Neurocomputing 566 (2024) 127054

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Intent recognition model based on sequential information and sentence features

Tiefeng Wu, Miao Wang *, Yunfang Xi, Zhichao Zhao

School of Information and Control Engineering, Qingdao University of Technology, Qingdao, 266520, Shandong, China

ARTICLE INFO

Communicated by G. Ciocca

Keywords:
Intent recognition
Sequential information
Sentence features
CNN
BILSTM
Multi-head attention mechanism
BERT

ABSTRACT

In human–computer dialogue systems, intent recognition is crucial for determining the intentions or purposes of users during interactions with the system, enabling the system to provide appropriate responses or actions. This paper proposes an intent recognition model that integrates sequential information and sentence structural features. Specifically, the approach utilizes a CNN to capture local salient features in the text, followed by a BILSTM to extract sequential information within the local context. The sequential information is then fed into a multi-head attention mechanism to focus on more relevant sequential details. Additionally, the original data is processed by BERT to extract sentence structural features. Finally, the sequential information features and sentence structural features are concatenated and fused to achieve enhanced intent recognition performance. This approach effectively leverages contextual and semantic information within the text, leading to improved accuracy in intent recognition. Experimental results demonstrate the effectiveness of the proposed method in intent recognition and its high relevance for practical applications.

1. Introduction

Intent recognition is a significant research area in the field of natural language processing, which aims to identify the intentions of users from natural language texts. Intent recognition finds extensive applications in human–computer interaction domains such as intelligent customer service, smart voice assistants, and chatbots. Typically, intent recognition techniques analyze user input texts to identify their underlying intentions and provide corresponding services or responses [1]. Commonly used methods for intent recognition include dictionary-based and rule-based classification methods, statistical-based classification methods [2–4], and deep learning-based classification methods.

In the dictionary-based and rule-based classification methods, Ramanand et al. [5] proposed a method based on rules and graphs to obtain intent templates for consumer intent recognition. Liu et al. [6] classified intents by matching core tuples. While these methods do not require training on large amounts of data, they are time-consuming, labor-intensive, and have limited scalability. They struggle to recognize intents that deviate slightly from the predefined templates, making them unsuitable for supporting truly open-ended conversations [7]. Later researchers, in order to solve the problem of weak scalability in rule-based classification, explored a large number of statistical learning methods, such as Naive Bayes Model (NBM), Support Vector Machine (SVM), Hidden Markov Model (HMM), Conditional Random Field (CRF), and so on [8–10]. Traditionally, intent recognition is defined as a classification problem, where the dialogue information at the current input serves as the primary classification data [11]. Therefore, almost all text classification methods can be applied in this domain by representing the input text as a vector and using a classifier for classification. For instance, Fuchun Peng et al. [12] applied the Naive Bayes (NB) method for intent recognition. B. Shen et al. [13] trained multi-class support vector machines (SVMs) with radial basis function (RBF) kernels and selected appropriate parameters for each SVM to classify different motion patterns. Li K et al. [14] proposed a novel algorithm combining Hidden Markov Model (HMM) and Bayesian filtering (BF) techniques to identify drivers' lane change intentions.

However, statistical models have a high dependency on a large amount of labeled data. They require sufficient training data to capture statistical relationships and patterns between different intents. If the training data is insufficient or not representative, the model may fail to accurately recognize intents or generalize to new data. Moreover, statistical models often rely on manual feature selection and design, converting textual data into suitable feature representations for modeling. This process of feature engineering consumes significant time and human resources, and it needs to be redone for intent data in different domains. Additionally, intent recognition tasks involve complex semantic relationships and contextual dependencies. Both rule-based and statistical models may struggle to capture long-term dependencies and

* Corresponding author.
E-mail address: wangmiao_qut@163.com (M. Wang).

https://doi.org/10.1016/j.neucom.2023.127054
Received 27 June 2023; Received in revised form 27 October 2023; Accepted 14 November 2023
Available online 17 November 2023

semantic relationships, particularly in multi-turn conversations. Models may overly rely on local features while neglecting global contextual information, leading to a decrease in performance. With the advancements in deep learning and pre-trained models, some deep learning-based methods have made progress in addressing the aforementioned challenges. Deep learning integrates classification and feature design into the learning process. Compared to traditional machine learning approaches, deep learning models can learn pattern structures within text sequences. As a result, many deep learning models have been applied in intent recognition research [15–18].

Following the successful application of Convolutional Neural Network(CNN) in text classification tasks [19], Hashemi et al. [20] employed CNN to extract query vector representations as query classification features for intent recognition in user search queries. Trewhela et al. [21] conducted a comprehensive comparative evaluation of various architecture variations of traditional neural networks for detecting problem intents. Considering that CNN can only capture local semantic features of intent texts and may not preserve semantic coherence, Bhargava et al. [22] utilized Recurrent Neural Network(RNN) to extract contextual information from dialogues, thereby improving the accuracy of intent recognition. RNN models consider the influence of preceding context on the subsequent context but may face challenges in learning text features effectively, including issues such as gradient explosion and vanishing gradients. Researchers began using Long Short-Term Memory (LSTM) models to achieve intent classification [23]. Kim J K et al. [24] enriched the neural network with external embeddings and built a bidirectional LSTM for intent detection on top of these embeddings. Experimental results indicated that incorporating semantic word embeddings from a lexical dictionary can enhance intent detection performance. Trewhela A et al. [25] conducted a comprehensive comparison of the application of various architectures in the aspect of intent detection in queries.

