IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 12, 2018, accepted September 3, 2018, date of publication September 10, 2018, date of current version October 12, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2869585

Query Intent Recognition Based on Multi-Class Features

LIRONG QIU[1], YIDA CHEN[1], HAORAN JIA[2], AND ZHEN ZHANG[1]
[1]School of Information Engineering, Minzu University of China, Beijing 100081, China
[2]International School, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Lirong Qiu (qiu_lirong@126.com)

This work was supported in part by the National Nature Science Foundation of China under Grant 61672553, in part by the Project of Humanities and Social Sciences of the Ministry of Education of China under Project 16YJCZH076, and in part by the Open Project Fund of the Key Laboratory of Trustworthy Distributed Computing and Service, Ministry of Education, Beijing University of Posts and Telecommunications, under Grant 2018KF01.

ABSTRACT In order to enhance the user search experience of the search engine, an intent recognition search based on natural language input is proposed. By using reality mining technology to obtain the potential consciousness information from the query expression, search engines can better predict the query results that meet users' requirements. With the development of conventional machine learning and deep learning, it is possible to further improve the accuracy of prediction results. This paper adopts a similarity calculation method based on long short-term memory (LSTM) and a traditional machine learning method based on multi-feature extraction. It is found that entity features can significantly improve the accuracy of intention classification. Second, the accuracy of intention classification based on the feature sequence constructed by key entities is up to 94.16% in the field of manual labeling by using the BiLSTM classification model.

INDEX TERMS Intent recognition, multi-class, long short term memory (LSTM), reality mining, deep learning.

I. INTRODUCTION

The Internet has become an indispensable part of people's daily life. People's access to network information is getting larger and larger, and more and more frequent. At present, people mainly use search engines to query information, enter websites, and obtain network services. However, with the exponential growth of Internet resources and websites, it is almost impossible for users to find resources manually. Therefore, the user's experience of the search engine largely determines the user's internet experience. The traditional search based on keyword matching lacks the understanding of the user's intention, and the retrieval results are quite different from the user's requirements. In order to improve the quality of the results returned by the search engine and provide users with more accurate results, it is necessary to understand what the user submits, so as to return the user's query targets through different search strategies.

In the traditional search engine, the system uses similarity calculation to obtain the most relevant web pages as the query results. But the query results obtained through text similarity calculations are sometimes inaccurate. The user information, the target of the query, and the motivation of the query may be included in the query statement, but traditional search engines do not consider these features. Therefore, in order to make the return information more relevant to the user's requirement, it is necessary to be able to automatically identify the user's query intention,so that the search engine can grasp the user's demand more accurately. Query statement is a search statement submitted by the user to the search engine, which contains the user's query intention implicitly. The user's query intention can be obtained by parsing the natural language text. This is the work of query intention identification.

With the improvement of users' requirements on search accuracy, the search engine needs to identify the query more accurately and efficiently. Therefore, how to identify users' demands more intelligently, allocate users' demands to the optimal content resources or application providers for processing, and finally return accurate and efficient results to users is a new research hotspot. In order to identify intention more accurately and efficiently, obtain more efficient retrieval results, and meet the needs of users, this paper uses conventional machine learning and deep learning method to identify intention from the perspective of query expression. The main contributions of this paper are as follows:

1) This paper uses the traditional machine learning method based on multi-feature extraction to identify intention,Through multiple experiments, it is found that entity features can significantly improve the accuracy of intent classification. It lays a foundation for the application and research of future multi-feature in intention recognition.

2) This paper proposes an intention classification method using LSTM similarity and time sequence model to identify the intention of the query scenario for the character events. By comparing with the traditional SVM classification algorithm, it is found that the LSTM model can improve the performance of the model. Serializing the query text can further improve the performance of the model. This model can be applied to classification problems in intelligent retrieval and dialog fields, and more accurately and efficiently return matching results to users.

II. RELATED WORKS

Before 2002, the academic circle thinks that the core purpose of traditional information retrieval is that the internal information demand of users prompts them to adopt information retrieval system and generate retrieval behavior. Therefore, the information requirements contained in the user query are defined as information classes in a narrow sense. As Broder [1] proposed that user performing retrieval is not just about getting information and proposed a new classification standard, the research on query intention began to be paid attention by the academic community. When the query intention category system is clear, people begin to study the information retrieval technology based on the query intention recognition from many angles. Feature selection techniques based on query expressions is one of the important research directions.

Analysis of query expressions helps to identify users' query intentions. Duan *et al.* [2] divide intention into navigation category and non-navigation category, and thinks that verbs co-present with nouns can express their intention, and the dependency relationship between verbs and nouns can be used to identify subclasses in non-navigational queries. Truran *et al.* [3] believe that if the query expression contains the words "price", "purchase", "sale", the query has a commercial purpose. Chien and Immorlica [4] analyzed the time sensitivity of queries and found a common phenomenon that queries are always popular in a small period of time, especially for certain types of queries, such as news. Bang *et al.* [5] pointed out that if a query can appear with a time and space name, or it can appear separately, this query is time-sensitive and geographically sensitive. Lau believed that the length of the user's query represented the degree of attention paid to the information sought. The longer the query was, the more professional the information sought.

At the same time, the related technologies of machine learning and deep learning are also developing rapidly. Mikolov *et al.* [6] proposed two novel model architectures for computing continuous vector representations of words from very large data sets. These vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities. Neculoiu *et al.* [7] proposed a deep architecture for learning a similarity metric on variable length character sequences. Mueller and Thyagarajan [8] presented a siamese adaptation of the Long Short-Term Memory (LSTM) network for labeled data comprised of pairs of variable-length sequences. The model they proposed is applied to assess semantic similarity between sentences, and can be applied in neural network systems of greater complexity. Ye *et al.* [9] focused on the measurement of semantic similarity. They constructed a dataset containing 4,322 labeled question pairs in Chinese which was the first open Chinese dataset for question similarity classification. And they proposed a novel framework for measuring the semantic similarity between sentences based on the architecture of a recurrent neural network (RNN), which does not require lexical or syntactic resources. Li *et al.* [10] proposed a new Multi-Glimpse LSTM(MG-LSTM) network, in which multi-scale contextual information is sequentially integrated to promote the human detection performance.

