



ARTICLE

# A Data-Enhanced Deep Learning Approach for Emergency Domain Question Intention Recognition in Urban Rail Transit

Yinuo Chen<sup>1</sup>, Xu Wu<sup>1</sup>, Jiaxin Fan<sup>1</sup> and Guangyu Zhu<sup>2,\*</sup>

<sup>1</sup>School of Traffic and Transportation, Beijing Jiaotong University, Beijing, 100044, China

<sup>2</sup>School of Automation and Software Engineering, Shanxi University, Taiyuan, 030006, China

\*Corresponding Author: Guangyu Zhu. Email: gyzhu@sxu.edu.cn

Received: 27 December 2024; Accepted: 17 April 2025; Published: 09 June 2025

**ABSTRACT:** The consultation intention of emergency decision-makers in urban rail transit (URT) is input into the emergency knowledge base in the form of domain questions to obtain emergency decision support services. This approach facilitates the rapid collection of complete knowledge and rules to form effective decisions. However, the current structured degree of the URT emergency knowledge base remains low, and the domain questions lack labeled datasets, resulting in a large deviation between the consultation outcomes and the intended objectives. To address this issue, this paper proposes a question intention recognition model for the URT emergency domain, leveraging knowledge graph (KG) and data enhancement technology. First, a structured storage of emergency cases and emergency plans is realized based on KG. Subsequently, a comprehensive question template is developed, and the labeled dataset of emergency domain questions in URT is generated through the KG. Lastly, data enhancement is applied by prompt learning and the NLP Chinese Data Augmentation (NLPCDA) tool, and the intention recognition model combining Generalized Auto-regression Pre-training for Language Understanding (XLNet) and Recurrent Convolutional Neural Network for Text Classification (TextRCNN) is constructed. Word embeddings are generated by XLNet, context information is further captured using Bidirectional Long Short-Term Memory Neural Network (BiLSTM), and salient features are extracted with Convolutional Neural Network (CNN). Experimental results demonstrate that the proposed model can enhance the clarity of classification and the identification of domain questions, thereby providing supportive knowledge for emergency decision-making in URT.

**KEYWORDS:** Emergency knowledge base for urban rail transit; emergency domain questions; intention recognition; knowledge graph; data enhancement

## 1 Introduction

When dealing with emergency events in the urban rail transit (URT) system, decision-makers or scheduling personnel need to consult the URT emergency knowledge base. In emergencies, questioners typically frame their inquiries as questions to consult the knowledge base. However, since the expression of questions is relatively random, rapidly and accurately identifying the consultation intentions of decision-making or scheduling personnel is crucial for ensuring the knowledge base produces accurate outputs. Intention recognition is essentially a text classification task focused on understanding the natural language input from users and mapping it to the predefined intention category [1]. At present, text classification methods can be roughly divided into rule-based, statistical learning-based, and deep learning-based methods, each of which is widely used across various fields [2–4]. The rule-based method [5] is straightforward and efficient but has great limitations in dealing with complex text data, which is why they are primarily used in



earlier text classification tasks. Meanwhile, advancements in computer technology have given rise to various statistical learning methods, such as the K-nearest neighbor algorithm [6], decision tree [7] and support vector machine [8]. Although these statistical models better learn the relationship between text features and categories, they rely on artificial feature engineering and struggle to manage high-dimensional feature spaces. In recent years, deep learning methods such as Word to Vector (Word2Vec) [9], Long Short-Term Memory Neural Network (LSTM) [10], Recurrent Convolutional Neural Network (CNN) [11,12] and capsule network [13] have been increasingly applied to overcome the limitations of traditional machine learning approaches. Moreover, the emergence of pre-trained models such as Bidirectional Encoder Representations from Transformers (BERT) [14], Generalized Auto-regression Pre-training for Language Understanding (XLNet) [15] have significantly improved the ability to represent complex semantic features for Natural Language Processing (NLP) tasks. They can autonomously extract and learn complex features from text data, resulting in better classification outcomes. With the ongoing development of pre-trained models, Large Language Models (LLMs) have shown remarkable capabilities in solving complex tasks [16]. Techniques such as prompt engineering and fine-tuning within LLMs will systematically enhance the accuracy and semantic generalization of intention recognition tasks.

The identification of question intentions in the field of natural language understanding has become a pressing issue, particularly in URT emergency scenarios. Several challenges hinder progress in this area: Firstly, there is a scarcity of publicly labeled question datasets specific to the URT emergency domain. The quality of available datasets is insufficient, and the labeling cost is high. Secondly, these domain-specific datasets typically suffer from insufficient diversity and contain small sample characteristics, which can lead to overfitting in question intention recognition and poor generalization performance. Thirdly, URT emergency knowledge is often stored in unstructured texts, such as URT emergency plans [17] and emergency cases [18]. This information is challenging to utilize effectively due to its large volume, complex content and low standardization. To address these challenges, it is essential to integrate URT emergency knowledge, efficiently generate question datasets, and develop a question intention recognition model that possesses strong robustness and generalization capabilities. This would enable the knowledge related to URT emergency management can be obtained quickly and accurately.

Considering the above factors, this paper focuses on the intention recognition method of questions for the URT emergency domain, utilizing data enhancement techniques. The aim is to realize the efficient classification of questions and provide theoretical and technical support for rapid access to emergency knowledge. The main contributions of this study are as follows:

1. This study designs a KG schema layer for URT emergency cases and emergency plans, linking two types of KGs. The emergency-related text is stored in the Neo4j graph database, creating an information-based and structured database for dataset generation and knowledge applications.
2. A series of comprehensive question templates are developed. By querying and storing the entities in the KG, combining regular expressions and random selection methods, the templates are filled to quickly generate the question dataset in the URT emergency domain.
3. This study employs the XLNet model to extract semantic information, combines the Recurrent Neural Network (RNN) for sequence modeling, and uses CNN to extract local features. At the same time, data back-translation and text transformation techniques are used to overcome template limitations and enhance the diversity of questions, improving their classification effectiveness.

## 2 Related Work

At present, deep learning has become the mainstream text classification method and is applied to intention recognition tasks in various fields. The application and comparison of different deep learning

networks have been widely studied to improve the effect of intention recognition. For example, Chen et al. [19] compared four deep learning models to understand users' intentions in a medical question and answering (Q&A) system, and CNN and Fast Text achieved the best results. Trehwela and Figueroa [20] explored the performance of many traditional neural networks in intention recognition of questions and proved that Recurrent Convolutional Neural Network (RCNN) is the most effective technology. Liu and Xu [21] combined Term Frequency-Inverse Document Frequency (TF-IDF), CNN and LSTM to solve the sparse problem, and used the attention mechanism to allocate model weights for short text intent recognition.

Pre-trained language models' general semantic representation ability gives them significant advantages in vertical applications. BERT [14] has implemented new effects in various NLP tasks. However, it has some limitations, such as the assumption of mask word independence and the inconsistency of pre-training and fine-tuning. XLNet is optimized for BERT's shortcomings, improving on Attentive Language Models Beyond a Fixed-Length Context (Transformer-XL) [22]. XLNet has many applications in named entity recognition [23] and less in intention recognition, while BERT has been widely used in intention recognition tasks in various fields [24,25]. XLNet is mostly used for text classification tasks such as emotion recognition. For example, Areshey and Mathkour [26] explored the performance of XLNet, BERT and their variants on emotion classification tasks, and the results showed that RoBERTa and XLNet had the best performance. Han et al. [27] combined XLNet, Bidirectional gated loop unit (BiGRU) and attention mechanism (Att) to effectively improve the model's performance in text emotion recognition tasks. There is relatively little attention paid to the task of intention recognition in the URT emergency domain, while identifying question intention is an important prerequisite for realizing tasks such as Q&A and dialogue. Therefore, it is necessary to conduct relevant research.

