Copyright ⓒ 2020 KSII

# Chatbot Design Method Using Hybrid Word Vector Expression Model Based on Real Telemarketing Data

**Jie Zhang[1], Jianing Zhang[1], Shuhao Ma[1], Jie Yang[1], and Guan Gui[1*]**

[1] College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China.

Corresponding author: Guan Gui (guiguan@njupt.edu.cn)

*Received July 17, 2019; accepted October 5, 2019; published April 30, 2020*

## *Abstract*

In the development of commercial promotion, chatbot is known as one of significant skill by application of natural language processing (NLP). Conventional design methods are using bag-of-words model (BOW) alone based on Google database and other online corpus. For one thing, in the bag-of-words model, the vectors are Irrelevant to one another. Even though this method is friendly to discrete features, it is not conducive to the machine to understand continuous statements due to the loss of the connection between words in the encoded word vector. For other thing, existing methods are used to test in state-of-the-art online corpus but it is hard to apply in real applications such as telemarketing data. In this paper, we propose an improved chatbot design way using hybrid bag-of-words model and skip-gram model based on the real telemarketing data. Specifically, we first collect the real data in the telemarketing field and perform data cleaning and data classification on the constructed corpus. Second, the word representation is adopted hybrid bag-of-words model and skip-gram model. The skip-gram model maps synonyms in the vicinity of vector space. The correlation between words is expressed, so the amount of information contained in the word vector is increased, making up for the shortcomings caused by using bag-of-words model alone. Third, we use the term frequency-inverse document frequency (TF-IDF) weighting method to improve the weight of key words, then output the final word expression. At last, the answer is produced using hybrid retrieval model and generate model. The retrieval model can accurately answer questions in the field. The generate model can supplement the question of answering the open domain, in which the answer to the final reply is completed by long-short term memory (LSTM) training and prediction. Experimental results show which the hybrid word vector expression model can improve the accuracy of the response and the whole system can communicate with humans.

***Keywords:*** chatbot, natural language processing (NLP), skip-gram, long-short term memory (LSTM), term frequency-inverse document frequency (TF-IDF).

**http://doi.org/10.3837/tiis.2020.04.001** **ISSN : 1976-7277**

## 1. Introduction

The enhance in acceptance of artificial intelligence (AI) has brought about needs for business services which assist people to bypass some useless data [1]. One of the technology is natural language processing (NLP) [2][3], which develops into chatbot, an important element of human-computer interaction [4][5]. Chatbot is recognized as a key element of the human-computer interaction [6]. This technology has been extensively used in smart hardware, smart home, intelligent robots and other fields. Some famous companies such as Amazon, Google and Apple are all focusing on the study of voice interaction. Other well-known companies have proposed a series of products related to speech recognition technology [7]. In China, chatbot has become one of the mainstream trends in social development, and it has increasingly become the largest artificial intelligence system for the recipients. Due to its great potential, many entrepreneurial teams have emerged in recent years to research and develop voice interaction technologies [8].

The earliest research work in natural language understanding was machine translation. In 1946, when British engineer A. D. Booth and American W. Weaver discussed the scope of application of electronic computers, they proposed the idea of using computers for automatic language translation. In 1949, W. Weaver first proposed a machine translation design scheme. Many countries had a large-scale research work in the aspect of machine translation [9]. However, the complexity of natural language was apparently underestimated. These research do with the so-called rule-based approach. But a large number of grammar rules and dictionary entries need to be compiled. It is too difficult for developing an actual system, little progress in these studies.

In the 1960s, the intelligent question answering system appeared. Bert F. Green, Jr. et al. proposed a computer program called Baseball to answer questions about storing data in ordinary English. In the early 1970s, a method for machines to adapt to ordinary natural English conventions was put forward by W. A. Woods et al., which can deal with the human-computer communication problems. This approach no longer requires people to fit in with the machine [10]. Massachusetts Institute of Technology has also presented a program called ELIZA that makes natural language dialogue between people and computers possible [11]. Afterwards Terry Winograd et al. proposed a system for answering questions, running instruction and accepting message in the responsive English dialog [12].

It was not until the 1990s that the domain of NLP was greatly developed. The trend of it research turned to a probabilistic thinking . As the increase in computer capabilities and data store, making access to a large amount of language data occurrence achievable. Combined with these language resources and language analyzers with good performance, we are able to extract useful language information and exploit practical systems for definite areas [13]. Deep Read [14] presented an early automatic reading comprehension system that agreed to input arbitrary text and answered questions about it and the research by the bag-of-words model (BOW). In 2000. Bengio et al. provided the word embedding technique which using the distributed representation for words to reduce the high dimensionality in big data context. Word2vec tools which put forward by Google at 2013 uses a three-layer neural network to achieve very great results. Besides, these chatbot design methods are based on processed corpus such as Google database and other online corpus. It is hard to apply in real applications. For Chinese, owing to the uniqueness of Chinese, many foreign mature technologies cannot be used in Chinese corpus. Chinese is different from English and other languages, it is written continuously, and lacks morphological changes such as voice and tense. At this stage, Chinese chatbot still has shortcomings like unanswered questions and limited response scenarios.

In this paper, we propose an improved chatbot design way using hybrid BOW and skip-gram model based on the real telemarketing data with Chinese. At first, we collect the telemarketing data. For the collected corpus, we use the language probability model for word segmentation. Then the decision tree is used for classification and word tagging. The word representation is adopted hybrid BOW and skip-gram model. When the BOW is used alone, the codes are independent of each other. Although this method is friendly to discrete features, it is not conducive to machine understanding of continuous statements due to the loss of connections between words in the encoded word vector. The skip-gram model maps synonyms in the vicinity of vector space. The correlation between words is expressed, so the amount of information contained in the word vector is increased. However, since the skip-gram model calculates the similarity based on the Euclidean distance, the output is also a continuous value, which is not sensitive enough to discrete features. Hybrid BOW and skip-gram model can make up for the shortcomings caused by using BOW alone, at the same time, it retains the excellent characteristics of BOW for discrete features. After this, we use the term frequency-inverse document frequency (TF-IDF) weighting method to adjust weights of words. The technique designed to ignore ordinary words and preserve representative words in the current context. Finally, the reply is produced using hybrid retrieval model and generate model. At present, deep learning is widely applied in the domain of communication [15]–[18], and it also has outstanding performance in voice dialogue. The retrieval model can accurately answer questions in the field. If the system receives an input, it will look up the input statement in the corpus when the input statement and corpus match are high, then output the answer with the highest matching degree. If not satisfied, the generate model used long short-term memory (LSTM) based multi-layer embedding mechanism can pick up the semantic information, generating a valid response. The purpose for the arrangement is to meet the characteristics of daily interaction between people and to provide personalized information services for each user.

