IEICE TRANS. INF. & SYST., VOL.E107–D, NO.4 APRIL 2024

PAPER Special Section on Information and Communication Technology to Support Hyperconnectivity

Conceptual Knowledge Enhanced Model for Multi-Intent Detection and Slot Filling

Li HE[†,††a)], **Jingxuan ZHAO**[†,††b)], ***Nonmembers***, **Jianyong DUAN**[†,††c)], ***Member***, **Hao WANG**[†,††,†††], ***and*** **Xin LI**[†,††], ***Nonmembers***

SUMMARY In Natural Language Understanding, intent detection and slot filling have been widely used to understand user queries. However, current methods tend to rely on single words and sentences to understand complex semantic concepts, and can only consider local information within the sentence. Therefore, they usually cannot capture long-distance dependencies well and are prone to problems where complex intentions in sentences are difficult to recognize. In order to solve the problem of long-distance dependency of the model, this paper uses ConceptNet as an external knowledge source and introduces its extensive semantic information into the multi-intent detection and slot filling model. Specifically, for a certain sentence, based on confidence scores and semantic relationships, the most relevant conceptual knowledge is selected to equip the sentence, and a concept context map with rich information is constructed. Then, the multi-head graph attention mechanism is used to strengthen context correlation and improve the semantic understanding ability of the model. The experimental results indicate that the model has significantly improved performance compared to other models on the MixATIS and MixSNIPS multi-intent datasets.

key words: *knowledge enhancement, multi-intent detection, semantic slot filling, joint model*

1. Introduction

With the advent of the information age, the application of human-machine dialogue is increasing, such as intelligent speakers and Siri voice assistants. Therefore, Natural Language Understanding (NLU) technology in dialogue systems is becoming increasingly important. In order to accurately answer users' questions using knowledge and contexts, the system must first accurately understand the user's question, which relies on the two important parts of NLU: intent detection and slot filling.

People's conversations and human-machine dialogues often express multiple intents and needs in daily life. Gangadharaiah et al. found that users' speech expressions in the Amazon dataset often have more than one intent [1], and each intent may contain multiple slot information. Therefore, the accuracy and efficiency of multi-intent detection and slot filling technology can greatly affect the intelligence of dialogue systems and are particularly important in evaluating the overall quality of a dialogue system.

However, there are still two major problems and shortcomings in current multi-intent detection and slot filling tasks. The first one is the problem of data sparsity. The data sources for multi-intent detection are scarce [25], the amount of data is insufficient, and the cost of annotating data is very high [2], [23], making it difficult to obtain annotated data. Additionally, the occurrence frequency of some intents or slots is relatively low, which leads to poor detection performance of certain intents or slots. The second issue is the problem of long-distance dependence. Traditional multi-intent detection and slot filling technology often only considers local information in the sentence. For example, the Bi-Model based RNN semantic framework model is more inclined to focus on short-term memory [3], which makes it difficult to handle the long-distance dependence problem in long sentences, resulting in the model can't capture the relationship between different parts of the sentence well, thus affecting the accuracy and robustness of the system. Especially in some sentences containing complex intents or slots or vocabulary containing deep and complex concepts, it is necessary to consider the relationship between multiple parts of the sentence.

In the absence of external knowledge, relying only on a limited word sequence may ignore the deep semantic information in the dialogue. At present, there has been significant progress in knowledge-based dialogue generation [4], which uses relevant literature or knowledge bases to assist models in understanding semantics, and uses denoising or filtering techniques [4] to refine knowledge to better understand semantics.

Table 1 lists some related concepts and knowledge triples in the external knowledge base based on a conversation, which is related to specific keywords in the conversation. If this related knowledge can be introduced, the model's understanding ability will be greatly improved, which is an effective method to solve the above problem. In this article, we introduce conceptual knowledge for multi-intent detection and slot filling models. Conceptual knowledge refers to a group of related concepts and their semantic relationships. It can be used to describe the relationship between different entities, entity attributes, entity categories, and other information. It is a common external knowledge that can provide

Manuscript received May 27, 2023.
Manuscript revised August 23, 2023.
Manuscript publicized October 25, 2023.

[†]The authors are with College of Informatics, North China University of Technology, Beijing, China.
[††]The authors are with CNONIX National Standard Application and Promotion Lab, Beijing, 100144 China.
[†††]The author is with Beijing Urban Governance Research Center, Beijing, China.
a) E-mail: heli@ncut.edu.cn
b) E-mail: jungxuan0224@163.com
c) E-mail: duanjy@ncut.edu.cn

DOI: 10.1587/transinf.2023IHP0004

Table 1 External knowledge triplets and concepts.

Utterance	**Add** the **song** to the soundscapes for gaming **playlist** and then **play** Signe Anderson chant music that is **newest.**
Knowledge	***(song**; related to; music)* ***(song**; related to; sing)* ***(playlist**; related to; play)* ***(playlist**; is a; list)* ***(play**; related to; broadcast)*
Concept	***add** (increase, improve, raise)* ***newest** (new, old, fresh)*

richer semantic information for multi-intent detection and slot filling tasks [5].