In the application of specific fields, intention recognition models often face the challenge of lacking publicly labeled datasets. Scholars often build datasets for specific fields or tasks according to research needs. The traditional method of generating domain datasets is generally manual annotation. For example, Weng et al. [28] sorted out 21 labels of judgment documents in the judicial field, and manually labeled each paragraph of the randomly selected 5000 judgment documents. Zhang et al. [29] invited three experts to manually annotate the category mapping of emergency supplies between the GB/T 38565 and the GPC taxonomy, establishing 798 fully mapped category pairs. Given that the traditional method is costly and inefficient, studying the fast-generation method of questions is necessary. By utilizing predefined intent types [30] and the dataset generation method based on question templates and entity substitution [31], the annotation costs can be effectively reduced, and the efficiency of question generation can be improved. A knowledge graph (KG) can store URT emergency knowledge in a structured and standardized way [32], thus providing structured information. Combined with the flexible retrieval function of KG [33], entity data can be quickly obtained to fill the question template, making it convenient to generate high-quality question datasets. Some scholars have transformed emergency-related text data into KG. For example, Liu et al. [34] constructed water conservancy project emergency plans as KG and provided retrieval of emergency knowledge such as risks and measures. Zhu et al. [35] constructed a KG for URT safety event cases as the basis of URT emergency response.

However, the domain datasets usually contain limited information and have the characteristics of small sample learning. In recent years, many scholars have paid attention to the small sample learning method. For example, Zhao et al. [36] used the SimBERT model to enhance the data of small sample datasets, improving multiple models' classification effects. Cao et al. [37] enhanced the ability of cross-language small-sample intent recognition through prompt tuning. Luo and He [38] fused the original sentence with an intention label to enrich semantic information and improve the model performance. Yan et al. [39] integrated the triples in the KG into the training data, effectively realizing the knowledge enhancement of

text representation. Therefore, it is necessary to consider data enhancement technology while studying the fast generation method of the dataset, thereby effectively improving the performance of the small sample intention recognition model under data scarcity.

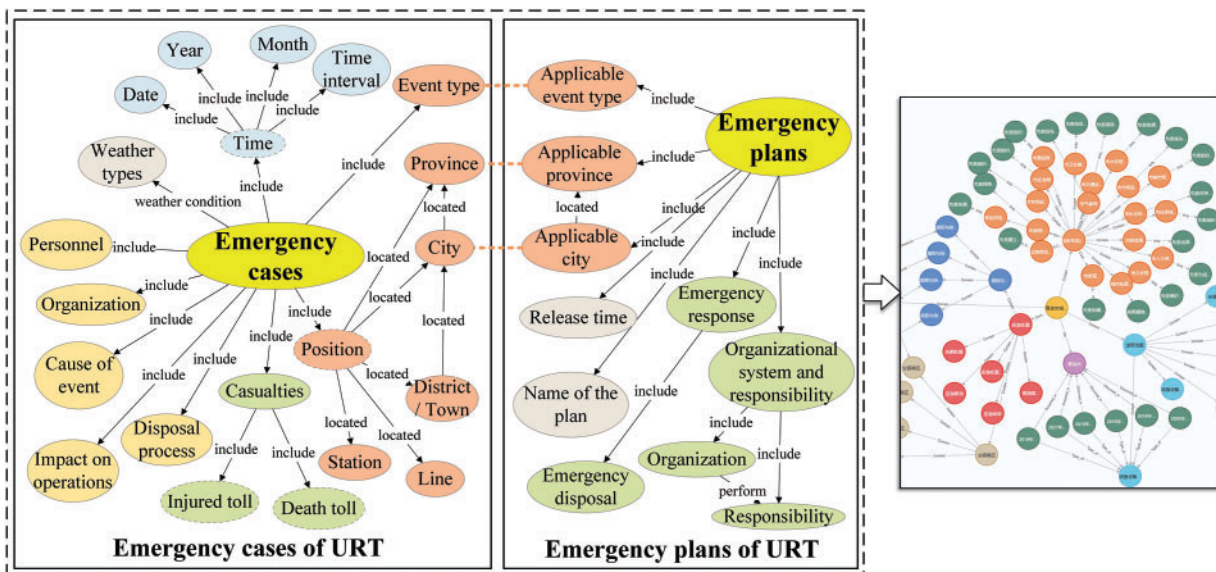
### 3 Proposed Methodology

#### 3.1 Data Preparation

##### 3.1.1 The KG for URT Emergency Domain

The URT emergency knowledge mainly exists in URT emergency cases and emergency plans, but these text data have not been organized and expressed standardized. The KG [40] can structurally store the emergency knowledge contained in these texts and provide a standardized knowledge database. This paper obtained more than 500 emergency cases and 56 emergency plans through Web Crawler and manual collection methods. On this basis, a KG for cases and plans is constructed. The construction process is as follows:

1. For the emergency cases, we analyzed the knowledge contained therein and constructed the pattern layer, as shown in the left half of the dashed box in Fig. 1. We cleaned and screened the data and annotated the dataset relying on the EasyDL platform. Most of the knowledge has the phrasal characteristics of entities, so deep learning-based entity knowledge extraction methods [35] are investigated. For regular knowledge like time, regular expression acquisition methods are used. For the weather that may not be indicated in the case, we obtained it by crawling the weather website through a web crawler.
2. Emergency plans have a relatively uniform composition, and we pay more attention to the content related to emergency response and take it as the core of the KG. The more critical content is abstracted into the pattern layer, as shown in the right half of the dotted box in Fig. 1. The knowledge in some nodes is directly adopted in a section of the plan, and the knowledge in other nodes is extracted from the paragraph by manually fine-grained.



**Figure 1:** The schema layer and local instance of the constructed KG

The acquired data is stored in the Neo4j database through Py2neo, and the same entity is shared, thus integrating the two knowledge systems. The local instance of the constructed KG is shown in the right part

of Fig. 1. The KG can quickly help generate a dataset and provide emergency personnel with consultation support based on historical experience and disposal rules.

### 3.1.2 Construction of Question Labeled Dataset

Based on the constructed database, this study designs 25 types of questions by considering the possible knowledge requirements in detail. The corresponding question template, slot number, and Cypher query template are designed for each question type and saved as an Excel file. There are 142 question templates. Each question template contains one or two slot entities, which are represented by placeholders “[ENT]” or “[ENT0]” and “[ENT1]”. The number of slots reflects the number of entities required for the question. All templates have six slot entities: city, organization, event level, event type, weather type, and event impact on operations. The question instance can be obtained by filling the slot entity in the question template, and then the dataset can be generated. The question metadata refers to the information describing the question and represents the type of each question, as shown in Table 1. The metadata structure is “knowledge source\_answer type\_key entity type”. Taking the label “0” as an example, the meaning of the question metadata is to query the responsibility of an organization in the emergency plan and use the city and organization as input conditions.

