An Intelligent Question and Answering System for Dental Healthcare

Yan Jiang, Yueshen Xu(✉), Jin Guo, Yaning Liu, and Rui Li

School of Computer Science and Technology,
Xidian University, Xi'an 710071, China
{yjiang_2, jguo_2, ynliul}@stu.xidian.edu.cn,
{ysxu, rli}@xidian.edu.cn

Abstract. The intelligent question and answering system is an artificial intelligence product that combines natural language processing technology and information retrieval technology. This paper designs and implements a retrieval-based intelligent question and answering system for closed domain, and focuses on researching and improving related algorithms. The intelligent question and answering system mainly includes three modules: classifier, Q&A system and Chatbots API. This paper focuses on the classifier module, and designs and implements a classifier based on neural network technology, mainly involving word vector, bidirectional long short-term memory (Bi-LSTM), and attention mechanism. The word vector technology is derived from the word2vec tool proposed by Google in 2013. This paper uses the skip-gram model in word2vec. The Q&A system mainly consists of two modules: semantic analysis and retrieval. The semantic analysis mainly includes techniques such as part-of-speech tagging and dependency parsing. The retrieval mainly relates to technologies such as indexing and search. The Chatbots API calls the API provided by Turing Robotics. The intelligent question and answering system designed and implemented in this paper has been put into use, and the user experience is very good.

Keywords: Question and answering · Word2vec · Skip-gram · Bi-LSTM · Attention mechanism · Part-of-speech

1 Introduction

The intelligent question and answering system is a new type of information service system that combines natural language processing and information retrieval technologies. The intelligent question and answering system can be divided into closed domain and open domain according to the knowledge field [1]. The closed domain focuses on answering specific domain questions. The questioner can only ask some domain related questions and get answers. The open domain system does not set the scope of the problem. The questioner can come up with any topic of interest to him, and can get the answer he wants from the system. According to the principle of answer generation, the question and answering system is divided into generation and retrieval.

At present, Most of the popular question and answering robots are based on open domain intelligent question and answering systems, such as Microsoft Xiaobing based on Internet corpus and user click logs, and Baidu voice assistants based on Baidu search logs. These open intelligent robots cannot accurately answer questions about specific domains such as taxation, finance, oral, etc., while traditional artificial services have problems such as slow response and high labor costs. This paper implements a specific domain (Prosthodontics), search-based intelligent question and answering system.

The intelligent question and answering system consists of a classifier, Q&A system, and Chatbots API. The workflow of the intelligent question and answering system is that the user asks questions, and then the classifier classifies the problems. This paper has set up two categories, ordinary chat and Prosthodontics. If the problem is assigned to the Prosthodontics, the Q&A system designed by this paper provides answers to the questions. If the question is assigned to the ordinary chat, the answer provided by the Chatbots API.

The classifier is the focus of this paper. In order to classify such short questions, this paper adopts the popular neural network technology in recent years to design classifier. At first, the Bi-LSTM neural network model is used to construct the classifier. The accuracy of the classifier can only reach about 80%. Later, this paper employs the attention mechanism, which is emerging in the field of machine translation and is now widely applied in various fields of artificial intelligence, and has achieved good results [10–13, 16]. The accuracy of the classifier can reach about 87%.

Another focus of this paper is the design of the Q&A system. The Q&A system has two modules, semantic analysis and retrieval. The semantic analysis module includes Chinese word segmentation, part-of-speech tagging, and dependency parsing. They are also the three basic tasks in natural language processing (NLP). After the dependency paring, according to the characteristics of the Prosthodontics data, this paper designed some filter rules for dependency, which has obvious effects in practical applications. The retrieval module is implemented by Appach Lucene. Lucene has incremental indexing technology, which allows users to add, change, or remove documents in their index without the need to re-index all contents within the index. By using timestamps, the index will identify the documents that were newly created, modified, or deleted, and what will be re-indexed are only those contents related to those documents. Lucene has a world-class search technology, especially its field search functionality is very suitable for the application scenario of this paper.

The last module of this paper is Chatbots API. This paper calls the API provided by Turing Robotics. After testing the user experience is very good, the user can ask any domain of questions, it can give a suitable answer.

2 The Architecture of the Intelligent Question and Answering System

Figure 1 presents the whole architecture of the intelligent question and answering system, which contains four modules and each module is given a brief explanation as follows.