We propose a **C**onceptual **K**nowledge **E**nhanced **M**odel for multi-intent detection and slot filling, named **CKEM**. This method utilizes a pre-trained language model and a conceptual knowledge base, ConceptNet, to enhance the model's ability to understand and express semantic relationships between different parts of a sentence. At the same time, the performance of the model can be further improved by introducing a graph attention mechanism [6] in NLU tasks, enhancing the model's ability to pay attention to important information. Our main contributions are as follows:

- We propose CKEM, which combines external knowledge ConceptNet with dialogue text for joint multi-intent detection and slot filling tasks.
- We validated the effectiveness of concept context graphs and the graph attention mechanism in understanding semantics, understanding user intentions, and slot filling.
- The experimental results indicate that the model has achieved good performance on multiple competitive baselines.

2. Related Work

Intent detection and slot filling are two important tasks in natural language processing, which have received a great deal of attention in recent years. Intent detection can be viewed as a classification problem, which aims to identify the user's initial intention expressed in the sentence. Multi-intent detection belongs to the multi-label classification problem [7], which needs to identify multiple intentions of the user. Many classification methods have been used for multi-intent detection tasks, such as Naive Bayesian Model (NBM), Support Vector Machine [8], etc. Slot filling can be viewed as a sequence labeling task, which aims to obtain semantic slots and their corresponding values in the user's speech. Popular methods include Hidden Markov Model (HMM), Conditional Random Fields (CRF), etc.

With the development of deep learning and neural networks, it has been found that methods based on deep learning can achieve better results in these tasks. For example, Hai et al. [9] used RNN and Long Short-Term Memory (LSTM) to process intent detection tasks, which showed that sequence features are helpful for intent detection tasks. Liu et al. [10] used RNN language models to predict semantic slot labels. Zheng et al. [11] proposed using capsule-based neural networks to solve intent detection classification problems. Ni et al. [12] used token-level information from the encoder to improve the performance of semantic slot filling tasks. Liu et al. [26] utilized structure consolidation networks (SCN) to continuously learn new ideas that arise in daily life.

For intent detection and slot filling tasks, traditional techniques usually separate the two tasks into two independent sub-tasks and process them separately. Recently, more academic approaches have been to jointly process intent detection and slot filling tasks, while capturing and learning the semantic dependencies between intent detection and slot filling tasks to achieve better performance. For example, the joint model proposed by Qin et al. [13] can perform both intent detection and slot filling tasks at the same time.

However, the above models mainly focus on single-intent scenarios and cannot handle complex multi-intent scenarios. Gangadharaiah et al. [14] first focused on multi-intent scenarios and proposed the first multi-intent detection model Joint Multiple ID-SF. The AGIF model proposed by Qin et al. [15] introduced an intent-slot graph interaction layer, using multi-intent information to guide slot filling, but these models heavily rely on autoregressive methods. Based on this problem, Qin et al. constructed the GL-GIN model based on the graph attention network [16] and used non-autoregressive methods to alleviate the problem of inconsistent slots. Bai et al. [23] proposed a memory based method to incrementally learn emerging intentions in order to address the high computational cost of storing new data and intentions each time and retraining the entire data. Jiang et al. [28] proposed a method of separation parsing, which divides a sentence into multiple clauses containing a single intention, performs loop parsing on each clause, and finally integrates the parsing results. However, there are still problems with difficult-to-handle complex semantic concepts and long-range dependencies.

To solve these problems, the use of external knowledge is an effective solution, and introducing external knowledge can improve the model's understanding ability. Yu et al. [24] need to acquire a large amount of common sense knowledge in order to understand users' intentions on e-commerce platforms. Therefore, they utilize the generation ability of large language models and human-in-the-loop annotations to semi-automatically construct knowledge graphs. In this paper, the knowledge source we use is ConceptNet [7], a large-scale knowledge graph that describes general human knowledge using natural language, and plays an effective role in dialogue-related tasks [17], mainly including tuples, concepts, and relationships. Each tuple includes four parts: head concept, relationship, tail concept, and confidence score. This knowledge source has been introduced and applied to many NLP tasks, such as dialogue, question answering, text classification, and sentiment analysis.

Currently, graph neural networks have been successfully applied to various NLP tasks. It can directly manipulate the graph structure and build models based on graph

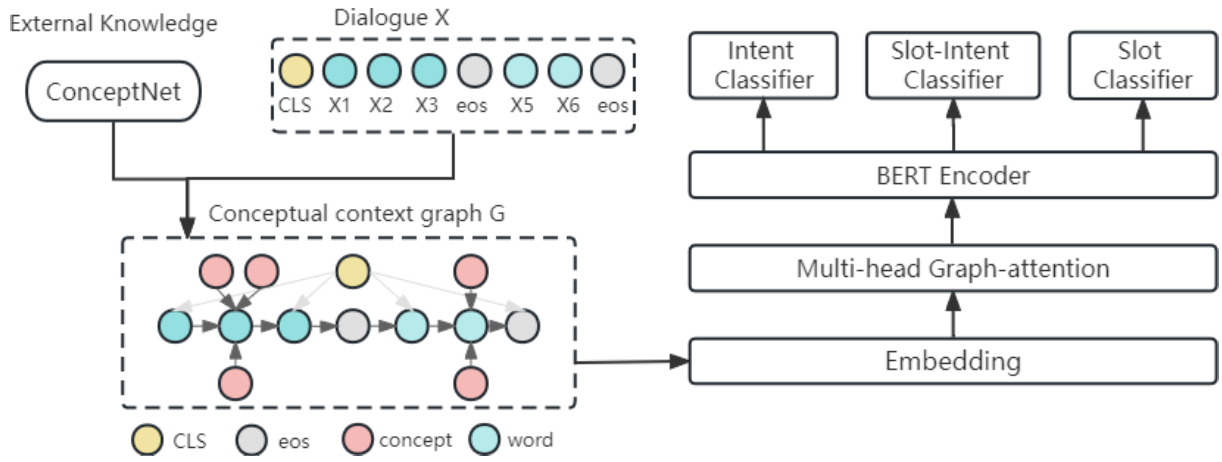


Fig. 1 CKEM model architecture.