**Table 1:** The question metadata

Label	Question metadata	Label	Question metadata
0	Case_Title_Weather type	13	Case_Organization name and its responsibility_Event type
1	Case_Cause_Event type	14	Case_Emergency resource name_City and event type
2	Case_Cause and consequence_City and weather type	15	Plan_Scope of application_City
3	Case_Consequence_Event type	16	Case_Title_City and event type
4	Case_Personnel names_Event type	17	Case_Personnel names_City and event type
5	Case_Organization name_Event type	18	Case_Personnel names and their measures_Event type
6	Case_Organization name_City and event type	19	Plan_Responsibility_City and organization
7	Plan_Organization name_City	20	Plan_Command and coordination measures_City and event level
8	Plan_Hierarchical response measures_City and event level	21	Case_Title_City and weather type
9	Case_Consequence_City and event type	22	Plan_Emergency measures_City
10	Case_Emergency resource name_Event type	23	Case_Title_City
11	Case_Personnel names and their measures_City and event type	24	Case_Organization name and its responsibility_City and event type
12	Case_Cause_City and event type		

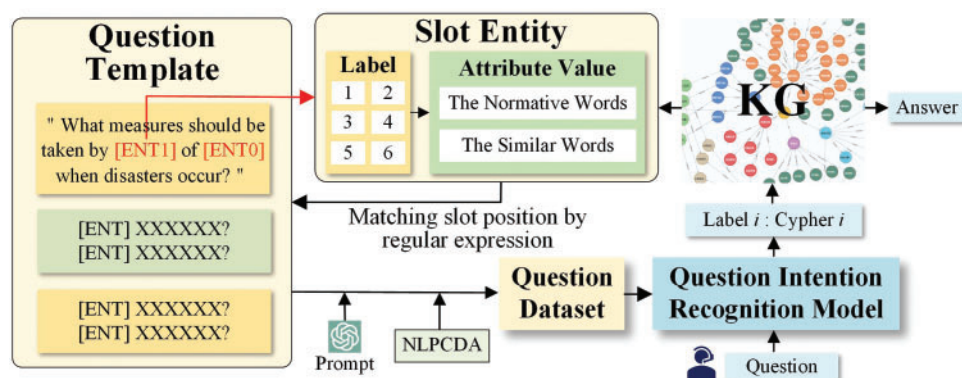


In order to fill the slot entity in the question template and get the question instance, it is necessary to obtain the attribute value and label of the slot entity. Firstly, the Graph class in Python's Py2neo library is used to establish a connection with the local Neo4j database. Then, Cypher query statements are defined for six types of slot entities. The query statement is executed using the run method, and the result is saved as an Excel file. Table 2 shows examples of attribute values for all types of entities in the Excel file. Finally, to improve the consistency and comparability of data, some attribute values with the same meaning are uniformly normalized, and a file of normative words and similar words is constructed.

**Table 2:** Examples of attribute values for slot entities

Label	Attribute value examples
City	Qingdao, Chengdu, Zhihua, Changchun
Organization	Municipal Emergency Bureau, Municipal Propaganda Department
Event level	Larger operational emergencies, major operational emergencies
Event type	Natural disasters, terrorist attacks, mass passenger flow
Weather type	Cloudy~sunny, cloudy~light rain, heavy rain~rainstorm
Event impact on operations	Interruption of operation, increased intervals between trains, temporary closure of stations

The question-labeled dataset for the intention recognition task is dynamically generated. Firstly, Python is used to read the question template file and the slot entity file (including normative words and similar words). Secondly, the label is extracted from the Cypher query statement of the slot entity through the regular expression. Then, the attribute value corresponding to the label is randomly selected. Finally, by replacing the placeholder in the problem template with the attribute value, the specific problem is generated. In order to ensure the quality of the dataset, each template is limited to be used no more than 10 times. Back-translation based on prompt engineering and text-transformation data based on NLPDA are introduced to ensure the dataset's diversity, resulting in 40,232 sentences participating in the training of intent recognition. The question intention recognition model can identify questions in predefined categories to match the corresponding Cypher query template for knowledge queries. The process is shown in Fig. 2.

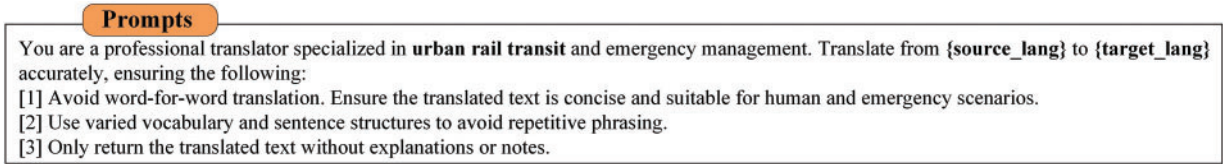


**Figure 2:** The process of dataset generation and question consultation

### 3.2 The Question Intention Recognition Model

#### 3.2.1 Data Enhancement Layer

Generating the structure of interrogative sentences through templates is highly controllable, but the diversity is insufficient, so data enhancement techniques need to be introduced. Through back-translation, the limitation of templates can be effectively broken, and the diversity of interrogative sentences can be enhanced by simulating the real data distribution while maintaining the semantic information. An approach based on GPT-4o-mini prompt learning is employed to achieve back-translation. The method is employed to generate translation data quickly and in batches. The semantic consistency of the back-translated data is ensured through the well-designed prompts and manual checking. The prompts are shown in Fig. 3. Replace the originally similar templated data with translated data while ensuring data quality to reduce the risk of overfitting due to low template diversity. Meanwhile, the NLP Chinese Data Augmentation (NLPCDA) tool is used to perform text transformation, including synonym replacement, near-sound word replacement, random word deletion and near-word replacement (Which means disrupting the relative position of the two words). This method can further enhance the robustness and generalization of the model.



**Figure 3:** Data back-translation prompts

#### 3.2.2 Semantic Representation Layer

XLNet [15] introduces two core innovations based on Transformer-XL: Permutation Language Modeling (PLM) and Two-Stream Self-Attention, which synergistically combine the advantages of autoregressive (AR) and autoencoding (AE) language models. PLM simulates diverse token permutations through attention masking, enabling the model to exploit bidirectional context during prediction. PLM aims to maximize the likelihood of generating the current token or character conditioned on its preceding inputs, as formalized in Eq. (1).

$$\max_{\theta} \mathbb{E}_{z \sim \mathcal{Z}_T} \left[ \sum_{t=1}^T \log p_{\theta}(x_{z_t} | x_{z_{<t}}) \right] \quad (1)$$

where  $\mathcal{Z}_T$  denotes the set of all possible permutations of a text sequence of length  $T$ ;  $z_t$  represents the  $t$ -th element in the permutation, and  $z_{<t}$  denotes the preceding  $t - 1$  elements, within a permutation  $z \sim \mathcal{Z}_T$ .

Traditional AR language models do not include the positional information of the predicted word used in prediction. To address this limitation, the positional information of the predicted word must be incorporated into the model, as formalized in Eq. (2).

$$p_{\theta}(X_{z_t} = x | x_{z_{<t}}) = \frac{\exp(e(x)^T g_{\theta}(x_{z_{<t}}, z_t))}{\sum_{x'} \exp(e(x')^T g_{\theta}(x_{z_{<t}}, z_t))} \quad (2)$$

where  $\theta$  denotes the model parameters;  $g_{\theta}(x_{z_{<t}}, z_t)$  integrates both the preceding context  $x_{z_{<t}}$  and the positional information  $z_t$  of the predicted word.