Several research studies have explored the combination of different types of neural networks to improve performance. For example, Yu et al. [26] cascaded attention mechanisms and RNN to achieve better results than using RNN or attention alone. Liu et al. [27] employed Bidirectional Encoder Representations from Transformers(BERT) as a pre-trained model and used Bidirectional Long Short-Term Memory(BiLSTM) to extract bidirectional features from text, constructing a task-oriented human–machine dialogue intent classification model. Huang et al. [28] proposed an effective multimodal representation and fusion method for intent recognition in multimodal scenarios. Liu et al. [29] introduced an attention-based multi-feature fusion approach that combines features extracted from Term Frequency-Inverse Document Frequency (TF-IDF), CNN, and LSTM to enrich short text features for intent recognition. Onan A [30] proposed a hierarchical graph-based text classification framework that leverages the power of contextual node embeddings and BERT-based dynamic fusion to capture complex relationships between nodes in the hierarchical graph. These studies demonstrate that cascading different types of neural networks has a strong competitive advantage in improving model performance.

To further enhance the information representation capability of features and improve intent classification performance, this paper proposes improvements based on CNN and BiLSTM. It introduces a multi-head attention mechanism to strengthen the focus on important feature information. Additionally, BERT model is incorporated to establish a global vector representation of text features, allowing for a comprehensive exploration of intent expression in the text. These enhancements aim to increase the accuracy of intent recognition.

2. Model construction

Intent recognition is typically performed by analyzing natural language texts inputted by users to identify the intended meaning or purpose behind their messages. Based on this intent, appropriate responses or actions are provided. Natural language texts inputted by users are context-dependent, meaning they are usually generated within a specific context or conversation. Therefore, their meanings and intents can be influenced by preceding or subsequent dialogue. Consequently, when performing intent recognition, it is necessary to consider contextual information and make the most of this information whenever possible.

The CBLMA-B (CNN-BiLSTM-Multi-head Attention and BERT) model described in this text first converts the textual data into vector representations. Next, the model utilizes a CNN to capture local key information, and a BiLSTM to learn sequential information from the key features. These learned features are then inputted into a multi-head attention mechanism for more precise learning and extraction of important information. Subsequently, the model feeds the text vectors into the BERT model to learn sentence-level features and obtain the CLS (classification) vector. Finally, the outputs from BiLSTM, multi-head attention, and BERT are concatenated to form the final feature vector, which is then subjected to intent classification using the softmax function. The model structure is illustrated in Fig. 1.

2.1. CNN

The core idea of CNN is to effectively process image data by reducing the number of parameters and computations through convolution and pooling operations. It consists of multiple convolutional layers, pooling layers, and fully connected layers. Fig. 2 illustrates the basic structure of a CNN.

The article defines three one-dimensional convolutional layers, each with different output channels and kernel sizes. Smaller kernels are able to capture local detailed information, while larger kernels can capture semantic information over a wider range. By setting up the model in this way, it captures features of the input data within different receptive fields to extract rich feature representations. Specifically, different kernel sizes (2, 3 and 4) are used to capture features of different lengths in the text, and these features are extracted through convolutional operations. For each kernel size, the model is configured with different numbers of output channels (50, 100 and 150) to increase its ability to perceive different features. The mathematical calculation formula is as follows.

Let us assume the input data is denoted as X_t. The computation of the ith convolutional layer can be represented as follows:

$$x^i_{conv} = Conv1d(X_t, W_i) + b_i \tag{1}$$

$$x^i_{relu} = ReLU(x^i_{conv}) \tag{2}$$

$$x^i_{pool} = AdaptiveMaxPool1d(x^i_{relu}) \tag{3}$$

Where *Conv1d* represents the one-dimensional convolution operation, W_i denotes the convolutional kernel weights for the ith convolutional layer, and b_i denotes the bias term for the ith convolutional layer. *ReLU* represents the activation function, and *AdaptiveMaxPool1d* represents the adaptive max pooling operation.

The outputs of all convolutional layers are concatenated to obtain the output of the entire model:

$$x_{cnn} = Concatenate([x^1_{pool}, x^2_{pool}, x^3_{pool}], axis = 1) \tag{4}$$

Where *Concatenate* represents the concatenation operation, and *axis*=1 indicates that the concatenation is performed along the channel dimension.

2.2. BiLSTM

BiLSTM is an extension of the standard LSTM that incorporates both forward and backward LSTM structures to capture more comprehensive

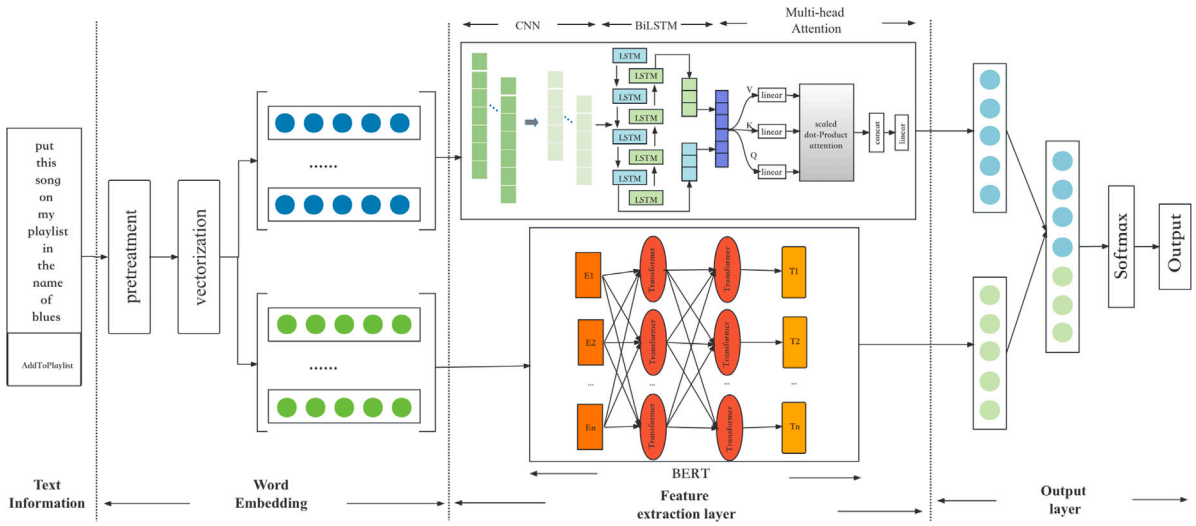


Fig. 1. Structure of the CBLMA-B model.