structure information. Lin et al. [18] used the Graph Attention Network (GAT) for text classification tasks to merge the dependency information of parsers. Liu et al. [19] used graph neural networks to model non-local contextual information in sequence labeling tasks. Feng et al. [20] applied graph neural networks to text generation tasks and successfully generated abstract information for text. These studies have shown that the effectiveness of graph neural networks in the field of dialogue. For better consideration of the inherent intra-class and inter-class relations, Zhang et al. [27] constructed an instance-level and a class-level graph neural network, which not only propagate label information but also propagate feature structure. Through the model based on graph attention networks proposed by Qin et al. [16], it can also be seen that graph neural networks and attention mechanisms play a significant role in multi-intent detection and slot filling tasks. Therefore, this article will use the multi-head graph attention mechanism to construct a model based on the constructed concept context graph, in order to strengthen the correlation between concepts and contexts and improve the semantic understanding ability of the model.

3. Models

The conceptual knowledge enhanced multi-intent detection and slot filling model proposed in this article is shown in Fig. 1, which consists of three parts: concept context graph, encoder, and classifier. First, one utterance, i.e., a word sequence X, is taken as input based on a given set of utterances $D = [X_1, \cdots, X_M]$. In the concept context graph part, we introduce external knowledge to enrich the utterance X and construct a concept context graph G. Secondly, the concept-enhanced concept context graph is transformed into word embeddings, and the utterance is encoded using a multi-head attention mechanism and BERT encoding layer. Finally, the two results of the BERT encoder are input into the classifier, the averaged pooling result is input into the Intent classifier to obtain the multi-intent detection result, and the sequence hidden state result is input into the Slot classifier for the slot filling task. The Intent-Slot classifier with intent constraint attention mechanism is used to complete the joint task of multi-intent detection and slot filling.

3.1 Conceptual Context Graph

We construct a conceptual context graph G by introducing the external knowledge source ConceptNet, which is a large-scale knowledge graph that describes human knowledge in natural language and plays an important role in related tasks of dialogue systems. It includes 5.9 million tuples, 3.1 million concepts, and 38 types of relationships. Each tuple consists of four parts, namely, the head concept, relationship, tail concept, and confidence score, denoted as $\tau = (x, r, c, s)$, for example, (birthday, RelatedTo, happy, 0.19).

Take one utterance X from the given utterance set D as input, such as the i-th utterance $X_i = [x_0^i, \ldots, x_m^i]$ containing a sequence of m words, can be represented as $X = [x_1, \ldots, x_m]$. Firstly, insert a CLS token at the beginning of the sentence sequence to obtain $X = [CLS, x_1, \ldots, x_m]$. Then, for each non-stopword $x_i \in X$, we introduce its corresponding conceptual knowledge. We need to retrieve a set of related tuples $T_i = \{\tau_i^k = (x_i, r_i^k, c_i^k, s_i^k)\}_{k=1,\ldots,K}$ from ConceptNet.

Then, we use three heuristic steps to refine the relevant knowledge: (1) Filter out irrelevant tuples based on confidence score (i.e., $s_i^k > 0.1$) and related relationship, and then extract a subset $\hat{T}_i$ of Ti, which includes the most relevant concept knowledge tuples with the word x_i; (2) Rank the candidate concept tuples based on the confidence score of retrieved concepts $\{c_i^k\}_{k=1,\ldots,K}$. For each word x_i, we select the top-k tuples and form a concept subgraph with them; (3) Construct the conceptual context graph G, where each word $x_i \in X$ and its corresponding concept c_i^k form the vertices $V = \{v_i\}_{i=1,\ldots,m}$ (m is the number of vertices), and the graph G contains three types of directed edges: temporary edges between two consecutive words, edges between the word x_i and its corresponding concept c_i^k, and global edges between the CLS token and other vertices. Finally, utterance X is

enriched with introduced conceptual knowledge and represented as the conceptual context graph G.

3.2 Multi-Head Graph Attention Network

This article first uses a custom shared word embedding layer and a position embedding layer to convert each vertex $v_i \in G$ into a vector, resulting in $E_w(v_i)$ and $E_p(v_i)$; Vertices need to be distinguished between those in discourse and those in external knowledge. Therefore, we have set up a state embedding layer to obtain the vector $E_s(v_i)$ to distinguish between the two types of vertices. From this, we can obtain the embedded representation of the vector of v_i, as shown in Formula 1.

$$\mathbf{v}_i = E_w(v_i) + E_p(v_i) + E_s(v_i) \tag{1}$$

Then, the multi-head graph attention mechanism in the graph attention network is used to update the node representation $\mathbf{v}_i$, and external knowledge vectors are more accurately and effectively introduced into the model. Graph Attention Network is a graph neural network that utilizes a self-attention mechanism. This network calculates the attention of each node in a graph relative to each adjacent node in a manner similar to the self-attention mechanism in a transformer, and connects the features of the node itself with the attention features as the features of the node. Based on this, it performs tasks such as node classification.

