Hindawi
Complexity
Volume 2020, Article ID 8858852, 11 pages
https://doi.org/10.1155/2020/8858852

Research Article

A Hybrid Neural Network BERT-Cap Based on Pre-Trained Language Model and Capsule Network for User Intent Classification

Hai Liu,[1,2] Yuanxia Liu,[1] Leung-Pun Wong,[3] Lap-Kei Lee,[3] and Tianyong Hao[1,4]

[1]*School of Computer Science, South China Normal University, Guangzhou 510000, China*
[2]*Guangzhou Key Laboratory of Big Data and Intelligent Education, Guangzhou 510000, China*
[3]*School of Science and Technology, The Open University of Hong Kong, Kowloon, Hong Kong SAR 999077, China*
[4]*Institute for Advanced Study of Educational Development in Guangdong-Hong Kong-Macao Greater Bay Area, South China Normal University, Guangzhou 510000, China*

Correspondence should be addressed to Tianyong Hao; haoty@m.scnu.edu.cn

Received 21 August 2020; Revised 21 September 2020; Accepted 23 October 2020; Published 21 November 2020

Academic Editor: Zhile Yang

User intent classification is a vital component of a question-answering system or a task-based dialogue system. In order to understand the goals of users' questions or discourses, the system categorizes user text into a set of pre-defined user intent categories. User questions or discourses are usually short in length and lack sufficient context; thus, it is difficult to extract deep semantic information from these types of text and the accuracy of user intent classification may be affected. To better identify user intents, this paper proposes a BERT-Cap hybrid neural network model with focal loss for user intent classification to capture user intents in dialogue. The model uses multiple transformer encoder blocks to encode user utterances and initializes encoder parameters with a pre-trained BERT. Then, it extracts essential features using a capsule network with dynamic routing after utterances encoding. Experiment results on four publicly available datasets show that our model BERT-Cap achieves a $F1$ score of 0.967 and an accuracy of 0.967, outperforming a number of baseline methods, indicating its effectiveness in user intent classification.

1. Introduction

In question-answering systems and task-driven dialogue systems, the classification of user intent is an essential task to understand the target of user questions or discourses. Spoken dialogue system enables users to use natural language as a medium of communication and more conveniently obtain information [1]. However, it is difficult for a computer to understand human natural language in dialogue. To solve this problem, spoken language understanding has become a public topic of research in recent years. Spoken language understanding usually involves two sub-tasks, namely, user intent classification and semantic slot filling [2]. In question-answering systems and task-driven dialogue systems, users express their purposes using short sentences. User intent classification is essential in the identification and analysis of users' intents from short sentences and predicts the intent labels of dialogue sentences to understand what the users truly want [3]. For example, in spoken dialogue systems, a sentence "*I need a forecast for Jetmore Massachusetts in 1 hour and 1 second from now*" expresses the propose of getting weather information and the pre-defined intent label of this sentence is "*acquireweather*." And a question "*How do I turn online numbers by default in TextWrangler on the Mac?*" corresponds to the pre-defined intent label "*seek guidance*" in question-answering systems.

In natural language processing, word encoding has evolved from one-shot to word2vec. The emergence of word2vec has greatly promoted the development of user

intent classification. Recently, it is relatively common to analyze user intent based on neural network methods [4–6]. The short user sentences as input are mapped into a high-dimensional semantic space through word2vec, which is the process of converting words into computable and structured vectors. In the semantic space, words with similar meanings demonstrate their similarity through a special distance [7]. However, one problem of word2vec encoding is that it cannot solve polysemy. The Bidirectional Encoder Representations from Transformers (BERT) [8] obtains a context-based language representation model by pre-training on a large number of corpora; thus, leveraging the pre-trained BERT may optimize sentence representation compared to encoding the sentences based on word2vec.

With the development of deep learning algorithms in natural language processing, deep neural networks such as Convolutional Neural Networks (CNN) [9] and Recurrent Neural Networks (RNN) [10] are frequently applied to text classification tasks. With sentence coding, there are usually some networks to extract higher-level features. CNN treats sentences as spatial sequences and extract deep features through convolution kernels in different sizes [4]. RNN treats sentences as time series and forward sentence information through hidden state cycles [6]. Capsule networks [11] are also used to extract key information for text classification. Encoded sentences are used as the low-level capsule input, and the high-level capsule output is obtained through dynamic routing [12]. Deep neural network methods are frequently used to extract user short sentence features and classify users' hidden intentions. In order to promote the development of natural language understanding, previous works have constructed many publicly available datasets. However, certain existing datasets have the problem of uneven distribution of category samples. Focal loss is an improved loss function based on the softmax function to improve the accuracy of classification task for uneven distribution datasets. It is initially used in image detection tasks and has a positive effect on solving the imbalance of category distribution [13].

To further study user intent classification, a model BERT-Cap is proposed in this paper combining focal loss to solve the problem of uneven distribution of data. This model uses stacked transformer encoder to encode sentences and utilizes the pre-trained BERT as the initial parameters of the encoder. The weight parameters are continuously adjusted to obtain a context-dependent sentence representation during the training process. The capsule network is used in this model to extract key information of the sentences. The sentence representation obtained by the encoder is converted into vectors as the input of the low-level capsule. Through the iterative process of a dynamic routing algorithm, the key features of the sentences are transferred to the high-level capsule as the output. The focal loss focuses on the samples that are difficult to classify. Four publicly available datasets are used to evaluate the performance of the model. The results on these datasets show the effectiveness of our model in user intent classification.

The main contributions include the following: (1) a new hybrid model BERT-Cap based on pre-trained BERT language model is proposed. (2) Capsule network captures deep features of sentence representation obtained by the encoder and transfers iteratively important information from the lower-level capsule to the higher-level capsule through the dynamic routing mechanism. (3) Extensive experiments on four public available datasets demonstrate the effectiveness of our model.