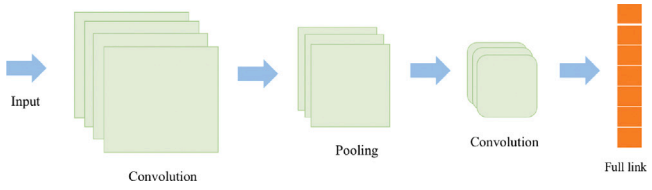


Fig. 2. Basic structure of a CNN.

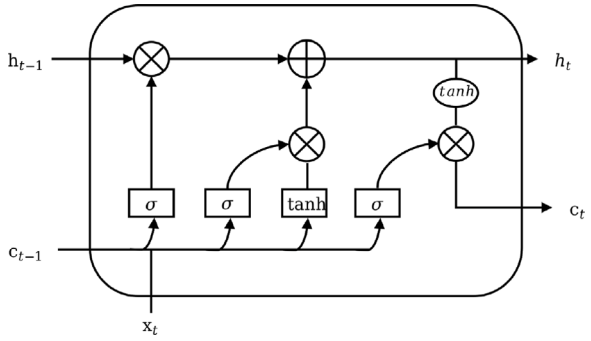


Fig. 3. LSTM architecture.

contextual information [31]. Specifically, BiLSTM operates on the input sequence using two independent LSTM networks: one in the forward direction and another in the backward direction. The forward LSTM processes the input sequence from the first element, while the backward LSTM processes it from the last element. The outputs of both LSTMs are then concatenated at each time step. This allows the output at each time step to contain both the preceding and succeeding context of the input sequence, enabling better capture of long-term dependencies in the sequence. Therefore, this paper adopts the bidirectional long short-term neural network, and Fig. 3 depicts its network structure.

In Section 2.1, the output of CNN, denoted as x_{cnn}, is passed through the BiLSTM layer, resulting in an output h_t.

$$\overrightarrow{h_t} = LSTM(x_t, \overrightarrow{h_{t-1}}) \tag{5}$$

$$\overleftarrow{h_t} = LSTM(x_t, \overleftarrow{h_{t-1}}) \tag{6}$$

$$h_t = \overrightarrow{W_t} \cdot \overrightarrow{h_t} + \overleftarrow{W_t} \cdot \overleftarrow{h_t} + b_t \tag{7}$$

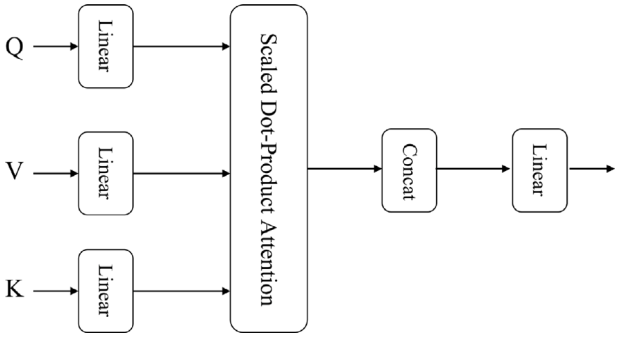


Fig. 4. Multi-head attention mechanism.

Let us denote the output of the BiLSTM as H:

$$H = [h_1, h_2, h_3, \ldots, h_T] \tag{8}$$

Where h_i represents the hidden state vector of the BiLSTM at the ith time step.

2.3. Multi-head attention mechanism

In the intent recognition model that incorporates the multi-head attention mechanism, self-attention can be employed to capture the correlations between different positions in the input sequence [32]. Additionally, multiple attention heads can be used to better utilize information from different aspects of the input sequence. Each attention head can focus on different aspects, such as vocabulary, syntax, context, and so on. Finally, the weighted representations obtained from each attention head are concatenated to obtain the representation of the entire input sequence. The multi-head attention mechanism helps the intent recognition model to better handle different parts of the input sequence and automatically weight them, thereby improving the model's performance. The structure of the multi-head attention mechanism is illustrated in Fig. 4.

The operation of attention is to map queries (Q) to key–value pairs (K-V). In the multi-head attention mechanism, each attention head employs self-attention. The calculation method of the multi-head attention is as follows: First, linear transformations are applied to Q, K and V. Next, these transformed values are inputted into the Scaled Dot-Product Attention mechanism h times, where each calculation represents one head. The parameters between different heads are not shared. The

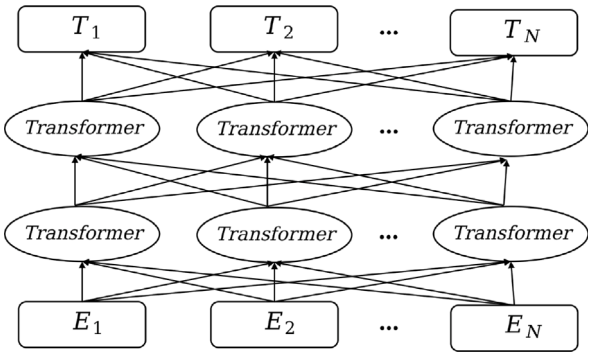


Fig. 5. BERT model architecture.