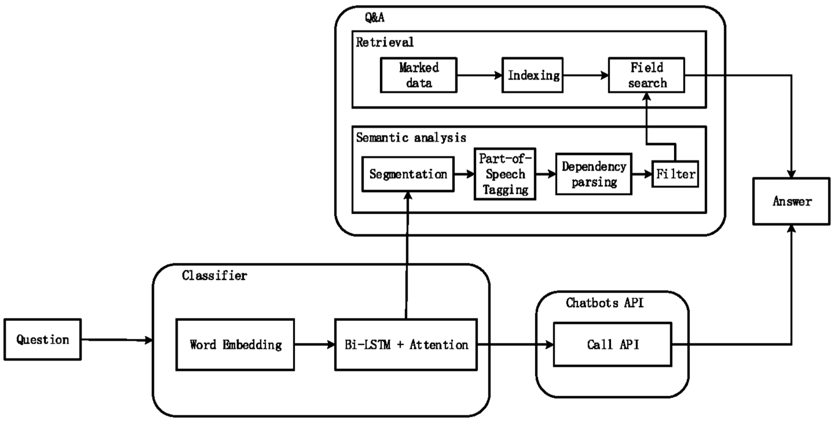


Fig. 1. The architecture of the intelligent question and answering system

Classifier. The classifier is an important module of the intelligent question answering system and the focus of this paper. The function of the classifier is to classify the questions asked by the user. This paper has set up two categories, ordinary chat and Prosthodontics. This paper designs and implements a classifier based on neural network technology.

Q&A System. The Q&A system is also the focus of this system. If the user's question is assigned to Prosthodontics, the Q&A system will answer the user's question. The Q&A system mainly involves two modules, semantic analysis and retrieval. After the semantic analysis of the user's question is completed, the search module will search with its result. If a similar problem is found, the search module will return the answer corresponding to the question.

Chatbots API. If the user's question is assigned to the ordinary chat category, this paper will call Turing Robot's API to answer the user's question.

3 The Classifier

This paper employs the 'jieba' Chinese segmentation tool to segment words for the Chinese Wikipedia dataset, and a 50-dimensional word vectors are trained using the skip-gram model in word2vec. For the professional dental dataset, this paper adopts the same tool. Through that process, the trained word vectors are obtained, which are used to encode the professional dental data. Then input those vectors into the classifier for training. The classifier model uses a Bi-LSTM model, incorporating the Attention mechanism.

The 'jieba' Chinese word segmentation supports three word segmentation modes: (1) the precise mode; it can cut the sentence most accurately and is suitable for text analysis; (2) the full mode; it can scan all the words that can be formed into words in sentences. It is very fast, but cannot solve ambiguity; (3) the search engine mode; it can split long words again on the basis of precise mode. It improves the recall rate and is suitable for search engine segmentation.

This paper adopts the search engine model. Although we are now in an era of data explosion, the data that can be used to train the classifier is still very small. The data used in this paper is labeled and processed by professional doctors. Search engine mode can increase the vocabulary and it has obvious effect in later practical applications. The ‘jieba’ word segmentation can be loaded into the developer’s own defined dictionary to contain words that are not in the ‘jieba’ lexicon. Although ‘jieba’ has the ability of recognizing new words, self-adding new words on its own can guarantee higher accuracy. Moreover, the ‘jieba’ word segmentation not only can dynamically modify the dictionary in the program, but can also adjust the word frequency of a single word so that it can (or cannot) be separated.

Word2vec is a two-layer neural net that processes text [3]. Its input is a text corpus and its output is a set of vectors: feature vectors for words in that corpus. It does so in one of two ways, either using context to predict a target word (a method known as continuous bag of words, or CBOW), or using a word to predict a target context, which is called skip-gram. So we use the latter method because it produces more accurate results on large datasets (Fig. 2).

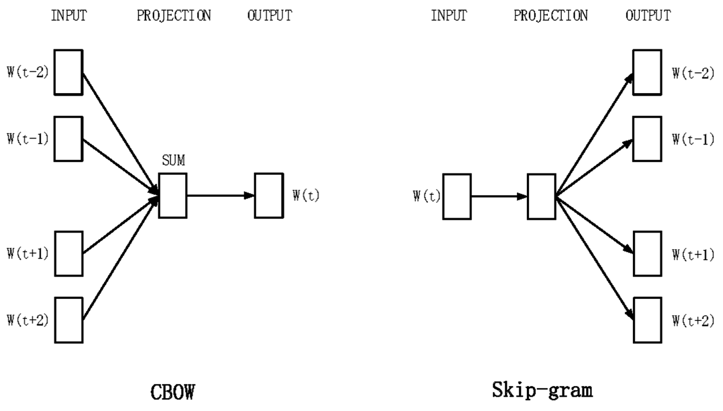


Fig. 2. Architecture of the CBOW and the Skip-gram.

