Agricultural Q&A System Based on LSTM-CNN and Word2vec

He Liu¹,²
¹ College of Information Technology, Jilin Agricultural University, Changchun 130000, Jilin, China
² Jilin precision agriculture and big data engineering research center, Changchun 130000, Jilin, China

Chenxi Li
College of Information Technology, Jilin Agricultural University, Changchun 130000, Jilin, China

Helong Yu¹²*
¹ College of Information Technology, Jilin Agricultural University, Changchun 130000, Jilin, China
² Jilin precision agriculture and big data engineering research center, Changchun 130000, Jilin, China
*Corresponding author (E-mail: yuhelong@aliyun.com)

Abstract

In order to realize the automation of agricultural question-and-answer, a deep learning model is designed to find the best answer to farmers’ questions by processing the accumulated agricultural question-and-answer data. Aiming at the shortcomings of the existing agricultural Q&A, proposed a Q&A model based on LSTM and CNN. For the agricultural question and answer data, the program of question and answer preprocessing and question and answer matching model training is designed with Python language. Pre-processing used Jieba tools to load the agricultural thesaurus for question and answer segmentation and separation of words. Genesis is used to compute word vector (word2vec), which is used to construct word vector containing semantic information. Word2vec solved dimensional explosion and sparse problem of previous word vectors. Finally, the LSTM-CNN fusion deep neural network model is constructed with Keras. Use questions, right and wrong answers as training data. The experimental results show that the Q&A model based on word2vec and LSTM-CNN can get better accuracy in dataset without manual annotation.

Key words: Text Classification, Feature Extraction, LSTM- CNN, Word2vec, Q&A System.

1. Introduction

Automatic question answering system is a common alternative to manual solution in many fields. Unlike search, it allows users to request information in natural language rather than in combinations of keywords. It integrates knowledge representation, information retrieval, natural language processing and other technologies, and can return correct answers from the knowledge base after analyzing user questions [1]. In view of the existing agricultural question and answer data, the agricultural question and answer system is constructed to analyze the problems encountered by farmers, calculate the similarity between them and the questions in the knowledge base, and return the most appropriate answers.

Traditional q&a matching mainly relies on the manual construction of q&a rules. Text representation includes probability model, latent semantic index model and vector space model [2]. Pacanaro et al. proposed the representation of word vectors [3]. Gongjing used improved tf-idf algorithm combined with naive bays for text classification [4]. Yoon Kim et al. processed sentence classification with convolutional neural network [5]. In front of text vector representation has problem which is sparse vector and no semantic relations between vectors. In recent years, Bengio Y. et al. proposed the neural network word vector method [6]. Mikolov et al. used word2vec tools to train word vectors with low dimensions and calculate semantic similarity [7]. Zhang D. et al. used word2vec to establish word vectors and SVM for text classification [8]. Tang Ming et al. used word2vec to calculate the document representation [9], and machine learning (KNN (k-nearest neighbor), Bayesian classifier, SVM (support vector machine)) is used to build the model of matching q&a pairs. This often depends on the quality of the selection of artificial features, which is limited by people’s extraction of question and answer features and lack of deeper semantic features.

Recurrent Neural Network (RNN) can effectively process time series data like natural language text, and make full use of context information for feature extraction. In order to solve the problem of gradient disappearance and explosion of RNN, the improved Long Short-Term Memory (LSTM) network can well record the context information, extract the characteristics of time series data, and connect the convolution layer and the full connection layer for text classification.
In this paper, word2vec combined with LSTM and CNN neural network is used to train similarity model for a certain agricultural question and answer data. Firstly, agricultural text corpus was constructed by preprocessing such as word segmentation, and word2vec word vectors were trained by genesis tool. Then, LSTM and CNN hybrid neural network model were used to train the data. Finally, the accuracy of the model was calculated.

2. Data Acquisition and Processing Methods

2.1. Data Sources

The data of inquiry and answer of some agriculture net of source is total more than 20 thousand data.