The rest of the paper is organized as follows: Section 2 introduces related work on user intent classification. Section 3 introduces the design process of the model in detail. Section 4 demonstrates the experimental details and experimental results, and Section 5 draws conclusions of this work.

2. Related Work

User intent classification is mainly used in question-answering systems and dialogue systems to identify users' potential purposes. Most of the current research focuses on short sentence text classification. Text classification as an important task of natural language processing has been studied by a large number of methods. In the early days, traditional machine learning methods used manually extracted features for text classification [14, 15]. However, short sentences cover fewer semantic features and are difficult to extract manually [16]. Furthermore, manually extracting features is very expensive and requires a lot of resources.

The deep neural networks [17–19] have shown the ability to automatically extract text features and are widely used in various text classification tasks. The deep neural networks models include CNN [4] to extract n-gram features in sentence sequences for text classification and RNN [6] to extract sensitive patterns and rules in the sentence sequence, model non-Markovian dependence, and capture useful information of the sentence sequence for text classification, attention-based long short-term memory networks (LSTM) [20] to focus on the key words of the sentence sequence and reduce the effect of other irrelevant words, and others.

Recently, the pre-trained language model has become a popular method in natural language processing by fine-tuning parameters during the training of downstream tasks to have a better effect compared with deep neural networks models. Based on the pre-trained BERT, He et al. [21] proposed the method by combining CNN for intent determination. With the development of transfer learning, some work focuses on discovering new intentions never seen before. Xia et al. [22] studied zero-shot intent classification by capsule neural networks and used category similarity to classify new intents. A model [23] was proposed to classify new unknown intent by the algorithm of Local Outlier Factor. In addition, some researches focused on the classification of user intent with few-shot learning. Casanueva et al. [24] proposed the intent classifier in few-shot setups by pre-trained dual sentence encoders. Lin et al. [25] tried to improve the performance of user intention classification through supervised and unsupervised alternating training based on few-shot learning.

Two English datasets, ATIS [26] and SNIPS [27], were widely used in user intent classification task which contained the pre-defined user intent categories and semantic slot values. The joint model was proposed based on the two English datasets to improve the performance of user intent classification and semantic slot slotting. The joint model [28] was proposed with recursive neural networks. Liu and Lane [29] proposed the joint model with attention-based recurrent neural network. And the joint model based on BERT [30] improved the performance of user intent classification. Compared with English, other languages rarely have datasets with semantic slot values and generally only contain intent category labels. Khalil et al. [31] explored the intention classification based on the multilingual transfer ability of English and French. Xie et al. [32] used the multiple semantic features to study Chinese user intention classification based on ECDT [33] dataset. Attention-based BiGRU-CNN [16] model was proposed for Chinese question classification based on the Fudan University Chinese question dataset.

However, the previous research is mostly based on distributed word embedding lacking contextual information for user intent classification tasks. Distributed word embedding expresses words as the same vectors by looking for pre-trained word embedding and cannot handle the problem of polysemous word in different contexts. The pre-trained language model can be used as an encoder to obtain context-dependent sentence representations and promote the development of natural language processing. In order to explore the effectiveness of the pre-trained model in the classification of user intentions, we propose the hybrid model based on Chinese and English datasets and previous researches mostly focused on Chinese or English only. The model applies stacked transformer encoder to obtain context-dependent sentence encoding representation, and the publicly available pre-trained language model is used as the initial parameters of the encoder. Our model uses the dynamic routing mechanism of the capsule network to capture the deep features of the sentence. In practice, there are some low-data few-shot scenarios where only a handful of annotated examples of certain intent are available. We design experiments to explore the impact on the performance of our model when some categories have few samples in the datasets to simulate few-shot learning scenarios. For the datasets with uneven distribution of categories, we focus on samples that are difficult to classify and improve the accuracy of user intent classification with focal loss.

3. The BERT-Cap Model

A BERT-Cap hybrid model with focal loss based on pre-trained BERT and capsule network is newly proposed for user intent classification. The BERT-Cap model consists of four modules: input embedding, sequence encoding, feature extraction, and intent classification. The architecture of our model is shown Figure 1. Given a sentence as input, the sentence is represented by the input embedding module to a sequence of embedding by retaining token information, position information, and segment information. The sequence encoding module loads the pre-trained language model obtained by transfer learning, using the encoder of transformer to perform sentence encoding. The sequence encoding module can obtain the context-dependent sentence representation by multi-head self-attention mechanism. In the feature extraction module, the capsule network extracts rich features of sentence representations from the sequence encoding module and the higher-level capsule outputs key information for subsequent module. The intent classification module maps the higher-level semantic capsule to the label space by a fully connected operation and uses the focal loss based on a softmax function to improve the performance of the model.

3.1. Input Embedding. The input embedding of our model consists of three parts: token embedding, positional embedding, and segment embedding. Our model splits original sentence sequence by WordPiece [34] into token sequences. At the beginning of a token sequence, the special character [CLS] is used to store the semantic information of the entire input sequence. At the end of the sequence, the special character [SEP] is used to indicate the end of the sentence sequence. In the token sequence, the i-th token is denoted as $t_i \epsilon R^H$ and H is the dimension of hidden layer. In order to use the sequential information of the sequence, position embedding is added to encode position information. The positional embedding is denoted as $P_i \epsilon R^H$. In the sequence, the segment embedding of i-th token is the same $S_i \epsilon R^H$ since the input of model is a single sentence. Token embedding, positional embedding, and segment embedding have the same dimension in the high-dimensional space, and the input embedding $E_i \epsilon R^H$ is the summation of the three embeddings.