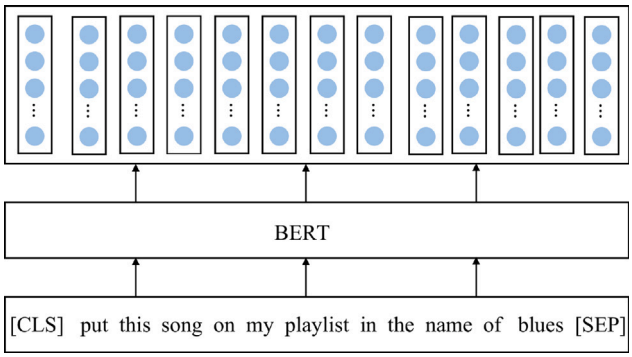


Fig. 6. Output of the BERT mode.

results of h Scaled Dot-Product Attention calculations are concatenated, and then a linear transformation is applied to obtain the attention matrix as the output of the multi-head attention mechanism.

In Section 2.2, the output H from the BiLSTM is fed into the multi-head attention mechanism. We linearly transform H using three parameter matrices, denoted as W^Q, W^K and W^V, to obtain the query, key, and value matrices respectively.

$$\begin{cases} Q = H \cdot W^Q + b^Q \\ K = H \cdot W^K + b^K \\ V = H \cdot W^V + b^V \end{cases} \tag{9}$$

Here, W^Q, W^K and W^V are learnable weight matrices, and b^Q, b^K and b^V are learnable bias vectors.

We split the query, key, and value matrices into 8 heads, each with a dimension of b_h. The split query matrix is denoted as $Q = [q_1, q_2, \ldots, q_h]$, the key matrix as $K = [k_1, k_2, \ldots, k_h]$, and the value matrix as $V = [v_1, v_2, \ldots, v_h]$. We calculate the attention weights for each head by performing a scaled dot product between the query matrix and the key matrix. Then, we multiply the attention weights with the value matrix to obtain the attention output for each head.

$$Head_i = SDA(Q, K, V) \tag{10}$$

By concatenating the outputs of multiple attention heads, we can obtain the final output of the multi-head attention mechanism.

$$M\_H = concat(Head_1, Head_2, \ldots, Head_h)W^T \tag{11}$$

2.4. BERT

BERT is a pre-trained language model based on the Transformer network architecture. It was introduced by Google in 2018 and has achieved groundbreaking performance on various NLP tasks [33]. BERT is a deep bidirectional Transformer encoder that can learn contextually aware word representations.

BERT employs two pre-training tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). The MLM task requires the model to predict the masked words in the input text, while the NSP task requires the model to determine whether two sentences are adjacent and in a specific order within the input sequence. Through training on these two tasks, BERT can learn universal word representations while preserving contextual semantic information. The breakthrough performance of BERT is attributed to its ability to model bidirectional context and the utilization of large-scale pre-training data. The model structure is illustrated in Fig. 5, where $E_1, E_2, \ldots, E_N$ represents the input vector to the model, and $T_1, T_2, \ldots, T_N$ denotes the vectorized output representation of the text obtained through bidirectional Transformer encoding.

The outputs of the BERT model consist of two forms: word embeddings and contextual embeddings. Word embeddings are representations generated by BERT for each input word. Contextual embeddings are obtained by BERT processing the entire input sequence as a whole. At the output layer of the model, BERT generates a set of contextual embeddings, where each position in the input sequence is associated with a corresponding vector representation. These contextual embeddings can be utilized for sentence-level tasks. The output of the BERT model is illustrated in Fig. 6.

After training the BERT model, it automatically adds a special token *[CLS]* at the beginning of the input text and *[SEP]* between every two sentences to differentiate the boundaries of different sentences. Each input token is assigned a corresponding vector representation after passing through the BERT model. Therefore, to obtain a sentence-level text vector representation, we simply need to retrieve the vector representation corresponding to the special token *[CLS]*. This vector contains the contextual information of the entire input sentence and has undergone multiple layers of non-linear transformations in the BERT model, making it a powerful representation that can be used for intent recognition tasks. The input data x_t, after going through BERT, obtains the *[CLS]* vector representation.

$$H_{cls} = BERT(x_t) \tag{12}$$

Concatenate the vector M_H and H_{cls}.

$$CBMB = [M\_H, H_{cls}] \tag{13}$$

Finally, pass it to the softmax layer for intent recognition.

$$P = softmax(W_f \cdot CBMB + b_f) \tag{14}$$

W_f and b_f are the weight and bias terms, respectively, and P represents the intent classification.

The specific details of the CBLMA-B model in this paper are as follows:

$$\begin{cases} x_{cnn} = CNN(X_t) \\ H = BiLSTM(x_{cnn}) \\ M\_H = Multi - headAttention(H) \\ H_{cls} = BERT(X_t) \\ CBMB = [M\_H, H_{cls}] \end{cases} \tag{15}$$

3. Experimental results and data analysis

3.1. Experimental dataset

The SNIPS (Spoken Natural Language Interaction with Programming Systems) dataset is a publicly available dataset used for speech recognition and natural language understanding. It aims to assist developers in training and evaluating natural language interaction applications such as voice assistants, chatbots, and voice automation systems. The dataset includes natural language intents from multiple domains, including ordering a taxi, restaurant reservations, playing music, querying weather,

Table 1
Intent and Utterance examples.

Intent	Utterance example
SearchCreativeWork	I'd like to see the show onion sportsdome
GetWeather	What's the weather in my current spot the day after tomorrow
BookRestaurant	Book a spot for 3 in mt
PlayMusic	I want to listen to seventies music
AddToPlaylist	Please add iris dement to my playlist this is selena
RateBook	I want to give this current textbook 4 points
SearchScreeningEvent	Give me a list of movie times for films in the area

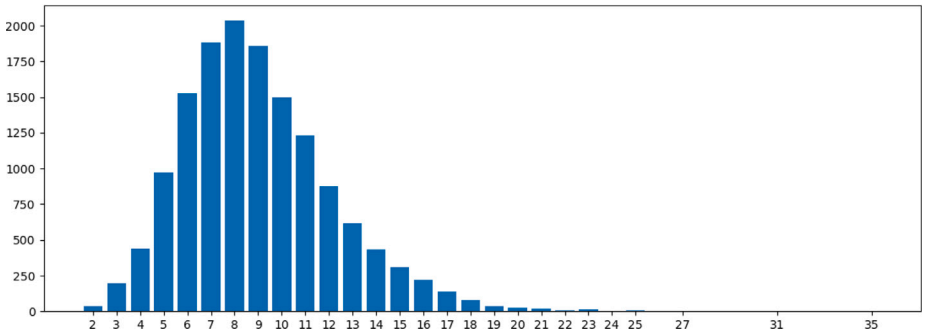


Fig. 7. Sentence length statistics.

