# User Input Classification for Chinese Question Answering System

Yongshuai Hou(✉), Xiaolong Wang, Qingcai Chen, Man Li, and Cong Tan

Key Laboratory of Network-Oriented Intelligent Computation,
Harbin Institute of Technology, Shenzhen Graduate School, Shenzhen 518055, China
houyongshuai@hitsz.edu.cn, wangxl@insun.hit.edu.cn,
qingcai.chen@gmail.com

**Abstract.** Restricted-domain question answering system gives high quality answer to questions within the domain, but gives no response or wrong answer for out of the domain questions. For normal users, the boundary of in-domain and out-domain is unclear. Most users often send out-domain inputs to the restricted-domain question answering system. In such cases, both no answer and wrong answer from the system will yield bad user experience. In this paper, an approach is proposed to solve the bad system response issue of the restricted-domain question answering system. Firstly, it uses a binary classifier to recognize in-domain user inputs and uses the restricted-domain question answering system to proved correct answer. Secondly, an user input taxonomy for out-domain user input is designed, and a classifier is trained to classify the out-domain user input based on the taxonomy. Finally, different response strategies are designed to response to different classes of out-domain user inputs. Experiments and actual application on a restricted-domain question answering system shows that the proposed approach is effective to improve user experience.

**Keywords:** User input classification · Restricted-domain · Question classification · Question answering system

## 1 Introduction

In recent years, researches on question answering (QA) systems progresses rapidly. Most QA systems focus on how to answer questions in a restricted domain or in a restricted sentence format. Those QA systems are called restricted-domain QA systems. This type of QA system answers user's questions with high accuracy in the system's restricted domains [1], [2]. Restricted-domain QA systems make information search easier for users. However, the system needs user's input questions which follow the system domain restriction and the specified sentence format. Users should know both the domain and the input format of the QA system. If user's questions are beyond the system limitation, the system returns no answer or wrong answer. For example, the system returns "Sorry, I don't understand your question. Please try another." if the user's question is beyond the system limitation.

However in the real world, most people are not professional users for the restricted-domain QA system. They even do not know the domain and the limitation of the QA


X. Wang et al. (Eds.): ICMLC 2014, CCIS 481, pp. 52–59, 2014.
DOI: 10.1007/978-3-662-45652-1_6

system when entering a query. Those users input questions following their daily expression habits and from all domains. Some users even use the QA system as a dialogue system. In these cases, the system cannot answer user's un-restricted inputs and can only respond with no answer or a predefined answer like "Sorry, I don't understand your question. Please try another". In some cases, the system will return wrong answers for out-domain user inputs which is even worse. Those cases reduce the accuracy of answers and give users bad user experience. So, it is necessary to distinguish users' inputs which meet the limitation of the restricted-domain QA system or not. Furthermore, designing diversify responses for those user inputs that do not meet the system limitation to improve user experience is also necessary.

There is a large amount of researches on question classification for QA systems. Widely used methods for question classification include rule-based methods, Naïve Bayes [3], Support Vector Machine (SVM) [4], [5] and KNN [6]. Usually, a question classification taxonomy is designed based on the question content, or a special question classification problem (like the UIUC taxonomy. It cannot be used in other question classification problems [7]. Fan Bu and Xingwei Zhu design a taxonomy suitable to general QA systems, and plan to use it to answer unrestricted-domain question by sending the question to different restricted-domain QA systems based on the question class [8]. Most researches on question classification focus on classifying questions by the domain, question format. Only very few of them focus on user inputs which are partly belonging to questions of the restricted-domain QA system.

In this paper, we propose a user input classification approach for restricted-domain QA systems to distinguish classes of users' inputs automatically. Firstly, a binary classifier is trained to judge whether the user input is a question in the required domain. If the question is in the domain, it is sent to the restricted-domain QA system. Secondly, we design a user input taxonomy including 14 classes for out of the system domain questions, and train a multi-class classifier to classify those user inputs. For the user input following the system restriction, we search answer from the restricted-domain QA system. For the user input which is out of the system domain, we design different response strategies for different user input classes. The processes for answering user inputs in the restricted-domain QA system with our method is shown in Figure 1.