3.2. Sequence Encoding. The pre-trained language model transfers knowledge learned in large unlabeled corpora to downstream tasks through transfer learning and accelerates the development of natural language processing. The BERT model proposed by Google can obtain context-dependent sentence representation by two pre-training tasks, namely, masking language model and predicting the next sentence. Google has released two public pre-trained models, namely, $\text{BERT}_{\text{base}}$ and $\text{BERT}_{\text{large}}$, based on abundant text corpus. In order to promote the development of Chinese natural language processing, Joint Laboratory of HIT and iFLYTEK (HFL) trained Chinese language models with whole word masking strategy based on a massive Chinese corpus and released BERT-WWM-Chinese [35] and RoBERTa-WWM-Chinese based on RoBERTa [36] which was an improved model of BERT. Our model employs BERT's multiple transformer encoder structure to obtain context-dependent sequence encoding and uses the public pre-trained model as the initial parameters of the encoder.

The sequence embedding $E = (E_1, E_2, \ldots, E_{m-1}, E_m)$ retaining token information, position information, and segment information is denoted as the input of the lowest-level transformer encoder. In the encoder, the sequence embedding first obtains three matrices of query matrices $Q \epsilon R^{m \times H}$, key matrices $K \epsilon R^{m \times H}$, and value matrices

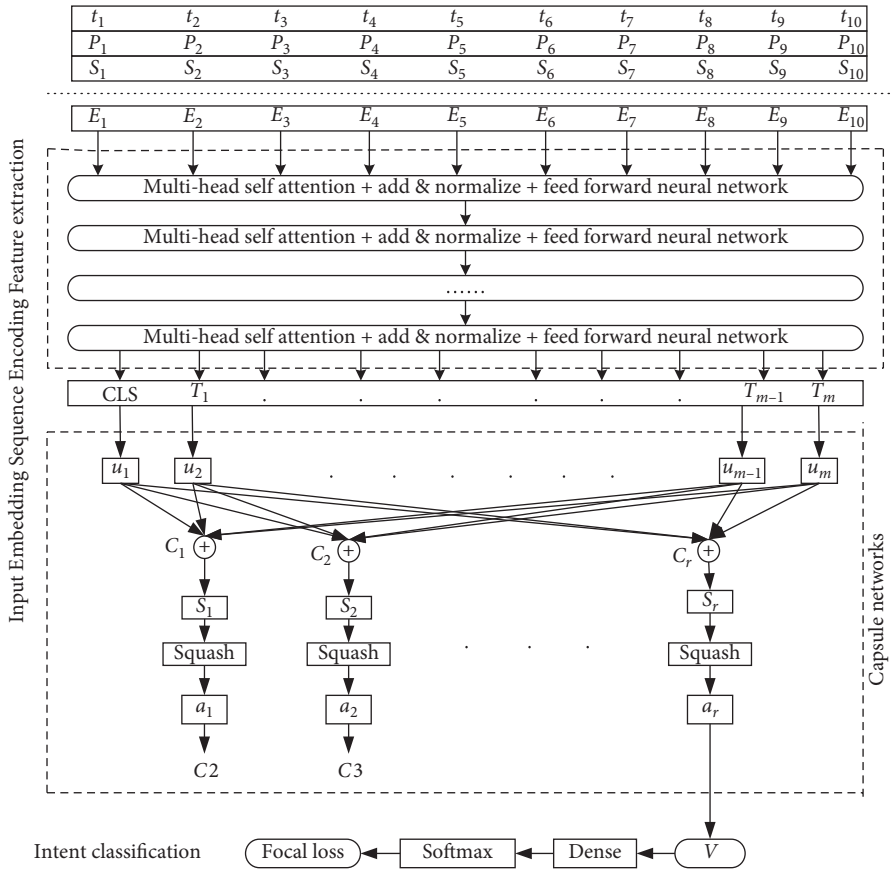

Figure 1: The architecture of our BERT-Cap model.

$V \epsilon R^{m \times H}$ through linear transformation. The linear transformation is as follows:

$$Q, K, V = W^Q E, W^K E, W^V E. \tag{1}$$

W^Q, W^K, and $W^V \epsilon R^{H \times H}$ are three different parameter matrices. Then, the query matrices Q, key matrices K, and value matrices V are the input of the scaled dot product attention function to obtain self-attention value. During calculating self-attention, the multi-head attention mechanism is used and Q, K, V are linearly mapped h groups to obtain different H/h-dimensional vectors. The self-attention calculation is as follows:

$$\text{Self-attention}\left(Q^i, K^i, V^i\right) = \text{Softmax}\left(\frac{Q^i K^{i^T}}{\sqrt{H/h}}\right) \cdot V^i. \tag{2}$$

The multi-head mechanism executes the self-attention function on the h groups of H/h-dimensional Q^i, K^i, and V^i in parallel. Each group produces an output result vector, and these result vectors are spliced together. The linear transformation is used to restore the vector of the H dimension.

$$\text{Mutli-head}\,(Q, K, V) = \text{Concat}\left(\text{attention}_1, \ldots, \text{attention}_h\right) W^H. \tag{3}$$

The result vectors of the multi-head self-attention operation are added with the self-attention input X for layer normalization as the input of the feed-forward neural network which contains two linear mapping functions and a nonlinear ReLU activation function. The layer normalization operation and feed-forward operation are as equations (4) and (5), respectively.

$$\text{Layer-normalization}\,(X) = \text{LayerNorm}\,(X + \text{multi} - \text{head}\,(X)), \tag{4}$$

$$F = f_1\left(\text{ReLU}\left(f_2\left(\text{layer} - \text{normalization}\,(X)\right)\right)\right). \tag{5}$$

Then, the output F is added with the input of the feed-forward neural network for layer normalization as the input of the next encoder. The number of transformer encoders is L in our model. The multiple transformer encoder structure obtains more sentence sequence syntax and semantic information in the process of sequence encoding.