The rest of this paper is set up as follows: Section II gives an account of the process model of the chatbot. Section III provides the experimental settings and results. Finally, Section IV introduces the conclusion and future works.

## 2. Process Model

Questionnaires and corresponding answers will be predefined in our corpus at first[19]. Then we perform a series of pre-processing on the defined corpus, including word segmentation and tagging, word embedding and word weighting. The "retrieval model" responds to the answer by comparing the user's statement with the existing data. It can provide more standardized and accurate responses to problems in specific areas to complete a specific task. However, user's statements cannot all be included in the predefined database. The generate model can handle flexible issues in different areas [20]. While, the model will be trained by a corpus with a large data content and a wide range of coverage, and the response statement is hard to keep smooth. As the result, we use a system hybrid retrieval model [21] and generate model in order to make up for the shortcomings of each system.

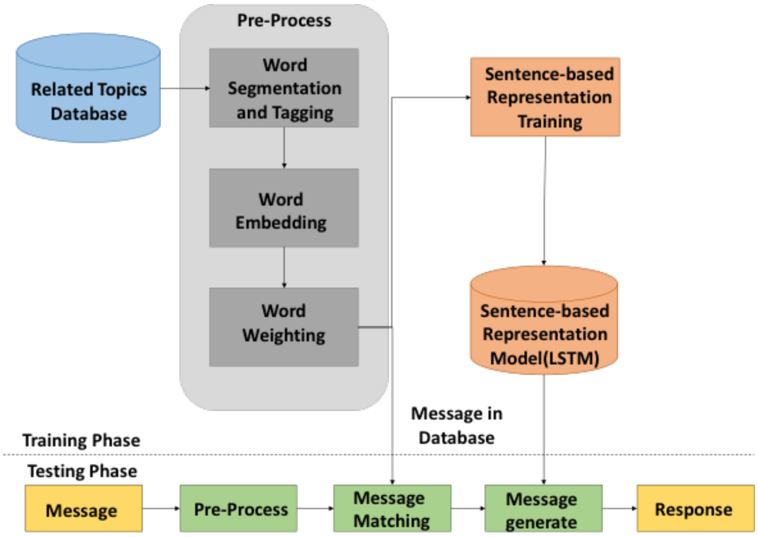


**Fig. 1.** The ultimate framework.

## 2.1 Data Processing

The first step of pre-process is to perform data processing, which is divided into two parts: word segmentation and word tagging. These technologies can help computer understand Chinese sentences quickly and accurately.

### 2.1.1 Word Segmentation

Word segmentation is an aspect of statistical probability. The input is a string $C = [c_1, c_2, \cdots, c_n]^T$ and the segmented is a string $S = [w_1, w_2, \cdots, w_m]^T$, $m \leq n$. For a given string $C$, it corresponds to a few of split ways $S$. The aim of segmentation is to look for the maximum likelihood among these $S$, that is,

$$Seg(C)\, argmax_{S\in G}\, P(S|C) = argmax_{S\in G} \frac{P(C|S)P(S)}{P(C)} \tag{1}$$

For example, there are two segmentation schemes $S_1$ and $S_2$. The conditional probabilities $P(S_1|C)$ and $P(S_2|C)$ are calculated, and then a segmentation way with the maximum likelihood is adopted. According to the Bayesian formula,

$$P(S|C) = \frac{P(C|S) \times P(S)}{P(C)} \tag{2}$$

where $P(C)$ is a fixed value. It shows the probability of the input string presence in the database. There is only one method from a segmented string to original, so $P(C|S) = 1$. So, the contrast between $P(S_1|C)$ and $P(S_2|C)$ is equivalent to contrast the value of $P(S_1)$ and $P(S_2)$.

For convenience, we presume that the probability of a element appearing is irrelevant with the context,

$$\begin{aligned} P(S) &= P(w_1, w_2, \cdots, w_m) \\ &\approx P(w_1) \times P(w_2) \times \cdots \times P(w_m) \\ &\propto log\, P(w_1) + log(w_2) + \cdots + log\, P(w_m) \end{aligned} \tag{3}$$

There will be different value of $m$ for different $S$, more concretely, the larger $m$ is, the smaller $P(S)$ is. This corresponds to the actual observation. This formula calculates the probability of a well-divided word

$$P(w_i) = \frac{n_{w_i}}{N} \tag{4}$$

where $n_{w_i}$ indicates the figure of appearance of $w_i$ in the corpus and $N$ suggests the whole number of element in the corpus.

### 2.1.2 Word Tagging

Words can be tagged by a decision tree [22]. The decision tree is a tree structure, as shown in **Fig. 2**. Through this structure, various combinations of situations are expressed. Each branch represents a selection until all selections have been made. Each leaf node represents a category that gives the final correct answer.

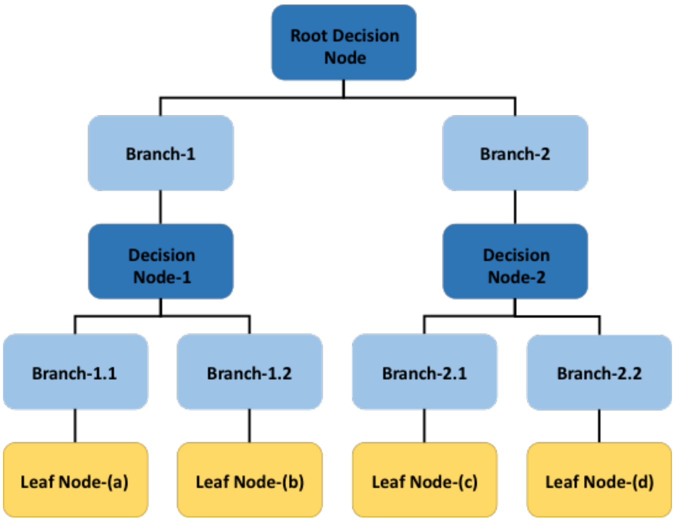


**Fig. 2.** The structure of decision tree.

The key question in decision tree is how to choose the optimal partitioning attribute. In general, in the process of classification, we hope that the samples included in the branch nodes is classified to the same category as much as possible, so that the "purity" of the nodes will keep rising. The important indicator of the branch is the attribute selection metric. It is a selection split criterion that determines how the data is divided. The selection metric of attributes is generally divided into three types: information gain, gain ratio, and *Gini* index. The calculation of these three indicators determines the priority of the classification attribute.