2.2. Data Processing

Text preprocessing of Chinese documents is different from English processing. English takes words as the basic unit, while Chinese takes characters as the basic unit. Words alone cannot express the complete meaning, so Chinese word segmentation is required.

Chinese word segmentation methods based on string matching, understanding or statistics. Agricultural text contains a large number of specialized agricultural words, which are difficult to be covered by ordinary word segmentation. The agricultural vocabulary of sago was introduced in the experiment.

The text was separated by the Jieba tool in Python, and Jieba loads the agricultural glossary thesaurus for the user thesaurus and the dictionary dict.txt file into a word search tree (Trie tree). We computing word frequency and generate all possible directed acyclic graphs of statements from the Trie tree. Full-mode segmentation of Jieba was performed using dynamic programming to find the maximum probability path. For words that did not exist in the dictionary corpus, HMM was used and Viterbi algorithm was used to solve the optimal state sequence.

After the word segmentation, stop word processing is also required. In general, stop words are not differentiated and representative for text classification. Delete the words with high number of documents and some function words, pronouns and stops from the rough word segmentation results, such as “yes”, “%”, “before” and so on. At the same time, the agricultural thesaurus was added to increase the word segmentation accuracy. The word segmentation results separated by Spaces are obtained as shown in table 1.

<table>
<thead>
<tr>
<th>Participle</th>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answer</td>
<td>Hello, Suzi, Root, Situation, Suggestions You, Detailed Description, Dial, No. Keys, Consultation, Platform, Experts, For Reference Only</td>
</tr>
</tbody>
</table>

The text features are expressed in numerical form for the convenience of computer understanding. The smallest unit for text to be understood is a word, which can be expressed as a vector of fixed dimensions, and the similarity between words can be obtained by calculating the distance between two vectors.

Word2vec uses distributed method to represent text, and the model implementation has two models based on CBOW and Skip-Gram. In this paper, Skip-Gram model word2vec is adopted. Skip-Gram model generates word vectors that are more accurate. The word vector generated by the model is more accurate, but it takes a little bit longer. The word vector dimension is set to 400, the training window is set to 5, and filter words with frequency less than 5, and the sampling rate is 0.001. Skip-Gram model is shown in figure 1 [10]. In the range of window size c=5, w (t) is used to predict the word of context.

\[
P(w_{t-2}w_{t-1}w_{t+1}w_{t+2} | w)
\]

(1)

The input layer is the word vector, and the iterative training maximizes the probability of formula (1). In order to accelerate the training, Negative Sampling is adopted. We marked \(W_{t-2}W_{t-1}W_{t+1}W_{t+2}\) as the context (w), w and context (w) are 1 positive example. Through negative sampling, we obtained the N negative examples that nonexistent word and context.
We conduct binary logistic regression for 1 positive and N negative examples, positive examples is for \( w_0, w_1, w_2, \ldots, \) N negative examples, we set the model parameters of each word be \( \theta_i \), \( i = 0, 1, 2, \ldots, n \), the initial parameters are random. For a given positive example, the maximization formula (2):

\[
L_s = \sum_{i=0}^{n} y_i \log(\sigma(X_{w_i}^T \theta_i)) + (1 - y_i) \log(1 - \sigma(X_{w_i}^T \theta_i))
\]

\( X_{w_i} \)— word vector

We use the random gradient rise to iterate to calculate \( X_{w_i}, \theta_i \), \( i=0, 1, 2, \ldots, n \), negative sample n is set to 10. Negative sampling uses one dimension to partition random sampling, divides a line segment of length 1 into M equal parts, \( M \gg V \), V is glossary size, the length of each word in the line segment is calculated by the formula (3):

\[
\text{len}(w) = \frac{\text{count}(w)^{3/4}}{\sum_{u \in \text{vocab}} \text{count}(u)^{3/4}}
\]

During sampling, n positions are randomly sampled from M positions, and the words of line segment corresponding to the positions are negative example words. After computing the cosine value between the word vectors obtained, the word with semantic similarity can be obtained after descending order. Table 2 shows similar words of rice. Figure 2, the similar words of Chinese medicinal materials and rice are obviously separated.
Table 2. Similar words of rice