Gradually, the researchers tried machine learning to identify the intention of the query. Baezayates *et al.* [11] classify query statements by using SVM and PLSA. Kathuria *et al.* [12] use k-means to classify the query intention. Figueroa and Atkinson [13] propose a new approach based on an ensemble of classifiers. The method combines syntactic and semantic features so as to effectively detect user intentions. The application of machine learning method to intention recognition has become a new research hotspot.

However, the machine learning model used in the existing consciousness recognition research is relatively simple. More features need to be extracted and utilized, and the classification accuracy needs to be improved. So this paper adopts a traditional machine learning method based on multi-feature extraction and multi-model fusion. And the paper proposes an intention classification method using LSTM similarity and time sequence model to identify the intention of the query scenario for the characters and events.

III. INTENTION CLASSIFICATION BASED ON MULTI-FEATURE FUSION

A. ENTITY EXTRACTION

The main task of entity extraction is to pick out words or phrases with specific meanings from the user's input text. Named entities can be divided into two categories according to their characteristics. One can be identified according to the rules of word formation and context, such as person name, place name and organization name. Another is a vertical field that can be exhaustive, such as movie title, fiction name, game name, etc. The paper is based on a domain-specific scenario and the entities in the query include Chinese person name, gender, age, ethnicity, Chinese address, time, time duration, train number, flight number, hotel, cyber bar, trip path, etc. In the paper, different identification methods are selected according to the characteristics of different entity dimensions, specifically as shown in Table 1.

TABLE 1. Entities information and extract methods.

Method	Demension	Name
CRF	Chinese Person Name	Person
	Chinese Address	Address
	Cyber bar	Cybercafe
	Hotel	Hotel
Regex	Identity	Identity
	Ethnicity	Ethnic
	Gender	Gender
	Airport	Airport
	Train Number	Trip
	Flight Number	Flight
	Relation	Relation
PCFG+ML	Identity	Identity
	Time	Time
	Time Duration	Duration
	Trip Path	Path
	Number	Number
	Age	Age

Conditional Random Field (CRF) is a conditional probability model for segmentation and sequences labeling [14]. It has a good effect on named entity recognition and The paper mainly uses CRF to identify person name, address names, hotel name and internet bar. RegEx uses regular expression methods to identify entities in text, suitable for identifying entities with limited values, such as ethnicity, gender, etc. Probabilistic Context Free Grammar (PCFG) can not only identify simple entities, but also complex composite entities, such as combinations between different types of entities [15]. Although entity extraction and standardization are critical to the accuracy of intent understanding and intent execution, they are not the focus of this paper. Thus, it won't go into details here.

B. INTENTION CLASSIFICATION BASED ON SIMILARITY CALCULATION

1) WORD VECTOR SIMILARITY CALCULATION

Word2Vec word vector embedding: Word vectors are very commonly used in NLP. Mikolov *et al.* [6] proposed to build a three-layer network structure with fewer hidden but more input and output layer nodes. This model uses the same context of different phrases to train the distributed representation of the phrase. The input is the One-Hot encoding of the phrase, and the output is the corresponding semantic representation.

In the process of similarity calculation, word vector and standard labeled data set are loaded firstly, and then vectoring the query text, the average of word vector in sentence is used as sentence vector representation in this paper. Secondly, similarity of the query statement and all intention corpora are calculated separately. The reciprocal of the average Euclidean distance between the target text and the annotated corpora is taken as the calculation method of similarity. The intention of largest similarity value is selected as the retrieval intention and then return to the user.

Algorithm 1 Base on Word2Vector SIMILARITY Calculation

Input: The User Query Text: sentence
Output: Query Intent: intent

$w2v \leftarrow load\ word2vec$
$intents, sentences \leftarrow load\ corpus$
$vector \leftarrow embedding(w2v, sentence)$
$similarities = \{\}$
for $intent$ **in** $intents$: **do**
  $matrix \leftarrow embedding(w2v, sentences[intent])$
  $distances \leftarrow Euclidean(matrix, vector)$
  $distance \leftarrow mean(distances)$
  $sim \leftarrow \frac{1}{1+distance}$
  $similarities \leftarrow (intent, sim)$
end for
$intent = max\_sim\_intent(similarities)$
return $intent$

2) SIMILARITY CALCULATION BASED ON LSTM

Sentence semantic model proposed by Mueller and Thyagarajan [8] is different from the word2vec model. The Model can be used directly to train the semantic vector features of sentences. The main idea is to train the LSTM recurrent neural network model by pre-labeling the corpus [17], and transform the sentence into vector representation by the model. Firstly, the sentence phrase is encoded, and the encoded sequence is used as the input of the model. The final output of the model is used as the sentence vector representation. Secondly, the average similarity of the query statement and the annotated corpus under various intents is calculated respectively, and the intention with the maximum similarity is selected as the query intention to return to the user. The algorithm is shown below in Algorithm 2.

Algorithm 2 Base on LSTM Similarity Calculation

Input: The User Query Text: sentence
Output: Query Intent: intent

$lstm \leftarrow load\ lstmRNN$
$intents, sentences \leftarrow load\ corpus$
$vector \leftarrow lstm\_embedding(lstm, sentence)(w2v, sentence)$
$similarities = \{\}$
for $intent$ **in** $intents$: **do**
  $matrix \leftarrow lstm_{e}mbedding(lstm, sentences[intent])$
  $ys \leftarrow exp^{\|matrix-vector\|}$
  $sim \leftarrow mean(ys)$
  $similarities \leftarrow (intent, sim)$
end for
$intent = max\_sim\_intent(similarities)$
return $intent$

C. CLASSIFIER ENSEMBLES FOR INTENT CLASSIFICATION

The single classifier has poor generalization ability, and it is easy to achieve better accuracy in the training set, but perform poorly in the test data set.Therefore, further classification based on the output results of multiple classifiers can greatly reduce the generalization risk of a single classifier.