If there is a variable $X$, there are $n$ possible values, and each of the obtained probabilities is $p_i$, then the entropy of $X$ is calculated by

$$H(X) = -\sum_{i=1}^{n} p(x_i) \, log \, p(x_i) \tag{5}$$

For corpus $D$, the random variable $X$ is the class of the sample. Suppose the sample has $k$ sorts, and the possibility of each sort is $|C_k|/|D|$, where $|C_k|$ shows the figure of samples of sort $k$, and $|D|$ denotes the whole figure of samples. The entropy of the corpus $D$ is

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} log_2 \frac{|C_k|}{|D|} \tag{6}$$

Information gain refers to the change of information via the corpus is assorted. The more information a feature brings, the more suitable it is for classification. By using feature $A$ to classify corpus $D$, the information gain can be defined as

$$g(D, A) = H(D) - H(D|A) \tag{7}$$

where $H(D|A)$ means the mutual information between data set $D$ and feature $A$. However, there is a big problem with information gain. When there are many values belonging to a feature, it is easier to obtain a subset with higher purity according to the feature division, as the result, the entropy after the division is lower. Since the entropy before division is a fixed value, it leads to a larger information gain. Therefore, the information gain is biased toward the feature with more values.

Gain ratio can solve the above problem. It is obtained by multiplying the penalty parameter and the information gain

$$g_R(D, A) = \frac{g(D,A)}{H_A(D)} \tag{8}$$

where $H_A(D)$ denotes the empirical entropy obtained by using the current feature A as a random variable for the sample set D. It is calculated by the following formula

$$H_A(D) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} log_2 \frac{|D_i|}{|D|} \tag{9}$$

The penalty parameter is the reciprocal of the entropy of data set $D$ with feature $A$ as the random variable. When the value of the feature is small, the value of $H_A(D)$ is small, so that the reciprocal is large, resulting in a relatively large information gain. That is why the gain ratio is biased toward a feature with a small value. Generally, the features with higher information gain than the average are found in the candidate features, and then we choose a feature with the highest gain ratio.

The third indicator is the *Gini* index, which shows the possibility of a sample that picked out at random is misclassification in the corpus. The smaller the *Gini* index is, the smaller the possibility that the picked out sample is classified wrong. That is, the higher the purity of the class, and vice versa, the less pure the set. The decision tree uses the *Gini* index to select the partitioning property,

$$Gini(p) = \sum_{k=1}^{K} p_k(1 - p_k) = 1 - \sum_{k=1}^{K} {p_k}^2 \tag{10}$$

The *Gini* index for dividing the sample set $D$ based on the feature $A$ is

$$Gini(D, A) = 1 - \sum_{i=1}^{n} (\frac{|D_i|}{|D|})^2 \tag{11}$$

From all $Gini(D, A_i)$, the smallest division of the *Gini* index is found. This division point is the best division point for classifying the sample set $D$ by using feature $A$.

## 2.2 Word Expression

In early research, natural language processing often viewed text as a collection of words, focusing only on how often words appear in the text and by default assuming each word in the document is independent, the bag-of-words model (BOW). An example is shown in **Fig. 3**, it can be find that only one value in the vectors is 1, and the rest are 0. The vectors of the cities are independent of one another. In the statement, this is not conducive to the machine to contact each word, thus understanding the sentences. Besides, the size of the vector dimension depends on the number of words in the corpus. If the vectors corresponding to whole number of cities around the world are represented by one matrix, this matrix will be very sparse, and it leads to dimension disasters. This increases the computational cost and may degrade performance due to overfitting.

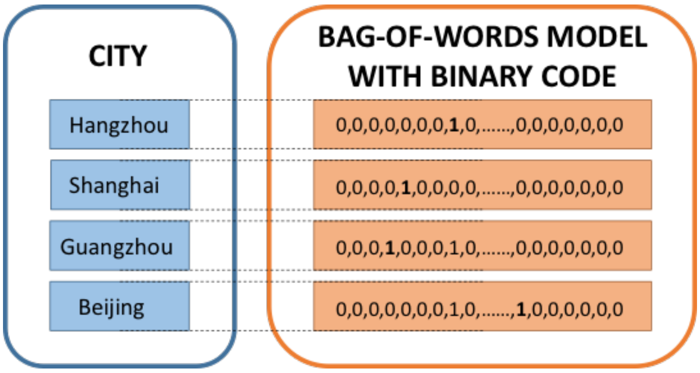


**Fig. 3.** Example of city name with BOW.

Word embedding converts BOW turn to low-dimensional continuous values, which called dense vectors. In addition, in the BOW, the positions where the words are placed are independent. For word embedding, the relationship between words can be expressed by distance. Synonym will be mapped nearby in vector space, so the amount of information contained in the word vector is increased, which is better for helping computers understand the relationship between words [23]. One of the most representative models of word embedding is the skip-gram model [24].

The skip-gram full name is the continuous skip-gram model. It supposes the content in which the current word is given. For a given sample $(w, Context(w))$, $\widetilde{\mathrm{w}}$ indicates the label of the word. The negative sample subset generated when dealing with the word $\widetilde{\mathrm{w}}$ is $\mathrm{NEG}^{\widetilde{\mathrm{w}}}(\mathrm{w})$. We want to maximize

$$g(w) = \prod_{\widetilde{w} \in Context(w)} \prod_{u \in \{w\} \cup NEG^{\widetilde{w}}(w)} p(u|\widetilde{w}) \tag{12}$$

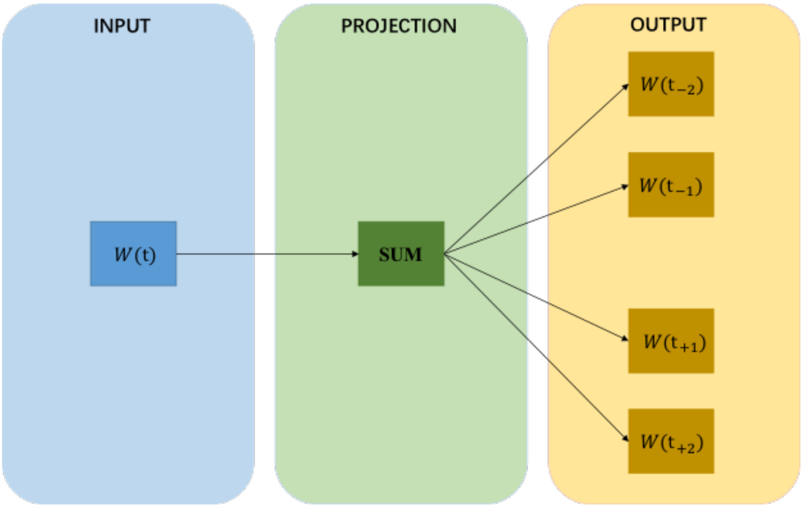
Where

$$p(u|\widetilde{w}) = \begin{cases} \sigma(v(\widetilde{w})^T \theta^u), L^w(u) = 1; \\ 1 - \sigma(v(\widetilde{w})^T \theta^u), L^w(u) = 0; \end{cases} \tag{13}$$