For node i, the first step is to strengthen context association by focusing on all its direct neighbors j and calculate the attention score of the node and its adjacent nodes, which is the attention feature of the node. The method for calculating the attention score between the two nodes is shown in Formula 2.

$$e_{ij} = a(\mathbf{v}_i, \mathbf{v}_j) \tag{2}$$

Among them, the attention score e_{ij} represents the importance of node j to node i, and a represents the self-attention mechanism, which is a single-layer feedforward neural network.

To make it easier to calculate and compare attention scores, softmax is introduced to regularization of all adjacent nodes j of node i, as shown in Formula 3.

$$\alpha_{ij} = softmax(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{z \in \mathcal{A}_i} \exp(e_{iz})} \tag{3}$$

Among them, A_i represents the set of neighboring nodes of node i, α is the coefficient used for weighted summation during each convolution. The specific calculation process is shown in Formula 4. The result of concatenating node i and its adjacent node j is multiplied by attention mechanism a, followed by a nonlinear mapping and finally normalized to use the obtained result as the attention feature of the current node.

$$\alpha_{ij} = \frac{\exp(LeakyReLU(a[\mathbf{v}_i \| \mathbf{v}_j]))}{\sum_{z \in \mathcal{A}_i} \exp(LeakyReLU(a[\mathbf{v}_i \| \mathbf{v}_z]))} \tag{4}$$

In order to make the learning process of self-attention more stable, we use a multi-head attention mechanism in graph attention networks. We use H independent attention mechanisms and connect their features to obtain the following feature representations:

$$\hat{\mathbf{v}}_i = \|_{n=1}^{H} \sum_{j \in \mathcal{A}_i} \alpha_{ij}^n W_v^n \mathbf{v}_j \tag{5}$$

Where $\|$ represents the concatenation of multi-head attention mechanisms, n represents the self-attention mechanism of the n-th head, H is the number of self-attention mechanisms, where W_v^n is a linear transformation.

Finally, the final node feature representation Vi is obtained by connecting the features of the node itself with the attention features.

$$\mathbf{V}_i = \hat{\mathbf{v}}_i + \mathbf{v}_i \tag{6}$$

3.3 BERT Encoder

Due to the previous operation only targeting local context (i.e. direct neighbors), we also need to update the vertex representation using global context information (i.e. all other vertices) for global interaction. We used BERT's Encoder layer and pooling layer [21] to inject global information into all vertices. The output of the BERT encoder mainly consists of two parts, one is the pooler_ output which is the hidden state of the first token and the last layer of the sequence. It is the global representation of h_{cls} in the concept graph. It is further processed by the linear layer and the Tanh activation function. The output h_{cls} is a good summary of the semantic content of the input. Alternatively, by performing an average pooling operation on the hidden state sequence of the entire input sequence, the resulting average pooling result can better represent a sentence. Therefore, we input h_{cls} and the results of average pooling into the intention classifier to complete the multi-intent detection task. The other is sequence_ output which is represented as $h = (h_{cls}, h_1, \ldots, h_m, h_{sep})$. This is the output result of the sequence of the last hidden layer in the BERT model. It is usually used for tasks such as naming entity recognition and sequence labeling. Therefore, we will use the result sequence_ output and input them into the slot classifier to handle the slot filling task.

3.4 Classifier

3.4.1 Intent Classifier and Slot Classifier

The multi-intent detection task is accomplished by the Intent Classifier. The Intent Classifier receives the global representation of the concept graph h_{cls} obtained from the encoder output, and uses the sigmoid activation function to classify the intent, obtaining the probability of each intent label, as follows:

$$y^i = Sigmoid(W^i h_{cls} + b^i) \tag{7}$$

In the slot-filling task, to predict the slot at position k, we input the sequence_output representation h, which is the output of the encoder, into a separate slot classifier and normalize it to obtain a probability distribution over the slots, using the following method:

$$y_k^s = Softmax(W^s h_k + b^s) \tag{8}$$

3.4.2 Slot-Intent Classifier

To clearly capture the relationship between slots and intents, we predict the intent of each slot s_m at the token level. After calculating the representation rm of slot s_m, we vectorially connect the global discourse representation h_{cls} with r_m to obtain the slot-intent detection result y_m^l, as follows:

$$y_m^l = Softmax(W^l[h_{cls}|r_m] + b^l) \tag{9}$$

To better align the above slot-intent detection results with the intent predicted by the intent classifier, we use intent-constrained attention to combine the intention classifier result y_i and the slot-intent detection result y_m^l yields the final result of the slot intention classifier as y_m^p.