Table 2
Sample counts of each dataset subset.

	Training set	Validation set	Test set	Aggregate
SearchCreativeWork	1847	100	107	2054
GetWeather	1896	100	104	2100
BookRestaurant	1881	100	92	2073
PlayMusic	1914	100	86	2100
AddToPlaylist	1818	100	123	2040
RateBook	1876	100	80	2056
SearchScreeningEvent	1852	100	107	2059
Aggregate	13 084	700	699	14 482

Table 3
Hyperparameter Settings.

Hyperparameter	Value
Maximum truncated text length	25
Initial learning rate	0.0001
Learning rate decay	0.99
Total training iterations	30
Batch size	64
Number of attention heads	8
Kernel sizes	(2, 3, 4)
Kernel numbers	(50, 100, 150)
Number of cross-validation folds	10

and more. Each intent consists of multiple example sentences [34]. The SNIPS dataset comprises 14,482 intents, covering 7 different domains: SearchCreativeWork, GetWeather, BookRestaurant, PlayMusic, AddToPlaylist, RateBook, and SearchScreeningEvent. Each intent has several samples, and each sample includes a natural language query, the corresponding slot values for that query, and the intent category. The data for each category has been cleaned, and a sample can be seen in Table 1.

Table 2 shows the number of subsets for each category of the Snips dataset used in this paper.

3.2. Hyper parameter

To select an appropriate text length, this study conducted statistical analysis on the text lengths in the dataset.

From Fig. 7, it can be observed that the sentence lengths in the dataset are mostly concentrated between 2 and 25. Therefore, this study truncates the maximum text length to 25. The settings of the hyperparameters during the experimental process are shown in Table 3.

3.3. Experimental comparison

In Fig. 8, the intent categories are abbreviated as Table 4.

From Fig. 8, it can be observed that the model in this study achieved a 100% accuracy and recall rate for category 2 on the validation set. For categories 4 and 5, both the accuracy and recall rates were 0.99, indicating a strong predictive capability of the model for these categories. This implies that the majority of samples predicted as these categories were correctly classified, and almost all actual samples belonging to these categories were accurately predicted. However, the performance of the model was relatively lower for category 6. The recall rate was approximately 87%, indicating a certain misclassification rate among the samples predicted as category 6. Additionally, a considerable proportion of samples from the actual category 6 were incorrectly predicted as other categories. The validation set report can be found in Table 5.

Table 4
Intent categories abbreviation.

Class	Abbreviation
AddToPlaylist	AP
RateBook	RB
BookRestaurant	BR
GetWeather	GW
SearchCreativeWork	SCK
SearchScreeningEvent	SSE
PlayMusic	PW

Table 5
Validation set report.

Class	Precision	Recall	F1-score	Support
AddToPlaylist	0.97	0.98	0.975	100
RateBook	1.000	1.000	1.000	100
BookRestaurant	0.990	0.980	0.985	100
GetWeather	0.971	0.99	0.98	100
SearchCreativeWork	0.876	0.99	0.93	100
SearchScreeningEvent	0.978	0.87	0.921	100
PlayMusic	0.99	0.95	0.969	100

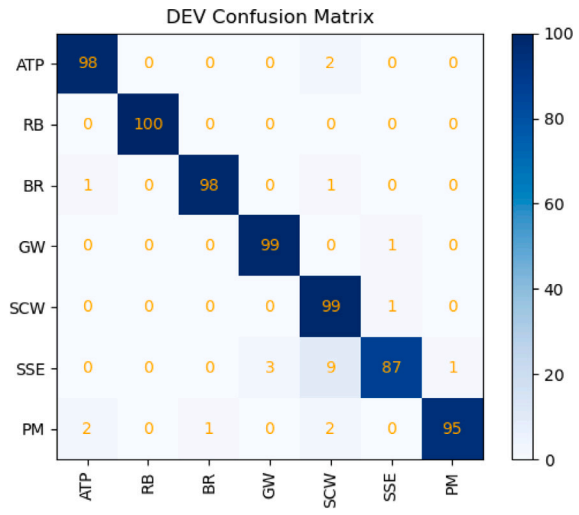


Fig. 8. The confusion matrix graph of CBLMA-B in the validation set.

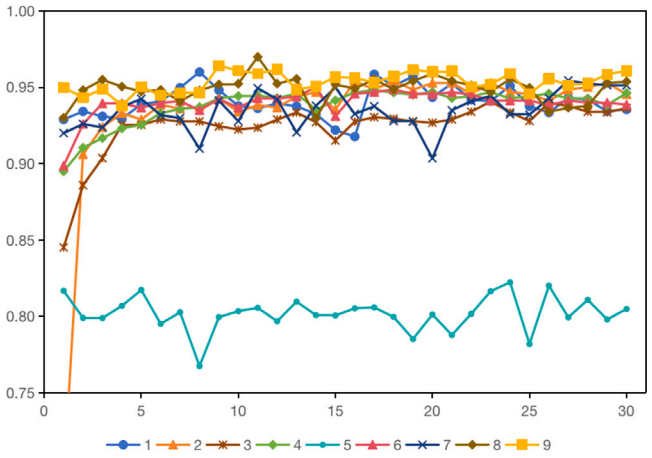


Fig. 9. The accuracy of each model on the validation set.