The training objective of the Skip-gram model is to find word representations that are useful for predicting the surrounding words in a sentence or a document [4]. More formally, given a sequence of training words $w_1, w_2, w_3, \ldots, w_T$, the objective of the Skip-gram model is to maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c \le j \le c, j \ne 0} \log p\left(w_{t+j}|w_t\right) \tag{1}$$

where c is the size of the training context (which can be a function of the center word w_t). The basic Skip-gram formulation defines $p\left(w_{t+j}|w_t\right)$ using the softmax function:

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^T v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_{w_O}}^T v_{w_I}\right)} \tag{2}$$

where v_w and v'_w are the "input" and "output" vector representations of w, and W is the number of words in the vocabulary. This formulation is impractical because the cost of computing $\nabla \log p(w_O + w_I)$ is proportional to W.

This paper uses the negative sampling [3] method in the skip-gram model to solve this problem. We define Negative sampling (NEG) by the objective

$$\log\sigma\left({v'_{w_O}}^T v_{w_I}\right) + \sum_{i=1}^{k} \mathrm{E}_{w_i \sim P_n(w)}\left[\log\sigma\left(-{v'_{w_I}}^T v_{w_I}\right)\right] \tag{3}$$

which is used to replace every $\log p(w_O|w_I)$ term in the Skip-gram objective. In this paper, the unigram distribution is used to select negative samples. The probability calculation formula for each word being selected as a negative sample is related to the frequency of its occurrence. The formula used for the implementation in the code is (Fig. 3):

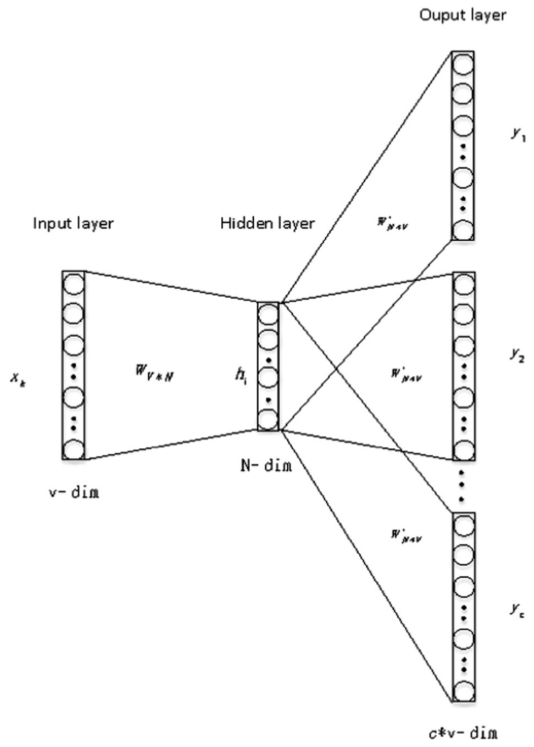


Fig. 3. Neural network model of Skip-gram

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=0}^{n}\left(f(w_i)^{3/4}\right)} \tag{4}$$

Among them, v is the vocabulary size; N refers to the dimension of the word vector; c is twice the size of the word vector window; w refers to the central word vector matrix; w' is the matrix formed by context words' vectors. In this paper, central word vector matrix w is used as the final word vector. This paper does not employ the traditional recurrent neural network model in the design of the classifier of neural network model. Because the traditional recurrent neural network model is prone to the problems of gradient disappearance and gradient explosion during training. In order to solve this problem, this paper adopts the LSTM network [4], a variant of the recurrent neural network.