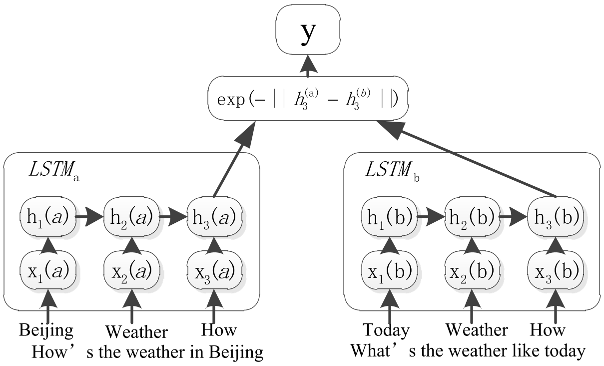


FIGURE 1. Based on LSTM similarity calculation.

If the selected model can not correctly represent the decision boundary, diversified model fusion can be considered, and the appropriate classifier combination can solve the complex data category decision boundary problem.

Effective model fusion requires each weak classifier to show various diversity, and different classifiers can generate different decision boundaries.If you choose the classifiers with some differences, each classifier will generate independent errors, and the total error will usually be reduced by combining these classifiers. A common framework for model fusion based on specific application areas is shown in Figure 2. Each classifier trains in different training subsets, which produces different errors, but the combined classifier can provide the best decision boundaries.

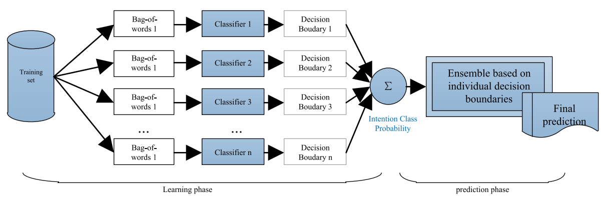


FIGURE 2. Based on LSTM similarity calculation.

Brown *et al.* [18] proposed three methods for constructing classifier differences in fusion models, which is changing the starting point in the hypothesis space, changing the training set of the weak classifier and changing basic classifier models or different fusion strategies. Rodriguez-Penagos *et al.* [19] focused on the analysis of expression level, so they used different training sets to construct differences for different basic classifiers.

In practice, we need to build a classifier to classify test data. In our research, once the basic classifier model is trained, the intention will be predicted through the average probability or the major voting.

1) RANDOM FOREST

Random forest [20] is to establish a forest in a random way. There are multiple decision tree classifiers in the forest, and each decision tree is not correlated, and its output category is

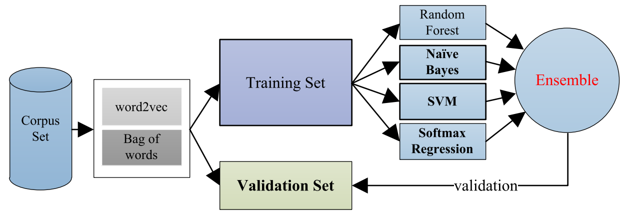


FIGURE 3. Schematic diagram of model fusion framework.

determined by the number of categories output by individual tree. The method to construct trees is as follows.

1. Use N to indicate the number of training sets, and M to indicate the number of features.
2. Enter the number of features m to determine the decision result of the previous node on the decision tree, and m should be much smaller than M.
3. Samples were taken N times from the N training samples by a sampling with replacement method to form a training set, and the unsampled samples were used to estimate the error.
4. For each node, m features are randomly selected, and the decision of each node on the decision tree is determined based on these features. According to these features, the best split mode is calculated.
5. Each tree grows intact without pruning.

2) SVM

Support Vector Machine (SVM) is a pattern recognition method based on statistical learning theory proposed by Vapnik [21] in 1963. SVM maps training set texts to high-dimensional spaces through nonlinear mapping. This mapping process transforms the linear non-separable problem in training sets into linear separable problems in high-dimensional space. SVM is a classification model for supervised learning, which judges the category according to the optimal classification hyperplane H. The optimal classification hyperplane can separate the data in the sample data set and make the distance between the data on the support vector to the hyperplane maximum.

Assuming that a training set $D = \{(x_i, y_i)\}$ is given, where x_i is a sample point, and y_i is the category label of x_i, $y_i \in \{-1, +1\}$, then the optimal hyperplane is defined as

$$H : w \cdot x + b = 0 \tag{1}$$

If the training data is linearly separable, then there are two parallel hyperplanes H_1 and H_2 that separate the two types of data and make the distance between the different categories of data as large as possible.

$$\begin{cases} H_1 : w \cdot x + b \geq 1 \\ H_2 : w \cdot x + b \leq -1 \end{cases} \tag{2}$$

The support vector machine model is based on the H_1 and H_2 distance to maximize the distance $\frac{2}{\|w\|^2}$ of these two hyperplanes, that is to minimize the distance $\frac{1}{2}\|w\|^2$ by finding the values of the model parameter w and the offset b.

$$\begin{cases} min\ \frac{1}{2}\|w\|^2 \\ s.t.\ y_i(w \cdot x + b) \geq 1 \quad (i = 1, 2, ..., n) \end{cases} \tag{3}$$

Support vector machines (SVM) can not only be applied to linear classification but also can solve the nonlinear classification by kernel function to conduct a higher dimensional space mapping. The model classification is determined by the support vector, so it has certain robustness and higher performance in the classification of small samples [22]. Therefore, the support vector machine model can solve the problem of multiple classification.

3) NAIVE BAYES

Naive Bayes classification model is a probabilistic classification model based on Bayes principle, which uses prior knowledge of sample data to predict the category of unknown samples. It uses the probability model to calculate the category probability of unknown samples and determine the type of samples by constructing the probability model of training sample feature. The naive Bayesian classification model sets the sample features independently of each other [21], and it is this setting that greatly improves the classification efficiency of naive Bayes.