3.3. Feature Extraction. The sequence encoder output $T = (\text{CLS}, T_1, T_2, \ldots, T_{m-1}, T_m)$ contains sentence sequence syntax and semantic information is used as the input of the feature extractor. The feature extractor consists of capsule networks with dynamic routing mechanism. The main characteristic of the capsule structure is vector in and vector out, while ordinary neuron is vector in and scalar out. The vector output from the capsule expresses richer features than the scalar output from neuron. The input sequence encoding T is first converted as the

lower-level capsules $U \epsilon R^m$ through linear transformation. The lower-level capsule u_i consists of n vectors and each vector has k dimensions. The lower-level capsule u_i is multiplied by the weight matrix c_i and summed to obtain a higher-level capsule. The squash activation function compresses this higher-level capsule s and determines what information is retained in each input vector of the lower-level capsule. The calculation of the squash activation function is as follows:

$$\text{Squash}(s) = \frac{s}{\sqrt{\|s\|^2 + e^{-7}}}. \tag{6}$$

The result of the squash activation function is multiplied with the lower-level capsule input to update the weight matrix c_r for the next routing process. The pseudocode of the dynamic routing Algorithm 1 is as follows:

The output of the dynamic routing mechanism is a higher-level capsule, which retains the important features of the sentence sequence during the iteration process and uses the weight matrix to continuously adjust the acquired features. Finally, a vector output is used to represent the sentence sequence as the input of the intent classifier.

3.4. Intent Classification. The input of the intent classifier is denoted as $O \epsilon R^{n \times k}$ containing important features of sentence sequence and we can calculate the intent representation $I \epsilon R^N$ by dense operation, where N is the number of pre-defined intent category labels. The calculation of dense operation is as follows:

$$I = \text{Dense}(WO + b). \tag{7}$$

W is the weight matrix and b is the bias. This dense operation maps the sentence sequence from the high-dimensional feature space to the low-dimensional category label space. Then, we use the softmax nonlinear activation function to convert the category label distribution obtained by the dense operation into a probability distribution. The category corresponding to the maximum value in the probability distribution is selected as the predicted intent label. The calculation of intent label prediction is as follows:

$$\text{label} = \arg\max(\text{Softmax}(I)). \tag{8}$$

3.5. Focal Loss. Focal loss is the loss function solving the problem of the category imbalance. Focal loss is an improvement on the standard softmax cross-entropy loss. Focal loss responds to smaller losses for easy-to-classify samples and pays more attention to difficult-to-classify samples by responding to larger losses. The formula of focal loss is as follows:

$$\text{loss} = -\alpha(1 - p_t)^{\gamma} \log(p_t). \tag{9}$$

α is the weight coefficient corresponding to each category and γ is the hyperparameter. The category with more samples has smaller weight coefficient α. The probability value p_t is the output of the softmax function. In our model, focal loss is used to replace the cross-entropy loss function. When the samples are easily classified, our model will reduce its proportion in the overall loss. For the samples that are difficult to classify, the larger loss value is calculated. The model focuses on these samples in the subsequent training process and gradient update process.

4. Experiments and Results

4.1. Dataset. Four real-word datasets are used in our experiments to evaluate the effectiveness of our proposed model. The SNIPS dataset contains 7 intent categories and 14,484 samples from voice assistants. The StackOverFlow dataset contains 20 intent categories and 20,000 samples about technical question originally released on Kaggle.com. The ECDT dataset contains 31 intent categories and 3,736 samples from human-machine dialogue systems, while the FDQuestion dataset contains 9 intent categories and 15,408 samples from the music entertainment field of Baidu Q&A. Table 1 shows the statistics of these datasets.

4.2. Evaluation Metrics. We choose four metrics including Precision (P), Recall (R), $F1$ score ($F1$), and Accuracy (Acc) that are widely used in classification tasks to evaluate the classification performance of our model. The higher the scores, the better the classification performance. The calculations are as equations (10)–(13). TP represents the number of samples predicted correctly, FP represents the number of samples that are incorrectly predicted, FN is the number of samples that are incorrectly predicted of other categories, and TN is the number of samples that are correctly predicted of other categories.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{11}$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{12}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}. \tag{13}$$

4.3. Baseline Methods. These baseline methods (http://ir.hit.edu.cn/SMP2017-ECDT-RANK) from the best models in the shared task from SMP conference are compared with our method to evaluate the performance of our model based on the ECDT dataset. These baselines include the following methods:

(1) CNN + domain template: the two-level system with domain template and convolutional neural network to perform multi-domain classifications.

(2) Lib-SVM + n-gram: this method designed a multi-feature user intent classification system based on the Lib-SVM classifier while feature selection adopted n-gram.

Input: all lower-level capsule $U = [u_1, u_2, \ldots, u_m]$
Output: the result of the squash activation function a
(1) **procedure** Routing (u_i, r)
(2) for all lower-level capsules u_i: $b_i = 0$
(3) **for** r iterations **do:**
(4) for all lower-level capsules u_i: $c_i = \text{softmax}(b_i)$
(5) The higher-level capsule: $s = \sum_i c_i u_i$
(6) The higher-level capsule: $a = \text{squash}(s)$
(7) for all lower-level capsules u_i: $b_i = au_i$
(8) **return** a

Algorithm 1: The dynamic routing algorithm.

(3) CNN + rules: two kinds of user intent classification methods were proposed based on semantic rules and based on CNN, respectively.

(4) CNN + ensemble learning: the method consisted of multiple residual convolution modules and the maximum pooling layer with ensemble learning to train classification models.

(5) LSTM + domain dictionary: the classification method adopted LSTM, while external domain knowledge table was constructed according to training datasets.

(6) LR + character + POS tags + target words + semantic tags [32]: the method used a traditional logistic regression with four feature expansions.