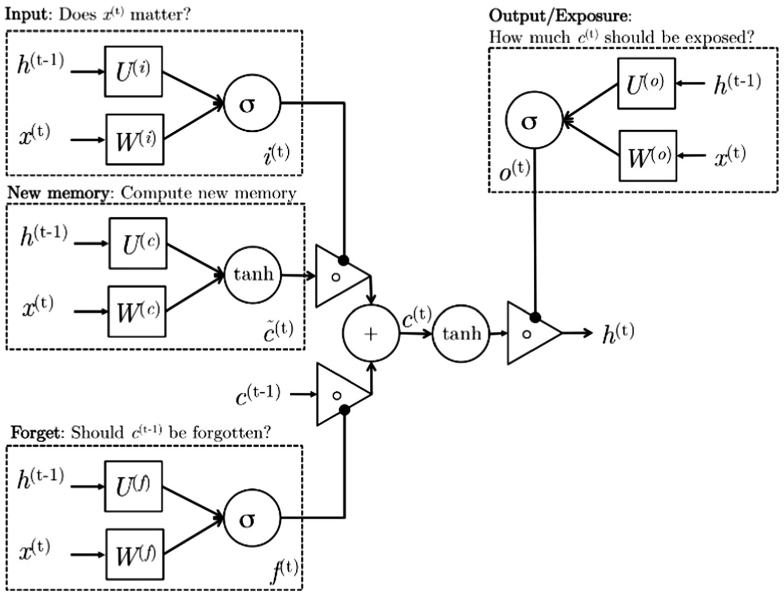


Fig. 4. Internal structure of LSTM

The Fig. 4 is described by the following formula:

$$\begin{aligned}
i^{(t)} &= \sigma\left(w^{(i)}x^{(t)} + U^{(i)}h^{(t-1)}\right) && \text{(Input gate)}\\
f^{(t)} &= \sigma\left(w^{(f)}x^{(t)} + U^{(f)}h^{(t-1)}\right) && \text{(Forget gate)}\\
o^{(t)} &= \sigma\left(w^{(o)}x^{(t)} + U^{(o)}h^{(t-1)}\right) && \text{(Output gate)}\\
\tilde{c}^{(t)} &= \sigma\left(w^{(c)}x^{(t)} + U^{(o)}h^{(t-1)}\right) && \text{(New memory cell)}\\
\mathrm{c}^{(t)} &= i^{(t)} \circ \tilde{c}^{(t)} + f^{(t)}c^{(t-1)} && \text{(Final memory cell)}\\
h^{(t)} &= o^{(t)} \circ \tan h\left(c^{(t)}\right)
\end{aligned} \tag{5}$$

The calculation process of LSTM is as follows:

1. The New memory cell uses the word $x^{(t)}$ at the current moment and the hidden state $h^{(t-1)}$ at the previous moment to generate a new memory $\tilde{c}^{(t)}$. So the new memory contains the attributes of the current word $x^{(t)}$.
2. The Input Gate uses the word $x^{(t)}$ at the current moment and the hidden state $h^{(t-1)}$ at the previous moment to determine how much the attribute of the word at the current moment should be kept. What is kept refers to $i^{(t)}$.
3. The Forget Gate uses the word $x^{(t)}$ at the current moment and the hidden state $h^{(t-1)}$ at the previous moment to determine how much the past memory $c^{(t-1)}$ should be forgotten. What is forgotten refers to $f^{(t)}$.
4. The Final memory cell adds up the new memory retained by Input Gate and the past memory forgotten by Forget Gate to generate the final memory $c^{(t)}$.
5. The Output Gate uses the word $x^{(t)}$ at the current moment and the hidden state $h^{(t-1)}$ at the previous moment to determine how much new memory $\tan\mathrm{h}\left(c^{(t)}\right)$ should be output, and the output amount is represented by $o^{(t)}$.

As the basic structure of the neural network model, the network structure diagram of the Bi-LSTM used in this paper is shown in Fig. 5 [14, 15]. In the built Bi-LSTM, the *jth* hidden state $h_j \rightarrow$ can only carry the *jth* word itself and a part of information in previous words. If the input is in reverse order, $h_j \leftarrow$ carries the *jth* word and a part of information in posterior words. Combining $h_j \rightarrow$ and $h_j \leftarrow$, $h_j\left[h_j \rightarrow, h_j \leftarrow\right]$ can contain the information before and after the *jth* word.

The built classifier aims to classify the questions raised by users, most of which are short sentences. Therefore, our system adds the attention mechanism [5] to the basic LSTM network. The Attention mechanism is to assign different weights to different words in a sentence, which is likely to improve the accuracy of the classifier, especially in the case of short sentences. Attention mechanism for the neural network in this paper