The model can predict  $x_{z_t}$  by leveraging the context  $x_{z_{<t}}$  and extracting information through the attention mechanism. But there is a contradiction: while  $g_{\theta}(x_{z_{<t}}, z_t)$  must exclude semantic information of  $x_{z_t}$  to predict  $x_{z_t}$ , it simultaneously requires this information to predict the subsequent word. To solve this problem, two hidden states are introduced. The content hidden state  $h_{\theta}(x_{z_{<t}}, z_t)$  (denoted  $h_{z_t}$ ) contains both the semantic information of  $x_{z_t}$  and the contextual information. The query hidden state  $g_{\theta}(x_{z_{<t}}, z_t)$  (denoted  $g_{z_t}$ ) contains only the positional information  $z_t$  and contextual information, excluding the semantic information of the predicted context. The updating processes are shown as Eqs. (3) and (4). The computational process [15] of the Two-Stream Self-Attention mechanism is shown in Fig. 4.

$$h_{z_t}^{(m)} \leftarrow \text{Attention}(Q = h_{z_t}^{(m-1)}, KV = h_{z_{\leq t}}^{(m-1)}; \theta) \quad (3)$$

$$g_{z_t}^{(m)} \leftarrow \text{Attention}(Q = g_{z_t}^{(m-1)}, KV = h_{z_{\leq t}}^{(m-1)}; \theta) \quad (4)$$

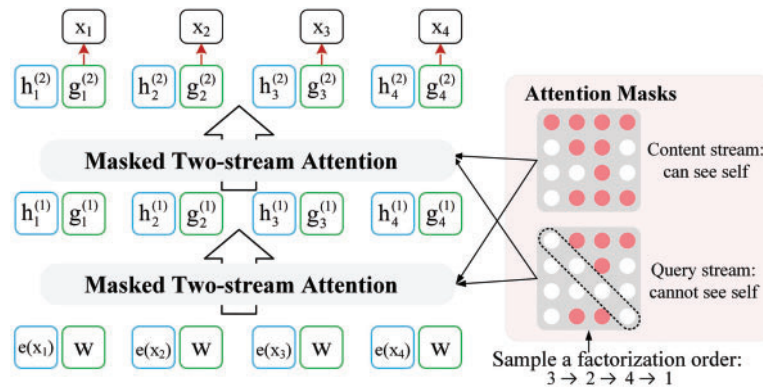


Figure 4: The structure of XLNet

### 3.2.3 Text Classification Layer

The word vectors output by the semantic representation layer are input into the TextRCNN model [41] for classification. The TextRCNN model can extract the local and global features of the text, and its structure is shown in Fig. 5.

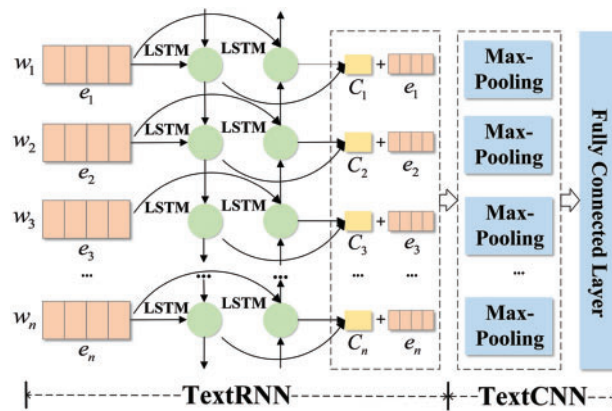


Figure 5: The structure of TextRCNN



Firstly, there is the Recurrent Structure Layer. The input text is represented as  $D = \{w_1, w_2, \dots, w_n\}$ , where  $w_i$  represents the  $i$ -th word. The embedding representation  $e(w_i)$  of the corresponding word is obtained using the XLNet model. The word vector sequence is input into the bidirectional RNN model, and the BiLSTM layer in the model captures the long-distance dependencies in the sentence. By processing the forward and reverse information, the BiLSTM learns the sentence's contextual information and obtains the context feature representation of each word. The formulas are shown in Eqs. (5) and (6).

$$c_l(w_i) = f(W^{(l)}c_l(w_{i-1}) + W^{(sl)}e(w_{i-1})) \quad (5)$$

$$c_r(w_i) = f(W^{(r)}c_r(w_{i+1}) + W^{(sr)}e(w_{i+1})) \quad (6)$$

where  $c_l(w_i)$  is the left-side context vector of the word  $w_i$ ;  $c_r(w_i)$  is the right-side context vector of the word  $w_i$ ;  $f$  indicates the activation function;  $W$  denotes the weight matrix.

By combining the word's context features and its embedding, the final vector representation for the word  $w_i$  can be obtained, as shown in Eq. (7). A linear transformation is performed on the comprehensive vector representation  $x_i$  of the word  $w_i$ . And the tanh activation function is applied to obtain the semantic vector  $y_i^{(2)}$ . The formula is shown in Eq. (8).

$$x_i = [c_l(w_i); e(w_i); c_r(w_i)] \quad (7)$$

$$y_i^{(2)} = \tanh(W^{(2)}x_i + b^{(2)}) \quad (8)$$

where  $y_i^{(2)}$  is the new semantic vector containing the semantic and context information of  $w_i$ ;  $b$  is the bias vector of the model.

Then, the transformed results are sent to the Max-Pooling Layer to extract prominent features and reduce the dimension of the feature map. This operation allows the model to handle a large amount of input data better. The operation is shown in Eq. (9).

$$y^{(3)} = \max_{i=1}^n y_i^{(2)} \quad (9)$$

where  $y^{(3)}$  is the output of the max-pooling operation, representing a vector of a fixed length;  $n$  is the number of words contained in the input text.

Finally, the output of the Max-Pooling Layer is sent to the Fully Connected Layer for linear operation and softmax function processing. Then, the final classification results are obtained. The formulas are shown in Eqs. (10) and (11).

$$y^{(4)} = W^{(4)}y^{(3)} + b^{(4)} \quad (10)$$

$$p_i = \frac{e^{y_i^{(4)}}}{\sum_{k=1}^n e^{y_k^{(4)}}} \quad (11)$$

where  $y^{(4)}$  is the result of linear operation;  $b^{(4)}$  is the offset vector;  $p_i$  denotes the probability value output by the softmax function.

## 4 Experiments and Results

### 4.1 Experiment Settings and Evaluation Index

The GPU model used in this study is GeForce RTX 4060, and the built environment is PyTorch1.13.1 and Python3.10. Divide the training set, validation set, and test set in the ratio of 6:2:2. This study employs

the Chinese XLNet model (“hfl/chinese-xlnet-base”) developed by the Harbin Institute of Technology & iFLYTEK Joint Laboratory (HFL). The Dropout strategy is applied before the fully connected layer of all models to ensure the fairness of the comparison. The stronger dropout strategy and L2 regularization strength are used to mitigate the model overfitting problem. Due to hardware resource limitations, a mini-batch training method is adopted. The gradient accumulation strategy is introduced to mitigate the oscillation phenomenon of the mini-batch training process. Detailed parameters are shown in [Table 3](#).