3.3.1. Contrast experiment

In this study, a comparison was made with several classical deep learning models and their combinations. The experimental models were labeled as follows: CNN as 1, BIGRU as 2, BILSTM as 3, BERT as 4, Transformer as 5, BiLSTM and multi-head attention as 6, CNN and BILSTM combination as 7, CNN + BIGRU + MUTI + BERT as 8, CBLMA-B as 9. MUTI stands for Multi-Head Attention Mechanism.

Fig. 9 shows the accuracy of each comparison model on the validation set. The curve of this model always oscillates around 0.95, the accuracy of Model 5 is the lowest, and the oscillation amplitude of the curves of the other models is relatively large.

From Fig. 10, the analysis and comparison of F1 scores on a set of validation data for different models are observed. Model 2 performs slightly better compared to Model 1. Model 3 exhibits a more outstanding performance compared to the previous two models. Model 7 shows better overall performance compared to individual Model 1 or Model 3. Model 5 performs relatively poorer with lower and more fluctuating scores. Model 6 demonstrates good performance in terms of overall effectiveness. Model 9 achieves the highest F1 score among all the models, ranging from 0.930 to 0.966 with a median of 0.952. This model demonstrates stable and overall superior performance.

In the comparative experiments, this paper utilizes the Snips and Atis datasets (Atis is a dataset commonly used to test the performance of Natural Language Processing (NLP) models in the aviation domain.

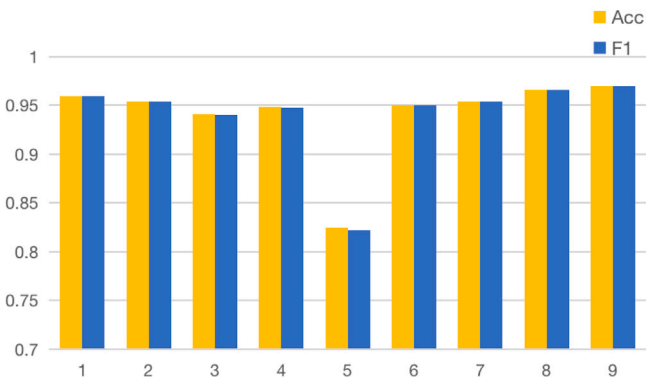


Fig. 10. Bar chart of accuracy and F1 score on the validation set.

It consists of a series of dialogue samples about flights, airports, dates, and other information [35]) as the input for models 1 to 9. The datasets are divided into training sets, validation sets, and test sets. The training set is used for adjusting the weights and parameters of the model, the validation set for selecting the optimal hyperparameter configuration and tuning the model, and the test set for evaluating the final performance of the model. After setting the architecture and hyperparameters of each model, we use the training set for backpropagation and gradient updates during the training process. Simultaneously, we use the validation set to monitor the model's performance and make adjustments and improvements based on its performance on the validation set. Upon the completion of model training, we calculate the values of various metrics to compare the performance differences between different models. Finally, we compile the experimental results of each model in Table 6. It can be observed that the model proposed in this paper achieves the best results in terms of loss rate and F1 score.

Table 6 presents the comparative results of different models on the test set. It can be observed that the model proposed in this study (Model 9, CBLMA-B) achieves the best performance in terms of both loss rate and F1 score.

3.3.2. Ablation experiment

In the ablation experiment, CNN is denoted as ①, BiLSTM as ②, MUTI as ③, and BERT as ④. The comparative results of accuracy, F1 score, and loss rate on the test set in the ablation experiment can be seen in Fig. 11.

In the first step, the performance metrics of the original model in this study were as follows: loss rate of 0.026, accuracy of 0.957, and F1 score of 0.958. These results indicate that the original model performs well in predictions, with a low loss rate and accuracy and F1 values close to 0.96. This suggests that the model is capable of accurately classifying and predicting the samples.

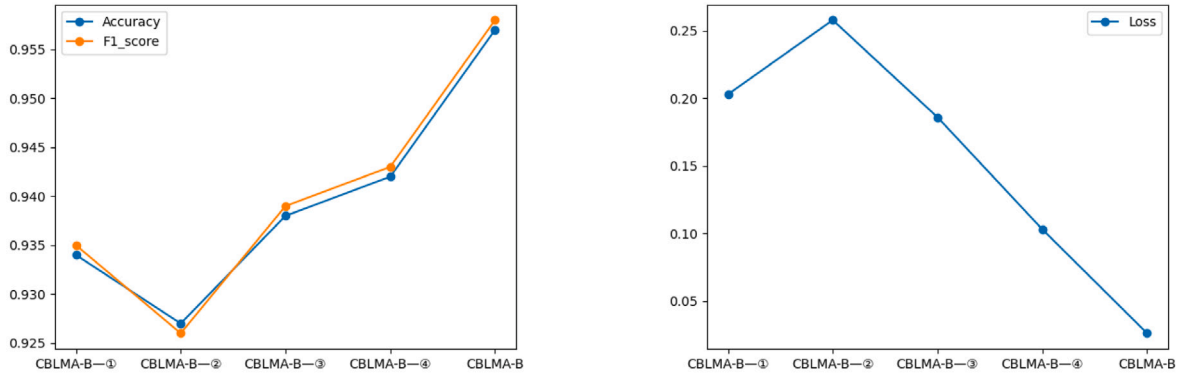
In the second step, the analysis was performed by removing the CNN component from the model. After removing the CNN layer, the model's loss rate increased to 0.203, accuracy decreased to 0.934, and F1 score decreased to 0.935. This indicates that the CNN component contributes significantly to the performance of the model. Removing it results in the model's inability to effectively extract features from the data, leading to a performance decline.

In the third step, the BiLSTM layer was removed from the model. After removing the BiLSTM layer, the model's loss rate increased to 0.258, accuracy decreased to 0.927, and F1 score decreased to 0.926. BiLSTM is a recurrent neural network structure that is capable of handling sequential data and capturing contextual information. By removing the BiLSTM layer, the model is unable to fully utilize the sequential information, resulting in a performance decline.