4.4. Parameter Settings. In our experiments, we applied the pre-trained $\text{BERT}_{\text{base}}$-uncased as the initial parameters of the sequence encoder for SNIPS dataset and the pre-trained $\text{BERT}_{\text{base}}$-case as the initial parameters of the sequence encoder for StackOverFlow dataset. We used the pre-trained RoBERTa-Chinese-WWM as the initial parameters of the sequence encoder for two Chinese datasets, namely, ECDT dataset and FDQuestion dataset. The parameter settings of our model are shown in Table 2.

4.5. The Results. To evaluate our model BERT-Cap for user intent classification, we firstly compared our model with base models with an ablation experiment. Table 3 shows the comparison of accuracy of these datasets using different methods for user intent classification on English datasets, while Table 4 shows the comparison result on Chinese datasets.

The results show that the performance of the BERT-Cap method of using BERT as the sequence encoder and capsule networks as the feature extractor surpassed that using BERT only or using BERT as the encoder with CNN as the feature extractor on all the four datasets. It illustrated that deep features of the context-dependent sentence representation obtained by the sentence sequence encoding from BERT could be extracted by capsule network for user intent classification. We replaced the softmax cross-entropy loss with focal loss on the basis of BERT-Cap. The accuracy of the model after adding focal loss had been improved compared with using cross-entropy loss on 2 out of 4 datasets and focal loss responded to different losses for training samples, making the model focused on these difficult-to-classify samples in the subsequent training process and gradient update process.

In order to study the impact of focal loss on the performance of user intent classification, we designed another experiment by comparing it with some commonly used models including CNN, RNN, RCNN, RNN + Attention, and Transformer in classification tasks on the two English datasets. The results of our experiments are shown in Table 5. The performance of the model with focal loss had been improved compared with the model without focal loss on SNIPS. However, the results on StackOverFlow have some reductions. They demonstrated that focal loss performs better on unbalanced datasets overall.

There were distribution differences of the categories on the datasets. To evaluate the classification accuracy of our proposed model on different categories, we calculated the recall value of each category obtained by the optimized model on the four datasets. Figure 2 shows the recall of each category achieved by the model of BERT-Cap with focal loss. In Figure 2(a), the recalls of two categories "*SearchScreeningEvent*" and "*SearchCreativeWork*" were 0.935 and 0.897, respectively, which were lower than other categories on the SNIPS. The reason might be that the two categories were relatively similar which made the model difficult to classify. For example, the sentence "*where can I find paranormal activity 3 playing near me 1 hour from now*" was difficult to classify with the sentence "*where can I see the movie across the line: the exodus of Charlie wright.*" In Figure 2(c), the recall of the category "是非类" (*Judgment*) was 0.676, lower than others on the FDQuestion. We found that larger proportion of the samples with the category label "是非类" (*Judgment*) were incorrectly classified as "评价类" (*Evaluation*). By looking at the original dataset, we found that the classification boundaries of the two categories were unclear and many samples were even difficult to distinguish by human annotators.

We designed the third experiment to analyze the relationship between the size of training data and the performance of the model and further analyzed the stability of our model based on the four datasets. We selected 0.5%, 1%, 5%, 10%, 25%, 50%, 75%, and 100% of the training data as the training subset, respectively. As shown in Figure 3, the accuracy of our model was significantly improved with the

Table 1: The statistics of the four datasets.

Datasets	Training	Developing	Testing	Total	#Categories
SNIPS	13,048	700	700	14,408	7
StackOverFlow	18,000	1,000	1,000	20,000	20
ECDT	2,299	770	667	3,736	31
FDQuestion	13,465	865	1,078	15,408	9

Table 2: The parameter settings of our model.

Parameter	Value
Sentence length	128
Batch size	32
Optimization	AdamW
Learning rate	$2e-5$
Dropout rate	0.25
Vector number	5
Vector dimension	10
Route	3
γ	2

Table 3: The performance of different methods on two English datasets.

Models\datasets	SNIPS				StackOverFlow			
	P	*R*	*F*1	Acc	*P*	*R*	*F*1	Acc
BERT	0.970	0.972	0.970	0.970	0.878	0.872	0.872	0.872
BERT + CNN	0.970	0.971	0.970	0.970	0.874	0.873	0.873	0.873
BERT-Cap	**0.977**	**0.978**	**0.977**	**0.977**	**0.882**	**0.880**	**0.880**	**0.880**
BERT-Cap + **focal loss**	**0.978**	**0.980**	**0.979**	**0.979**	0.883	0.878	0.879	0.878

Table 4: The performance of different methods on two Chinese datasets.

Models\datasets	ECDT				FDQuestion			
	P	*R*	*F*1	Acc	*P*	*R*	*F*1	Acc
BERT	0.962	0.952	0.956	0.955	0.791	0.819	0.801	0.843
BERT + CNN	0.968	0.956	0.96	0.960	0.798	0.824	0.808	0.849
BERT-Cap	**0.969**	**0.960**	**0.963**	**0.963**	**0.808**	**0.831**	**0.814**	**0.856**
BERT-Cap + **focal loss**	**0.972**	**0.972**	**0.971**	**0.967**	0.778	0.847	0.803	0.845

Table 5: The results of model comparison on the two English datasets.