3.5 Training Objective

The training objective of the CKEM model is to maximize P. In the Eq. (10), the first two terms are the objectives of intent detection and slot filling, and the last term is the objective of slot-intent classification. The multi-intent detection task is trained using binary cross-entropy loss, and the other two tasks are trained using conventional cross-entropy loss. The loss of the entire CKEM model is the weighted sum of the losses of these three classifiers.

$$P = p(y^i|x)\prod_{k=1}^{n} p(y_k^s|x)\prod_{m} p(y_m^p|y^i, y_{s_m}^s, x) \tag{10}$$

4. Experiments

4.1 Datasets

In this experiment, we use the multi-intent datasets MixSNIPS and MixATIS constructed by Qin et al. [15]. The MixSNIPS dataset is based on the SNIPS dataset, which comes from the Snips personal voice assistant. The MixSNIPS dataset uses some connecting words such as "and" to connect sentences with different intents, and finally obtains 45,000 utterances for training, 2,500 for validation, and 2,500 for testing. Similarly, another multi-intent dataset MixATIS is based on the ATIS dataset. The ATIS dataset mainly consists of audio recordings of flight booking users, with 18,000 utterances in the training set, 1,000 in the validation set, and 1,000 in the test set. The proportion of utterances with 1, 2, and 3 intents in the two datasets is 3 : 5 : 2, and the data set division is detailed in Table 2.

Table 2 Distribution of user utterances.

Dataset	Train	Validation	Test
MixATIS	18000	1000	1000
MixSNIPS	45000	2500	2500

Table 3 Hyperparameters setting.

Parameters	Value
batchsize	32
max_seq_len	45
max_con_len	10
dropoutrate	0.2
learningrate	5×10^{-5}

4.2 Experiment Setup

In this experiment, we used the English uncased BERT-Base model, which contains 12 layers, 768 hidden states, and 12 heads. The maximum sequence length was set to 45, and the maximum concept length was set to 10. The training batch size was set to 32. We used random search to adjust the hyperparameters based on the semantic frame accuracy on the validation set. The dropout rate for the output layer of all three classifiers was 0.2. The hyperparameters of the experimental setup are shown in Table 3.

4.3 Equations

In this paper, Intent Accuracy (Intent Acc) is used to evaluate the performance of multi-intent detection tasks, Slot F1 is used as the performance metric for semantic slot filling tasks, and Semantic Frame Accuracy (SeFr Acc) is used to measure the overall performance of the joint model. When both the slot and intent are accurate, "SeFr Acc" considers the prediction of the utterance to be correct. The formulas for calculating these metrics are as follows:

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \tag{11}$$

$$F1 = \frac{2Precision * Recall}{Precision + Recall} \tag{12}$$

4.4 Experiment Results and Analysis

4.4.1 Experiment Design

To verify the effectiveness of the proposed concept knowledge-enhanced method for intent detection and slot filling tasks, we compared our model with several other joint models, including:

(1) Stack-Propagation [22]: A Stack-Propagation joint model to capture intent semantic knowledge and perform token-level intent detection to alleviate error propagation.

Table 4 Comparison of experimental results.

Models	MixATIS			MixSNIPS		
	Intent Acc	Slot F1	SeFr Acc	Intent Acc	Slot F1	SeFr Acc
Stack-Propagation	72.1	87.8	40.1	96.0	94.2	72.9
JointMultiple ID-SF	73.4	84.6	36.1	95.1	90.6	62.9
AGIF	74.4	86.7	40.8	95.1	94.2	74.2
GL-GIN	76.3	88.3	43.5	95.6	94.9	75.4
CKEM	**77.1**	**89.7**	**46.6**	**96.1**	**96.7**	**79.7**

Table 5 Results of ablation experiment.

Models	MixATIS			MixSNIPS		
	Intent Acc	Slot F1	SeFr Acc	Intent Acc	Slot F1	SeFr Acc
CKEM	**77.1**	**89.7**	**46.6**	**96.1**	**96.7**	**79.7**
-CCG	74.7	85.9	43.8	94.9	94.1	74.7
-multi-head GAT	75.4	86.2	44.1	95.3	94.6	75.8

(2) Joint-Multiple ID-SF [14]: A multitask framework that uses attention-based models to identify intent and generate slot labels at the token level.
(3) AGIF [15]: An adaptive Graph-Interactive framework for joint multi-intent detection and semantic slot filling, extracting intent information for token-level slot filling.
(4) GL-GIN [16]: A global-local graph interaction network that accelerates model decoding time and uses non-autoregressive methods to address incoherent semantic slot issues.

Table 4 summarizes the performance of different models on MixATIS and MixSNIPS. We observed that our model outperforms other models on all three metrics. For the intent detection accuracy metric (Intent_Acc), our model exceeded GL-GIN by 0.8% and 0.6%, respectively. For the Slot-F1 metric, our model has brought significant improvements (1.4% and 1.8%), which proves that the concept knowledge augmentation module, after improving the performance of multi-intent detection, guides the slot filling task together with the results of intention detection. Our model has a strong ability to recognize intentions and fill slots, and effectively improves the accuracy of the joint task of intention recognition and slot filling. In addition, our model has improved by 3.1% and 4.3% compared to GL-GIN in terms of more stringent metrics, namely semantic accuracy. This indicates that the model has a strong ability to understand semantics, verifying the effectiveness of understanding intention and utilizing the relationship between intent-slots.

4.4.2 Results of the Ablation Experiment

In order to verify the contributions of the proposed improvement factors in the model to the multi-intent detection and slot filling tasks, we conducted ablation experiments again, mainly considering two improvement factors, namely the concept context graph and the multi-head graph attention mechanism. The experimental results are shown in Table 5.