In the fourth step, the Multi-Head Attention layer was removed from the model. After removing the Multi-Head Attention layer, the

Table 6
Comparative results of different models on the test set.

Index	Model	Snips				Atis			
		Loss	Acc	Recall	F1	Loss	Acc	Recall	F1
1	CNN	0.144	0.94	0.941	0.940	0.229	0.936	0.908	0.922
2	BiGRU	0.263	0.935	0.937	0.936	0.267	0.936	0.931	0.933
3	BiLSTM	0.302	0.934	0.936	0.935	0.231	0.936	0.931	0.933
4	BERT	0.169	0.914	0.916	0.915	0.172	0.921	0.920	0.920
5	Transformer	0.662	0.780	0.784	0.782	0.449	0.872	0.688	0.769
6	BiLLSTM+MUTI	0.170	0.937	0.940	0.938	0.292	0.927	0.906	0.916
7	CNN+BiLSTM	0.147	0.942	0.943	0.942	0.243	0.938	0.933	0.935
8	CNN+BiGRU+MUTI+BERT	0.131	0.941	0.942	0.941	0.207	0.947	0.948	0.947
9	**CBLMA-B**	**0.026**	**0.957**	**0.959**	**0.958**	**0.112**	**0.949**	**0.951**	**0.950**



(a) (b)

Fig. 11. Comparative results of ablation experiment on the test set.

model's loss rate increased to 0.186, accuracy decreased to 0.938, and F1 score decreased to 0.939. The Multi-Head Attention layer helps the model better capture relevant information from the input and improves performance.

In the fifth step, the BERT component was removed from the model. After removing the BERT component, the model's loss rate decreased to 0.103, accuracy decreased to 0.942, and F1 score decreased to 0.943. BERT is a pre-trained language model with powerful semantic understanding capabilities. The introduction of BERT in the original model clearly contributes to the performance of the model.

In summary, the results of the ablation experiment show that different components have varying effects on the performance of the model. CNN and BiLSTM play important roles in this task, as they are effective in feature extraction and capturing sequential information. Multi-Head Attention and BERT have shown significant improvements in the model's performance in this task.

4. Conclusion

This study focuses on the problem of intent recognition in human–machine dialogue systems and proposes a novel model called CBLMA-B. The objective of this model is to accurately determine the intent or purpose of users during interactions with the system by considering both the sequential information and structural features of the sentences. This enables the system to provide more appropriate responses or actions. Specifically, the proposed method employs a multi-level processing approach. It uses CNN to capture local important information in the text and BiLSTM to extract sequential features from the local information. Then, through the multi-head attention mechanism, the sequential features are further enriched, focusing more on useful sequence information. Additionally, the BERT model is utilized to extract the structural features of the sentences. Finally, the sequential features and sentence structure features are concatenated and fused to achieve more accurate intent recognition. Through this approach, contextual and semantic information in the text is successfully captured. The comparative experiments and ablation experiments demonstrate that the proposed method achieves good classification performance in intent recognition tasks.

With the development of artificial intelligence, the future of natural language processing will not only be its own domain but also the collective development of the entire field of artificial intelligence, such as the integration of text data and image data [36,37]. Therefore, under the multimodal paradigm, future research will focus first on optimizing the performance of the CBLMA-B model, especially its performance in large-scale data or real-time applications; secondly, further exploring the combination of the CBLMA-B model with image processing technology to achieve more comprehensive and accurate intent recognition. In addition, in-depth research on the robustness, real-time performance, and user experience of the model will be conducted to promote the joint development of natural language processing and image processing fields, making a positive contribution to the overall progress of artificial intelligence.

CRediT authorship contribution statement

Tiefeng Wu: Conceptualization, Writing – review & editing. **Miao Wang:** Methodology, Data curation, Writing – original draft. **Yunfang Xi:** Visualization, Investigation. **Zhichao Zhao:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] A. Celikyilmaz, D. Hakkani-Tur, G. Tur, A. Fidler, D. Hillard, Exploiting distance based similarity in topic models for user intent detection, in: 2011 IEEE Workshop on Automatic Speech Recognition & Understanding, IEEE, 2011, pp. 425–430.

[2] A. De, S.K. Kopparapu, A rule-based short query intent identification system, in: 2010 International Conference on Signal and Image Processing, IEEE, 2010, pp. 212–216.

[3] A. Fujii, Modeling anchor text and classifying queries to enhance web document retrieval, in: Proceedings of the 17th International Conference on World Wide Web, 2008, pp. 337–346.

[4] B.J. Jansen, D.L. Booth, A. Spink, Determining the user intent of web search engine queries, in: Proceedings of the 16th International Conference on World Wide Web, 2007, pp. 1149–1150.

[5] J. Ramanand, K. Bhavsar, N. Pedanekar, Wishful thinking-finding suggestions and'buy'wishes from product reviews, in: Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, 2010, pp. 54–61.

[6] X.-m. Liu, L. Liu, Question classification based on focus, in: 2012 International Conference on Communication Systems and Network Technologies, IEEE, 2012, pp. 512–516.

[7] Y. Niimi, T. Oku, T. Nishimoto, M. Araki, A rule based approach to extraction of topics and dialog acts in a spoken dialog system, in: Seventh European Conference on Speech Communication and Technology, 2001.

[8] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.

[9] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.

[10] J. Lafferty, A. McCallum, F.C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, 2001.

[11] O. Dušek, F. Jurčíček, A context-aware natural language generator for dialogue systems, 2016, arXiv preprint arXiv:1608.07076.

[12] F. Peng, D. Schuurmans, S. Wang, Augmenting naive bayes classifiers with statistical language models, Inf. Retr. 7 (2004) 317–345.

