERT is illustrated in Figure 4.4. It consists of three main components: the input layer, the encoding layer, and the output layer. For the input layer, the input representation vectors are composed of word embeddings, position embeddings, and segment embeddings. Segment embeddings are used to distinguish between different sentences in the conversation of the question answering system, and both the segment embeddings and position embeddings require learning by the model. The special token [CLS] (classification token) is placed at the beginning of the sequence to serve as a classifier. After passing through the final Transformer layer, this classification token aggregates the representation information of the entire sequence. The [SEP] token acts as a separator placed between two sentences. The specific input representation of BERT is detailed in Figure 3.

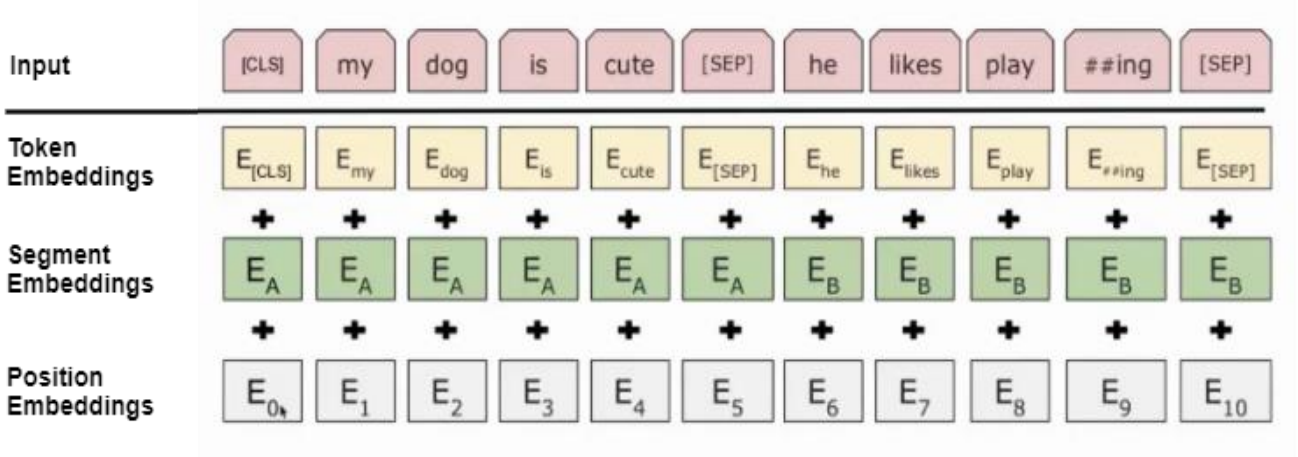

Figure 3. BERT model input diagram.

C. Model architecture

The construction site safety knowledge question answering system based on BERT has a model architecture that primarily includes an input layer, an Embedding layer, a Transformer Encoder, and an output layer. The structure is shown in Figure 4. The input layer receives preprocessed question inputs, with special [CLS] and [SEP] tokens added before the questions. In the Embedding layer, each token is mapped to a high-dimensional vector space, incorporating a combination of Token Embedding, Segment Embedding, and Position Embedding. The Transformer Encoder is the core component of BERT, consisting of multiple identical layers of multi-head self-attention (Multi-Head Attention) and feed-forward neural networks (Feed-Forward Neural Networks). This layer is responsible for capturing the semantic dependencies and structural features within the text. For the output layer, the output vector at the [CLS] position is taken as the representation of the entire sequence, followed by a fully connected layer (Fully Connected Layer) and a SoftMax activation function to predict the start and end positions of the best answer.

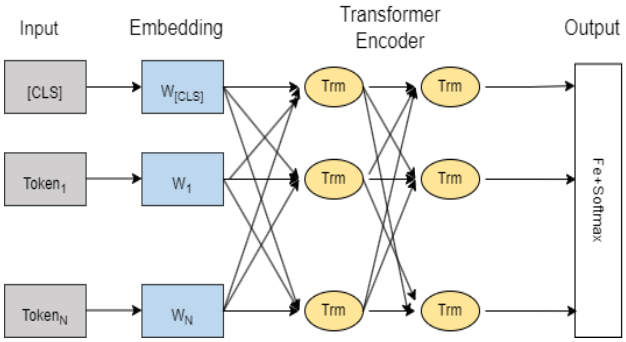

Figure 4. Model architecture of BERT-based Q&A system.