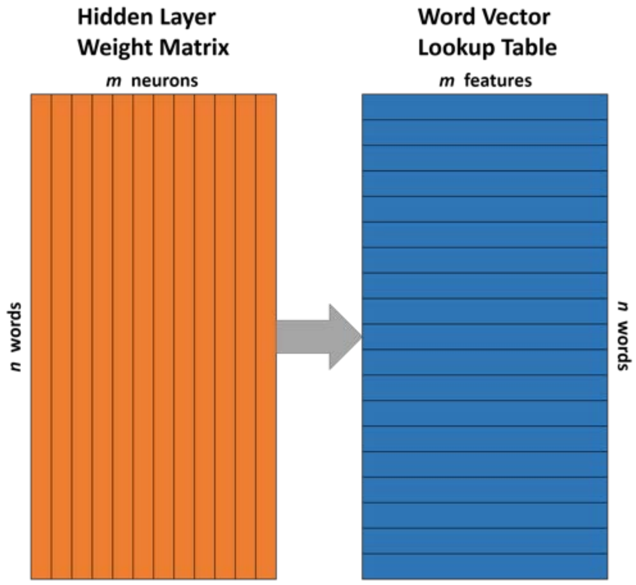
$$L^w(u) = \begin{cases} 1, u = w; \\ 0, u \neq w; \end{cases} \tag{14}$$

The definition $\sigma\left(\mathrm{v}(\widetilde{\mathrm{w}})^{\mathrm{T}} \theta^{\mathrm{u}}\right)$ indicates the probability that the context is $Context(w)$ when the central word is $w$.

The neural network structure of the skip-gram is shown in **Fig. 4**.



(a)



(b)

**Fig. 4.** (a) Skip-gram model. Input layer is denoted as W word vector v(w), projection layer is still v(w), there is a weight matrix in it, which output an "embedded word vector" for each input word. The output of the hidden layer is just the "word vector" of the input word. Output layer is a Huffman tree where the leaf nodes are words that exist in the database and the weight is the figure of appearances; (b) the hidden layer operates as a lookup table.

## 2.3 Word Weighting

Term frequency-inverse document frequency (TF-IDF) is a statistically based way to judge the significant of a word to a document in the corpus. Term frequency (TF) indicates the figure of appearances of a known word in the document. This figure is frequently normalized,

$$tf_{ij} = \frac{m_i}{M} \tag{15}$$

where $tf_{ij}$ is the figure of features existing in the document, $m_i$ is the quantity of times a word $w$ appears in the document. $M$ is the total quantity of words in the document. Inverse document frequency (IDF) [25] means that in case the figure of documents contain smaller term $t$, the IDF will be larger. It shows that the word has a superior differentiate sort performance,

$$idf_i = log\left(\frac{N}{n_i+1}\right) \tag{16}$$

where $idf_i$ is the reciprocal of the feature term, $N$ is the whole figure of documents in the database as well as $n_i$ is the amount of papers involving of the item $w$. The denominator of the formula is added to prevent it from being zero.

If a word appears many times in a file and has a very low file frequency throughout the data set, it can result in a highly weighted TF-IDF. Thus, the role of TF-IDF is to retain important words and drop common words [26],

$$w_{ij} = tf_{ij} \times idf_i = \frac{m_i}{M} \times log\left(\frac{N}{n_i+1}\right) \tag{17}$$

We use normalization to rule out the effect of document length on weight calculations, and get the following formula

$$w_{ij} = \frac{tf_{ij} \times log\left(\frac{N}{n_i+1}\right)}{\sqrt{\sum_{l-1}^{N} tf_{ij}{}^2 \times log^2\left(\frac{N}{n_i+1}\right)}} \tag{18}$$

## 2.4 LSTM Model

A suitable model for processing principal events with relatively long intervals and delays in time series [27] called the storage unit LSTM model. The architecture is given in Fig. 5 below. The memory cell composed of four parts: the input gate, the neurons with autoregressive connections (connected to itself), the forget gate together with the output gate. This operation allows the state of the memory cell to keep fixed over time, but does not preclude external interference. The role of the gate is to regulate the interplay between the memory unit itself and its environment. The input gate may permit the input signal to alter the state of the memory cell or prevent it. Furthermore, the output gate can permit the state of the memory cell to have influence on other neurons or prevent it. Eventually, the forget gate can control the self-reverting connection of the storage unit so as to the unit alters the memory of the previous state according to demand.

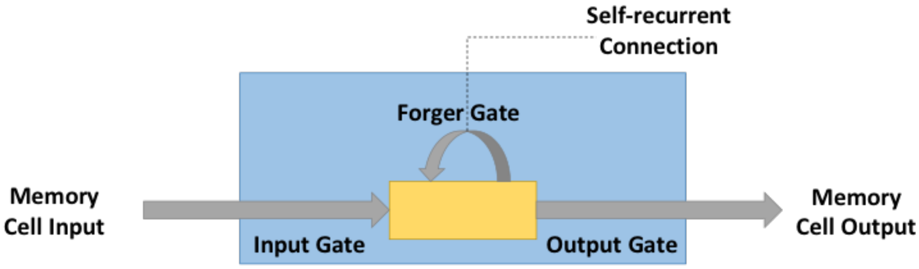


Fig. 5. LSTM storage unit.

These formula shows how to renew a layer of memory cells $\boldsymbol{t}$ at every time step [28]. In these formula, $\boldsymbol{x_t}$ means the input to the memory cell layer, $\boldsymbol{W_i}, \boldsymbol{W_f}, \boldsymbol{W_c}, \boldsymbol{W_o}, \boldsymbol{U_i}, \boldsymbol{U_f}, \boldsymbol{U_c}, \boldsymbol{U_o}$and $\boldsymbol{V_o}$ are the weight matrix. $\boldsymbol{b_i}, \boldsymbol{b_f}, \boldsymbol{b_c}$, and $\boldsymbol{b_o}$ are the offset vector. Above all, we calculate the value $\boldsymbol{i_t}$ of the memory cell state over time, input gate and candidate $\widetilde{\boldsymbol{C}}_{\boldsymbol{t}}$,

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{19}$$

$$\widetilde{C}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{20}$$

Second, we work out the activation value of the forget gate at time in the memory cell $f_t$,