Models	SNIPS				StackOverFlow			
	P	*R*	*F*1	Acc	*P*	*R*	*F*1	Acc
CNN	0.965	0.966	0.965	0.964	0.827	0.816	0.819	0.816
RNN	0.947	0.946	0.946	0.946	0.822	0.818	0.818	0.818
RCNN	0.966	0.967	0.966	0.966	0.851	0.839	0.842	0.839
RNN + Atten	0.961	0.961	0.961	0.961	0.845	0.827	0.831	0.827
Transformer	0.966	0.966	0.966	0.966	0.832	0.819	0.820	0.819
CNN + focal loss	0.969	0.969	0.969	0.969	0.838	0.831	0.832	0.831
RNN + focal loss	0.957	0.957	0.957	0.957	0.811	0.806	0.805	0.806
RCNN + focal loss	0.967	0.968	0.968	0.967	0.852	0.837	0.840	0.837
RNN + Atten + focal loss	0.970	0.970	0.970	0.97	0.819	0.805	0.808	0.805
Transformer + focal loss	0.971	0.970	0.970	0.97	0.824	0.815	0.816	0.815

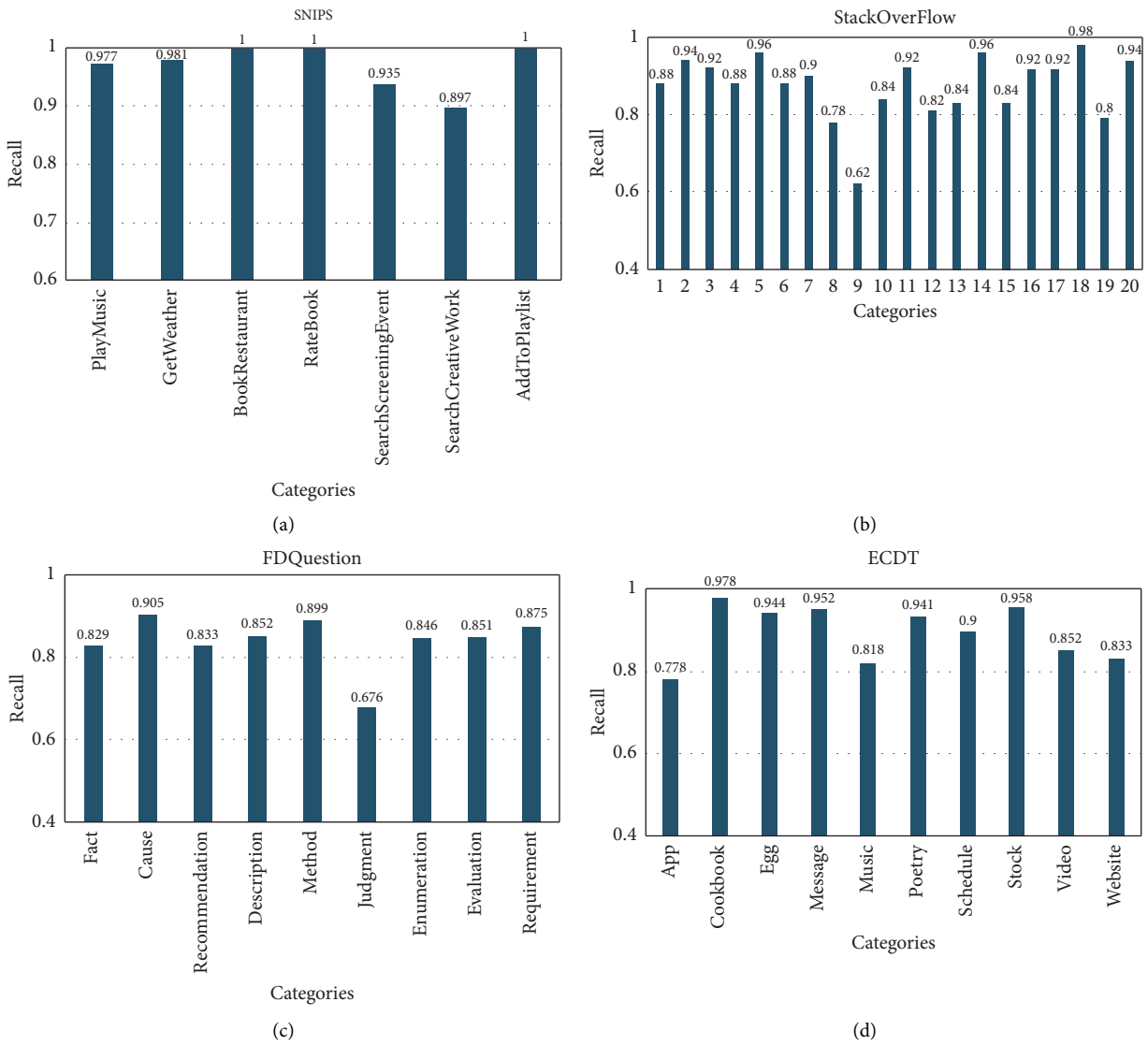

Figure 2: The recall comparison of each category on the four datasets including (a) SNIPS, (b) StackOverFlow, (c) FDQuestion, and (d) ECDT.

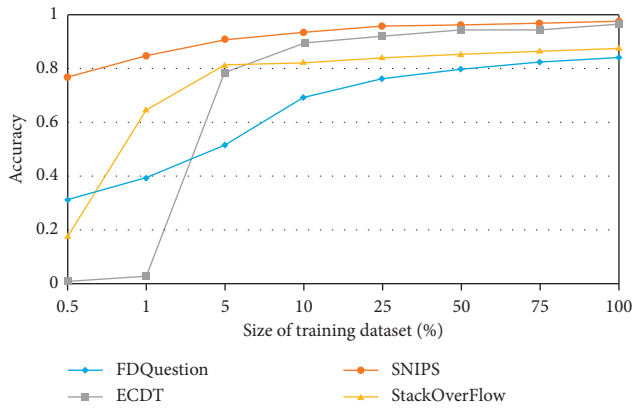

Figure 3: The accuracy of our model with different sizes of training samples on the four datasets.

increase of training data when the percentage was less than 25%. When the percentage exceeded 25%, the accuracy of the model changed relatively smoothly. Figure 3 illustrates that our model had a stable performance in the tasks of user intent classification.

