Customer Service Automatic Answering System Based on Natural Language Processing

Xia GONG[1], Zhu Jun ZHANG[1], Zi Xiong ZHANG[1], Xiang Yi KONG[2], Lin TAN[2]
[1]Beihang University, 100191,China
[2]Beijing Institute of Technology, 100081,China
[2]Beijing Foreign Studies University, Department of finance, 100089,China
Email: soooda@buaa.edu.cn, mojinxinzhu@163.com, zzx_2015@buaa.edu.cn,
1120161805@bit.edu.cn, tanlin@bfsu.edu.cn

Abstract: *With the rapid development of Internet, information grows explosively, and traditional search engine have failed to meet the needs of users. This paper proposes a customer service automatic answering system with a high-quality knowledge base. First of all, based on unsupervised learning algorithm, this system extracts the question and answer pairs from documents and store them in the knowledge base. Then employing semantic analysis module and the method of Natural Language Processing (NLP), this system gains the meaning of the customers' question accurately, then retrieve the knowledge base and return a high-resolution answer to the user. Furthermore, we construct a dialog management module, which makes reasonable guesses on issues that cannot be matched, and records the dialogue history so that the question-answering system can give more intelligent responses. Finally, due to the diversity of the document structure and the complexity of Chinese natural language, this system adds an edifying function that can add, delete, and modify the question and answer pair in the knowledge. Therefore, our customer service automatic answering system can be more intelligent and efficient than the existing question and answer system.*

Keywords: *Natural Language Processing, unsupervised learning algorithm, Question-answering system, Customer-service*

1 Introduction

Questioning and answering is one of the fundamental ways for human beings to acquire knowledge. In order to gain the information in a more efficient way, an automatic question answering system (QA System) are designed to answer the questions raised by the users on the internet. As a typical application of natural language processing, this system no longer needs the users to retrieve content by keywords query and can automatically provide the professional answers. However, current QA systems often focus on a small part of the professional field. Therefore, designing a smart customer service answering system dedicated to professional fields becomes very important. The knowledge base has been one of the key competitiveness of the intelligent customer service answering system.

Recent years, the rapid growth of information on the Internet and the formation of community websites such as Wikipedia and Quora provides a favorable condition to build a common knowledge base. Many companies and research institutes are committed to developing intelligent question-and-answer robots based on a common knowledge base, such as Apple Siri, Microsoft Cortana, and so on. In the academic field, in 2011, the Watson robot developed by IBM won the championship in the popular intelligence quiz TV program "Dangerous Edge" in the United States. Wolfram Alpha, a new search engine launched by Wolfram Research, can directly return the answer to the user. This type of open domain question answering system is more suitable for answering factual questions such as names, time, events, etc. However, it can't respond accurately to a slightly more complicated problem.

At present, scholars have not yet paid enough attention and practice to the construction of the QA system in more professional field. There are a large number of document formats, such as product documentation and user guides in the website. However, users need to search them one by one to find answers to their own problems. It's not efficient enough. Some companies also establish some manual customer service systems for answering questions, but the manual labor costs are very high. Researchers realize that the professional QA system can play the role of intelligent customer service, saving lots of costs, having the widespread commercial value and the prospects for development. Therefore, it is very meaningful to develop a professional QA system.

Under this background, we designed a customer service automatic answering system that can enter the standardized product documentation, automatically extract question and answer pairs, build a knowledge base, and answer user questions in a professional way. The system takes the advantages of less requirement of data, no manual processing, and scalability, and can well meet the needs of many enterprises to build intelligent customer service systems.

This paper will be arranged as follows. The first section will briefly introduce the research background and our work. The second section will introduce some related work. The third section will detail the methods used to build our QA system and show the effect of our QA system. The final section is the conclusion and summary for the whole paper.

2 Related Work

Our QA system is aimed at automatically constructing the professional domain knowledge base, offering the interactive question and answer to the users. It can replace the role of the traditional product service. However, there are two main challenges for the construction of this system.

First, the construction of the knowledge base is a main difficulty. The formats of the documents, which are entered to the knowledge base are not always consistent. The question-answer pairs may contain sub-problems, tables or lists, which are not convenient to store in the base.