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{21}$$

Calculating the new state $C_t$ of the memory cell over time based on the input gate activation value $i_t$, the forget gate activation value $f_t$ and the candidate state value $\widetilde{C}_t$,

$$C_t = i_t \times \widetilde{C}_t + f_t \times C_{t-1} \tag{22}$$

Based on the new status of memory cells, we can work out the values of the output gates and the subsequent output are

$$o_t = \ \sigma(W_o x_t + U_o h_{t-1} + V_O C_t + b_o) \tag{23}$$

$$h_t = o_t \times tanh(C_t) \tag{24}$$

## 3. Experimental Results and Analysis

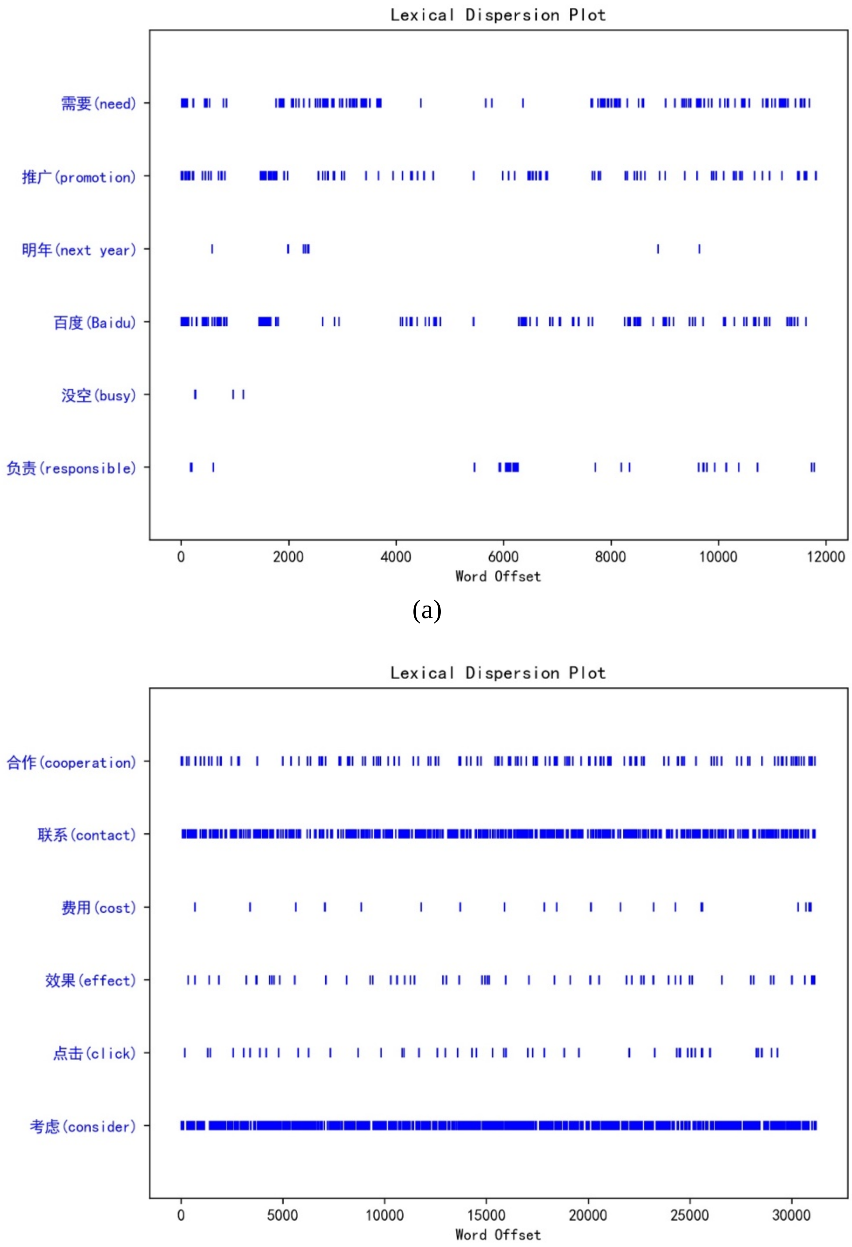
The aim of the experiment is to construct a commercial automated chatbot which can answer questions from users accurately and quickly, in addition, the response scene is flexible.

### 3.1 Data Processing Results

The main source of the data set is the question and answer corpus provided by commercial users. There are generally two types. The first is the industry dialogue template library. The second is the daily conversation corpus provided by the user, which is generated and collected during the dialogue between actual customer service and users. Our answer corpus provides a variety of different answers to the same question in order to meet multiple needs.

The Chinese word segmentation used in this paper is based on the model of probability and statistics. The task of word segmentation is to look for the most likely of these splitting schemes. The following figures show the frequency of six common word segments in two corpuses.

After the word segmentation, the word tagging is executed in corpus. The number of nouns, verbs and adjectives are counted in **Fig. 7**. The same question may correspond to different answers, so the figure of different types of words in the answer corpus is more than question corpus in the results.



(a)

(b)

**Fig. 6.** (a) The frequency of six common word segments in question corpus; (b) The frequency of six common word segments in answer corpus.

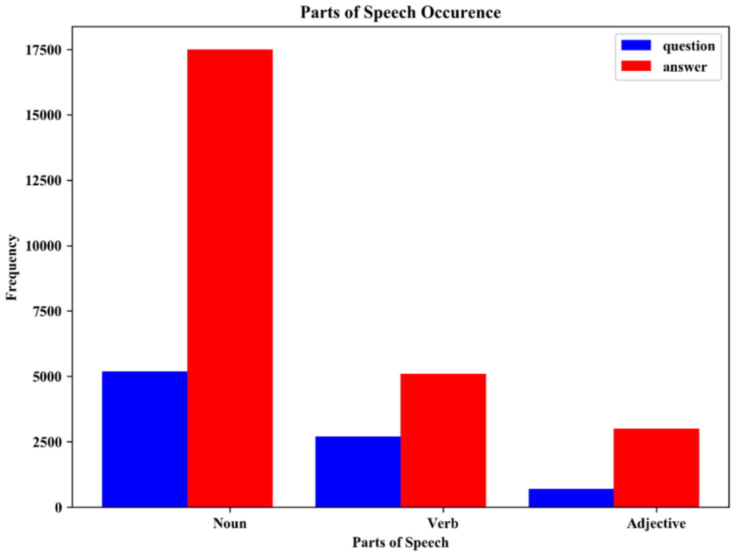
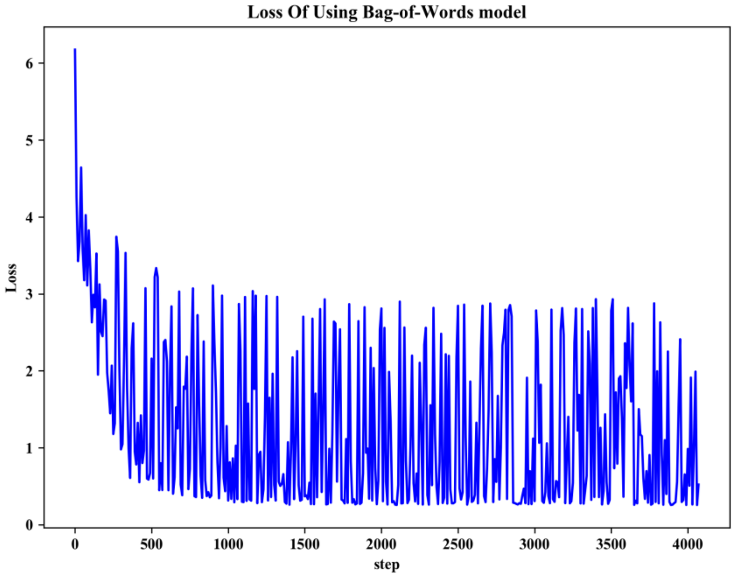