We compared our model with versions that removed two enhancement components separately (-CCG and -multi-head GAT) to analyze the impact of these components on the model's performance. From Table 5, we can see that removing these two components lowered the performance of our model. If external conceptual knowledge is not introduced and only the original textual input is used for model training, the decrease in Intent Acc score is even greater, indicating that external knowledge has a greater impact on the intent detection task, and injecting external knowledge is crucial for understanding intent. After replacing our model's encoder with a BERT encoder (-multi-head GAT) in the encoder part, both the Slot F1 score and the overall semantic accuracy significantly deteriorated, demonstrating the effectiveness of the multi-head graph attention mechanism in the entire multi-intent detection and slot filling task.

4.4.3 Effect of the Sequence Length on the Experiment

Since the sequence length of each sentence in the MixATIS and MixSNIPS experimental data sets is different, according to statistics, the sentence length ranges from about 7 to 100. Therefore, the above multi-intent detection and slot filling models will unify the sequence length during the experiment. We set the sequence length max_seq_len from 40 to 52 in MixATIS and MixSNIPS for experiments, and the experimental results (only SeFr_Acc experimental data are listed) are shown in Fig. 2.

By observing the line chart, it is found that when the sequence length increases from small to large, the result of SeFrAcc rises first and then decreases. It can be seen that the

Table 6 Utterance example.

Utterance Example														
Utterance	I	need	a	**cheap**	food	place	tomorrow	between	1	and	**3**	**pm**	in	Seattle
-CKEM	O	O	O	**O**	O	O	B-date	O	B-starttime	O	**O**	**O**	O	B-city
GL-GIN	O	O	O	**O**	O	O	B-date	O	B-starttime	O	**B-starttime**	**I-starttime**	O	B-city
CKEM	O	O	O	**B-pricing**	O	O	B-date	O	B-starttime	O	**B-starttime**	**I-starttime**	O	B-city

Table 7 The weight of some conceptual knowledge.

Knowledge		
Cheap	**Tomorrow**	**Seattle**
rel, affordable (0.99) rel, chintzy (3e-7) rel, gimcrack (8e-6)	is a, day (0.4) rel, later on (5e-2) rel, morrow (7e-3)	part of, washington (0.87) rel, emerald city (9e-2) is a , city (8e-3)

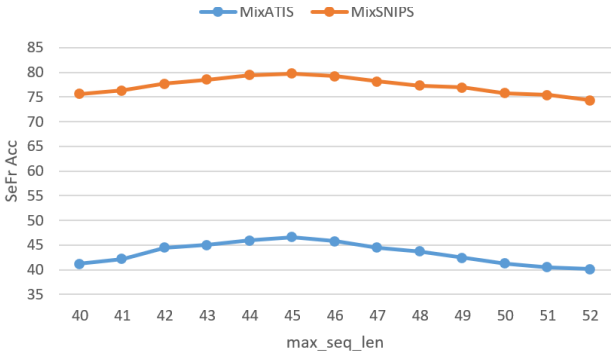


Fig. 2 Effect of the sequence length on the experiment.

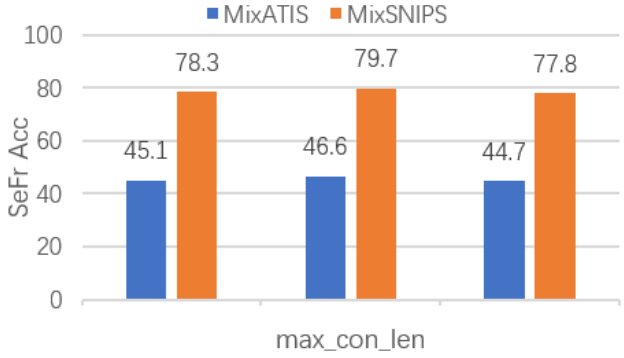


Fig. 3 Effect of the number of concepts on the experiment.

experimental effect is not good when the sequence length is shorter (less than 45). But it's not as if the longer the better. When the sequence length exceeds 45, the result of SeFr Acc will decrease with the increase of the length. The possible reason for this line chart is that the information contained in the sentence is not rich enough when the sequence length is short, the introduction of knowledge is more effective for the experiment. When the sequence length is long, we speculate that the model may introduce some noise during the zero-filling operation in the short sentences, which may result in low-performance improvement. Therefore, we chose 45 as the optimal sequence length value.

4.4.4 Effect of the Number of Concepts on the Experiment

We set the number of introduced concepts max_con_len to 5, 10, and 15 for experiments in MixATIS and MixSNIPS, and the experimental results (only the SeFr_Acc experimental data are listed) are shown in Fig. 3. It can be found that when we introduce a maximum of 5 concepts, the SeFr_Acc score is the lowest. Therefore, we speculate that the introduction of a maximum of 5 concepts has little effect on enriching knowledge, and more conceptual knowledge needs to be introduced; However, when a maximum of 15 concepts are introduced, the score is higher than 5, but not as good as when the number is 10. The possible reason is that some of the introduced 15 concepts have a low correlation with the present words, which will mislead the model and lead to poor results.