**Table 3:** Parameter setting of the intention recognition model

Parameter	Parameter value	Parameter	Parameter value
Number of classes	25	Regularization warmup	0.1
Number of epochs	5	Gradient accumulation steps	2
Batch size	64	Max length of sentence	32
Number of iterations	1860	Learning rate	$1.1 \times 10^{-5}$
Regularization method	L2	Dropout rate	0.5
Regularization strength	0.5	Number of RNN layers	2

## 4.2 Results and Discussions

### 4.2.1 Performance Comparison with Baseline Models

This study selected multiple classes of baseline models for comparative analysis, including two major classes: non-pretrained and pre-trained models. For the non-pretrained models, we included several variants of RNN-based models. As the baseline pre-trained model, we chose the BERT framework along with its variants. The evaluation results are shown in [Table 4](#).

**Table 4:** Comparative results between the models

Model	Accuracy	Average Precision	Average Recall	Average F1-Score
BiLSTM	0.9448	0.9347	0.9293	0.9315
BiGRU	0.9531	0.9444	0.9431	0.9436
BiLSTM_Att	0.9530	0.9440	0.9461	0.9449
BiLSTM_CNN	0.9607	0.9534	0.9539	0.9535
ALBERT_BiLSTM_CNN	0.9530	0.9430	0.9451	0.9439
DistilBERT_BiLSTM_CNN	0.9663	0.9586	0.9610	0.9596
BERT_BiLSTM_CNN	0.9676	0.9633	0.9638	0.9633
XLNet	0.9671	0.9607	0.9616	0.9610
XLNet_BiLSTM_CNN	0.9698	0.9645	0.9681	0.9662

**BERT** [14]. A deep bidirectional language model constructed by stacking multiple layers of Transformer encoders, utilizing only the encoder component of the Transformer architecture.

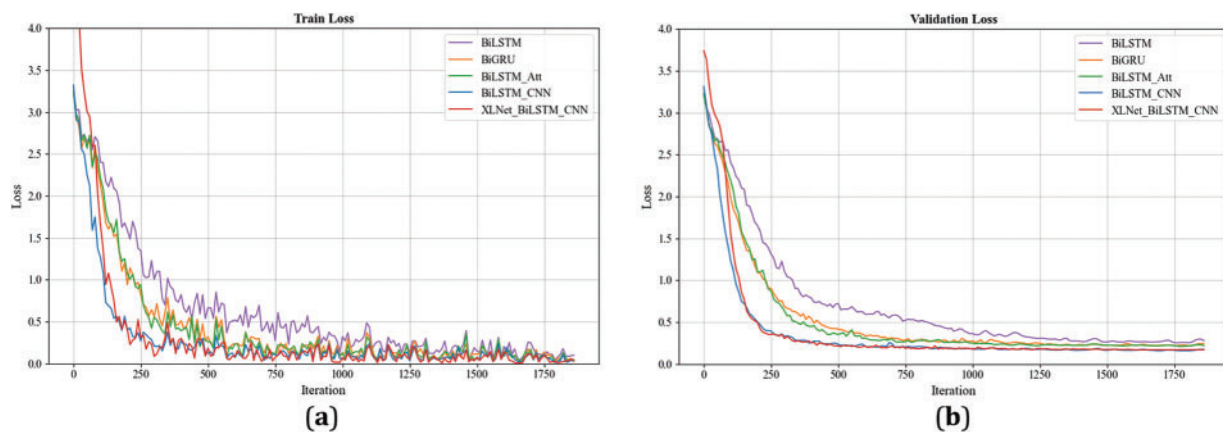
**ALBERT** [42]. Reducing parameter size through Factorized Embedding Parameterization and cross-layer parameter sharing whilst introducing self-supervised loss for Sentence Order Prediction focuses on modeling inter-sentence coherence.

**DistilBERT** [43]. Knowledge Distillation techniques were used in the pre-training phase, while the BERT model architecture was streamlined.

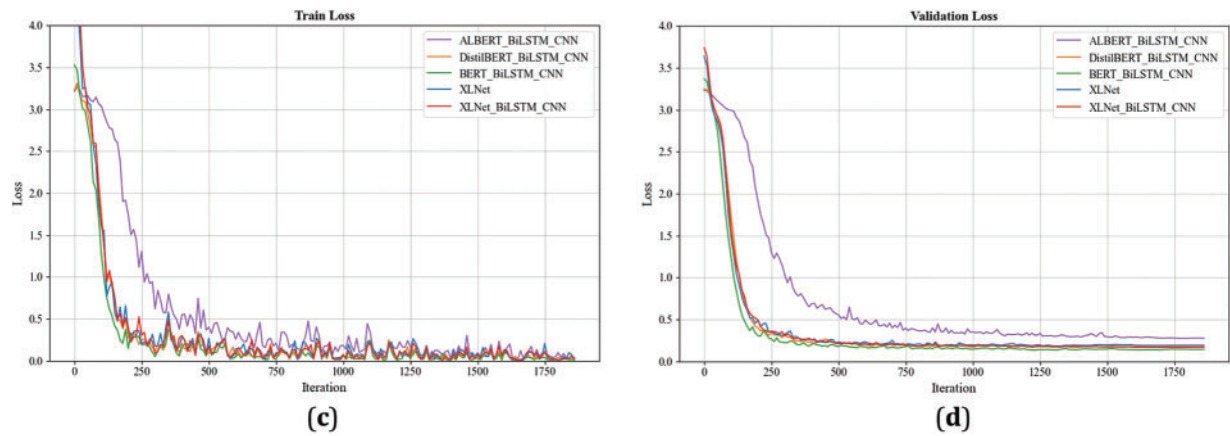
The BiLSTM\_CNN has significant advantages among the non-pretrained models. Its structure better captures the long-distance dependency of sentences internally by using the memory of BiLSTM, and effectively extracts local features through CNN. Introducing the pre-training phase generally improves the model's performance. The reason for the lack of performance improvement of ALBERT\_BiLSTM\_CNN may be that ALBERT focuses on inter-sentence logical relations rather than single-sentence internal structure. XLNet\_BiLSTM\_CNN has the best overall performance regarding overall prediction ability, accuracy in identifying positive samples, coverage of positive samples and balance performance. The reason is that the XLNet architecture is superior to the BERT architecture and its variants in bidirectional context modeling capabilities and fine-grained semantic capture. With the introduction of BiLSTM\_CNN, the proposed model realizes the collaborative optimization of sequence dependent modeling and local feature extraction.

To ensure the clarity of the image, the models are divided into two groups to plot the loss curves of the training set and the validation set, as shown in Fig. 6. XLNet\_BiLSTM\_CNN achieved the fastest convergence rate when the baseline model was non-pretrained model. The introduction of pre-training architecture generally speeds up the convergence of the model. The loss function curve of XLNet\_BiLSTM\_CNN performs well on both training and verification sets. The proposed model can quickly adapt to the training data and show good learning and generalization ability.