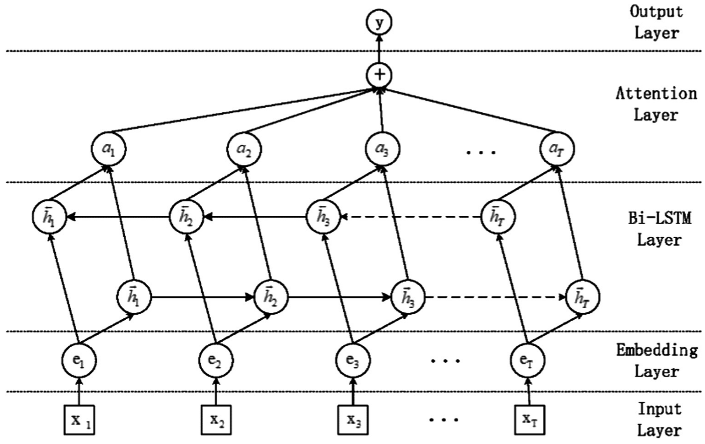


Fig. 5. Neural network model of the classifier

is to add a parameter to each moment of Bi-LSTM after the Bi-LSTM layer. This parameter is randomly given at the beginning, and then it is revised through continuous training. Figure 5 shows the neural network model of the classifier. The diagram contains Bi-LSTM and Attention mechanism.

4 Q&A System

In Sect. 3.1, we focus on how to classify the questions asked by users. In this section and the next section, we will focus on how to provide answers to questions asked by users. The answer comes from two modules: the self-designed search-based Q&A system; Call the more mature Chatbots API of the current technology. If the user's question is assigned to the oral prosthetics field, the Q&A system will provide the answer, otherwise the Chatbots API will provide the answer. The Q&A system consists of two modules, semantic analysis and retrieval.

4.1 Semantic Analysis

The semantic analysis module mainly performs two tasks, part-of-speech and dependency parsing. This paper uses the Stanford Corenlp tool to accomplish these tasks.

Syntactic parsing is one of the key techniques in natural language processing (NLP). Its basic task is to determine the syntactic structure of a sentence or the dependence of words in a sentence. In general, syntactic parsing is not the ultimate goal of NLP task, but it is often an important link or even a key link to achieve the ultimate goal.

Syntactic parsing is divided into two types: syntactic structure parsing and dependency parsing. Syntactic structure parsing refers to determining whether a composition of a word sequence (usually a sentence) conforms to a given grammar and analyzes a syntactic structure that conforms to grammar. Syntactic structure is generally represented by the tree data structure, typically referred to as syntactic parse trees. Dependency parsing refers to the analysis of the dependence between words and words for a sentence. In fact, we sometimes don't need or need to know the phrase structure tree of the whole sentence, and we must know the dependence between words and words in the sentence, so does this paper. "Dependence" refers to the dominant and dominated relationship between words. This relationship is not equal and has a direction. The dominant component is called the governor, and the dominated component is called the modifier. There are four rules for dependency parsing [6]:

(1) A sentence has only one independent component;
(2) The other components of a sentence are subordinate to a certain part;
(3) No single component can be dependent on two or more components;
(4) If component A is directly subordinate to component B, and component c is located between A and B in the sentence, component c is either subordinate to A, or subordinate to B, or subordinate to a component between A and B.

These four rules are equivalent to the formal constraints on the dependency graph and the dependency tree: single, headed, connective, acyclic and projective, and used to ensure that the dependency parsing result of the sentence is a tree with “root”. This lays the foundation for the formal description of dependent grammar and its application in computer linguistics. Currently, there are two ways to do dependency parsing: rules based (rules designed by people), based on neural networks [2, 9]. With the rapid growth of marking datasets, neural network-based dependency parsing has become increasingly popular. Labeling datasets is time-consuming and labor-intensive, but with the development for many years, the number is growing. Everyone prefers to use an annotated dataset because its work for each process can be reused, such as part-of-speech tagging, named entity recognition, and so on. There are many shortcomings in rule-based dependency analysis: the rules are relatively rigid; moreover, there are many rules, which are difficult to unify. Therefore, this paper uses a neural network-based dependency parser to do dependency parsing analysis. This paper uses the dependency parsing analyzer of the Stanford Corenlp toolkit to implement dependency parsing. Figure 6 shows the dependency tree after dependency parsing of a sentence.

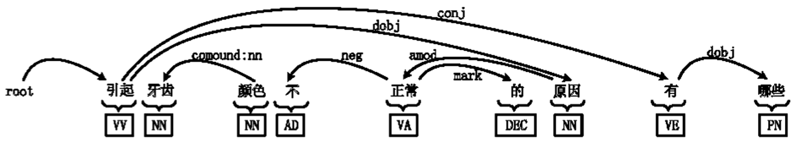


Fig. 6. Dependency tree

The dependencies contained in Fig. 6 are shown in Table 1:

Table 1. Dependency results

Dependency
(root,ROOT,引起#VV)(compound:nn,颜色#NN,牙齿#NN)(nsubj,正常#VA,颜色#NN)(neg, 正 常 #VA, 不 #AD)(amod, 原 因 #NN, 正 常 #VA)(mark, 正 常 #VA, 的#DEC)(dobj,引起#VV,原因#NN)(conj,引起#VV,有#VE)(dobj,有#VE,哪些#PN)

Analysis of the results of a large number of dependency parsing, combined with the analysis of the search results in next section, the paper finally determined that only eight dependencies were retained: amod (adjective modifier), compound:nn (noun modifies noun), advmod (adverb modifies adjective), nsubj (nounsubjective), dobj (direct objective), det (determiner), prnmod (parenthetical modifier), assmod (associative modifier). This paper also establishes filtering rules for each of these dependencies. For example, amod’s filtering rules:

a. Both modified words and modifiers can only be Chinese characters at the same time, and cannot contain meaningless symbols such as punctuation.

b. Only retain the modified word part as NN (common noun), and modify the part of speech as JJ (adjective or numeral, ordinal).

Due to the limited length of the paper, the filtering rules for each dependency are not described here. The results of the dependency filtering for the example sentences in Fig. 6 are shown in Table 2.

Table 2. Selected dependency

Selected dependency
(dobj, 引起#VV, 原因#NN)(compound:nn, 颜色#NN, 牙齿#NN)(nsubj, 正常#VA,颜色#NN)(amod,原因#NN,正常#VA)(dobj,有#VE,哪些#PN)

The results after the combination of Table 2 dependency filter results are shown in Table 3.

Table 3. Sentence

Sentence
引起 牙齿 颜色 正常 原因 有 哪些

4.2 Retrieval

The retrieval module consists of two main steps, indexing and searching. Indexing means that the system uses the question and answer pairs marked by the professional doctor to establish an index; searching refers to the system using the results of the semantic analysis module to search for similar problems. If it is found, the answer corresponding to the question is returned. Implementation of the retrieval module employs Apache Lucene.

Apache Lucene is an ultra-fast, powerful, and full-featured text search engine library developed to empower mobile applications, websites, and solutions with search capabilities so they can deliver a seamless search experience. Written in Java, Apache Lucene functions as an indexing and search technology that allows the implementation of full-text search capabilities and efficient indexing processes on applications and websites that are running on multiple platforms.

As an indexing technology, Apache Lucene offers a feature called incremental indexing. This feature allows users to add, change, or remove documents in their index without the need to re-index all contents within the index. By using timestamps, the index will identify the documents that were newly created, modified, or deleted, and what will be re-indexed are only those contents related to those documents. This feature is very advantageous for users who are managing a large database and who need to change their data. The indexing capability of Apache Lucene is scalable and has a high performance. In fact, it lets users process a large number of documents and index them in less time. This is because the search engine library supports the indexing of over 50 GB of documents or data within an hour. In addition, as they perform indexing, they only need to allocate a minimal amount of RAM for such task which is 1 MB heap.

Apache Lucene makes it easy for searchers to find any information or content they need. Through the aid of its powerful, accurate, and efficient search algorithms, users will be able to precisely deliver whatever searchers are looking for. This makes the search engine library a truly world-class search technology that can handle sophisticated searches and generate search results in an advanced and modern way. For example, accurate searching can be implemented by enabling a fielded search functionality on websites and applications. Basically, searchers are looking for pieces of information that are stored in a database and are defined and labelled in the search index during the indexing process. These pieces of information are organized in specific areas called fields. In this paper, question, answer and disease fields are established when indexing. Using Lucene's fielded search functionality to search only in question field. In the process of searching, this paper also adds some rules, for example: import synonym dictionary to improve the hit rate; Repeated searches for some keywords help to improve the ranking of keyword-related questions [8].