Assuming that D is the sample set, one of the samples is $d = \{t_1, t_2, ..., t_n\}$ and the categories set is $C = \{c_1, c_2, ..., c_k\}$, we can calculate the probability that the sample d belongs to the category c_i according to the Bayes principle.

$$p(c_j|d) = \frac{p(c_j)p(d|c_j)}{p(d)} \tag{4}$$

In the formula, $p(c_j)$ represents the proportion of samples that belong to c_j in the sample. $p(d)$ is the probability of occurrence of sample d in the sample. $p(d|c_j)$ represents the conditional probability of the sample,and the formula is as follows.

$$p(d|c_j) = p(t_1, t_2, ...t_n|c_j) \prod_{i=1}^{N} p(t_i|t_1, t_2, ...t_n, c_j) \tag{5}$$

Because the feature items in Naive Bayes are independent of each other, it can be inferred:

$$p(t_i|t_1, t_2, ...t_n, c_j) = p(t_i|c_j) \tag{6}$$

$$p(d|c_j) = \prod_{i=1}^{N} p(t_i|c_j) \tag{7}$$

Formula (6) and (7) can be used to obtain the probability formula of text belonging to the category.

$$p(c_j|d) = p(c_j) \prod_{i=1}^{N} p(t_i|c_j) \tag{8}$$

When predicting the intent category of the search statement, the probability of each category is calculated separately, and the category with the highest probability is the intent category to which the search statement belongs.

$$d \in maxp(c_j) \prod_{i=1}^{N} p(t_i|c_j),\ c_j \in C \tag{9}$$

4) SOFTMAX REGRESSION

The Softmax regression model is a generalization of logistic regression models on multi-classification problems. In multi-classification problems, class labels y can take more than two values. In logistic regression, the training set consists of m labeled samples: $\{(x^{(1)}, (y^{(1)}), ..., (x^{(m)}, (y^{(m)})\}$, where the input feature $x^{(i)} \in \Re^{(n+1)}$. Since logistic regression solves the problem of two classifications, the class label $y^{(i)} \in \{0, 1\}$. The hypothesis function is as follows.

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \tag{10}$$

Training the model parameter θ to minimize the loss function:

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))] \tag{11}$$

Softmax regression solves multi-classification problems. Class label y can take k different values. For the given test input x, the hypothesis function is used to estimate the probability value p for each category j, that is, the probability of x appearing in each category. Therefore, our hypothesized function is going to output a vector of k dimensions (the sum of vector elements is 1) to represent the probability values of k estimates. Therefore, our hypothesized function is going to output a vector of k dimensions (the sum of vector elements is 1) to represent the probability values of these k estimates. The function is as follows.

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1|x^{(i)}; \theta) \\ p(y^{(i)} = 2|x^{(i)}; \theta) \\ \cdots \\ p(y^{(i)} = k|x^{(i)}; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \cdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix} \tag{12}$$

Among them, $\theta_1, \theta_1, ...\theta_k \in \Re^{n+1}$ are the parameters of the model, and $\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}$ is used to normalize the probability distribution.When implementing Softmax regression, use the matrix of $k \times (n+1)$ to simplify the θ representation, which is represented by the matrix θ as follow:

$$\theta = \begin{bmatrix} \theta_1^T \\ \theta_2^T \\ \cdots \\ \theta_k^T \end{bmatrix} \tag{13}$$

The cost function of the Softmax regression algorithm is as follows.

$$\begin{aligned} J(\theta) &= -\sum_{i=1}^{m} \sum_{c=1}^{k} sign(y^{(i)} = c) \log p(y(i) = c|x^{(i)}; \theta) \\ &= -\sum_{i=1}^{m} \sum_{c=1}^{k} sign(y^{(i)} = c) \log \frac{e^{\theta_c^T x^{(i)}}}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \end{aligned} \tag{14}$$

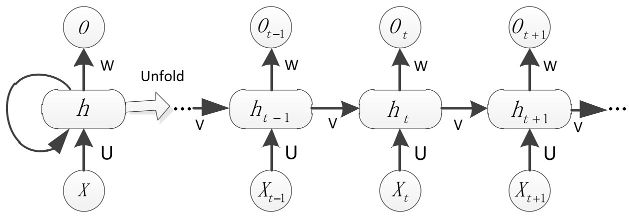


FIGURE 4. The basic recurrent neural network.

It can be seen that the Softmax cost function is very similar in form to the logistic cost function, except that the k possible values of the class flags may be accumulated in the Softmax loss function. The probability that Softmax classifies x as category j is

$$p(y^{(i)} = j|x^{(i)}; \theta) = \frac{e^{\theta_c^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}} \tag{15}$$

For the problem of $J(\theta)$ minimization, this paper uses the iterative gradient descent optimization algorithm. After derivation, the gradient formula can be obtained as follows,and in the equation, $1(\cdot)$ is the "indicator function", so that 1(a true statement)=1, and 1(a false statement)=0. For example, 1(2+2=4) evaluates to 1; whereas 1(1+1=5) evaluates to 0.

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[x^{(i)}(1\{y(i)=j\} - p(y^{(i)}=j|x^{(i)};\theta))] \tag{16}$$

Use partial derivative formula to iteratively update $\theta_j := \theta_j - \alpha\nabla_{\theta_j} J(\theta), (j = 1, 2, ..., k)$, then put the θ into gradient descent algorithm and obtain the maximize $J(\theta)$ by multiple iterations.