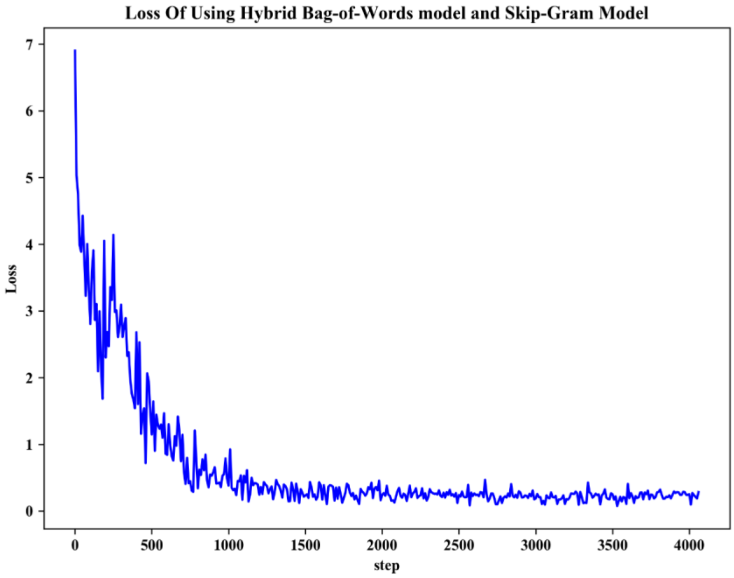
**Fig. 7.** The word tagging in question corpus and answer corpus based on commercial promotion.

## 3.2 Comparison of Different Word Vector Representations

When expressing words, we compared the performance of a model that uses BOW alone and the BOW combined with skip-gram model.



(a)



(b)

**Fig. 8.** (a) The loss function used bag-of-words model alone;
(b) The loss function used the model combined with bag-of-words model and skip-gram model.

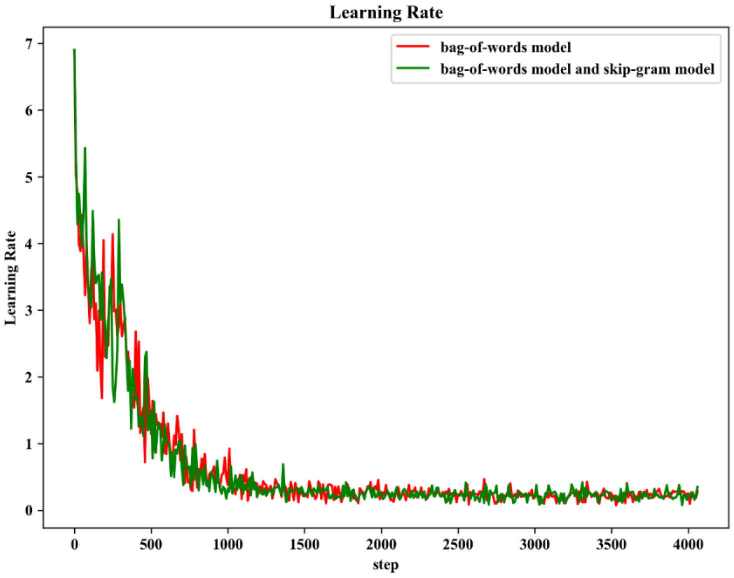


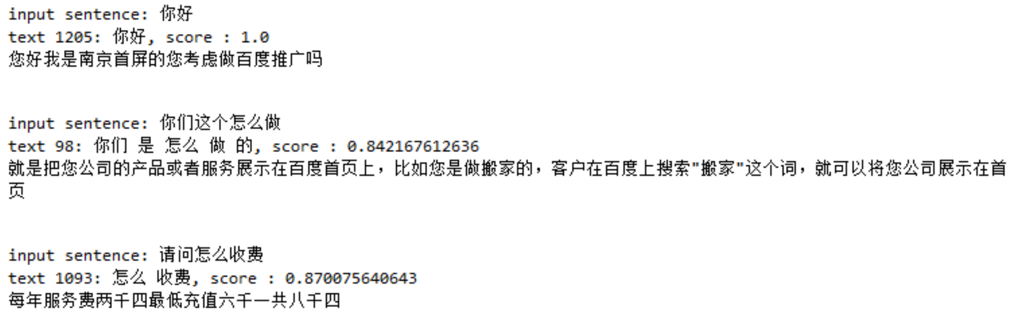
**Fig. 9.** Learning rate of two models.

The comparison is made to bring the results more obvious when the learning rate is approximately same. The skip-gram model combined with the word bag model uses multiple central words to generate the final word vector. Specifically, we first select a word as the first central word and then find the word which closest to the first central word as the second central word. In this way, we can find the third central word based on the second central word, and so on. Considering the problem of repeated words, if the word that is most similar to the i-th central word has appeared, the second similar to it is selected, if it also exists, the third similar word is selected, and so on.

We use the loss function to describe the performance, and the number of training steps is 4100 steps. At this point, the results have stabilized. The experiments prove that the model combined with BOW and skip-gram model training word vector can significantly improve the accuracy.

### 3.3 Performance with Combined Model

The switching between the two models is achieved by the question similarity threshold, that is, a threshold is set according to the similarity of the questions obtained by the retrieval model. If the similarity is greater than the threshold, the result of the retrieval model is invoked, and the generate model is invoked instead. After multiple experiments, we found that when the retrieval model finds the corresponding sentence with a matching degree of 0.8 or higher in the corpus, the sentence can be output as the final reply.