<table>
<thead>
<tr>
<th>Word</th>
<th>Cosine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rice</td>
<td></td>
</tr>
<tr>
<td>paddy</td>
<td>0.686254</td>
</tr>
<tr>
<td>Hybrid rice</td>
<td>0.602272</td>
</tr>
</tbody>
</table>

In order to compare the different text representations of word2vec and tfidf (word frequency-inverse frequency) and bow (bag of word), the two-dimensional viewable text vectors of tfidf, bow and word2vec under three random text representations were calculated and drawn respectively. Word2vec represents text by averaging all word vectors of the text, and the results are shown in figure 3. The text vectors of tfidf and bow have obvious overlap, while the boundaries of the three types of text represented by word2vec are obvious.

By experimental comparison, word2vec is stronger in the expression of text semantics and solves the high dimension sparse problem of tfidf and bow vectors. In this paper, word2vec is used as the text feature input of the classification model.

3. Question and Answer Model Based on LSTM-CNN

In recent years, with the rapid development of deep neural network (DNN), it has achieved success in audio recognition, image recognition, natural language processing and other fields. Long Short-Term Memory (LSTM) network has been widely used as a neural network model for time series data processing as an improvement of Recurrent Neural Network (RNN) [11]. Text features are obtained by word vector calculation in word2vec. Lstm-cnn extracts the central features of the whole text.

LSTM is an improvement of RNN. RNN neural network can save historical information, understand time series data, and apply the information of the previous layer to the information of the lower layer. There are problems of gradient disappearance and gradient explosion in RNN training [12]. Hochreiter et al improved the RNN to obtain LSTM neural network model [13]. By improving the original unit structure of RNN, long-term information can be effectively remembered, favorable data can be selected and useless data can be discarded,
and historical information can be fully utilized. CNN retains the features of LSTM to further extract the central features of the text to make feature extraction more accurate.

LSTM adds gate unit to the original RNN structure to complicate the structure of the unit (hidden layer). The data are processed through the input gate, the forgetting gate and the output gate within the cell, and maintain the cell state and cell output at each moment. The overall structure of LSTM is shown in figure 4(top), taking the 4-length sequence as an example, and the unit structure is shown in figure 4(bottom).

In the gate structure of LSTM, mark ① is forgotten gate, mark ② is input gate and mark ④ is output gate as shown in figure 4. The forgetting gate decides to forget the information. By calculating $h_{t-1}$ and $x_t$, we get values on the scale of 0 to 1 and apply them to the cell state $C$, where 1 is completely preserved and 0 is completely discarded. Formula (4) is the calculation of the forgetting gate.

$$f_t = \sigma \left( w_f \cdot [h_{t-1}, x_t] + b_f \right)$$

where $h_{t-1}$ is Previous Cell Output, $x_t$ is Current Cell Input, $\sigma$ is Sigmoid Function, $w_f$ is Forgetting Gate Weight and $b_f$ is Forgetting Gate Bias.

The input gate determines how much new information is added to the cell state. There are two steps. First, sigmoid layer decides the information that needs to be updated, then tanh layer decides the alternative information, and finally the dot product decides to update the information. Formula (5) and (6) are calculated for the input gate.

$$i_t = \sigma \left( w_i \cdot [h_{t-1}, x_t] + b_i \right)$$

$$\tilde{c}_t = \tanh \left( w_c \cdot [h_{t-1}, x_t] + b_c \right)$$

where $h_{t-1}$ is Previous Cell Output, $x_t$ is Current Cell Input, $\sigma$ is Sigmoid Function, $w_i$ is Input Gate Sigmoid Layer, $w_c$ is Input Gate Tanh Layer, $b_i$ is Sigmoid Bias and $b_c$ is Tanh Layer Bias.