III. DATA SET CONSTRUCTION AND PRE-PROCESSING

A. Data set construction

Firstly, the objective of constructing the dataset for this paper is to encompass a broad and specific range of construction site safety knowledge,

ensuring diversity and practicality in the question-answer pairs. Data was collected using web crawlers from professional websites, forums, and blogs in the construction industry, focusing on safety questions and their answers that workers might encounter in real-life situations. In addition, common question-answer pairs were obtained from journal articles, e-books, and legal regulations related to construction safety. For the collected dataset, the model only requires the pure conversational information from the corpus. Therefore, text cleaning was performed to remove irrelevant characters, punctuation marks, unify the text format, and categorize the original materials into different safety topics to facilitate the creation of targeted question-answer pairs. The final number of items in the dataset is shown in TABLE I.

TABLE I. NUMBER OF SAMPLES IN THE DATA SET

Data Set Number	**Dataset Name**		
	training set	***validation set***	***test set***
67700	47750	9975	9975

B. Pre-processing steps

Preprocessing is a crucial step in this project, directly impacting the model's training effectiveness and the ultimate performance of the question answering system. Initially, tokenization is required, which involves breaking continuous text into individual words or lexical units. This is particularly important for Chinese text, as there are no clear word boundaries. In this paper, we utilize the Full-Tokenizer that comes with the BERT model. The process involves first applying Basic-Tokenizer to obtain a relatively coarse list of tokens, followed by Word-Piece Tokenizer to achieve the final tokenization results. Subsequently, the tokenized text needs to be converted into Tokens from the model's vocabulary. For words not found in the vocabulary, the UNK (unknown) Token is used, or sub-word encoding is applied to transform the text into a numerical form that the model can interpret.

Wherever Times is specified, Times Roman or Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times. Avoid using bit-mapped fonts if possible. True-Type 1 or Open Type fonts are preferred. Please embed symbol fonts, as well, for math, etc.

IV. EXPERIMENTS AND ANALYSES

A. Experimental environment

The model in this study is implemented in Python and utilizes the deep learning framework PyTorch. The training process was conducted under the Ubuntu operating system. To ensure stable operation and good performance of the model, we meticulously configured the software and hardware environment for our experiments. The specific environmental configurations are detailed in TABLE II, including the operating system, version of the programming language, version of the deep learning framework, and versions of other necessary software libraries.

Additionally, we specified the GPU model and memory capacity to ensure adequate computational resources during the model training process. Through carefully configured experimental settings, we were able to effectively manage and optimize the training process, thereby ensuring the smooth progress of our research.

TABLE II. EXPERIMENTAL ENVIRONMENT CONFIGURATION PARAMETERS

Experimental Environment	**Configure**
operating system	Ubuntu
development language	Python3.8.8
development framework	Pytorch1.8.0
CPU	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz2.21 GHz
GPU	NVIDIA GeForce GTX3070Ti 8G
random access memory (RAM)	Kingston 2400Mhz 16.0 GB

B. Experimental parameter settings

For the training of the designed question answering model, the learning rate was set to 10e-5, the batch size was set to 16, and the number of epochs for the training set was set to 20. The

Adam optimizer was used for optimization, and a redesigned loss function based on cos (u, v) was adopted, as shown in Equation (1). Here, t ∈ {0, 1} indicates whether the samples are similar, where u and v represent the sentence feature vectors of Question 1 and Question 2, respectively. The purpose of the loss function is to maximize the similarity for positive sample pairs and minimize the similarity for negative sample pairs.

$$L = t \cdot \left(1 - cos(u,v)\right) + (1-t) \cdot \left(1 + cos(u,v)\right) \quad (1)$$

The experimental training parameters are set as shown in TABLE III:

TABLE III. EXPERIMENTAL ENVIRONMENT CONFIGURATION PARAMETERS

Experimental Parameters	Retrieve A Value
Learning rate	2e-5
Batch Size	16
Num of epoch	20
Length of Maxseq	128