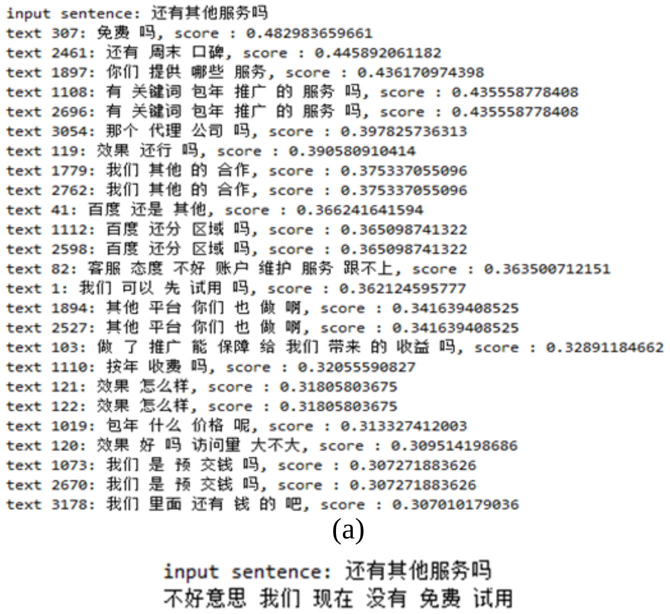
```
input sentence: 你好
text 1205: 你好, score : 1.0
您好我是南京首屏的您考虑做百度推广吗


input sentence: 你们这个怎么做
text 98: 你们 是 怎么 做 的, score : 0.842167612636
就是把您公司的产品或者服务展示在百度首页上，比如您是做搬家的，客户在百度上搜索"搬家"这个词，就可以将您公司展示在首
页


input sentence: 请问怎么收费
text 1093: 怎么 收费, score : 0.870075640643
每年服务费两千四最低充值六千一共八千四
```

**Fig. 10.** Retrieval model answer some questions.

When the matching degree is lower than 0.8, the generate model is used to generate the reply. The sentences shown in **Fig. 11** (a) are arranged from high to low by the similarity with the input sentences. **Fig. 11** (b) shows the case of using the generate model to form a corresponding response.



```
input sentence: 还有其他服务吗
text 307: 免费 吗, score : 0.482983659661
text 2461: 还有 周末 口碑, score : 0.445892061182
text 1897: 你们 提供 哪些 服务, score : 0.436170974398
text 1108: 有 关键词 包年 推广 的 服务 吗, score : 0.435558778408
text 2696: 有 关键词 包年 推广 的 服务 吗, score : 0.435558778408
text 3054: 那个 代理 公司 吗, score : 0.397825736313
text 119: 效果 还行 吗, score : 0.390580910414
text 1779: 我们 其他 的 合作, score : 0.375337055096
text 2762: 我们 其他 的 合作, score : 0.375337055096
text 41: 百度 还是 其他, score : 0.366241641594
text 1112: 百度 还分 区域 吗, score : 0.365098741322
text 2598: 百度 还分 区域 吗, score : 0.365098741322
text 82: 客服 态度 不好 账户 维护 服务 跟不上, score : 0.363500712151
text 1: 我们 可以 先 试用 吗, score : 0.362124595777
text 1894: 其他 平台 你们 也 做 啊, score : 0.341639408525
text 2527: 其他 平台 你们 也 做 啊, score : 0.341639408525
text 103: 做 了 推广 能 保障 给 我们 带来 的 收益 吗, score : 0.32891184662
text 1110: 按年 收费 吗, score : 0.32055590827
text 121: 效果 怎么样, score : 0.31805803675
text 122: 效果 怎么样, score : 0.31805803675
text 1019: 包年 什么 价格 呢, score : 0.313327412003
text 120: 效果 好 吗 访问量 大不大, score : 0.309514198686
text 1073: 我们 是 预 交钱 吗, score : 0.307271883626
text 2670: 我们 是 预 交钱 吗, score : 0.307271883626
text 3178: 我们 里面 还有 钱 的 吧, score : 0.307010179036
```

(a)

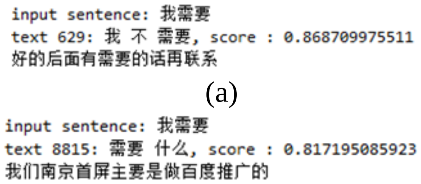
```
input sentence: 还有其他服务吗
不好意思 我们 现在 没有 免费 试用
```

(b)

**Fig. 11.** (a) The degree of matching of a particular question; (b) Generate model gives a similar answer.

### 3.4. Performance with Weight Adjustment

We found that the entire system answer was not ideal at the time, due to insufficient feature extraction. So we used TF-IDF technology to improve this feature.



```
input sentence: 我需要
text 629: 我 不 需要, score : 0.868709975511
好的后面有需要的话再联系
```

(a)

```
input sentence: 我需要
text 8815: 需要 什么, score : 0.817195085923
我们南京首屏主要是做百度推广的
```

(b)

**Fig. 12.** (a) The answer given with the usual weighting method; (b) The answer given with the TF-IDF.

In the usual weighting method, the word *need* is considered to be an important word because of the high number of occurrences, thus giving a higher weight. It will cause sometimes *not need* to be recognized as *need* by the system, outputting the wrong answer. TF-IDF method improves this performance.

## 5. Conclusion

The objective of this paper is to propose a Chinese-based commercial automated chatbot system, which includes word segmentation, word tagging, word code as vector representation, and uses the hybrid retrieval model and generate model to meet the needs of daily interaction, in addition by improving the weighting method, the weight of each word is more reasonable. Then the model will proceed to be improved to make the answer faster and more accurate.

## References

[1] Y. Takebayashi, "A consideration of learning in speech recognition from the viewpoint of AI class-description learning," in *Proc. of the Twenty-First Annual Hawaii International Conference on System Sciences*, pp. 705–714, 1988. Article (CrossRef Link)

[2] I. Kushchu, "Web-Based Evolutionary and Adaptive Information Retrieval," *IEEE Trans. Evol. Comput.*, vol. 9, no. 2, pp. 117–125, 2005. Article (CrossRef Link)

[3] H. S, E. N, T. M, Y. S, M. F, K. T,T. Kasami, "A processing system for programming specifications in a natural language," in *Proc. of the Twenty-First Annual Hawaii International Conference on System Sciences*, pp. 754–763, 1988. Article (CrossRef Link)

[4] M. Vargas-Vera and M. D. Lytras, "AQUA: hybrid architecture for question answering services," *IET Softw.*, vol. 4, no. 6, pp. 418–433, 2010. Article (CrossRef Link)

[5] S. A. Abdul-Kader and J. Woods, "Question answer system for online feedable new born Chatbot," in *Proc. of 2017 Intelligent Systems Conference(IntelliSys)*, pp. 863–869, 2017. Article (CrossRef Link)

[6] A. Guerrieri, G. Ghiani, and A. Manni, "A Tourist Advisor based on a Question Answering System," in *Proc. of the 2017 Intelligent Systems Conference (Intellisys)*, pp. 1173–1176, 2017. Article (CrossRef Link)

[7] F. Casacuberta *et al.*, "Human interaction for high-quality machine translation," *Commun. ACM*, vol. 52, no. 10, pp. 135–138, 2009. Article (CrossRef Link)