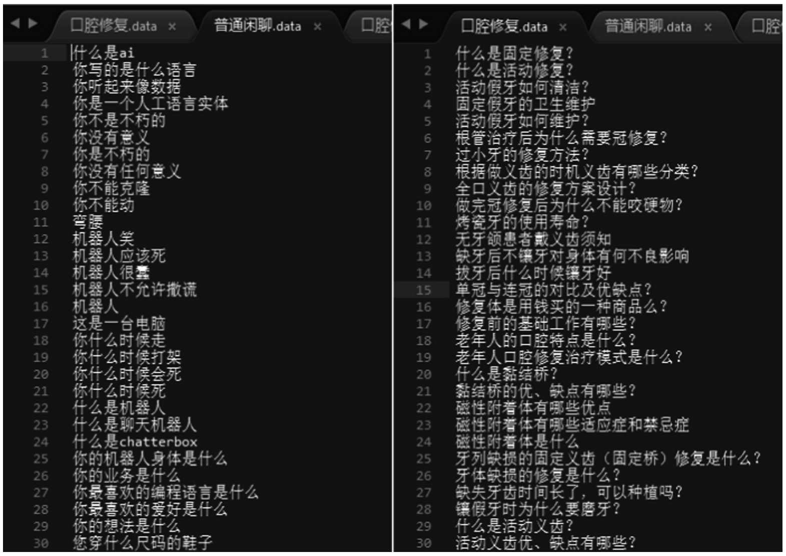
Lucene enables users to rank their search results so that searchers will be able to see first the contents that are most relevant to what they are looking for. In addition, the search engine library allows users to use different ranking models as they rank their search results which include Vector Space Model and Okapi BM25 [3]. This paper uses Vector Space Model to rank.

5 Chatbots API

When the classifier in Sect. 3.1 divides the problem into the ordinary chat field, then the user's question is answered by the chatter robot. This paper calls Turing Robotics' Chatbots API to implement this function. The details of the technology are not covered here.

6 Experimental Results

In this section, this paper tests the performance of the classifier described in Sect. 3 and delay of the intelligent question and answering system. The data set used in this paper has two parts, ordinary chat data sets and Prosthodontics data sets. Prosthodontics data is some real patient problems crawled from the medical website, such as Chunyuyisheng, Haodaifu, etc., and then marked by a professional dentist. Ordinary chat data is crawled from the web. The two-part data format is shown in Fig. 7:



```
什么是ai
你写的是什么语言
你听起来像数据
你是一个人工语言实体
你不是不朽的
你没有意义
你是不朽的
你没有任何意义
你不能克隆
你不能动
弯腰
机器人笑
机器人应该死
机器人很蠢
机器人不允许撒谎
机器人
这是一台电脑
你什么时候走
你什么时候打架
你什么时候会死
你什么时候死
什么是机器人
什么是聊天机器人
什么是chatterbox
你的机器人身体是什么
你的业务是什么
你最喜欢的编程语言是什么
你最喜欢的爱好是什么
你的想法是什么
您穿什么尺码的鞋子
```

```
什么是固定修复?
什么是活动修复?
活动假牙如何清洁?
固定假牙的卫生维护
活动假牙如何维护?
根管治疗后为什么需要冠修复?
过小牙的修复方法?
根据做义齿的时机义齿有哪些分类?
全口义齿的修复方案设计?
做完冠修复后为什么不能咬硬物?
烤瓷牙的使用寿命?
无牙颌患者戴义齿须知
缺牙后不镶牙对身体有何不良影响
拔牙后什么时候镶牙好
单冠与连冠的对比及优缺点?
修复体是用钱买的一种商品么?
修复前的基础工作有哪些?
老年人的口腔特点是什么?
老年人口腔修复治疗模式是什么?
什么是黏结桥?
黏结桥的优、缺点有哪些?
磁性附着体有哪些优点
磁性附着体有哪些适应症和禁忌症
磁性附着体是什么
牙列缺损的固定义齿(固定桥)修复是什么?
牙体缺损的修复是什么?
缺失牙齿时间长了, 可以种植吗?
镶假牙时为什么要磨牙?
什么是活动义齿?
活动义齿优、缺点有哪些?
```

Fig. 7. Ordinary chat data sets and Prosthodontics data sets

6.1 Classifier Performance Test

The performance metric of the evaluation classifier is generally the classification accuracy, which is defined as the ratio of the number of samples correctly classified by the classifier to the total number of samples for a given test data set. The formula is as follows:

n – the number of samples correctly classified
N – the total number of samples

$$Accuracy = \frac{n}{N} \tag{6}$$

Evaluation indicators commonly used for the two-class classification are precision and recall [7]. Usually, the class of interest is positive, the other classes are negative, and the prediction of the classifier on the test data set is either correct or incorrect, the total number of occurrences of the four cases is recorded as:

TP – Positive class is predicted as positive class
FN – predict positive classes as negative classes
FP – predict negative classes as positive classes
TN – predict negative classes as negative classes.

The precision is defined as:

$$P = \frac{TP}{TP + FP} \tag{7}$$

The recall is defined as:

$$R = \frac{TP}{TP + FN} \tag{8}$$

In addition, there is an F1 value, which is the harmonic mean of the precision and recall. It is defined as:

$$\frac{2}{F_1} = \frac{1}{P} + \frac{1}{R} \tag{9}$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{10}$$

When both the precision and the recall are high, the F1 value will be high.