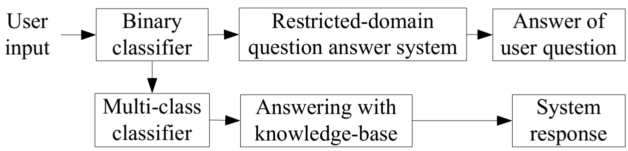


**Fig. 1.** Answering processes for user inputs

We define the *user input* as all cases of user inputs to the QA system, such as all questions, dialogue sentences, keywords for information search. The *in-domain user input* is defined as the user questions which meet the input limitation of the restricted-domain QA system. All other user inputs that do not follow the limitation of the QA system are defined as the *out-domain user input*.

## 2 User Input Taxonomy

Because restricted-domain QA systems can only give high quality answers to questions in their restricted domains, so it is necessary to distinguish the questions which a system can answer. Most users do not usually use the QA system. They are not familiar with the usage rules and domain restriction for the system. So those users cannot ensure their inputs are following the system restriction. Even more, some users think the QA system knows everything and input queries from all domains, or use the QA system as a dialogue system and send dialogue sentences to the system. Most QA systems cannot answer those out-domain user inputs. There are two ways in those systems to response to the out-domain user inputs: the first one was response nothing or response answers like "there is no answer, please try other questions". This response way makes a bad user experience for the system; the second one is finding an answer which is closest in meaning to the user input. This way cannot get right answers in most case, and reduced answer accuracy of the system.

To improve the user experience and answer accuracy for restricted-domain QA system, we designed an answer strategy to response those out-domain user inputs. We firstly classified those out-domain user inputs, and designed different responses basing the user input classes. We designed a user input taxonomy for classifying user input for restricted-domain QA system, and the taxonomy is shown in figure 2.

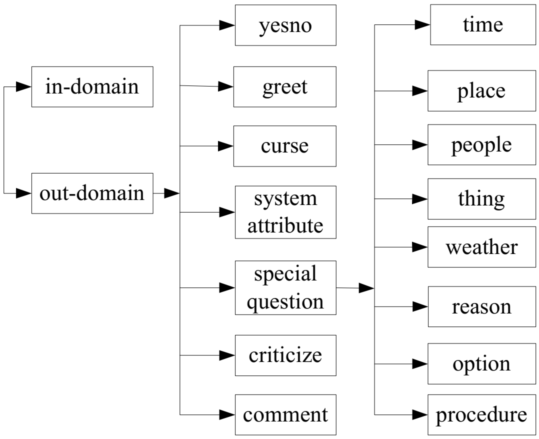


**Fig. 2.** User input taxonomy for restricted-domain QA system

## 3 User Input Classification

To answer user input separately basing on user input for restricted-domain QA system, we proposed an approach to classify and response user input for the restricted-domain QA system. The process for user input classification is shown in figure 1. First, we trained a binary classifier to recognize *in-domain* class question from user input and filter the *out-domain* class user input for the restricted-domain QA system. Second, we trained a classifier to classify the *out-domain* user input into 14 classes following the user input taxonomy we designed. Finally, we designed different response for different class user input to ensure the restricted-domain QA system can send appropriate system response for most of user inputs, including *in-domain* and *out-domain*.

a) User input domain classification

A binary classier was trained here to recognize user input domain for restricted-domain QA system. The features for the *in-domain* class user input to the restricted-domain QA system are clear and effective, and they are much suitable for the classifier based on rules. Here we used rule-based classifier to recognize the *in-domain* user input.

The restricted-domain QA system we used here is a financial-orient QA system (BIT Chinese Financial QA System)[1] we developed. So features we extracted for the binary classifier are financial-orient.

The features list we used to classify the domain of user input is shown as following:

(1) Financial code: The code here includes stock code, fund code, bond code and so on. Regular expressions were used to match whether the user input contain financial code;
(2) Financial named entity: Constructing a financial named entity dictionary includes name of stock, fund, bond, name of financial organize and so on, and judging whether the user input contain financial named entity in the dictionary;
(3) Financial term: Constructing a financial term dictionary for those terms that are most used in financial domain, such as "interest on deposit", "loan interest", "exchange rate" and so on, and judging whether the user input contain financial term in the dictionary;
(4) Special sentence structure: Using regular expressions to match whether user input is the special expression or sentence structure for financial domain;
(5) Combination of the 4 features above.