C. Evaluation indicators

This paper adopts commonly used evaluation metrics in deep learning, namely accuracy, recall, and the F1 score. Accuracy represents the ratio of correctly predicted positive samples to all positive samples. The F1 score is the harmonic mean of precision and recall. Recall represents the ratio of correctly predicted positive samples to all actual positive samples. The formulas for these three metrics are given in Equations (2), (3), and (4):

$$P_{precision} = \frac{TP}{TP + FP} \quad (2)$$

$$r_{recall} = \frac{TP}{TP + FN} \quad (3)$$

$$F = 2 * \frac{r_{Recall} * P_{precision}}{r_{Recall} + P_{precision}} \quad (4)$$

In the above equations: TP represents the number of true positives, where the model correctly predicts positive samples as positive; TN represents the number of true negatives, where the model correctly predicts negative samples as negative; FP represents the number of false positives, where the model incorrectly predicts negative samples as positive; FN represents the number of false negatives, where the model incorrectly predicts positive samples as negative.

D. Experimental results and analyses

For this study, to measure the accuracy of the BERT-based question answering system, it was compared with several baseline models, and the results are shown in TABLE IV:

TABLE IV. COMPARATIVE EFFECTS OF DIFFERENT BASELINE MODELS

Modelling	Evaluation Metrics		
	P/%	***R/%***	***F1/%***
LSTM	72.72	68.63	70.61
Text-CNN	73.40	70.23	71.78
BERT	80.5	80.96	81.65

According to TABLE IV, it can be observed that the LSTM and Text-CNN models exhibit comparable performance on the question answering task. However, the pre-trained BERT model demonstrates significantly better results than the other baseline models. Furthermore, during the training process, BERT's performance tends to improve with the increase in model size, as illustrated in Figure 5. This suggests that larger variants of BERT are more capable of capturing complex patterns in the data, which leads to higher accuracy in answering questions related to construction site safety. The superior performance of BERT can be attributed to its ability to understand context and semantic relationships within the text, which is critical for accurately interpreting the nuances of safety-related queries.

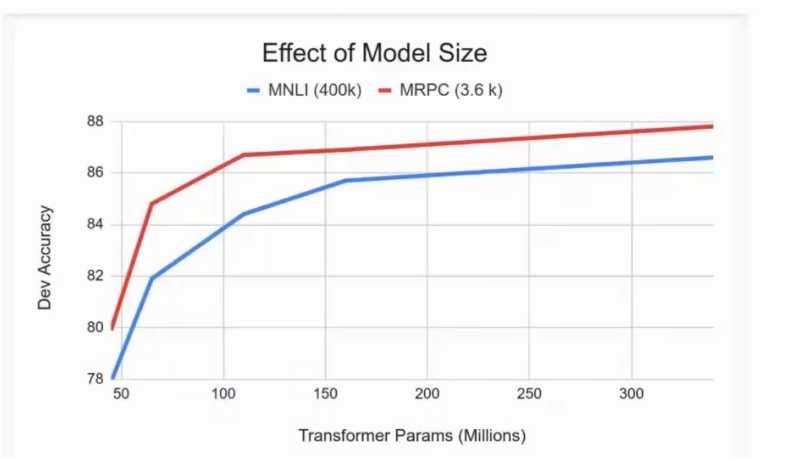

Figure 5. Effect of BERT with increasing model size.

V. WEB PAGE DESIGN

A simple graphical user interface (GUI) was constructed based on the model trained in this study, designed to facilitate user interaction. The entire web page project was created using the Next.js template configuration within WebStorm. The web design encompasses components such as a toolbar panel, button definitions and implementations, dialog lists, dialog messages, and dialogue handling.

Initially, a small sidebar was designed in this paper, which influenced the overall structure definition and page routing management of the interface. Two buttons were placed on the left sidebar: one for chat and another for role selection. The chat button facilitates dialogue handling, while the role button allows users to choose from various scenarios, including roles such as engineer, project manager, worker, etc. The coding of the webpage was carried out using an object-oriented approach, defining functions, methods, properties, and incorporating packages similar to how it is done in Java.

Subsequently, button functionality was introduced to achieve zooming effects on the page, enhancing the web UI's infrastructure modules. Although implementing a single button might appear minor, the inclusion of multiple buttons necessitated the design of a generic button framework along with configuration storage. When operating these buttons, configurations could be set and utilized, thereby adjusting the interface scale. Buttons serve as a small feature point that leads to the realization of various modules within the overall web UI architecture, encompassing aspects such as button definition, CSS design, and TypeScript syntax.