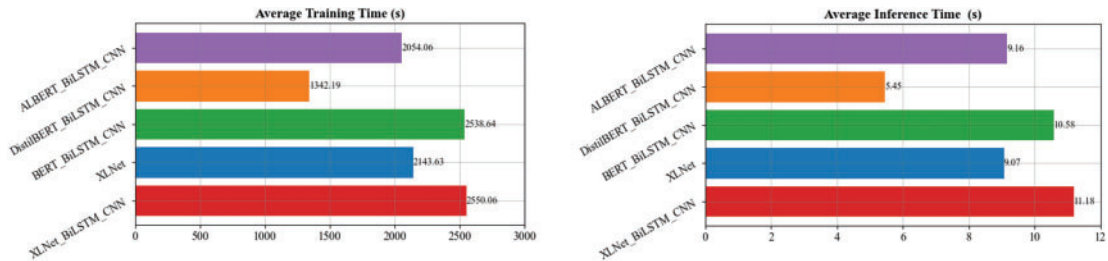
Based on the above analysis, the hybrid architecture has higher accuracy and generalization ability, so the performance of the hybrid architecture model is further compared. Fig. 7 shows the model's average training time and inference time, and Fig. 8 represents the complexity of the model. Although XLNet\_BiLSTM\_CNN has the largest training time, inference speed, and model parameters, it does not exceed other models by much. As can be seen, higher model complexity generally brings higher time costs and advantages for accuracy.



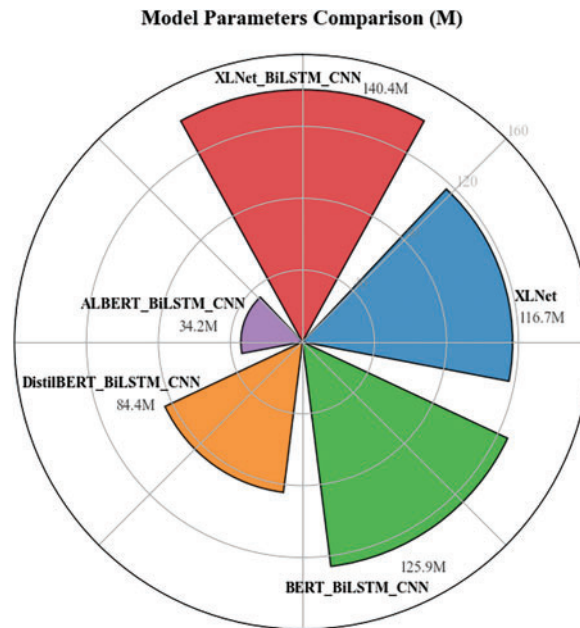
**Figure 6:** (Continued)



**Figure 6:** Loss function. (a) Training loss function with non-pretrained baseline model; (b) Validation loss function with non-pretrained baseline model; (c) Training loss function with pre-trained baseline model; (d) Validation loss function with pre-trained baseline model



**Figure 7:** Average training time and average inference time



**Figure 8:** Model parameters

#### 4.2.2 Performance of Single Model

The performance of the proposed model on the test set is evaluated. The results are shown in Table 5.

**Table 5:** Evaluation results of the proposed model on the test set

Question label	Precision	Recall	F1-Score	Support	Question label	Precision	Recall	F1-Score	Support
0	0.9690	0.9690	0.9690	129	13	0.8956	0.9532	0.9235	171
1	0.9783	0.9890	0.9836	182	14	0.9969	0.9877	0.9922	648
2	0.9789	0.9957	0.9872	466	15	0.9505	0.9796	0.9648	98
3	0.9778	0.9724	0.9751	181	16	0.9623	0.9708	0.9665	342
4	0.9508	0.9355	0.9431	186	17	0.9528	0.9498	0.9513	638
5	0.9314	0.9532	0.9422	171	18	0.8854	0.8797	0.8825	158
6	0.9568	0.9519	0.9543	582	19	0.9919	0.9973	0.9946	370
7	1.0000	0.9835	0.9917	121	20	0.9931	0.9897	0.9914	290
8	0.9873	0.9936	0.9904	313	21	0.9784	0.9706	0.9745	374
9	0.9829	0.9875	0.9852	641	22	0.9402	0.9649	0.9524	114
10	0.9760	0.9879	0.9819	165	23	0.9608	0.9608	0.9608	102
11	0.9619	0.9456	0.9537	588	24	0.9651	0.9540	0.9595	609
12	0.9877	0.9808	0.9842	572					

Most indicators are above 0.9. For most smaller categories (such as those with 0, 7, 15, and 23), their Precision, Recall, and F1-scores exceed 0.95. This indicates that the model performs well on rare query types. For the label “18”, its performance in three aspects is not more than 0.9. This may be attributed to the high semantic ambiguity and a limited number of questions, which means the model has not fully learned its semantic information. To assess the model’s robustness, four text transformation methods were employed to introduce varying levels of noise to the test set. As can be seen from Table 6, performance degradation correlates positively with noise intensity, and the sensitivity of different transform types is significantly different. Near-sound word replacement has the substantial impact on the model, with effects at an intensity of 0.3 being significantly greater than those of other types. Near-word replacement has the least influence on the model, and the model accuracy decreases almost linearly. Synonym replacement and random word deletion are relatively stable. It can be seen that the robustness of the model to structural changes is better than that of semantic substitutions. The variation in accuracy of most experiments is less than 1%, indicating that the model has good robustness and generalization ability.

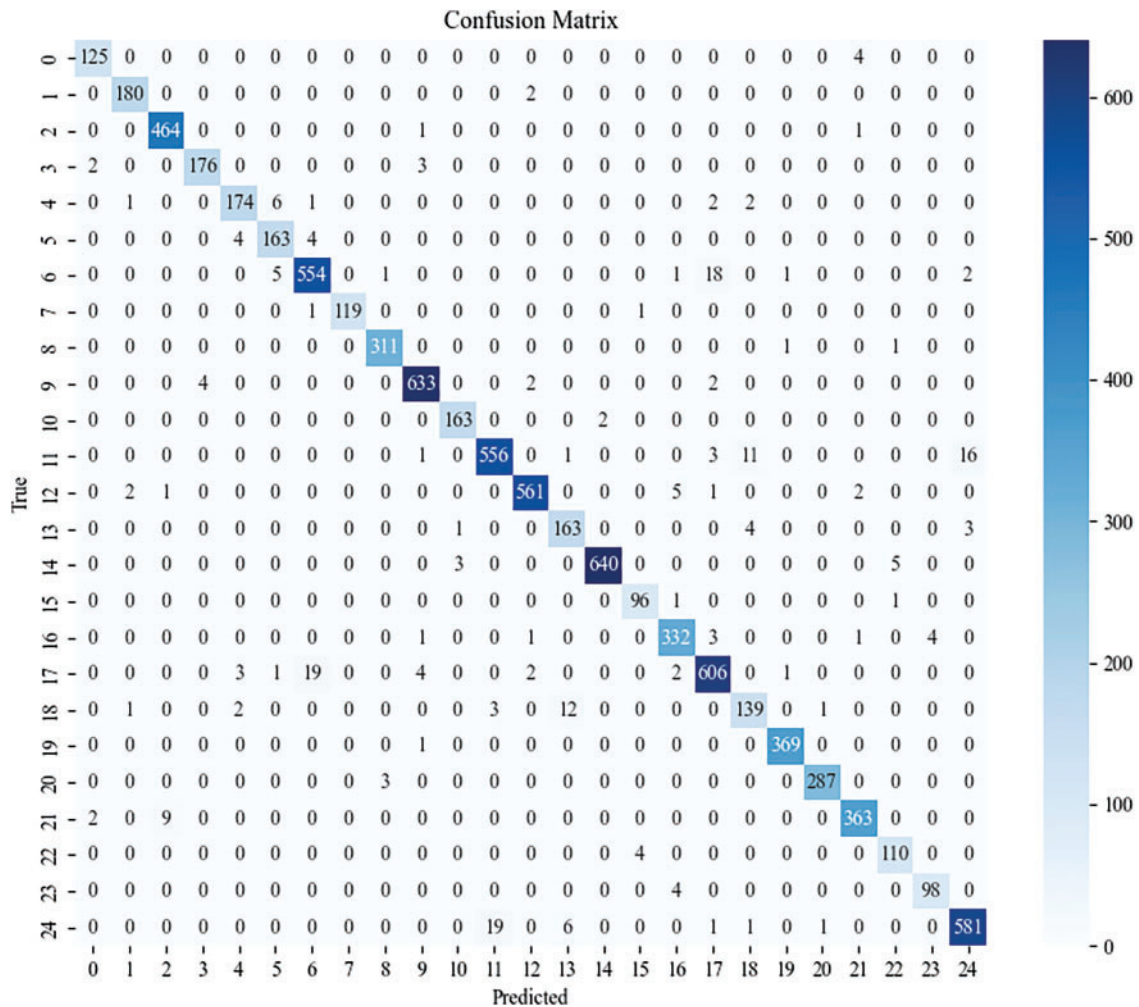
**Table 6:** The performance of the model under different noise intensities