4.5 Case Study

In order to better illustrate this model, we use sentences from the dataset for case analysis, as shown in Table 6. We will visualize the introduced conceptual knowledge and its weight. In this case, we will analyze the three keywords that have the greatest impact on the semantics of the dialogue: 1, cheap, tomorrow, and Seattle.

Firstly, it can be observed that the model without the incorporation of our proposed method predicts "1" as the label "O". This is because, without the knowledge enhancement and graph attention mechanism, it is challenging to

predict that "1" represents a time entity. In contrast, our approach accurately predicts its slot label as "B-starttime". We believe that this is because the knowledge enhancement and graph attention mechanisms we proposed can assist the model in acquiring abundant knowledge and contextual information. Additionally, our method leverages knowledge related to "pm" to correctly predict the label "I-starttime" for the entity "pm".

Secondly, it can be seen that in the GL-GIN model, "cheap" is predicted as the label "O", and our model correctly predicts that the slot label "cheap" is "B-pricing". This correct prediction carries detailed information of the discourse. In the discourse example, the word "cheap" has a strong correlation and close relationship with the knowledge "affordable" we have introduced, which helps to determine the intention and mark the correct slot label. Our model also utilizes the fact of "later on" and "day" to determine the slot of "tomorrow" is "B-date", and uses knowledge related to "Seattle" to help identify its slot as "B-city".

5. Conclusion

This article proposes a conceptual knowledge enhanced multi-intent detection and slot filling model. We introduce external knowledge and construct a concept context graph G. Then, we encode the discourse using the multi-head graph attention mechanism and BERT encoding layer. Finally, the results of the BERT encoder are inputted into the classifier to obtain the results of intent detection and slot filling. The experimental results show that the model achieves the best results on two multi-intent datasets, and its understanding ability is greatly improved. In addition, we also validated the effectiveness of two components, the conceptual context graph and the multi-head graph attention mechanism.

Acknowledgments

This work is supported by R&D Program of Beijing Municipal Education Commission (KM202210009002), the National Natural Science Foundation of China (61972003), the Beijing Urban Governance Research Base of North China University of Technology (2023CSZL16), and the North China University of Technology Startup Fund. We would also like to thank the anonymous reviewers for their helpful comments. We would like to thank the referees for their comments, which helped improve this paper considerably.

References

[1] R. Gangadharaiah and B. Narayanaswamy, "Joint multiple intent detection and slot labeling for goal-oriented dialog," NAACL-HLT, vol.1, pp,564–569, Association for Computational Linguistics, 2019.

[2] T.-W. Wu, R. Su, and B. Juang, "A label-aware BERT attention network for zero-shot multi-intent detection in spoken language understanding," Proc. EMNLP, Online and Punta Cana, Dominican Republic, pp.4884–4896, Association for Computational Linguistics, Nov. 2021,

[3] H. Liu, F. Zhang, X. Zhang, S. Zhao, and X. Zhang, "An Explicit-Joint and Supervised-Contrastive Learning Framework for Few-Shot Intent Classification and Slot Filling," Findings of the Association for Computational Linguistics: EMNLP 2021, 2021.

[4] J. Wu, I. Harris, and H. Zhao, "Spoken language understanding for task-oriented dialogue systems with augmented memory networks," Proc. NAACL, pp.797–806, Association for Computational Linguistics, June 2021.

[5] Y. Wang, Y. Wang, X. Lou, W. Rong, Z. Hao, and S. Wang, "Improving dialogue response generation via knowledge graph ?lter," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.7423–7427, 2021.

[6] L. Qin, F. Wei, T. Xie, X. Xu, W. Che, and T. Liu, "GL-GIN: Fast and Accurate Non-Autoregressive Model for Joint Multiple Intent Detection and Slot Filling," Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, vol.1, pp.178–188, 2021.

[7] R. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: An Open Multilingual Graph of General Knowledge," AAAI, vol.31, no.1, pp.4444–4451, 2017.

[8] P. Haffner, G. Tur, and J.H. Wright, "Optimizing SVMs for complex call classification," Proc. IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2003), vol.1. pp.632–635, 2003.

[9] E. Haihong, P. Niu, Z. Chen, and M. Song, "A Novel Bi-directional Interrelated Model for Joint Intent Detection and Slot Filling," Proc. 57th Annual Meeting of the Association for Computational Linguistics, pp.5467–5471, 2019.

[10] B. Liu and I. Lane, "Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling," Interspeech 2016, pp.685–689, 2016.

[11] W. Zheng, N. Milic-Frayling, and K. Zhou, "Knowledge grounded dialogue generation with term-level de-noising," Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp.2972–2983, Association for Computational Linguistics, Aug. 2021.

[12] J. Ni, T. Young, V. Pandelea, F. Xue, and E. Cambria, "Recent advances in deep learning based dialogue systems: A systematic survey," Artificial Intelligence Review, vol.56, no.4, pp.3055–3155, 2022.

[13] L. Qin, T. Liu, W. Che, B. Kang, S. Zhao, and T. Liu, "A co-interactive transformer for joint slot filling and intent detection," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.8193–8197, 2021.

[14] R. Gangadharaiah and B. Narayanaswamy, "Joint Multiple Intent Detection and Slot Labeling for Goal-Oriented Dialog," North American Chapter of the Association for Computational Linguistics, pp.564–569, Association for Computational Linguistics, 2019.