In order to solve these problems, people have proposed many different methods. The most traditional method is based on regular expression, such as Tang X et al (2010) [11] This method uses the regular expression to extract the string which match the question and answer templates from the web. However, this method also has some problems. The question-answer pairs matched by the regular expression are too single, and the characteristics of the web page are not fully used.

Aiming at effectively utilizing the structural features of the web page, we can also match question-answer pairs based on the characteristics of the HTML template. We can develop a set of rules that extract all the information in a web page of the same structure at a time. However, as the information on the Internet becomes more and more abundant, the information extraction based on the CSS selector has different rules for different structural pages, and it is obviously unsustainable to make rules manually. Therefore, the method of automatically extracting information based on machine learning has received more and more attention.

Generally, the machine learning based web information extraction algorithms can be categorized into three kinds: web information extraction algorithm based on heuristic and unsupervised learning algorithm, web information extraction algorithm based on classifiers and web information extraction algorithm based on web page template Chang C et al.(2006) [9]

The web information extraction algorithm based on heuristic and unsupervised learning algorithm is the easiest and best algorithm to implement. This algorithm has high versatility, which makes it effective on various web pages with different languages and structures. Some early algorithms, such as MSS algorithm Jeff P et al. (2009) [6], don't parse the web pages into a DOM tree, but a token sequence. Its goal is to find a subsequence in which the sum of the scores of the corresponding tokens reaches the maximum. And this subsequence will be considered as the body of the web page. However, this method also has its weakness, for example, using the token sequence as a unit of computing features not only destroys the structure of the web page but doesn't fully utilize the features of the web page. Therefore, the node of the DOM tree begins to be used as the basic unit of feature calculation, such as Wu G et al. (2013) [2], Sun F et al. (2011) [7]

The web information extraction algorithm based on the classifier, needs to label the text and non-text parts for a large number of web pages, and select appropriate features and classifiers for training. These two steps are vital for the efficiency of the model but requiring a lot of manual labor. Therefore, we believe that this kind of algorithm is not suitable for our system with the automated question answering function.

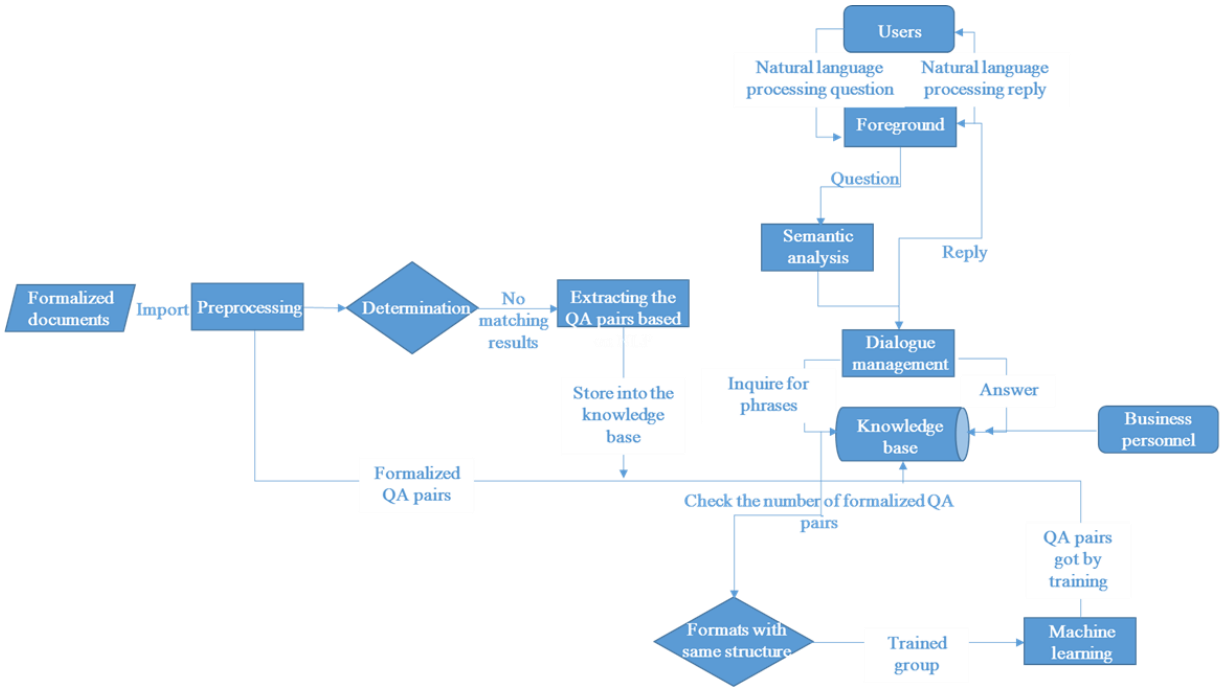
The web information extraction algorithm based on web page template has many ways to implement, such as Borut Set al. (2013) [8] introduced in his paper. By comparing multiple webs with same structures, it treats the common parts of all the webs as non-text and parts with large difference as text. This algorithm can be applied to extract the information from the multiple pages of the same web. However, it can't be used for a single web page.