[8] S. A. and D. John, "Survey on Chatbot Design Techniques in Speech Conversation Systems," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 7, pp. 72–80, 2015. Article (CrossRef Link)

[9] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 295–302, 2007. Article (CrossRef Link)

[10] J. Bert F. Green, A. K. Wolf, C. Chomsky, and K. Laughery, "BASEBALL: AN AUTOMATIC QUESTION-ANSWERER Bert," in *Proc. of IRE-AIEE-ACM Computer Conference*, pp. 219–224, 1961. Article (CrossRef Link)

[11] W. A. Woods, "Progress in natural language understanding: an application to lunar geology," *National computer conference and exposition*, pp. 441–450, 1973. Article (CrossRef Link)

[12] J. Weizenbaum, "Eliza A computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, no. 1, pp. 36–45, 1966. Article (CrossRef Link)

[13] H. Isahara, "Resource-based Natural Language Processing," in *Proc. of 2007 International Conference on Natural Language Processing and Knowledge Engineering*, pp. 11–12, 2007. Article (CrossRef Link)

[14] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Toward understanding natural language directions," in *Proc. of HRI '10 Proc. of the 5th ACM/IEEE international conference on Human-robot interaction*, pp. 259–266, 2010. Article (CrossRef Link)

[15] G. G. Yu Wang, Miao Liu, Jie Yang, "Data-Driven Deep Learning for Automatic Modulation Recognition in Cognitive Radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, 2019. Article (CrossRef Link)

[16] G. G. H. Huang, Y. Song, J. Yang, "Deep-Learning-Based Millimeter-Wave Massive MIMO for Hybrid Precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, 2019. Article (CrossRef Link)

[17] H. S. G. Gui, H. Huang, Y. Song, "Deep learning for an effective non-orthogonal multiple access scheme," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8440–8450, 2018. Article (CrossRef Link)

[18] G. G. M. Liu, J. Yang, T. Song, J. Hu, "Deep learning-inspired message passing algorithm for efficient resource allocation in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 641–653, 2019. Article (CrossRef Link)

[19] M. H. Su, C. H. Wu, K. Y. Huang, Q. B. Hong, and H. M. Wang, "A chatbot using LSTM-based multi-layer embedding for elderly care," in *Proc. of the 2017 International Conference on Orange Technologies, ICOT 2017*, pp. 70–74, 2018. Article (CrossRef Link)

[20] S. Quarteroni and S. Manandhar, "Designing an Interactive Open-Domain Question Answering System," *Nat. Lang. Eng.*, vol. 15, no. 1, pp. 73–95, 2009. Article (CrossRef Link)

[21] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional Neural Network Architectures for Matching Natural Language Sentences," in *Proc. of Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pp. 2042–2050, 2014. Article (CrossRef Link)

[22] U. M. Fayyad and K. B. Irani, "On the Handling of Continuous-Valued Attributes in Decision Tree Generation," *Mach. Learn.*, vol. 8, no. 1, pp. 87–102, 1992. Article (CrossRef Link)

[23] D. Grangier, J. Keshet, and S. Bengio, "Discriminative Keyword Spotting," *Speech Comm.*, vol. 51, no. 4, pp. 317–329, 2009. Article (CrossRef Link)

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Proc. of Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pp. 1–9, 2013. Article (CrossRef Link)

[25] K. Spärck Jones, "A Statistical Interpretation of Term Specificity and its Retrieval," *J. Doc.*, vol. 28, no. 1, pp. 11–21, 1972.

[26] A. Guo and T. Yang, "Research and improvement of feature words weight based on TFIDF algorithm," in *Proc. of 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2016*, pp. 415–419, 2016.

[27] S. Hochreiter and J. J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. Article (CrossRef Link)

[28] A. Graves and N. Jaitly, "Towards End-To-End Speech Recognition with Recurrent Neural Networks," in *Proc. of International Conference on Machine Learning*, pp. 1764–1772, 2014. Article (CrossRef Link)

**Jie Zhang** received her B.S. degree in In and Engineering from Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, China in 2017. She is currently working toward the M.Sc degree in NUPT. His research interests include deep learning and its application in natural language processing.

**Jianing Zhang** is the member of FocusLab, Nanjing University of Posts and Telecommunications (NJUPT) since 2018. She is currently working toward the becholar degree in School of Computer Science, NJUPT. Her research interests include deep learning for wireless communication.

**Shuhao Ma** is the member of FocusLab, Nanjing University of Posts and Telecommunications (NJUPT) since 2018. His current research interests include reservoir computing, neural networks, and deep learning architectures with applications in the domains of time series prediction, classification, and nonlinear system identification.

**Jie Yang** received the B.Sc. degree, M.Sc degree, PhD degree in communication engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2003 and 2006, 2018, respectively. She is Assistant Professor with Nanjing University of Posts and Telecommunications, Nanjing China.

**Guan Gui** received the Dr. Eng degree in Information and Communication Engineering from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2012. From October 2009 to March 2012, with the financial supported from the China scholarship council (CSC) and the global center of education (ECOE) of Tohoku University, he joined the wireless signal processing and network laboratory, Department of Communications Engineering, Graduate School of Engineering, Tohoku University as for research assistant as well as postdoctoral research fellow, respectively. From September 2012 to March 2014, he was supported by Japan society for the promotion of science (JSPS) fellowship as postdoctoral research fellow at same laboratory. From April 2014 to October 2015, he was an Assistant Professor in Department of Electronics and Information System, Akita Prefectural University. Since November 2015, he has been a professor with Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, China.

He is currently engaged in research of deep learning, compressive sensing and intelligent wireless techniques. Dr. Gui has published more than 200 international peer-reviewed journal/conference papers. He received Member and Global Activities Contributions Award in IEEE ComSoc and eight best paper awards, i.e., ICEICT 2019, CSPS 2019, ADHIP 2018, CSPS 2018, ICNC 2018, ICC 2017, ICC 2014 and VTC 2014-Spring. He was also selected as for Jiangsu Specially-Appointed Professor (2016), Jiangsu High-level Innovation and Entrepreneurial Talent (2016), Jiangsu Six Top Talent (2018), Nanjing Youth Award (2018). Dr. Gui was an Editor of Security and Communication Networks (2012--2016). He has been the Editor of IEEE Transactions on Vehicular Technology, since 2017, the Editor of IEEE Access, since 2018, the Editor of Physical Communication Journal, since 2019, the Editor of KSII Transactions on Internet and Information Systems since 2017, the Editor of Journal of Communications, since 2019, and the Editor-in-Chief of EAI Transactions on Artificial Intelligence, since 2018. He is IEEE Senior Member.