[15] L. Qin, X. Xu, W. Che, and T. Liu, "AGIF: An Adaptive Graph-Interactive Framework for Joint Multiple Intent Detection and Slot Filling," Findings of the Association for Computational Linguistics: EMNLP 2020, pp.1807–1816, 2020.

[16] L. Qin, F. Wei, T. Xie, X. Xu, W. Che, and T. Liu, "GL-GIN: Fast and Accurate Non-Autoregressive Model for Joint Multiple Intent Detection and Slot Filling," Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, vol.1, pp.178–188, 2021.

[17] D. Ghosal, D. Hazarika, A. Roy, N. Majumder, R. Mihalcea, and S. Poria, "KinGDOM: Knowledge-Guided DOMain Adaptation for Sentiment Analysis," Proc. 58th Annual Meeting of the Association for Computational Linguistics, pp.3198–3210, 2020.

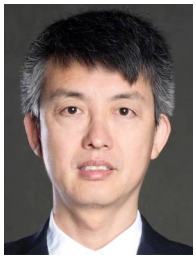
[18] H. Linmei, T. Yang, C. Shi, H. Ji, and X. Li, "Heterogeneous graph attention networks for semi-supervised short text classi?cation," Proc. 2019 Conference on Empirical ethods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp.4821–4830, Hong Kong, China, Association for Computational Linguistics, 2019.

[19] P. Liu, S. Chang, X. Huang, J. Tang, and J.C.K. Cheung, "Contextualized non-local neural networks for sequence learning," Proc. AAAI Conference on Arti?cial Intelligence, vol.33, no.1, pp.6762–6769,

2019.
[20] X. Feng, X. Feng, B. Qin, X. Geng, and T. Liu, "Dialogue discourse-aware graph convolutional networks for abstractive meeting summarization," 2020.
[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," NAACL-HLT(1), pp.4171–4186, Association for Computational Linguistics, 2019.
[22] L. Qin, W. Che, Y. Li, H. Wen, and T. Liu, "A stack-propagation framework with token-level intent detection for spoken language understanding," Proc. 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp.2078–2087, Hong Kong, China, Association for Computational Linguistics, 2019.
[23] G. Bai, S. He, K. Liu, and J. Zhao, "Incremental Intent Detection for Medical Domain with Contrast Replay Networks," Findings of the Association for Computational Linguistics: ACL 2022, pp.3549–3556, Dublin, Ireland, Association for Computational Linguistics, 2022.
[24] C. Yu, W. Wang, X. Liu, J. Bai, Y. Song, Z. Li, Y. Gao, T. Cao, and B. Yin, "FolkScope: Intention Knowledge Graph Construction for E-commerce Commonsense Discovery," Findings of the Association for Computational Linguistics: ACL 2023, pp.1173–1191, Toronto, Canada, Association for Computational Linguistics, 2023.
[25] N. Moghe, E. Razumovskaia, L. Guillou, I. Vulić, A. Korhonen, and A. Birch, "Multi3NLU++: A Multilingual, Multi-Intent, Multi-Domain Dataset for Natural Language Understanding in Task-Oriented Dialogue," Findings of the Association for Computational Linguistics: ACL 2023, pp.3732–3755, Toronto, Canada, Association for Computational Linguistics, 2023.
[26] Q. Liu, Y. Hao, X. Liu, B. Li, D. Sui, S. He, K. Liu, J. Zhao, X. Chen, N. Zhang, and J. Chen, "Class Lifelong Learning for Intent Detection via Structure Consolidation Networks," Findings of the Association for Computational Linguistics: ACL 2023, pp.293–306, Toronto, Canada, Association for Computational Linguistics, 2023.
[27] F. Zhang, W. Chen, F. Ding, and T. Wang, "Dual Class Knowledge Propagation Network for Multi-label Few-shot Intent Detection," Proc. 61st Annual Meeting of the Association for Computational Linguistics, vol.1, pp.8605–8618, Toronto, Canada, Association for Computational Linguistics, 2023.
[28] S. Jiang, S. Zhu, R. Cao, Q. Miao, and K. Yu, "SPM: A Split-Parsing Method for Joint Multi-Intent Detection and Slot Filling," Proc. 61st Annual Meeting of the Association for Computational Linguistics, vol.5, pp.668–675, Toronto, Canada, Association for Computational Linguistics, 2023.

Li He is an associate professor, graduated from Yanshan University in 2002 with a master's degree. Now she works in the Department of Computer Science, North China University of Technology. The main research interests include data warehouse and data mining, large database processing.

Jingxuan Zhao is a master student in College of Informatics, North China University of Technology. Her major research field is Natural Language Processing and Knowledge Graph.



Jianyong Duan professor, born in 1978. He graduated from Department of computer science, Shanghai Jiao Tong University by 2007. His major research field includes natural language processing and information retrieval.

Hao Wang received the Ph.D. degree in Computer Application Technology from Tsinghua University in 2013. He is now an associate professor in College of Informatics, North China University of Technology. His research interests include machine learning and data analysis.

Xin Li received the Ph.D. degree in Physics, Electrical and Computer Engineering from Yokohama National University in 2020. He is now a lecturer in College of Informatics, North China University of Technology. His research interests include knowledge extraction from nonuniform skewed data, deep learning, and artificial intelligence applications.