The second difficulty of the construction of the system is to understand the questions raised by the users accurately and give the right answer. It is defined as question-and-match task, which is to give a question and a set of candidate answers and select the answer that best matches the semantic relationship. The complexity of Chinese grammar, the lack of standardization of users' questions, and the immature natural language processing in Chinese all pose great challenges to us.

For question-and-answer matching tasks, the early methods were mainly based on machine learning models, which required artificially constructed features. These models based on statistical machine learning usually define related features including lexical, syntactic, grammatical, and use machine learning classifiers to match question-answer pairs. The results often depend on the quality of feature extraction. The generalization ability of data in different fields is poor, and the ability to learn deep semantic information of data is lacking.

With the rise of natural language processing based on deep learning, many methods have been developed to convert text into vectors, such as Word2Vec, Glove, and so on. These methods are very convenient for calculating the similarity between texts, so they are suitable for question-and-answer matching. Wang et al. (2015) [12] proposed a joint vector method based on the LSTM model, which transform the question-answer matching problem into a classification or sorting learning problem. Feng et al.(2015) [13] proposed a method training question-answer pair based on shared-CNN. The model achieved breakthrough experimental results in the InsuranceQA (English data set). For Chinese question-and-answer matching, a method based on deep learning Rong G et al.(2017) [10] proposed three models based on deep learning, which also achieved good results in the Chinese open-domain question and answer data set NLPCC-ICCPOL 2016.

Figure I System architecture diagram



3 Detailed design of Customer Service Automatic Answering System

3.1 Question and answer pair extraction

When customers raise questions, instead of extracting directly from the document on the webs, it will be much more convenient and efficient to search database. Therefore, the construction of the database is vital for such customer service automatic answering system. This part will introduce how the system extracts the question-answer pairs from the normalized documents and store them in the database.

3.1.1 Preprocessing

Before extracting the question-answer pairs, we firstly remove the useless information from the html documents. According to Wang Zhong [1], even the html documents in a normalized format can contain lots of useless information, including the header of the corporate website, the content in the footer, hyperlinks and images. We handle all these useless information by using crawler crawling and regular expressions.

After dealing with the above process, if there is still some useless information, we will also take advantage of the method of web page text analysis to extract the body part, then further extract the question and answer pairs.

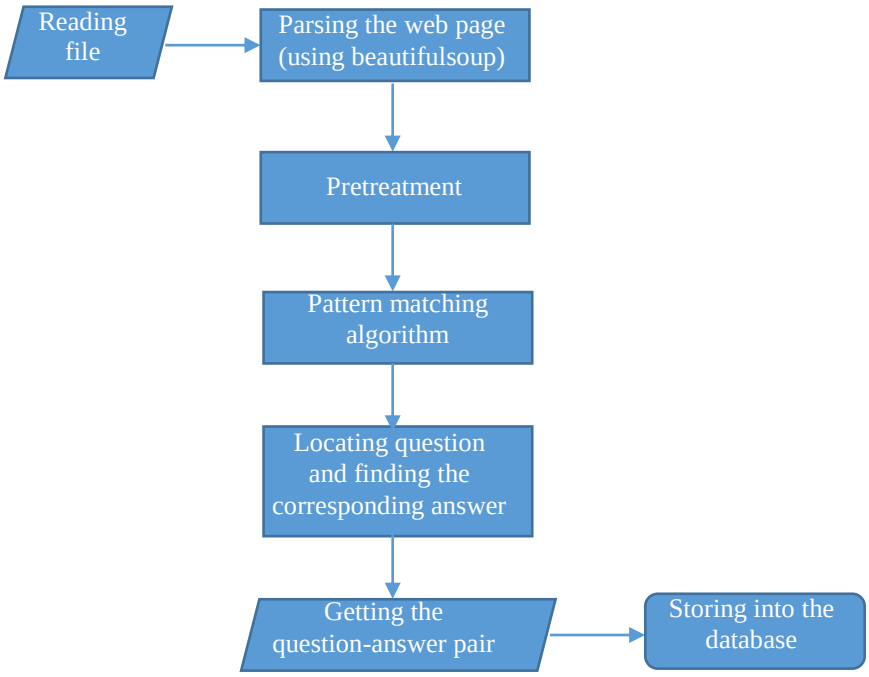
3.1.2 Standard document extraction methods