Before the output gate calculates the output signal, the cell state $C$ needs to be updated, as shown in the ③ area (figure 4). Formula (7) is the updated formula of cell state $C$.

$$C_t = f_t C_{t-1} + i_t \tilde{c}_t$$

where $f_t$ is Forgetting Gate Vector, $C_{t-1}$ is Cell State, $\tilde{c}_t$ is Input Gate Tanh Layer Output and $i_t$ is Input Gate Sigmoid Layer Output.

The output gate $h$ is solved based on the updated $C$. Formula (8) and (9) are output gate $h$ updating formula.

$$o_t = \sigma \left( W_o [h_{t-1}, x_t] + b_o \right)$$

$$b_t = o_t \tanh C_t$$
The final output of LSTM is usually $h_t$ (h of the last unit), and the final output is connected to the full connection layer. If the result of LSTM is output to the convolutional layer (CNN), the $h$ of all sequences can be output as a matrix to the convolutional layer and then connected to the full connection layer.

In the training, the weight parameters of LSTM are updated by neural network back propagation and gradient descent method to minimize the loss function value.

4. Experiment

4.1. Data Processing

Manually filter the question and answer content to remove useless question and answer information, and write it into the text according to a line and a pair of question and answer texts. Read the data with Python Panda library, divide the text into words, remove stop words, punctuation and Numbers, and generate the question and answer corpus. The word vector of word2vec was calculated by Gensim tool and used as the default weight of the embedded layer in the neural network model. At the same time, the weight no longer participates in the parameter adjustment of the whole back propagation.

In order to obtain a fixed-length text length, the histogram of the text length of the question and answer database is shown in figure 5. 200 are manually selected as the text length to cut or supplement the text. The uniform text length is convenient for input.

![Figure 5. Text length bar chart](image)

4.2. Model Building

The agricultural classification model adopts the model structure as shown in figure 6, and the steps are as follows:

1. Data input: the fixed matrix parameters of embedding layer are the word vectors trained by word2vec, and all text is processed into time-series data of fixed length and network input, which is transformed into a two-dimensional matrix through the embedding layer, and each line is a word.

2. Model training: Embedded layer parameters do not participate in the training model uses the obtained
word2vec directly. All the text sequences of the training set are taken as the input layer data; convert to two-dimensional time series data through the embedded layer. Then, the data is passed to the LSTM layer and then to CNN layer. The convolution layers consist of three convolution layer. After maximum pooling, it is connected to the common cosine similarity computing layer, and the loss function is finally defined. The formula (10) of loss function is as follows:

\[ L = \max \left\{ 0, m - \cos \left( V^Q, V^A \right) + \cos \left( V^Q, V^A \right) \right\} \]  

(10)

- \( m \) — Threshold Value
- \( V^Q \) — Problem Eigenvector
- \( V^A \) — Right Answer Eigenvector
- \( V^A \) — Wrong Answer Eigenvector

Using back propagation to update the network parameters, in order to improve the generalization ability of models and avoid over fitting, partial neural connections were randomly discarded and Batch Normalization was added [14, 15].

(3) Model validation: Use the trained model to evaluate the problem. Compare the real answers and verify the accuracy of the model.

4.3. Model Training

We divided 20,000 pieces of data into training set and test set data according to the ratio of 7:3. Batch Size was set as 64 and epoch was set as 50 for model training. The overall training situation is shown in figure 7. With the increase of training times, the value of loss function in the test set is inversely proportional to the accuracy [16]. The value of loss function is on the whole declining trend, and the accuracy is on the rise. The parameters of the neural network in the whole training process are obviously optimized. When the training time reaches about 80 generations, the loss reaches minimum, and then it is in the state of small fluctuation.