[13] B. Shen, J. Li, F. Bai, C.-M. Chew, Motion intent recognition for control of a lower extremity assistive device (LEAD), in: 2013 IEEE International Conference on Mechatronics and Automation, IEEE, 2013, pp. 926–931.

[14] K. Li, X. Wang, Y. Xu, J. Wang, Lane changing intention recognition based on speech recognition models, Transp. Res. C 69 (2016) 497–514.

[15] M. Firdaus, S. Bhatnagar, A. Ekbal, P. Bhattacharyya, Intent detection for spoken language understanding using a deep ensemble model, in: PRICAI 2018: Trends in Artificial Intelligence: 15th Pacific Rim International Conference on Artificial Intelligence, Nanjing, China, August 28–31, 2018, Proceedings, Part I 15, Springer, 2018, pp. 629–642.

[16] H.B. Hashemi, A. Asiaee, R. Kraft, Query intent detection using convolutional neural networks, in: International Conference on Web Search and Data Mining, Workshop on Query Understanding, 2016, pp. 134–157.

[17] S. Ravuri, A. Stoicke, A comparative study of neural network models for lexical intent classification, in: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU, IEEE, 2015, pp. 368–374.

[18] D. Guo, G. Tur, W.-t. Yih, G. Zweig, Joint semantic utterance classification and slot filling with recursive neural networks, in: 2014 IEEE Spoken Language Technology Workshop, SLT, IEEE, 2014, pp. 554–559.

[19] Y. Kim, Convolutional Neural Networks for Sentence Classification Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Emnlp 2014, October 25–29, 2014, Doha, Qatar, a Meeting of Sigdat, a Special Interest Group of the ACL, Association for Computational Linguistics, Doha, Qatar, 2014.

[20] H.B. Hashemi, A. Asiaee, R. Kraft, Query intent detection using convolutional neural networks, in: International Conference on Web Search and Data Mining, Workshop on Query Understanding, 2016.

[21] A. Trewhela, A. Figueroa, Text-based neural networks for question intent recognition, Eng. Appl. Artif. Intell. 121 (2023) 105933.

[22] A. Bhargava, A. Celikyilmaz, D. Hakkani-Tür, R. Sarikaya, Easy contextual intent prediction and slot detection, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2013, pp. 8337–8341.

[23] S. Ravuri, A. Stolcke, Recurrent neural network and LSTM models for lexical utterance classification, in: Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[24] J.-K. Kim, G. Tur, A. Celikyilmaz, B. Cao, Y.-Y. Wang, Intent detection using semantically enriched word embeddings, in: 2016 IEEE Spoken Language Technology Workshop, SLT, IEEE, 2016, pp. 414–419.

[25] A. Trewhela, A. Figueroa, Text-based neural networks for question intent recognition, Eng. Appl. Artif. Intell. 121 (2023) 105933.

[26] Y. Wang, Y. Shen, H. Jin, A bi-model based rnn semantic frame parsing model for intent detection and slot filling, 2018, arXiv preprint arXiv:1812.10235.

[27] D. Liu, Z. Zhao, L.-D. Gan, Intention detection based on bert-bilstm in task-oriented dialogue system, in: 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing, IEEE, 2019, pp. 187–191.

[28] X. Huang, T. Ma, L. Jia, Y. Zhang, H. Rong, N. Alnabhan, An effective multimodal representation and fusion method for multimodal intent recognition, Neurocomputing (2023) 126373.

[29] C. Liu, X. Xu, AMFF: a new attention-based multi-feature fusion method for intention recognition, Knowl.-Based Syst. 233 (2021) 107525.

[30] A. Onan, Hierarchical graph-based text classification framework with contextual node embedding and BERT-based dynamic fusion, J. King Saud Univ.-Comput. Inf. Sci. (2023) 101610.

[31] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, LSTM: A search space odyssey, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2016) 2222–2232.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[33] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[34] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, et al., Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces, 2018, arXiv preprint arXiv:1805.10190.

[35] G. Tur, D. Hakkani-Tür, L. Heck, What is left to be understood in atis? in: 2010 IEEE Spoken Language Technology Workshop, IEEE, 2010, pp. 19–24.

[36] X. Huang, T. Ma, L. Jia, Y. Zhang, H. Rong, N. Alnabhan, An effective multimodal representation and fusion method for multimodal intent recognition, Neurocomputing (2023) 126373.

[37] A. Rahate, R. Walambe, S. Ramanna, K. Kotecha, Multimodal co-learning: Challenges, applications with datasets, recent advances and future directions, Inf. Fusion 81 (2022) 203–239.

Tiefeng Wu was born in Jiamusi, Heilongjiang, P.R. China, in 1974. He received the Ph.D. degree from Xidian University, P.R. China. Now, he works in School of Information and Control Engineering, Qingdao University of Technology, His research interests include machine learning, deep learning and EDA.

E-mail: wutiefeng@qut.edu.cn

Miao Wang was born in Huairen, Shanxi, P.R. China, in 1998. Master candidate. Now, he studies in School of Information and Control Engineering, Qingdao University of Technology, His research interests include machine learning, deep learning and Natural Language Processing (NLP).

E-mail: wangmiao_qut@163.com

Yunfang Xi was born in Dezhou, Shandong, P.R. China, in 1999. Master candidate. Now, she studies in School of Information and Control Engineering, Qingdao University of Technology, Her research interests include machine learning, deep learning and Natural Language Processing (NLP).

E-mail: xi_yunfang@163.com

Zhichao Zhao was born in Jiamusi, Heilongjiang, P.R. China, in 1979. Her received the M.S. degree with electronic science and technology from Harbin Institute of technology, P.R. China, in 2008. Now, she was a associate professor with School of Information and Control Engineering, Qingdao University of Technology. Her research interests include computational intelligence and big data analysis.

E-mail: zhaozhichao@qut.edu.cn