Following this, a dialog module was added to implement a list of dialog boxes. Corresponding test data was also included, and new sessions were created upon clicking the "+" button. Building upon the completed sidebar implementation, the development of the dialog list window was initiated. When users engage in dialogue with the model, different roles may emerge, such as engineer, project manager, or worker. These windows need to be displayed in a list format within the dialog box list, akin to how conversations are presented with different contacts in WeChat. Consideration was given to what information should be displayed, such as who was being chatted with, the number of chat messages exchanged, and when the last conversation ended, and how to present this information on the interface.

Subsequently, the focus was placed on implementing the dialog message feature, including the setup of corresponding sub-routing, page navigation, message transmission, interface design, and realization. After establishing the dialog box list, the content of the dialog box message panel needed to be realized. This meant that when a user clicked on an element in the list of dialog boxes, a corresponding dialog box message would appear on the right side, along

with an input field for messages. Additionally, the dialog box list was stored locally within the browser to preserve the user's conversation history. Such information could also be retrieved via server-side APIs. However, relying more on the browser's local storage rather than server storage reduces server load, particularly beneficial for non-corporate users who wish to deploy such services with minimal dependencies. Similarly, message information was stored locally in the user's browser, eliminating the need for server-side storage. Future expansions might include server-side storage options. Finally, concerning message transmission and presentation, considering that the data returned by the model is in Markdown format, especially for code snippets, rendering this data becomes necessary for better readability. Thus, extending Markdown rendering capabilities was essential.

For the of dialogue roles, this paper predefines these roles within the program, allowing users to initiate a dialogue operation with a specific role by simply clicking on it. Additionally, clicking the role button navigates users to a list of roles. Each role introduces its own functionalities, and once a user chooses to converse with a particular role, a chat message is created in the user's dialog box.

The web system's user interface described above is illustrated in Figure 6.