Text transformation	Noise intensity					
	0.05	0.1	0.15	0.2	0.25	0.3
Synonym Replacement	0.9671 (−0.27%)	0.9659 (−0.39%)	0.9633 (−0.65%)	0.9625 (−0.73%)	0.9644 (−0.54%)	0.9586 (−1.12%)
Near-Sound Word Replacement	0.9674 (−0.24%)	0.9664 (−0.34%)	0.9627 (−0.71%)	0.9592 (−1.06%)	0.9609 (−0.89%)	0.9551 (−1.47%)
Random Word Deletion	0.9677 (−0.21%)	0.9682 (−0.16%)	0.9660 (−0.38%)	0.9635 (−0.63%)	0.9613 (−0.85%)	0.9594 (−1.04%)
Near-Word Replacement	0.9689 (−0.09%)	0.9675 (−0.23%)	0.9658 (−0.40%)	0.9633 (−0.65%)	0.9614 (−0.84%)	0.9602 (−0.96%)

The Fig. 9 presents the confusion matrix, which matrix reveals some confusing categories. The number of times that “Case\_Organization name and its responsibility\_City and event type” is incorrectly classified



as “Case\_Personnel names and their measures\_City and event type” and “Case\_Personnel names\_City and event type” is incorrectly classified as “Case\_Organization name\_City and event type” type is 19. Moreover, there are some similar mistakes. The reason may be that the concepts of “organization name” and “personnel” are generally asked in a similar way. Overall, misclassification errors for question types with more complex answers and more entities tend to be worse than for simple questions. This shows that it is necessary to improve the data quality and the model’s complex semantic understanding ability.



**Figure 9:** Confusion matrix

## 5 Conclusions

The data in the URT emergency domain mostly exists in scattered text, which makes accessing emergency knowledge challenging. Furthermore, there is a lack of publicly available question-labeled datasets in this area. In order to solve these problems, this study integrates KG and NLP technologies to develop methods for dataset construction and question intention recognition. Firstly, the KG is utilized to create a structured representation of emergency data and to establish deep correlations within emergency knowledge. Moreover, the question-labeled dataset is generated efficiently and with minimal manual annotation by filling entities in the KG into the predefined question template. Finally, the data-enhanced question intention recognition model is proposed. By leveraging the complex semantic understanding capabilities of

XLNet\_BiLSTM\_CNN, along with the diverse training samples enabled by data enhancement techniques, we improve the accuracy and robustness of the model. Compared with the baseline models, the proposed model improves the efficiency of question classification and shows superiority. However, there are still limitations to the study:

First of all, the knowledge sources in the KG include only emergency cases and emergency plans. It would be beneficial to introduce additional text data (such as maintenance logs and expert experience) and multimodal data (such as images and videos) to enrich the sources of intention. In addition, the questions generated by the template lack diversity. It is essential to enrich the original dataset and explore methods for rapidly generating the diversified high-quality questions.

Secondly, in the emergency response scenario, user-generated questions are often not standardized, requiring the model to possess a high degree of generalization to accurately identify user intentions. The proposed model does not have a high inference speed, which is critical in emergency situations. Future studies should focus on enhancing the model's generalization performance and response speed.

Finally, knowledge retrieval and Q&A processes can be realized by combining KG and the proposed model. The intention recognition model in this study is part of the pipeline Q&A system, and utilizing multi-task learning can optimize related tasks cooperatively. By considering intention recognition as one of the tasks in a multi-task learning framework, the Q&A process can become more efficient and robust, ultimately providing users with convenient emergency knowledge acquisition services in URT.

**Acknowledgement:** The authors would like to thank all the researchers and reviewers who helped to improve the quality of the manuscript.

**Funding Statement:** This work was supported in part by the National Natural Science Foundation of China. The funding numbers 62433005, 62272036, 62132003, and 62173167.

**Author Contributions:** The authors confirm contribution to the paper as follows: study conception and design: Yinuo Chen, Xu Wu, Jiaxin Fan; data collection: Jiaxin Fan, Guangyu Zhu; analysis and interpretation of results: Yinuo Chen, Xu Wu; draft manuscript preparation: Yinuo Chen, Guangyu Zhu. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The data that support the findings of this study are available from the corresponding author, Guangyu Zhu, upon reasonable request.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest to report regarding the present study.

## References

1. Hu M, Peng J, Zhang W, Hu J, Qi L, Zhang H. An intent recognition model supporting the spoken expression mixed with Chinese and English. *J Intell Fuzzy Syst.* 2021;40(5):10261–72. doi:10.3233/JIFS-202365.
2. Liang S, Sun F, Sun H, Chen T, Du W. A medical text classification approach with ZEN and capsule network. *J Supercomput.* 2024;80(3):4353–77. doi:10.1007/s11227-023-05612-6.
3. Occhipinti A, Rogers L, Angione C. A pipeline and comparative study of 12 machine learning models for text classification. *Expert Syst Appl.* 2022;201(7):117193. doi:10.1016/j.eswa.2022.117193.
4. Cui H, Wang G, Li Y, Welsch RE. Self-training method based on GCN for semi-supervised short text classification. *Inf Sci.* 2022;611:18–29. doi:10.1016/j.ins.2022.07.186.
5. Taha K, Yoo PD, Yeun C, Homouz D, Taha A. A comprehensive survey of text classification techniques and their research applications: observational and experimental insights. *Comput Sci Rev.* 2024;54(7):100664. doi:10.1016/j.cosrev.2024.100664.