b) Out-domain user input classification

The *out-domain* user input classification classified user input to 14 classes based on the taxonomy we designed. And machine learning methods were used on the classification. The features were selected based on Chinese short text classification methods [9] and the special characters for the QA system user input. Features list is shown as following:

(1) Word feature: uni-gram, bi-gram, tri-gram of the user input;
(2) Special word: dirty words, positive evaluation words, negative evaluation words, criticism words and so on;
(3) Sentence templates and sentence combination patterns: Collecting and summarizing special sentence templates and sentence combination patterns from user input;
(4) *Wh* word feature: *Wh* word, word before *Wh* word and word after *Wh* word, front word and tail word of user input;
(5) Part of speech (POS): POS of word before and after *Wh* word, number of Verb, Noun and Adj. word in user input;
(6) Semantic feature: semantic of word before and after *Wh* word;
(7) Pronouns: whether there is personal pronouns in user input;
(8) Syntactic structure: whether the syntactic structure of user input is complete;
(9) Tail word: whether the tail word of user input is function word;

[1] bit.haitianyuan.com

# 4 Dataset and Experiment

a) Dataset

Most public dataset for QA system are just useful for question classification, and the classification taxonomy is aim on the question domains or syntactical structure. But there is few public user log for QA system as we know. And at same time, public dataset about classifying user input of QA system focus on the in-domain and out-domain case is also lacking.

In our experiments, we collected user logs from the financial-orient QA system (BIT Chinese Financial QA System) we developed. In this data set, financial-orient questions were used as *in-domain* questions and those didn't belong to financial domain input in the user log were used as *out-domain* user input. After removing repeated and invalid user inputs, we got 2196 different user input from the log of BIT QA system. Then we asked 3 persons to manually annotate classes of the user input and extracted the user input that at least 2 annotated results are same. Finally, we got 1669 manually annotated user input for experiment on classifying *in-domain* and *out-domain* user input, including 838 financial questions and 831 *out-domain* user inputs.

To train classifier for classifying the *out-domain* user inputs into 14 classes, using 831 user inputs as dataset is too small. So we extended the dataset scale by including the user inputs in AIML daily dialogue knowledge-base. After annotating and extracting, we got total 3024 different *out-domain* user inputs for classifier training. The detail user input distribution on the 14 classes is shown in table 1.

**Table 1.** Out-domain user input class distribution

| Classes of out-domain user input | Number of user input |
|---|---|
| Yesno | 303 |
| Greet | 215 |
| Curse | 218 |
| System attribute | 329 |
| Criticize | 181 |
| Comment | 149 |
| Time | 214 |
| Place | 157 |
| People | 188 |
| Thing | 152 |
| Weather | 204 |
| Reason | 285 |
| Option | 191 |
| Procedure | 238 |
| Total | 3024 |

b) Experiment

The user input classification performance is measured by precision (*P*), recall (*R*), F1 score (*F1*), calculate method are shown as formula (1), formula (2) and formula (3).

$$P = \frac{\sum_i \#correctly\_classified\_as\_c_i}{\sum_i \#classified\_as\_c} \tag{1}$$

$$R = \frac{\sum_i \#correctly\_classified\_c_i}{\sum_i \#labeled\_as\_c_i} \tag{2}$$

$$F1 = \frac{2PR}{P+R} \tag{3}$$

There are rules to follow while classify financial-orient question. So we used rules based method to training the binary classifier. And the user input domain classification results with the binary classifier is shown in table 2.

**Table 2.** Experimental results of user input domain classification

| | In-domain | Out-domain | Average |
|---|---|---|---|
| P | 98.74% | 94.52% | 96.52% |
| R | 94.22% | 98.81% | 96.52% |
| F1 | 96.43% | 96.62% | 96.52% |

The classification precision for *in-domain* class user input is up to 98.74%, and the F1 score is 96.43%. The result shows that our binary classification method can ensure most user inputs send to the financial-orient QA system are financial questions. This is the basement for the financial-orient QA system to return right answers and appropriate system response. If there was no classifier to filter the out-domain user input for financial-orient QA system, the answer precision would decline and the user experience would become poor.

The binary classifying for user input can improve the answer precision. But to improve the system user experience, it must strengthen the response capacity to the *out-domain* class user input for the financial-orient QA system. We designed different response strategy to different user input class for the QA system.