Our system uses four kinds of standard document extraction methods: Regular expression matching algorithm, Algorithm based on the document structure rules, Abstract generation based on NLP seq2seq model, and Unsupervised machine learning. This paper will introduce them one by one.

a. Regular expression matching algorithm

This algorithm is normally used for questions with many standard issues and questions which includes many sub-problems, such as the common question and answer pairs. This method is easy to apply and can achieve a good result if the given documents are of similar format.

Figure II Regular expression matching algorithm flow chart



Through this algorithm, we can get 692 question-answer pairs with high quality. This method is suitable for the questions and answers with standard formation

b. Algorithm based on the document structure rules

The template matching method shows that in the body of HTML, if there are titles such as <h1>,< h2>,< h3>,< p>, we can extract the questions and answers in a hierarchical way, matching the questions and the following answers. The advantage of this method is that it is relatively simple and can be matched to an accurate question and answer pair. However,

In order to cover more question-answer pairs comprehensively, a variety of approximate problem templates can be defined for syntactic transformation, such as changing the position of the question word, the flip structure of the sentence, etc. [2]. Although the weakness is that the question-answer pairs that conform to the rules may not be much, and further methods are needed for further extraction, for most users, the questions won't be particularly strange.

We define t as a webpage similar to the following structure.

```
Webpage example
1. <div id="main">
2.     <div id="article">
3.         <div class="columnGroup first">
4.             <h1 class="articleHeadline">
5.             BBC Episode Examines…
6.             </h1>
6.         </div>
6.         <div class="articleBody">
7.             <p itemprop="articleBody">
8.             The unusual spectacle…
9.             <a> Britain </a>
10.            </p itemprop="articleBody">
11.            Previous BBC associates…
12.        </div>
13.    </div>
14.</div>
```

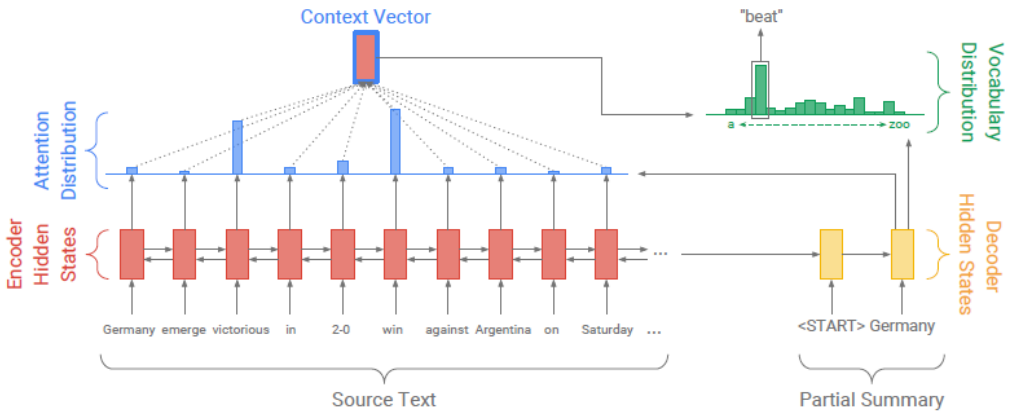
Our method is to save the title into an array, traverse the DOM tree, find all the child nodes with text, update the title array when traversing, and output the title and content when it encounters something like <p>, but only if the page is a canonical page (the title size is proportional to the title level).

```
Procedure FindQA()
Parameters: Html dom tree t, answer a (a data set), question q (a data set)，a temp title tag set P，a QA set T
Function: find questions and answers from web pages
for child in T:
if child is a tag and child has only one decendent and child's content is not empty
    if child is text node like <p> or <pre>:
        a ← a + child.string
    else:
        if mycontent != "":
            q ← P[0] + P[1] + P[2] + P[3]
            add q and a to T
            a ← Null
            P ← ∅
        if child's tag is h1:
            P[0] ← child's content
            P[1] ← Null
            P[2] ← Null
            P[3] ← Null
        elif child's tag is h2:
            P[1] ← child's content
            P[2] ← Null
            P[3] ← Null
        elif child's tag is h3:
            P[2] ← child's content
            P[3] ← Null
        elif child's tag is h4:
            P[3] ← child's content
```

