Convolutional neural network model based on text similarity for customer service

Wenshuang Du\textsuperscript{1*}, Jiawei Ge\textsuperscript{2}, Xuchen Liu\textsuperscript{3} and Junjie Ai\textsuperscript{4}

\textsuperscript{1}College of Computer Internet of Things Engineering, University of Hohai, Changzhou Campus, 213022, China

jsdws1999@gmail.com, 2298544852@qq.com, njliuxc@163.com, AiJunJie0014@163.com

Abstract. Customer service is a good way for companies to communicate with customers, high-quality customer service can improve customer satisfaction and dependence on the enterprise. Text matching is a core problem in natural language understanding, and it can be applied to a large number of natural language processing tasks, such as information retrieval, question answering systems, repetition questions, dialogue systems, machine translation. These natural language processing tasks can be approximately abstracted into text matching problems. This paper combines text matching and convolutional neural networks to build an intelligent customer service model, and finally achieves ideal results in F1 value, recall rate and accuracy rate.

1. Introduction
As the scale of the company continues to expand, the number of users continues to increase, and the problems that customer service needs to deal with continue to increase. The manual customer service training cycle is long, the labor cost is high, it is susceptible to emotions, and the problem handling efficiency is low. Due to the popularity of smartphones, the continuous enrichment of communication channels and software functions, traditional corporate customer service is facing more service requirements and more fragmented and diversified customer needs.

Many of the intelligent customer service currently used in the society are unable to accurately answer questions or cannot understand the question correctly, so they still rely heavily on labor.

We plan to build a convolutional neural network model to handle customer service-related business, and find the best answer from a given question Q and an answer candidate pool. Take the cosine similarity operation based on the text vector for Q & A matching, and take out the most matching answer.

The rest of the paper is arranged as follows: Section 2 analyzes the principles of text matching, convolutional neural networks, and word vectors and Bag-of-words models. Section 3 provides a description of the convolutional neural network model used in this article, and the results are shown in Section 4.

2. Algorithm principle and model analysis
This paper uses a parallel structure similar to the Inception convolutional layer, using word vectors and training the convolutional neural network model through text similarity matching. It aims to deepen the depth of the convolutional layer, extract more features, and shorten the model training by matching the word vector dimension and the hidden layer size.
2.1. Text similarity matching

There are many words such as prepositions, adverbs, articles, or conjunctions in sentence expressions that connect words, but they have no specific meaning in sentence expressions and occupy huge vector memory. These are stopwords [3]. We can use sentence segmentation to process sentences, and consider removing stopwords to extract the required keywords. These keywords can reflect the meaning of the whole sentence, and then word vector training is based on this. Establishing a Bag-of-words model for these keywords, and then calculate the matching degree between the question posed by the user and the Q & A database through the cosine similarity calculation, and extract the most matching answer.

\[
similarity = \cos(\theta) = \frac{A \cdot B}{||A|| \cdot ||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}
\]

Figure 1. Cosine similarity calculation.

2.2. Convolutional neural network model

Convolutional neural networks are a type of feedforward neural networks that include convolutional calculations, and are one of the most successful areas of deep learning algorithm applications. Generally, they all include convolution Layer, non-linear transformation layer, pooling layer, and fully connected layer. By extracting the features of the original input, and then extracting and compressing the features, higher-level features are obtained to facilitate the final processing: such as classification, regression.

2.3. Word vector and Bag-of-words model

Bag-of-words model [4] first appeared in the fields of Natural Language Processing and Information Retrieval. The model ignores elements such as grammar and word order of the text, and treats it as just a collection of several words. The appearance of each word in the document is independent. The Bag-of-words model uses an unordered set of words to express a piece of text or a document. The Bag-of-words model of the text is "the histogram obtained from the feature vector [5] of all words in the text".

3. Neural network model description

The network model used in this paper is trained on Q and A in the hidden layer [6] and the CNN layer, and then pooling and non-linear mapping are performed separately. Finally, the cosine similarity between the two is calculated through the hidden layer changes.

First, doing word vector conversion and embedding based on the bag of words model for customer’s ask in the database VQ. We add a negative answer VA on the basis of matching the positive answer VA+. Subsequently, the cosine similarity \( \cos (V_Q, V_+ (A+)) \) and \( \cos (V_Q, V_- (A-)) \) are calculated so that \( \cos (V_Q, V_+ (A+)) \cos (V_Q, V_- (A-)) < m \) (formula a). Set the value of margin (m) to determine whether to update the parameters. If the formula a does not hold, the parameters will not be updated.