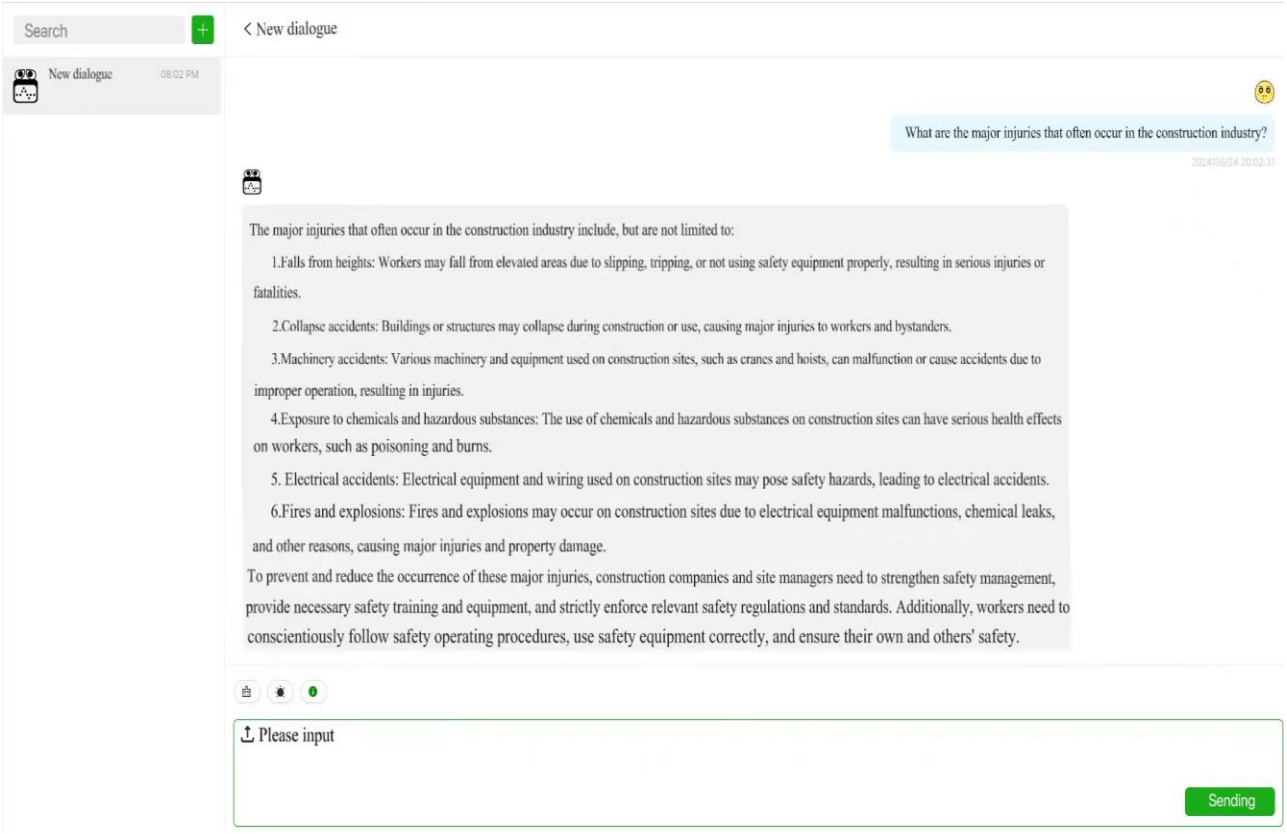

Figure 6. Graphical interface of the construction site Q&A system.

VI. CONCLUSIONS

The primary contribution of this paper is the proposal of a construction site safety knowledge question answering system based on the BERT model. Reliable safety knowledge from the construction industry was gathered from the internet to build a testing dataset, and experiments were conducted on this dataset with the proposed model. Comparisons with other baseline models demonstrated that this model can be effectively applied in the construction site industry. The system enables workers to quickly and accurately acquire safety knowledge, with a deeper understanding of the textual queries provided by users, resulting in more precise answers and

significantly improving the efficiency and accuracy of safety information retrieval for workers.

However, the limitations of the BERT model in terms of computational resource consumption, domain-specific knowledge constraints, handling of long texts, and interpretability cannot be ignored. The BERT model requires substantial computational resources for training and may lack flexibility when dealing with domain-specific knowledge. Additionally, its performance on long texts is inferior to that on short texts, and the model's decision-making process lacks transparency and interpretability. Therefore, future research will explore efficient question answering retrieval model architectures and algorithm implementations to further enhance the question answering capabilities of the system and improve the performance of its responses.

REFERENCES

[1] Love, Rachel et al. "Natural Language Communication with a Teachable Agent", CoRR (2022).

[2] Qian Yangge et al. "A review of deep learning-based text semantic matching." Software Guide 21.12 (2022): 252-261.

[3] Sun, Simeng, and Mohit Iyyer. "Revisiting Simple Neural Probabilistic Language Models", North American Chapter of the Association for Computational Linguistics abs/2104.03474 (2021): 5181-5188.

[4] Zhang, Min, and Li, J. T." Generative pre-training model." Chinese Science Foundation 35.03 (2021): 403-406. doi: 10.16262/j.cnki.1000-8217.2021.03.014.

[5] Jingsheng Zhao, et al. "A study of text representation in natural language processing." Journal of Software 33.01 (2022): 102-128. doi:10.13328/j.cnki.jos.006304.

[6] Zeng, Yanhong et al. "Learning Pyramid-Context Encoder Network for High-Quality Image Inpainting", 2019 IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR 2019) abs/1904.07475 (2019): 1486-1494.

[7] Linyang, Li et al. "BERT-ATTACK: Adversarial Attack Against BERT Using BERT", Conference on Empirical Methods in Natural Language Processing 2020.emnlp-main (2020): 6193-6202.

[8] Lee, Jinhyuk et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining", Bioinformatics 36.4 (2020): 1234-1240.

[9] Zhang, Zhengyan et al. "Ernie: Enhanced Language Representation with Informative Entities", 57TH ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS (ACL 2019) abs/1905.07129 (2019): 1441-1451:

[10] Subakan, Cem et al. "Attention is All You Need in Speech Separation", IEEE International Conference on Acoustics, Speech, and Signal Processing abs/2010.13154 (2021): 21-25.

[11] Sufeng, Duan, and Zhao Hai. "Attention Is All You Need for Chinese Word Segmentation", Conference on Empirical Methods in Natural Language Processing 2020.emnlp-main (2020): 3862-3872.