In real scenarios, there were some low-data problem and certain intent categories had only few annotated examples available. We designed the fourth experiment to simulate low-data scenarios and explored the performance of our model in the scenarios. We selected 50% of the categories as the low-data categories and 0.5%, 1%, 5%, 10%, 25%, 50%, 75%, and 100% of these categories original training data were selected adding to training dataset, and other categories training data remained unchanged. Figure 4 shows how the recall of each category changes on the SNIPS and the FDQuestion.

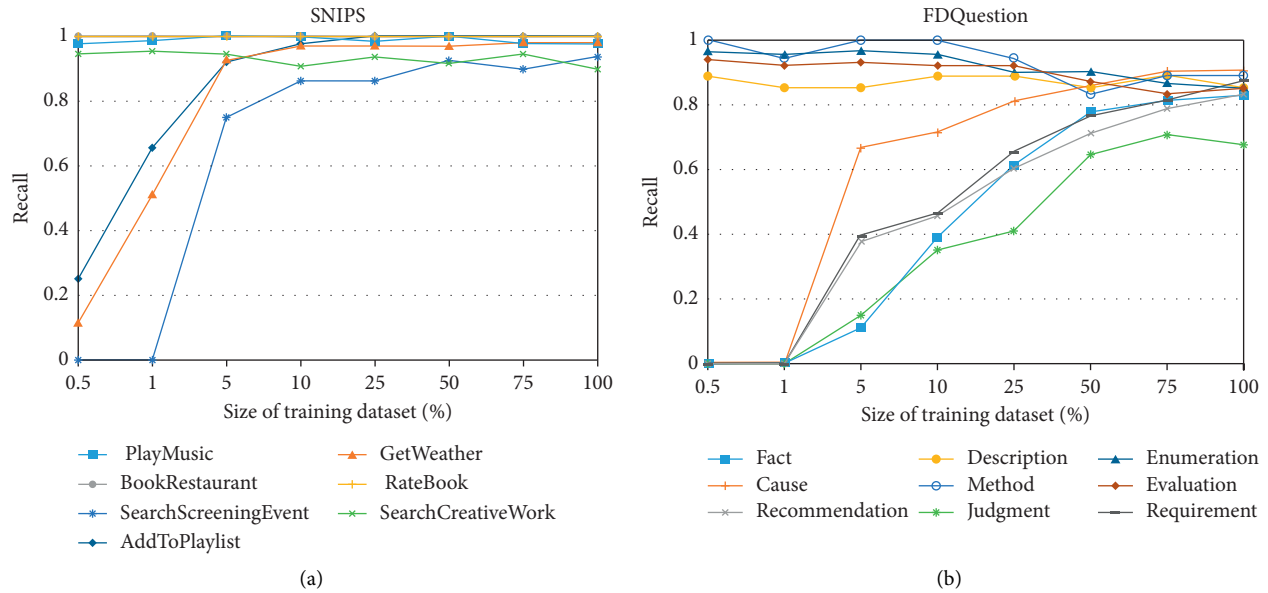

Figure 4: The change of the recall on SNIPS and FDQuestion. (a) SNIPS. (b) FDQuestion.

Table 6: The performance comparison of different methods using $F1$ score measure.

Methods	$F1$ score
CNN + domain template	0.899
Lib-SVM + n-gram	0.912
CNN + rules	0.926
CNN + ensemble learning	0.929
LSTM + domain dictionary	0.941
LR + character + POS tags + target words + semantic tags	0.945
BERT-Cap + focal loss	0.967

From Figure 4, the recall of these selected categories improved obviously with the percentage changing on the two datasets. When the percentage was less than 5%, the recall of these selected categories had improved dramatically. When these selected categories were low-data, the model could classify these samples into the categories with abundant samples. Therefore, the problem of intent categories had few-shot samples needed to be further studied in the future. We observed that the recall of these not-selected categories dropped as the samples of these selected categories increase on the FDQuestion in Figure 4(b). The main reason was that the classification boundaries between many categories on the FDQuestion were not obvious. With the addition of these categories samples, it was difficult for the model to accurately classify these samples. Finally, compared with FDQuestion, the recall values on these high-data categories were relatively stable on the SNIPS in Figure 4(a).

The $F1$ scores of the seven different methods are shown in Table 6. Our proposed model achieved a $F1$ score of 0.967, a 2.2% improvement compared with the baseline methods. Our model could obtain context-dependency sentence representation by using the pre-trained language model and the capsule networks captured key features during the process of dynamic routing. The result proved the effectiveness of our proposed model for improving the performance of user intent classification and solving the problem of uneven distribution of categories.

5. Conclusions

This paper proposed a hybrid model BERT-Cap using a pre-trained BERT to encode sentence sequence, applying capsule networks with dynamic routing mechanism to capture higher-level features, and combining a focal loss to improve the performance of user intent classification. Experimental results have demonstrated the performance improvement of our model compared with other baselines. In the future, we will try to introduce knowledge graphs to enhance sentence representation for improving the performance of user intent classification.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61772146), National Social Science Fund of China (AGA200016), Guangzhou Key Laboratory of Big Data and Intelligent Education (201905010009), and Katie Shu Sui Pui Charitable Trust-Research and Publication Fund (KS 2018/2.8).

References

[1] K. Jokinen and M. F. Mctear, "Spoken dialogue systems," *Synthesis Lectures on Human Language Technologies*, Morgan & Claypool Publishers, Bristol, UK, 2010.

[2] J. Liu, P. Pasupat, Y. Wang, S. Cyphers, and J. Glass, "Query understanding enhanced by hierarchical parsing structures," in *Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Olomouc, Czech Republic, 2013.