c. Abstract generation based on NLP seq2seq model

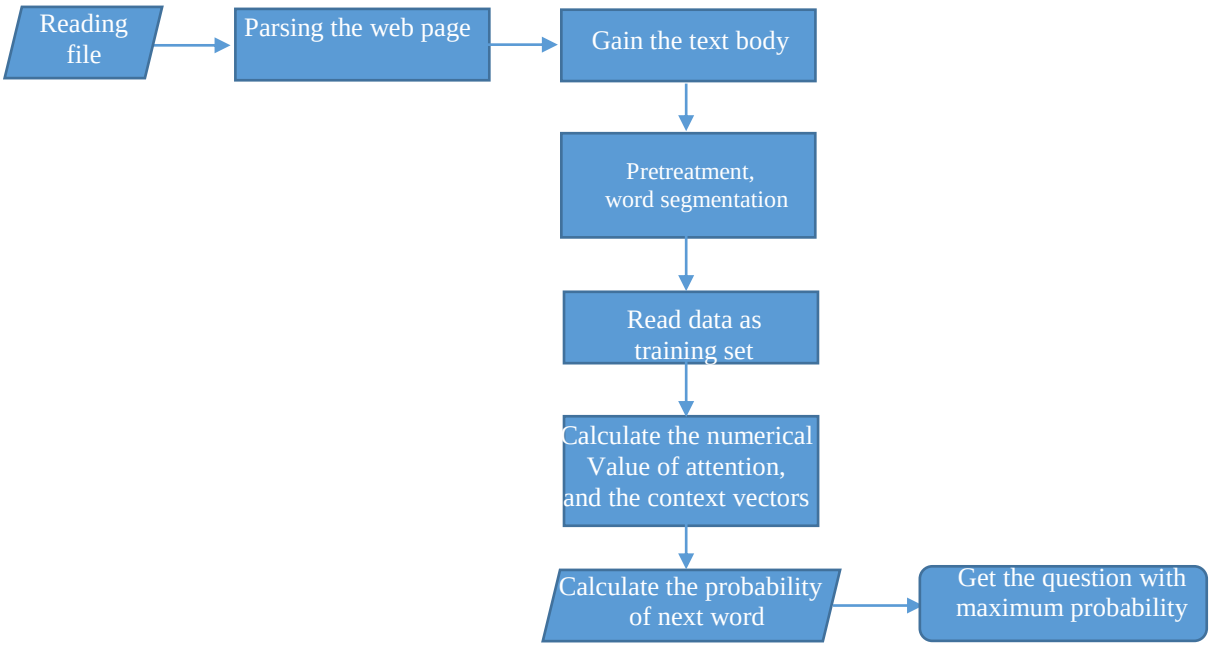
Refer to See A, et al. [3], Sutskever I, et al. [4], Vaswani A, et al. [5] , we gain our core model which is as following.

Figure III Pointer-generator model



This is an sequence-to-sequence attention model with a hybrid pointer-generator network and with coverage. First, we have some web pages summarized manually and take them as the training set. Then, we put web pages as the input and get the summary through this model. We take these summaries as questions.