D. INTENTION CLASSIFICATION BASED ON LSTM

1) RECURSIVE NEURAL NETWORKS

Recursive neural networks are increasingly applied to classification tasks. For sequence modeling tasks such as intent recognition, the key is to capture long distance information from the sentence. The following figure6 is a typical structure of RNN, where x_t stands for the input of time step t, h_t for a hidden state, and the output of the previous layer h_{t-1} in hidden layer is involved in the calculation of this layer.

$$h_t = f(W_x x_t + W_h h_{t-1} + b_n) \tag{17}$$

Theoretically,the RNN neural network model can capture dependence of any text length, but when the length is too long, it is easy to cause the network gradient to disappear or explode, which limits the length of the accessible context. Secondly, the hidden layer of the RNN model and the processing logic of the current input are too simple to describe the complex information in the sequence. Not all the information of the neural units in RNN network nodes has effect on the model, and the model's effect can be improved by choosing to retain the positive information and forget the useless. Given the problems above, the paper adopts long and short term memory (LSTM) as the main training model [17]. As a variant of RNN, this model has the characteristics of maintaining long-distance dependence. LSTM uses a gating mechanism to control information, and uses a forgotten gate to select the unimportant information to forget. There are many variants of the LSTM neural unit and the most widely used neuron structure is used here. As is shown in the following figure5, i_t stands for the input gate, f_t for the forgotten gate, o_t for the output gate, c_t for the storage cell, h_t for the hidden state, t for the time step.

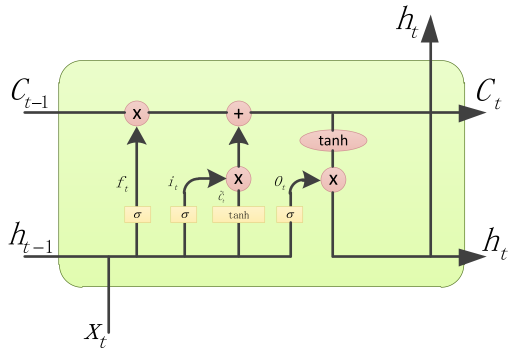


FIGURE 5. The LSTM unit.

$$i_t = \sigma(W^{(i)}X_t + U^{(i)}h_{t-1} + b^{(i)}) \tag{18}$$
$$f_t = \sigma(W^{(f)}X_t + U^{(f)}h_{t-1} + b^{(f)}) \tag{19}$$
$$o_t = \sigma(W^{(o)}X_t + U^{(o)}h_{t-1} + b^{(o)}) \tag{20}$$
$$u_t = \tanh(W^{(u)}X_t + U^{(u)}h_{t-1} + b^{(u)}) \tag{21}$$
$$c_t = i_t \odot u_t + f_t \odot c_{t-1} \tag{22}$$
$$h_t = o_t \odot \tanh c_t \tag{23}$$

Among them, $X_t \in R^d$ and $h_t \in R^d$ are respectively the input and output vector of the time step t. $W^{(k)}$, $U^{(k)}$ $(k = z, i.f, o)$ are respectively the weight of the input part and the weight of the gating in the multiple loop structure. $b^{(k)}(k = z, i.f, o)$ is the bias vector, function σ is a nonlinear function, such as sigmoid or tanh, and $\odot$ is Element-by-element multiplication.

2) BI-DIRECTIONAL LSTM

BLSTM neural network can not only capture the hidden sequence information of training data, but also maintain the past and future context features of the sequence [23]. Different from the LSTM network, the BLSTM network has two neural network layers with different transmission directions. Each layer is trained in forward and backward directions in manner of conventional neural network delivery. And the model can memorize the sequence information in both directions.

$$h_{(ft)} = H(W^{(h_f)}X_t + U^{(h_f)}h_{f(t-1)} + b^{(h_f)}) \tag{24}$$
$$h_{(bt)} = H(W^{(h_b)}X_t + U^{(h_b)}h_{b(t-1)} + b^{(h_b)}) \tag{25}$$

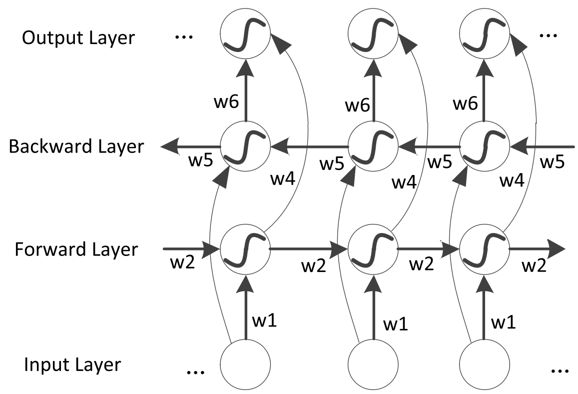


FIGURE 6. The net structure of the BiLSTM.

$h_f \in R^d$ and $h_b \in R^d$ represent the final output vector of the forward layer and the backward layer respectively, where the two outputs are combined as the final output $y_t = [h_{ft}, h_{bt}]$, $y_t \in R^{2d}$.

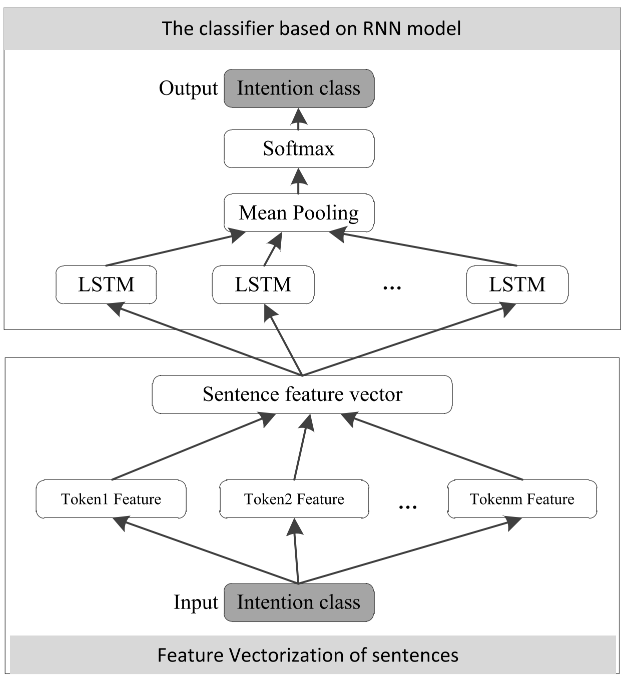


FIGURE 7. The flow of intent classification base on LSTM.