Figure 2. Block diagram of the model used in this article
This means that by using the margin value as an indicator, we can make the similarity between the positive answer and the standard question higher and higher, and the lower the similarity between the negative answer and the standard question. Next, Q & A enters the HL layer for non-linear transformation. The results of the non-linear changes are then passed to the convolutional layer, the information is extracted, and the Relu transform is performed [7]. Finally, the optimal weight matrix is improved by calculating the cosine similarity of the output in vector form.

According to the data set test, adding the HL layer greatly improves the model effect, but it will significantly increase the length of the training model. We have defined 2000 filters in the CNN layer. And use the Relu function for activation, then using max_pool dimension reduction sampling for the activated vector. We analyze the Inception structure. In the code, the structure of Inception [8] is used to make the convolution layers in parallel. Finally, the vectors will be obtained after the dimensionality reduction sampling are stitched to deepen the depth of the convolution layer.

Calculate the similarity between the user's question and the positive and negative answers, and define the loss function as shown in equation:

\[ \text{Loss} = \max\{0, m - \cos(V_{Q}, V_{A+}) + \cos(V_{Q}, V_{A-})\} \]

In addition, it should be noted that when building the model, parameters need to be modified to avoid gradient explosion or the disappearance of the gradient, so that the trained model has more accurate parameters. The L2 canonical formula is used in the process of training the model. (weight 0.001, learning rate 0.01, judgment threshold 0.009) and dropout to avoid overfitting problems, improve the generalization ability of the model and make the result more stable. The dropout retention rate needs to be weighed according to the accuracy rate and training time out.

4. Results and conclusions
The data we obtained through web crawlers was used as the test set. On the test set, our model can reach an average F1 value of 0.8879, a recall rate of 0.8909, a TOP 1 accuracy rate of 0.8865, and a TOP5 accuracy rate of 0.9973. When optimizing the model, we reached the following conclusions:

- The use of Dropout does not have a great impact on the highest accuracy, but the results using Dropout are more stable, and the fluctuation of accuracy will be smaller, so we choose to use Dropout. However, Dropout should not be used too much. For example, if the keep_prob probability of Dropout is set to 0.25, the model will converge more slowly, the training time will be much longer, the effect will be worse, and it is difficult to set. In the optimization of the model, we use a keep_prob of 0.5, while ensuring accuracy and training time. In addition, Dropout is only applied to the results of max-pooling, not using Dropout too much.

- The effect of the HL layer is more obvious, but if the size of the HL layer is 200 and the word vector is 100, then the HL layer is equivalent to double the word vector, and the available information is limited. Therefore, the length of the word vector and the size of the HL layer must be matched,
eliminating the size conversion of the HL layer, which can speed up the model training speed without reducing the accuracy too much.

- The larger the number of filters does not mean that the effect is better, but basically it is difficult to improve to a certain degree, but it will reduce the training speed.

References

[1] Blodgett, Jeffrey G., K. L. Wakefield, and J. H. Barnes. "The effects of customer service on consumer complaining behavior." Journal of Services Marketing 9.4(1995):31-42.

[2] Li, Baoli, and L. Han. Distance Weighted Cosine Similarity Measure for Text Classification. Intelligent Data Engineering and Automated Learning – IDEAL 2013. Springer Berlin Heidelberg, 2013.

[3] Wilbur, W. J., and K. Sirotkin. "The automatic identification of stop words." Journal of Information Science 18.1(1992): 45-55.

[4] Heap, Bradford, Bain, Michael, Wobcke, Wayne, Krzywicki, Alfred, & Schmeidl, Susanne. Word vector enrichment of low frequency words in the bag-of-words model for short text multi-classification problems.

[5] Walcher, Sebastian. "Eigenvectors of Tensors—A Primer." Acta Applicandae Mathematicae.

[6] Guang-Bin Huang. "Learning capability and storage capacity of two-hidden-layer feedforward networks." IEEE Trans Neural Netw 14.2(2003):274-281.

[7] Li, Yuanzhi, and Yuan, Yang. "Convergence Analysis of Two-layer Neural Networks with ReLU Activation."

[8] Szegedy, Christian, Ioffe, Sergey, Vanhoucke, Vincent, & Alemi, Alex. Inception-v4, inception-resnet and the impact of residual connections on learning.