Figure IV Algorithm flow chart



d. Heuristic rules and unsupervised learning-based web extraction

Web page extraction algorithm based on heuristic rules and unsupervised learning is an algorithm with best effect, simple implementation and wide application. Taking the Node of the DOM tree as a computing unit, the nodes are classified to problem nodes, answer nodes and irrelevant nodes. And the classification is shown as follows. Each computed node will be graded according to a given condition, and the higher scores it gets mean the greater possibility it is the most appropriate node.

Answer nodes

Answer nodes are classified according to their score, their evaluation standard includes:

- Standard deviation of characters in leaf nodes
- The density of plain text characters minus the hyper link characters
- Number of text characters
- The number of <p> Tags

The bigger those 4 standards tend to be, the more possible that they are the answer.

It was further found that the most useful information tends to contain mare punctuation, especially in Chinese. Most of the irrelevant information in web pages contains no punctuation or only one comma. Therefore, punctuation information is added when calculating node information. The score is calculated by the following equation:

$$score = log(var) * pureTextDensity * log(textCount - linkTextCount + 1) * log10(pCount + 2) * log(punctuation)$$

In the code, we traverse all nodes in <body> tag and find text node in the html page. We then calculate the number of characters and punctuations and add them to a HashMap for faster store and query.

Figure V Core code to obtain the answer content and calculate the score

```
CountInfo ComputeInfo(Node node):
   contInfo = new ContInfo()
   if node is Element:
       for childnode in Node.childnodes:
            childInfo = computeInfo(childNode)
            countInfo += childInfo
       contInfo.tagCount ++;
       if node.tagName =='P': contInfo.pCount ++
       else if node.tagName =='a': contInfo.linkTagCount ++
       contInfo.density = (countInfo.textCount - countInfo.linkTextCount)/(contInfo.tagCount - countInfo.linkTagCount)
       Map: node and node's countInfo
       return countInfo
   else if node is TextNode:
       countInfo.textCount = text_length
       return countInfo
   else return new CountInfo()
```

This method is very adaptive and we can extract the keywords and answers accurately from almost every given web page. It is also the most important algorithm at present.

Question nodes

With the highest scoring element as the answer part, finding the corresponding question can be divided into the following situations:

- The most likely match problem is the <h> tag which has a high approximation to the title of the whole page and is nearest to the node where the answer is located. Therefore, we traversed the title elements based on this principle.
- When there is no <h> tag. The title is the Question we need.
- Some titles may repeat. We use crumbs of the web page to identify different pages.
- If there's no title in the web page, return the first text with an appropriate length.

This method has strong adaptability, almost every given page can extract keywords and answers better.

3.2 Dialogue management

One of the most important function for the intelligent question answering system is to deal with people's dialogue. As an intelligent system, it should not only answer the questions raised accurately, but also need to be able to remember, have divergent thinking, and be user-friendly.

Our system uses both methods and also be supplemented by other new methods. We train all the questions in the knowledge base into vectors. When users raise their questions, we will firstly find the most similar question-answer pairs and return them to the users. If the knowledge base doesn't have such similar questions, we will use the keywords of the questions to search the base and return the question-answer pairs which contains the same keywords. In the meanwhile, we will save the history of the conversation. For the questions, which have been answered, we can quickly return them to the user without having to retrieve the database. We also store some common question-answer pairs for daily communication. Furthermore, considering that enterprises may add, delete and modify the demand for question-answer pairs at any time, we provided the corresponding interface.