E. MODEL TRAINING

The specific processing flow is shown in Figure 7. Firstly, the tokenized sentence is obtained through preprocessing. In the paper, tokenization includes word segmentation and entity serialization, in which the word segmentation mainly uses Jieba word segmentation. Based on the word segmentation, entity serialization replaces the identified entity text with the dimension name by the entity recognition processing mentioned above, thus a set of both part of phrase and entity sequence is obtained. Secondly, the corresponding feature vectors are obtained through the Embedding layer, and the Token vectors of these sequences are used as time step inputs of the LSTM cyclic neural network. Finally, the output vector of each neural node is performed with mean pooling, and then the intention classification result is obtained through linear variation of the Softmax layer. The paper regards intention recognition as a multi-classification task, uses the cross-entropy of the model output value and the standard labeled value as the loss function [24], and uses the Adadelta adaptive learning rate and the Mini-batch Gradient Descent training model to minimize the loss function of the model through multiple epochs. During the training process, the performance of the model is verified and the parameters are saved for every 10 batches of training data. When the performance of the model is not improved after multiple epochs, the model training is terminated and best parameters are produced. The model training process is shown below.

To construct the LSTM (BiLSTM) cyclic neural network structure as shown in Figure 7, firstly, the input data processing of the model is required, and the Tokens of corpus sequence need to be performed one-hot encoding as the initial input of the model. Secondly, the output and target loss function of the network model are constructed. The output of the model will perform by the Softmax linear change processing after the LSTM giving the output. The target function uses the probability output of the model and the cross entropy of the one-hot annotated value of the corpus. Finally, the Adadelta algorithm is used to optimize the target function. After multiple iterations, the loss value can be minimized and the model performance tends to be stable.

IV. EXPERIMENT AND RESULT ANALYSIS

The query data corpus used in this paper is mainly from the query requirements of a specific domain. The domain query involves finding people based on character attributes and event attributes like finding person information based on train flight events, hotel accommodation events, Internet cafe events, and mobile phone numbers. The character attributes mainly include id card number, mobile phone number, name, gender, age, nationality and height, among which id card and mobile phone number are the main attributes to determine the people that need to be searched, while other attributes are used as query constraints. According to the query requirements of the business, the first seven intents are set as follows and considering that users would input irrelevant query contents, the unknown_type intention is added to identify those intentions that could not be clearly understood.

A. EXPERIMENT SETTINGS

Experiments are conducted using a real-world scene dataset in this paper. The dataset involves character attributes and event retrieval, and a total of 6680 sentences and 8 types of tags are obtained by label processing. The dataset is randomly divided into a training set for 80%, a validation set for 10%, and a test set for 10%. The parameters of SVM and LSTM models are trained on the training set, and the

Algorithm 3 The Training of The LSTM Model

Input: Labeled Corpus: corpus

Output: LSTM Model: lstm_model

$intents, sentences \leftarrow load\ corpus$

$dictionary \leftarrow sentences$

for $sentence$ **in** $sentences$: **do**

  $X \leftarrow embedding(dictionary, sentence)$

  $y \leftarrow tocategorical(intents)$

end for

$model \leftarrow build\_lstm(hidden\_size, layer\_size, batch\_size)$

$out, logits \leftarrow build\_output(lstm\_output, in\_size, out\_size)$

$loss \leftarrow \frac{1}{1+e^{label_i+\ln logits_i+(1-label_i)*\ln(1-logits_i)}}$

$optimizer \leftarrow AdadeltaOptimizer(loss, learning_rate)$

for $epoch$ **in** $epochs$: **do**

  for $batch\_x, batch\_y$ **in** $get\_batch(X, y, batch\_size)$ **do**

    $model.run(loss, feed = input : batch\_x, target : batch\_y)$

    $model.run(optimizer, feed = input : batch\_x, target : batch\_y)$

  end for

end for

return $model$

TABLE 2. The demonstrate of intentions.

Intent name	Example
person_attribute	Zhang's telephone number
person_by_train_event	The person who takes the train to Beijing in March 2017
person_by_hotel_event	A Man who stayed at the seven-day hotel in August
person_by_internetcafe_event	A person named Zhang who has an online record in the Internet cafe
person_by_flight_event	The man who took the flight in February of CA281
vehicle_attribute	Find a car with the Jing B12345 license plate
phoneNO_attribute	The person whose phone number is 185180xxxx
unknown_type	

performance of the model is verified on the test set after every update iteration when training the model. If the accuracy of the model remains unchanged on the validation set in 10 training iterations, the current model parameters will be saved. In the experiment, SVM adopts Sklearn open source library, LSTM adopts TensorFlow open source framework, and word2vec word vector are use as feature representation. Word vectors are obtained by 40G news text corpus training, and random vectors are used to represent non-existent word vectors.

B. EVALUATION INDEX

The data used in this experiment are all labeled data, so the accuracy, recall rate and F1 values are used as the experimental evaluation indexes. The accuracy refers to the ratio of the user number with the correct group division results in the total in the entire experimental user sample. The recall rate refers to the ratio of the user number that has the correct group division result in the same type of users. The F value is the harmonic average of the accuracy and the recall rate. The specific formula is as follows:

$$Accuracy = \frac{\sum T_i}{N} \tag{26}$$

$$Recall = \frac{\sum T_i}{\sum(T_i + F_i)} \tag{27}$$

$$F1 = \frac{2 \times Accuracy \times Recall}{Accuracy + Recall} \tag{28}$$

Among them, T_i refers to the number of the correct divisions, $F1$ refers to the number of wrong divisions, and N is the total number of samples in the experiment.

C. EXPERIMENTAL RESULTS AND ANALYSIS

1) EXPERIMENTAL RESULTS OF INTENTION CLASSIFICATION BASED ON SIMILARITY

The performance of the intention classification is evaluated on the Word2Vec word vector and the LSTM model. As shown in Figure 8, the average accuracy of the word vector model in this test set is 70%, while LSTM similarity calculation is 78.5%.