[3] T. Y. Hao, W. X. Xie, Q. Y. Wu, H. Weng, and Y. Y. Qu, "Leveraging question target word features through semantic relation expansion for answer type classification," *Knowledge-Based Systems*, vol. 133, pp. 43–52, 2017.

[4] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, 2014.

[5] P. Xu and R. Sarikaya, "Convolutional neural network based triangular CRF for joint intent detection and slot filling," in *Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Olomouc, Czech Republic, 2013.

[6] P. Liu, X. P. Qiu, and X. J. Huang, "Recurrent neural network for text classification with multi-task learning," 2016, https://arxiv.org/abs/1605.05101.

[7] G. E. Hinton, "Learning distributed representations of concepts," in *Proceedings of the 8th Annual Conference of the Cognitive Science Society*, Amherst, MA, USA, 1989.

[8] J. Devlin, M. W. Chang, K. Lee et al., "BERT: pre-training of deep bidirectional transformers for language understanding," 2018, https://arxiv.org/abs/1810.04805.

[9] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *NIPS*, Curran Associates Inc, Brooklyn, NY, USA, 2012.

[10] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association*, Chiba, Japan, 2010.

[11] S. Sabour, F. Nicholas, and G. E. Hinton, "Dynamic routing between capsules," 2017, https://arxiv.org/abs/1710.09829.

[12] M. Yang, W. Zhao, J. Ye et al., "Investigating capsule networks with dynamic routing for classification," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018.

[13] T. Y. Lin, P. Goyal, R. Girshick et al., "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 42, pp. 2999–3007, 2017.

[14] T. Y. Hao, W. X. Xie, and F. F. Xu, "A WordNet expansion-based approach for question targets identification and classification," *Lecture Notes in Artificial Intelligence*, vol. 9427, pp. 333–344, Springer, Berlin, Germany, 2015.

[15] W. X. Xie, D. F. Gao, and T. Y. Hao, "A feature extraction and expansion-based approach for question target identification and classification," *Lecture Notes in Computer Science*, vol. 10390, pp. 249–260, Springer, Berlin, Germany, 2017.

[16] J. Liu, Y. Yang, S. Lv et al., "Attention-based BiGRU-CNN for Chinese question classification," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, 2019.

[17] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, 2017.

[18] Y. Zhao, Y. Shen, and J. Yao, "Recurrent neural network for text classification with hierarchical multiscale dense connections," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence IJCAI-19*, Macao, China, 2019.

[19] S. V. Ravuri and A. Stolcke, "Recurrent neural network and LSTM models for lexical utterance classification," in *Proceedings of the 2015 IEEE Automatic Speech Recognition and Understanding Workshop*, Scottsdale, AZ, USA, 2015.

[20] P. Zhou, W. Shi, J. Tian et al., "Attention-based bidirectional Long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2016.

[21] C. He, S. Chen, S. Huang et al., "Using convolutional neural network with BERT for intent determination," in *Proceedings of the 2019 International Conference on Asian Language Processing (IALP)*, Shanghai, China, 2019.

[22] C. Xia, C. Zhang, X. Yan et al., "Zero-shot user intent detection via capsule neural networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018.

[23] T. E. Lin and H. Xu, "Deep unknown intent detection with margin loss," in *Proceedings of the 2019 Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019.

[24] I. Casanueva, T. Temčinas, D. Gerz, M. Henderson, and I. Vulić, "Efficient intent detection with dual sentence encoders," 2020, https://arxiv.org/abs/2003.04807.

[25] T. E. Lin, H. Xu, and H. Zhang, "Discovering new intents via constrained deep adaptive clustering with cluster refinement," 2019, https://arxiv.org/abs/1911.08891.

[26] G. Tur, D. Hakkani-Tur, and L. Heck, "What is left to be understood in ATIS?" in *Proceedings of the 2010 Spoken Language Technology Workshop*, Berkeley, CA, USA, 2010.

[27] A. Coucke, A. Saade, A. Ball et al., "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," 2018, https://arxiv.org/abs/1805.10190.

[28] D. Guo, G. Tur, W. Yih, and G. Zweig, "Joint semantic utterance classification and slot filling with recursive neural networks," in *Proceedings of the IEEE Spoken Language Technology Workshop*, South Lake Tahoe, NV, USA, 2014.

[29] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," in *Proceedings of the 2016 Automatic Speech Recognition and Understanding Workshop*, Okinawa, Japan, 2016.

[30] Q. Chen, Z. Zhuo, and W. Wang, "BERT for joint intent classification and slot filling," 2019, https://arxiv.org/abs/1902.10909.

[31] T. Khalil, K. Kielczewski, G. C. Chouliaras et al., "Cross-lingual intent classification in a low resource industrial setting," in *Proceedings of the 2019 International Joint Conference on Natural Language Processing*, Hong Kong, China, 2019.

[32] W. X. Xie, D. F. Gao, R. Y. Ding, and T. Y. Hao, "A feature-enriched method for user intent classification by leveraging

semantic tag expansion," in *Proceedings of the 2018 Natural Language Processing and Chinese Computing*, Hohhot, China, 2018.

[33] W.-N. Zhang, Z. Chen, W. Che, G. Hu, and T. Liu, "The first evaluation of Chinese human-computer dialogue technology," 2017, https://arxiv.org/abs/1709.10217.

[34] Y. Wu, M. Schuster, Z. Chen et al., "Google's neural machine translation system: bridging the gap between human and machine translation," 2016, https://arxiv.org/abs/1609.08144.

[35] Y. Cui, W. Che, T. Liu et al., "Pre-training with whole word masking for Chinese BERT," 2019, https://arxiv.org/abs/1906.08101.

[36] Y. Liu, M. Ott, N. Goyal et al., "RoBERTa: a robustly optimized BERT pretraining approach," 2019, https://arxiv.org/abs/1907.11692.