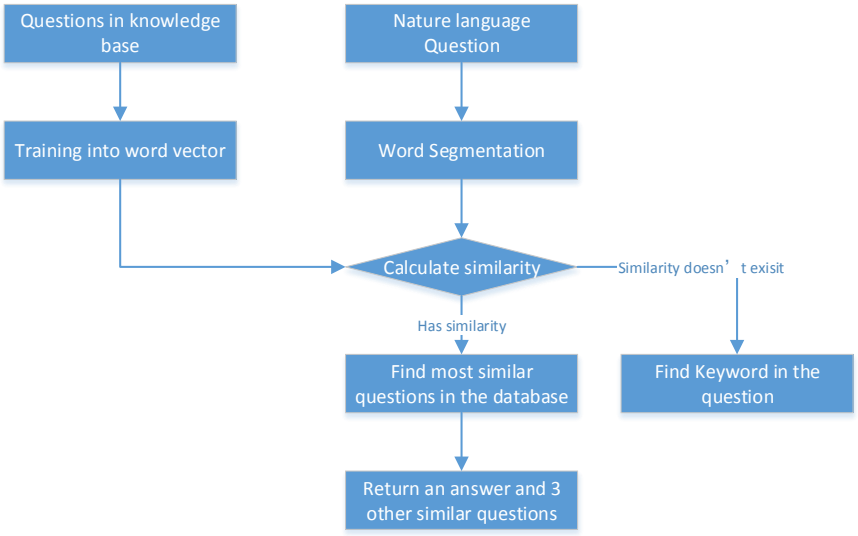
a. Search the proper answer

After extracting the question and answer pairs, we train the problem into word vectors and store it as a model. When the user asks questions to the system, it first performs word segmentation, then calculates the similarity according to the model, finding the closest four questions. Finally, it gets the responding answer from the database and return it and 3 other similar questions to the users. If the similarity doesn't work, the system will find out the key words of the questions, search them in the database and return the answer of the most similar question and 3 other similar questions.

In the similarity calculation, the n_similarity of Word2Vec is used to calculate the similarity between the natural language question and the two-word sequence of the question in the library. The sentence vector is calculated by summing the sequence vector and gaining the average value for each sentence. For the sentences in the context of this article used, this method has a good effect because there are few ambiguities and the sentences are relatively short.

If the similarity algorithm failed to find the proper answer, we split the words and remove the stop words, and rank the remaining words according to importance. The importance can be calculated by a professional vocabulary, since the web pages are of a certain field. A dictionary is easy to train or obtain. Then we search the database with the keywords and find the answer.

Figure VII Natural Language Process



b. Store the dialogue history

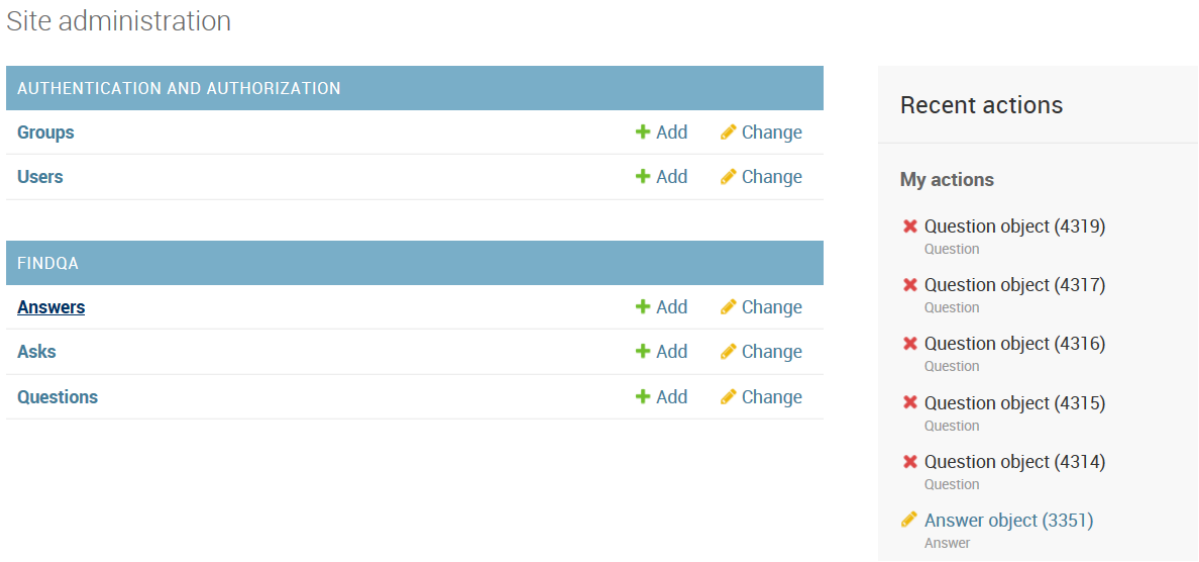
This QA system saves all the questions, which asked by users. Every time when a new logged-in user asks some questions, it firstly retrieve the questions it saved and if it finds a problem with very high similarity, there is no need to send user's problem information to the server. Instead, you only need to reply to the user "The question has been answered", but if the user ask the same question once again, the system then will send post request to the server. This can increase the intelligence of the system.