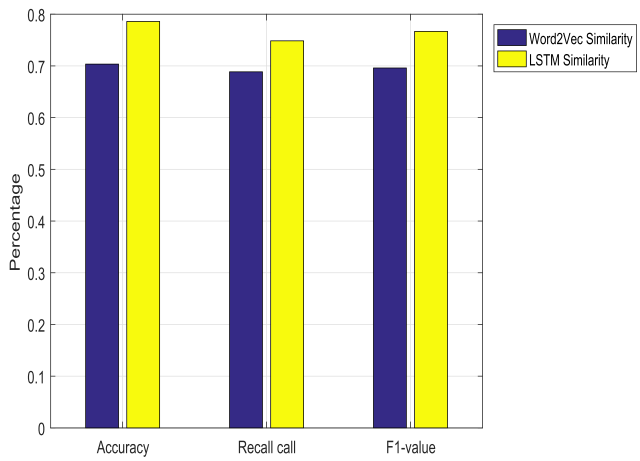


FIGURE 8. Comparison of the Word2Vec and LSTM similarity.

The effect of the word vector model is very dependent on the training corpus so that the accuracy of the word vector based on the news corpus training is lower when participate in similarity calculation directly. While the LSTM output vector is based on the corpus learning and can fully learn the implicit semantics of various query statements, so the accuracy of the intention recognition is relatively high. However, Since the similarity calculation is required for each statement in the standard set when calculating the similarity, the execution efficiency remains low and the accuracy is also deviated.

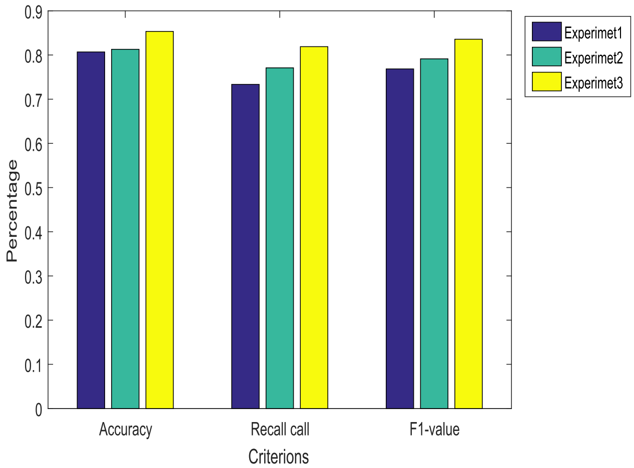


FIGURE 9. Comparison of the Word2Vec and LSTM similarity.

2) EXPERIMENTAL RESULTS OF INTENTION CLASSIFICATION BASED ON CLASSIFIER ENSEMBLES

In the paper, the fusion model is used for intent recognition. The main features are Bag of words. The following is experimental effects of each model and ensemble classifier. As shown in Table 3, the ensemble classifier can significantly improve the classification results.

TABLE 3. Comparative experimental results of each model and ensemble classifier.

Experiment	Accuracy(%)	Recall(%)	F1 value(%)
Random Forest	77.54	73.20	75.31
Naive Bayes	77.84	94.16	85.23
SVM	78.29	79.34	78.81
Softmax Regression	78.59	80.84	79.70
Classifier ensemble	84.58	92.06	88.16

By comparing the five experimental results above, each independent model has its own advantages and own differences, for example, Naive Bayes has a low accuracy and a high recall rate, while SVM and Softmax Regression has a high accuracy and low recall rate.Inadequacies of the models were reduced by ensembling model. It can be seen that the accuracy rate of the ensemble training model can reach 84.58% and the recall reaches 92.06% which inherits accuracy of the Softmax Regression and highly recall of Naive Bayes.

3) EXPERIMENTAL RESULTS OF INTENTION CLASSIFICATION BASED ON SVM

In the paper, the SVM model is used for intent recognition. The main features are word vector, stop word and entity dimension. The following experiment is performed with summing the phrase vectors in the sentence (Experiment 1) and taking the average as the sentence feature representation(Experiment 2), and adding extra entity dimensions as the extended dimensions(Experiment 3).

By comparing the three experimental results above, it can be seen that the accuracy rate of the feature training model

TABLE 4. The experiments based on LSTM.

Experiment	Model
1	Word vector + LSTM
2	Word vector + BiLSTM
3	Entity sequence vector +LSTM
4	Entity sequence vector +BiLSTM

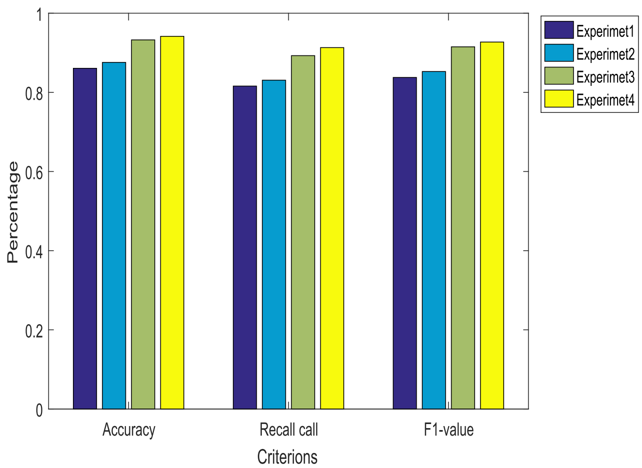


FIGURE 10. Comparison of the Word2Vec and LSTM similarity.

can reach 80.69% through the word vector only. The improvement effect of the model is not significant after removing the stop words because they are less used in user retrieval input, while by adding entity dimension features, the classification effect is improved, which indicates the entities in the retrieval statement are helpful to understand the query intention.

4) EXPERIMENTAL RESULTS OF INTENTION CLASSIFICATION BASED ON TIME SERIES MODEL

Considering the influence of the ordered sequence of the text's input on the intention expression, The LSTM and Bi-LSTM intention recognition methods are proposed. The word vector and entity sequence are respectively taken as the input of the time sequence model, in which the entity sequence is the input sequence formed by replacing the entity text with the entity dimension name and uses self-encoding to encode Token in the sequence and acts as the input to the time sequence model. For example, after entity serialization, "An Uyghur male who checked into the Hanting hotel in August this year" transforms to "Ethnic Gender who checked into the Hotel Time" where the bold texts represent the entity dimension.