6. Zhang D. A novel text classification model combining time correlation principle and rough set theory IEEE Access. 2023; 11:135797–810. doi:10.1109/ACCESS.2023.3332909.
7. Chen X, Yu D, Fan X, Wang L, Chen J. Multiclass classification for self-admitted technical debt based on XGBoost. IEEE Trans Rel. 2022;71(3):1309–24. doi:10.1109/TR.2021.3087864.
8. Lu Q, Wang Y. Latent semantic text classification method research based on support vector machine. Int J Inf Commun Technol. 2019;15(3):243. doi:10.1504/IJICT.2019.102998.
9. Chen J. A multi granularity information fusion text classification model based on attention mechanism. J Intell Fuzzy Syst. 2023;45(5):7631–45. doi:10.3233/JIFS-233388.
10. Gong J, Zhang J, Guo W, Ma Z, Lv X. Short text classification based on explicit and implicit multiscale weighted semantic information. Symmetry. 2023;15(11):2008. doi:10.3390/sym15112008.
11. Abas AR, Elhenawy I, Zidan M, Othman M. BERT-CNN: a deep learning model for detecting emotions from text. Comput Mater Contin. 2022;71(2):2943–61. doi:10.32604/cmc.2022.021671.
12. Kim Y. Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014 Oct 25–29; Doha, Qatar. p. 1746–51. doi:10.3115/v1/D14-1181.
13. Kim J, Jang S, Park E, Choi S. Text classification using capsules. Neurocomputing. 2020;376(8):214–21. doi:10.1016/j.neucom.2019.10.033.
14. Devlin J, Chang MW, Lee K, Toutanova K, Hulburd E, Liu D, et al. BERT: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805. 2018.
15. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R, Le QV. XLNet: generalized autoregressive pretraining for language understanding. arXiv:1906.08237. 2019.
16. Zhao X, Zhou K, Li JY, Tang TY, Wang XL, Hou YP. A survey of large language models. arXiv:2303.18223. 2023.
17. Zhu G, Yang R, Wu EQ, Law R. Extraction of emergency elements and business process model of urban rail transit plans. IEEE Trans Comput Soc Syst. 2024;11(2):1744–52. doi:10.1109/TCSS.2023.3235338.
18. Zhu G, Huang X, Yang R, Sun R. Relationship extraction method for urban rail transit operation emergencies records. IEEE Trans Intell Veh. 2023;8(1):520–30. doi:10.1109/TIV.2022.3160502.
19. Chen N, Su X, Liu T, Hao Q, Wei M. A benchmark dataset and case study for Chinese medical question intent classification. BMC Med Inform Decis Mak. 2020;20(Suppl 3):125. doi:10.1186/s12911-020-1122-3.
20. Trehwela A, Figueroa A. Text-based neural networks for question intent recognition. Eng Appl Artif Intell. 2023;121(8):105933. doi:10.1016/j.engappai.2023.105933.
21. Liu C, Xu X. AMFF: a new attention-based multi-feature fusion method for intention recognition. Knowl Based Syst. 2021;233:107525. doi:10.1016/j.knosys.2021.107525.
22. Dai Z, Yang Z, Yang Y, Carbonell J, Le QV, Salakhutdinov R. Transformer-XL: attentive language models beyond a fixed-length context. arXiv:1901.02860. 2019.
23. Yan R, Jiang X, Dang D. Named entity recognition by using XLNet-BiLSTM-CRF. Neural Process Lett. 2021;53(5):3339–56. doi:10.1007/s11063-021-10547-1.
24. Zhang T, Wang D. Classification of crop disease-pest questions based on BERT-BiGRU-CapsNet with attention pooling. Front Plant Sci. 2023;14:1300580. doi:10.3389/fpls.2023.1300580.
25. Wu T, Wang M, Xi Y, Zhao Z. Intent recognition model based on sequential information and sentence features. Neurocomputing. 2024;566(2):127054. doi:10.1016/j.neucom.2023.127054.
26. Areshey A, Mathkour H. Exploring transformer models for sentiment classification: a comparison of BERT, RoBERTa, ALBERT, DistilBERT, and XLNet. Expert Syst. 2024;41(11):e13701. doi:10.1111/exsy.13701.
27. Han T, Zhang Z, Ren M, Dong C, Jiang X, Zhuang Q. Text emotion recognition based on XLNet-BiGRU-att. Electronics. 2023;12(12):2704. doi:10.3390/electronics12122704.
28. Weng Y, Gu SY, Li J, Wang F, Li JL, Li X. Paragraph context-based text classification approach for large-scale judgment text structuring. J Tianjin Univ. 2021;54(4):418–25 (In Chinese). doi:10.11784/tdxbz202002021.
29. Zhang Q, Hou H, Ju Y, Yuan J, Zhang K, Wang H, et al. Category mapping of emergency supplies classification standard based on BERT-TextCNN. Systems. 2024;12(9):358. doi:10.3390/systems12090358.

30. Guo S, Wang Q. Application of knowledge distillation based on transfer learning of ERNIE model in intelligent dialogue intention recognition. *Sensors*. 2022;22(3):1270. doi:10.3390/s22031270.
31. Chen YN, Chen ZT, Xie YG, Shang HB, Wang ZY, Zhang X. Construction of a question-answering system for enterostomy-related complications based on knowledge map. *Military Nurs*. 2024;41(11):9–12 (In Chinese). doi:10.3969/j.issn.2097-1826.2024.11.003.
32. Lu J, Wen G, Lu R, Wang Y, Zhang S. Networked knowledge and complex networks: an engineering view. *IEEE/CAA J Autom Sinica*. 2022;9(8):1366–83. doi:10.1109/JAS.2022.105737.
33. Li X, Zhao T, Wen J, Cai X. Many-objective emergency aided decision making based on knowledge graph. *Appl Intell*. 2024;54(17):7733–49. doi:10.1007/s10489-024-05557-0.
34. Liu X, Lu H, Li H. Intelligent generation method of emergency plan for hydraulic engineering based on knowledge graph—take the South-to-North Water Diversion Project as an example. *LHB Hydrosoci J*. 2022;108(1):2153629. doi:10.1080/27678490.2022.2153629.
35. Zhu G, Fan J, Huang X, Zhang N, Wu B, Wei Y. Knowledge acquisition method of urban rail transit safety event case base for intelligent emergency response. *IEEE Trans Autom Sci Eng*. 2023;2023:1–9. doi:10.1109/TASE.2023.3337135.
36. Zhao YM, Pan P, Mao J. Recognizing intensity of medical query intentions based on task knowledge fusion and text data enhancement. *Data Anal Knowl Discov*. 2023;7(2):38–47 (In Chinese). doi:10.11925/infotech.2096-3467.2022.0919.
37. Cao P, Li Y, Li X. Cross-language few-shot intent recognition via prompt-based tuning. *Appl Intell*. 2024;55(1):60. doi:10.1007/s10489-024-06089-3.
38. Luo SC, He J. Few shot multi-intent recognition with intent information. *J Chin Inf Process*. 2023;37(7):61–70 (In Chinese). doi:10.3969/j.issn.1003-0077.2023.07.008.
39. Yan X, Jian F, Sun B. SAKG-BERT: enabling language representation with knowledge graphs for Chinese sentiment analysis. *IEEE Access*. 2021;9:101695–701. doi:10.1109/ACCESS.2021.3098180.
40. Lin J, Zhao Y, Huang W, Liu C, Pu H. Domain knowledge graph-based research progress of knowledge representation. *Neural Comput Appl*. 2021;33(2):681–90. doi:10.1007/s00521-020-05057-5.
41. Lai S, Xu L, Liu K, Zhao J. Recurrent convolutional neural networks for text classification. In: *Proceeding of the 29th Association-for-the-Advancement-of-Artificial-Intelligence (AAAI) Conference on Artificial Intelligence*; 2015 Jan 25–30; Austin, TX, USA. Palo Alto, CA, USA: AAAI Press; 2015. p. 2267–73. doi:10.1609/aaai.v29i1.9513.
42. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R, et al. ALBERT: a lite BERT for self-supervised learning of language representations. *arXiv:1909.11942*. 2019.
43. Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108*. 2019.