c. Add General dialogue scene

In addition to storing and accessing information about product documentation, the knowledge base also needs to store some simple conversations that conform to normal daily communication, such as "Hello", "Hello"; "What is your name?", "I am a small customer service i". When the user no longer enters a question for a long time, it should output a polite expression such as "I hope my answer is helpful to you" to enhance the user's experience. A daily conversation data set can be trained to adapt to this demand.

d. Knowledge Base Management

Because the structure of the document varies widely, the method of extracting the question and answer pair proposed in this paper can't guarantee the 100% correct rate. Therefore, the database provides an interface, you can view the question and answer pairs, manually add question-answer pairs and similar questions, delete the unqualified question-answer pairs. In addition, when entering into the knowledge base, it is guaranteed that there can be no complete repetitive problems, so as to avoid the ambiguity of multiple answers to one question.



4 Conclusion

This paper proposes a customer service automatic answering system which can better meet people's needs. Compare with the existing DBQA systems, we sum up 3 ways to extract QA pairs from given documents. The method based on Heuristic unsupervised machine learning achieve a good result among these 3 methods. Of 3501 documents, this method filtered invalid documents and get 3265 QA pairs with high accuracy. Our code and results are on https://coding.net/u/Pacsiy/p/QAServer/git.

In general, this system overcomes two difficulties, building a professional knowledge base and having a clear understanding of users' questions. Taking advantage of unsupervised learning algorithm, we extract QA pairs from online documents and store them in the professional knowledge base. Based on natural language processing, this system can have an accurate understanding of the questions, then return the corresponding answers from the knowledge base. This system also improves the efficiency and accuracy through storing the dialogue history and adding the edifying function. We sincerely believe that our system will be in use in the near future.

References

[1] Wang Zhong. XHTML - An Extensible Hypertext Markup Language [J]. Computer Science, 2000 (10): 16-18.) standard

[2] Wu G, Li L, Hu X, et al. Web News Extraction via Path Ratios[J]. 22nd International Conference on Information and Knowledge Management, 2013:2059-2068.

[3] See A, Liu P J, Manning C D. Get to the Point: Summarization with Pointer-Generator Networks[J]. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017:1073-1083.

[4] Sutskever I, Vinyals O, Le Q V. Sequence to Sequence Learning with Neural Networks[J]. Advances in Neural Information Processing Systems on Neural Information Processing Systems, 2014, 4:3104-3112.

[5] Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need[J]. Advances in Neural Information Processing Systems on Neural Information Processing Systems, 2017:6000-6010.
[6] Jeff P, Dan R. Extracting Article Text from the Web with Maximum Subsequence Segmentation[J]. Proceedings of the 18th International Conference on World Wide Web, 2009:971-980.
[7] Sun F, Song D, Liao L, et al. Dom based content extraction via text density [J]. Proceeding of the 34th International Conference on Research and Development in Information Retrieval, 2011:245-254.
[8] Borut S, Miha G. URL Tree: Efficient Unsupervised Content Extraction from Streams of Web Documents[J]. 22nd International Conference on Information and Knowledge Management, 2013:2267-2272.
[9] Chang C, Kayed M, Girgis G. M, Shaalan F. K. A Survey of Web Information Extraction Systems[J]. IEEE Transactions on Knowledge and Data Engineering, 2006 (18): 1411-1428.
[10] Rong G, Huang Z. Question answer matching method based on deep learning[J]. Journal of Computer Applications, 2017(37):2861-2865.
[11] Tang X, Zeng Q, Cui T, Wu Z. Regular Expression-based Reference Metadata Extraction from the Web[J]. IEEE 2nd Symposium on Web Society, 2010.
[12] WANG D，NYBERG E. A long short-term memory model for answer sentence selection in question answering[C]. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015:707—712.
[13] FENG M，XIANG B，GLASS M R，et a1. Applying deep learning to answer selection: a study and an open task[C]. Proceedings of the 2015 IEEE Workshop on Automatic Speech Recognition and Understanding, 2015:813—820.