Compared with the above experiments, it is found that the time sequence model can greatly improve the effect of the model, especially after entity serialization because the entity serialization can eliminate the difference of entity values and extract the common features of the query. By analyzing and identifying the wrong sentences, it can be found that there is entity misidentification in the sentence such as misidentifying the "Seven Days" of "the Seven Days Hotel" as a time dimension. Besides, there is lack of key entity information.

For example, after serialization, "Searching for men who surf the Internet in 2017" changes into "Searching for Gender who surf the Internet Time", which lacks the key entity dimension information of the Cyber bar.

V. CONCLUSION

The paper proposes an intention classification method using LSTM similarity and time sequence model to recognize the intention of the query scenario for the character events. Compared with the traditional SVM classification algorithm, the LSTM model can improve the performance of the model, and the model performance can be further improved after query text serialization. It is hoped that the algorithm model and the accuracy of entity recognition can be further improved in the future and it can be applied to classification problems in the field of intelligent retrieval and dialog.

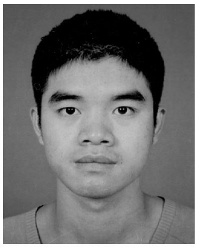
REFERENCES

[1] A. Broder, "A taxonomy of Web search," *ACM SIGIR Forum*, vol. 36, no. 2, pp. 3–10, 2002.
[2] R. Duan *et al.*, "Dependency relation based detection of lexicalized user goals," in *Proc. Int. Conf. Ubiquitous Intell. Comput.*, vol. 6406. Springer-Verlag, 2010, pp. 167–178.
[3] M. Truran, J.-F. Schmakeit, and H. Ashman, "The effect of user intent on the stability of search engine results," *J. Assoc. Inf. Sci. Technol.*, vol. 62, no. 7, pp. 1276–1287, 2011.
[4] S. Chien and N. Immorlica, "Semantic similarity between search engine queries using temporal correlation," in *Proc. Int. Conf. World Wide Web*, 2005, pp. 2–11.
[5] V. N. Bang and M.-Y. Kan, "Functional faceted Web query analysis," in *Proc. Workshop 16th Int. Conf. World Wide Web Conf. (WWW)*, 2007.
[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. ICLR Workshop*, 2013.
[7] P. Neculoiu, M. Versteegh, and M. Rotaru, "Learning text similarity with siamese recurrent networks," in *Proc. 1st Workshop Represent. Learn. NLP*, 2016, pp. 148–157.
[8] J. Mueller and A. Thyagarajan "Siamese recurrent architectures for learning sentence similarity," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2786–2792.
[9] B. Ye, G. Feng, A. Cui, and M. Li, "Learning question similarity with recurrent neural networks," in *Proc. IEEE Int. Conf. Big Knowl.*, Aug. 2017, pp. 111–118.
[10] H. Li, J. Liu, G. Zhang, Y. Gao, and Y. Wu. (Nov. 2017). "Multi-glimpse LSTM with color-depth feature fusion for human detection." [Online]. Available: https://arxiv.org/abs/1711.01062
[11] R. Baeza-Yates, L. Calderón-Benavides, and C. González-Caro, "The intention behind Web queries," in *Proc. Int. Conf. String Process. Inf. Retr.*, vol. 4209. Springer-Verlag, 2006, pp. 98–109.
[12] A. Kathuria, B. J. Jansen, C. Hafernik, and A. Spink, "Classifying the user intent of Web queries using *k*-means clustering," *Internet Res.*, vol. 20, no. 5, pp. 563–581, 2010.
[13] A. Figueroa and J. Atkinson, "Ensembling classifiers for detecting user intentions behind Web queries," *IEEE Internet Comput.*, vol. 20, no. 2, pp. 8–16, Mar./Apr. 2016.
[14] J. D. Lafferty, A. Mccallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Mach. Learn.* San Mateo, CA, USA: Morgan Kaufmann, 2001, pp. 282–289.
[15] C. D. Manning and H. Schütze, "Probabilistic context-free grammar," in *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999, ch. 11.
[16] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *Proc. AAAI*, 2016, pp. 2786–2792.
[17] A. Graves, *Supervised Sequence Labelling With Recurrent Neural Networks*. Berlin, Germany: Springer, 2012, pp. 1735–1780.
[18] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: A survey and categorisation," *Inf. Fusion*, vol. 6, pp. 5–20, Mar. 2005.
[19] C. Rodríguez-Penagos *et al.*, "FBM: Combining lexicon-based ML and heuristics for social media polarities," in *Proc. 4th Int. Workshop Semantic Eval.*, Atlanta, Georgia, 2013, pp. 483–489.
[20] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[21] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 2000.
[22] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul./Aug. 2008.
[23] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
[24] L.-Y. Deng, "The cross-entropy method: A unified approach to combinatorial optimization, Monte–Carlo simulation, and machine learning," *Technometrics*, vol. 48, no. 1, pp. 147–148, 2006.

LIRONG QIU received the Ph.D. degree in computer sciences from the Chinese Academy of Sciences in 2007. She is currently an Associate Professor of computer sciences with the Information Engineering Department, Minzu University of China. Her current research interests include different aspects of natural language processing, artificial intelligence, and distributed systems.

YIDA CHEN is currently pursuing the master's degree with the School of Information Engineering, Minzu University of China. His current research interests include machine learning, big data analysis, and artificial intelligence.

HAORAN JIA is currently pursuing the bachelor's degree with the Beijing University of Posts and Telecommunications. Her current research interests include machine learning and big data analysis.



ZHEN ZHANG is currently pursuing the master's degree with School of Information Engineering, Minzu University of China. His current research interests include natural language processing, artificial intelligence, and sentiment analysis.