In this paper, the four performance indicators of the three classifiers described in Sect. 3.1 are tested separately. The results are shown in Table 4:

Table 4. Classifier performance test

	Precision	Recall score	F1 score	Accuracy
The classifier	0.8923	0.9134	0.9027	0.8780

6.2 System Delay Test

The intelligent question and answering system provides a restful interface that can be easily embedded in any program. The intelligent question and answering system includes classifiers, Q&A, and chat API. These modules all involve certain delay problems. In this paper, the response time of the system is tested for different lengths of the user's question. The results are shown in Fig. 8. It can be seen from the Fig. 8 that

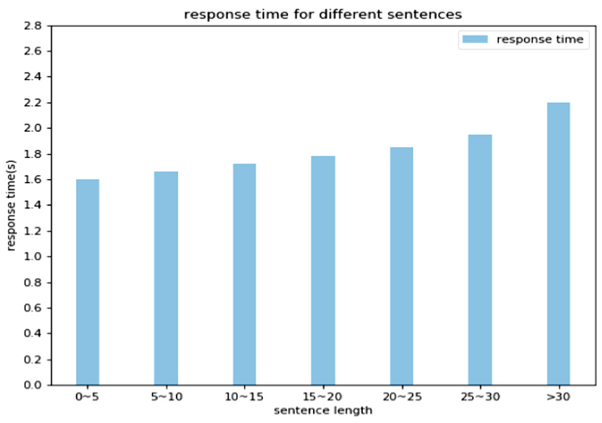


Fig. 8. System delay test

as the length of the user's problem increases, the response time of the system gradually increases. However, the overall response of the system is still very fast, which will give users a good experience.

7 Conclusion

This paper proposes three modules of classifier, Q&A and Chatbots API, which constitute a complete intelligent question and answering system. The paper focuses on the design process of the classifier. It is concluded that the effect of neural network technology on small and medium-sized data sets is not as good as the traditional statistical learning method. In the case of the data set used in this paper, AdaBoost is better than non-linear SVM. As for the dependency parsing in the semantic analysis module of Q&A, this paper does not use rule-based dependency parsing while adopts neural network technology-based dependency parsing, the effect is really better. According to the characteristics of users' questioning question and the search module, this paper designs some filtering rules of dependency relationship, so that the semantic analysis module analysis is very good.

In the future, this intelligent question and answering system will continue to collect corpus marking it by a professional dentist, and then import it into the database, so that the intelligent question and answering system can answer more questions. Will continue to work to improve the accuracy of the classifier on short problem data sets, to study dependency parsing to make it even better. The search algorithm will continue to be improved, so that users to get a more appropriate answer.

Acknowledgement. This paper is supported by Fundamental Research Fund for Central Universities (No. JBX171007), National Natural Science Fund of China (No. 61702391), and Shaanxi Provincial Natural Science Foundation (No. 2018JQ6050).

References

1. Pundge, A.M., Khillare, S.A., Mahender, N.: Question answering system, approaches and techniques: a review. Int. J. Comput. Appl. **141**, 0975–8887 (2016)
2. Li, H., Zhang, Z., Ju, Y., Zhao, H.: Neural character-level dependency parsing for Chinese. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI), pp. 5205–5212 (2018)
3. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of Workshop at ICLR (2013)
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. 1735–1780 (1997)
5. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv, pp. 1409–0473 (2014)
6. Chengqing Zong: Statistical natural language processing. 155–158 (2013)
7. Li, H.: Statistical Learning Methods. Tsinghua Press, Beijing (2012)
8. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. J. Doc. (1972)

9. Chen, D., Manning, C.D.: A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 740–750 (2014)
10. Vaswani, A., et al.: Attention is all you need. Neural Information Processing Systems (NIPS) (2017)
11. Mnih, V., Heess, N., Graves, A.: Recurrent model of visual attention. Advances in Neural Information Processing Systems 27 (NIPS) (2014)
12. Ba, J., Mnih, V., Kavukcuoglu, K.: Multiple object recognition with visual attention. arXiv (2014)
13. Xu, K., et al.: Show, attend and tell: neural image caption generation with visual attention. In: International Conference on Machine Learning (ICML) (2015)
14. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
15. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. **18**(5–6), 602–610 (2005)
16. Zhang, M., Wang, N., Li, Y., Gao, X.: Neural probabilistic graphical model for face sketch synthesis. TNNLS (2019)