![Figure 7. The curve of loss function value and accuracy of test set](image)

4.4. Model Evaluation

Different from the usual accuracy evaluation method, precision and recall are often used to calculate the performance of the model classification in the multi-classification model. The formula is as follows(11)(12):

\[ \text{Precision (P)} = \frac{TP}{TP + FP} \]  

(11)

\[ \text{Recall (R)} = \frac{TP}{TP + FN} \]  

(12)

In the formula, TP is true positive; the model predicted the correct number of positive samples. FP is false positive; the model predicted the wrong number of negative sample. FN is false negative; the model predicted the wrong number of positive samples. In order to comprehensively consider the index evaluation classification model of precision rate and recall rate, f-score (harmonic mean) is obtained. Usually, the accuracy rate is as important as the recall rate. We can set \( \beta \) as 1 to calculate and obtain the value of f1-score. The larger the f1-score, the better the model performance.
score value, the better the classification performance of the model. The formula is as follows (13).

\[
F1\text{-Score} = \frac{2(P \times R)}{(P+R)}
\]  

(13)

4.5. Analysis Result

In the evaluation of the model, we calculated the accuracy rate, recall rate and f1-score under each experiment, and drew the ROC curve to visualize the classification effect of the classification algorithm.

The experiment compares LSTM+CNN with other classification algorithms based on Support Vector Machine (SVM), Multinomial NB and Convolutional Neural Network (CNN). SVM uses linear kernel function to classify multiple classifications. Multinomial NB classifies text using statistical classification methods. CNN has the characteristics of local perception, global sharing and multi-convolution kernels. In the CNN model experiment, a convolution layer with a three-layer convolution kernel of 128, a maximum pooling layer and a three-layer full connection layer (RELU activation function) were used. Finally, the soft max layer was used to output the classification results. In the LSTM model experiment, the LSTM connection with 500 units in one layer and the whole connection layer (RELU activation function) in three layers were used to output the classification results through the soft max layer. LSTM+CNN model integrates the structure of CNN and LSTM, and takes the sequence output of LSTM as the input data of CNN. The classification results are shown in table 3.

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision rate</th>
<th>Recall rate</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>75.55</td>
<td>75.41</td>
<td>75.45</td>
</tr>
<tr>
<td>Multinomial NB</td>
<td>76.78</td>
<td>77.13</td>
<td>76.95</td>
</tr>
<tr>
<td>CNN</td>
<td>77.86</td>
<td>77.74</td>
<td>77.80</td>
</tr>
<tr>
<td>LSTM</td>
<td>78.13</td>
<td>78.09</td>
<td>78.07</td>
</tr>
<tr>
<td>LSTM+CNN</td>
<td>80.36</td>
<td>80.26</td>
<td>80.26</td>
</tr>
</tbody>
</table>

According to the data in table 3, LSTM+CNN have the best effect. Meanwhile, in order to analyze the influence of word vector dimension and the number of LSTM cells (lstm-units) on the classification effect, the change curves of f1-score with the word vector dimension were drawn respectively, and the change curves of f1-score with the number of LSTM cells under the word vector 400 dimension were shown in figure 8.

Figure 8. Change curves of F1-Score with the dimension of word vector and the number of units in LSTM cells

As can be seen from the figure 9, F1-score is slightly affected by the number of units in lstm. Here 500 are chosen as the number of units in lstm. Different dimension of word vector has greater impact on classification performance. The model selects 400 as the dimension of training word vector.

5. Conclusions

(1)This research combines Word2vec with LSTM and CNN to classify agricultural texts. Word2vec vector feature representation is better than TFIDF and bow in resolving the sparse data and the semantic relationship between words and words in text representation. It greatly solves the shortcomings of dimension explosion and insufficient semantic relationship in traditional text feature representation.
The results show, based on the text features of word2vec and the features of lstm-cnn agricultural intelligent question and answer can achieve a good accuracy to a certain extent without the need of artificial extraction of the question and answer data, which can be applied in the question and answer big data.

Acknowledgements

This work was supported by Natural science fund project of science and technology department of Jilin province (20170101051JC); Changchun science and technology project (17DY009); Project of education department of Jilin province (JJKH20190923KJ); Changchun science and technology plan project science and technology innovation “double tenth project” major science and technology project (18SS018); Science and technology development program of Jilin province (20190301024NY).

References