With respect to the machine learning model, Naïve Bayes [3], Support Vector Machine (SVM) [4], [5] and KNN [6] are used in this paper. We compared the performance of the three machine learning methods on *out-domain* class user input classification and choose the best classifier.

The three machine learning methods we used in this paper are from the toolkit RainBow[2], and all the parameters were set as default.

We used the 3024 annotated *out-domain* user inputs as experiment dataset. The dataset was divided into 3 parts, 2 parts were used as training data and 1 part was used as test data. We run the experiment with 3 folds cross validation. And the experiment results are shown as in table 3 and figure 3.

[2] http://www.cs.cmu.edu/~mccallum/bow/rainbow/

**Table 3.** Experimental results of out-domainn user input classification with different algorithms

| | Naïve Bayes | SVM | KNN |
|---|---|---|---|
| P | 68.84% | 65.47% | 40.33% |
| R | 63.50% | 60.73% | 46.76% |
| F1 | 61.99% | 59.85% | 43.27% |

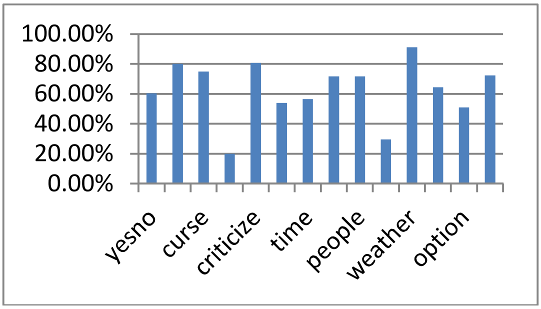


**Fig. 3.** F1 score of out-domain user input classification with Naïve Bayes

The results in table 3 show that the result of Naïve Bayes method is better than the other two machine learning methods. Basing on the experiment results we choose Naïve Bayes classifier to classify the *out-domain* user input for the financial-orient QA system. Figure 3 is the detail F1 score of each class in Naïve Bayes method results. Figure 3 shows that the F1 score of class *Greet*, *Curse*, *Criticize*, *Place*, *People*, *Weather* and *Procedure* are above 70%, and F1 score of other classes are low. Classification result of half number of classes is not satisfied to real requirement.

The accuracy of the whole classification results is low. The main reason is that the training data is insufficient. There are 14 classes but only 3024 instances to use, only 216 user inputs for each class in average. The training data is poor for each class. Another reason is that the feature for some classes is not discriminating to classification. For example, all the user inputs about system information are belong to class *System attribution*, but those user inputs has same features with other classes user input such as *Person* class. So some classes in the taxonomy should be adjusted.

In the practical application on the financial-orient QA system, to ensure the system give the right response, we designed some classification rules for some *out-domain* user input classes to avoid the influence of the low accuracy of classifier. With the user input, we first used the rules to classify some *out-domain* user inputs, then used the *out-domain* user input classifier to classify the user input that rules cannot cover. The real system we developed was shown in web http://bit.haitianyuan.com.

## 5 Conclusions

In this paper, we study the problem of how to classify the *in-domain* and *out-domain* user inputs and how to response the *out-domain* user input for the restricted-domain QA system. Firstly, we use a binary classifier to recognize *in-domain* user inputs for the restricted-domain QA system, and send *in-domain* user inputs to the system to ensure that the system provide accurate answers. Then, we design a 14 classes taxonomy to classify *out-domain* user inputs and different response strategies for different user input classes. This step ensures that the restricted-domain QA system yields appropriate system responses to most of *out-domain* user inputs. Experimental results demonstrate that the classification of user input domain is effective, but more rules need to be added to the classification of the 14 classes *out-domain* user inputs before application in real systems to ensure the accuracy of the classification and accurate responses.

In our future work, we will collect and annotate more user input data for the classifier training on *out-domain* user inputs. We also plan to improve the 14 classes taxonomy by analyzing more user input data to make the class boundaries in the taxonomy more clear.

**Acknowledgements.** This paper is supported in part by grants: NSFCs (National Natural Science Foundation of China) (61173075 and 61272383), Scientific and Technological Research and Development Fund of Shenzhen (JC201005260175A), Strategic Emerging Industry Development Special Funds of Shenzhen (ZDSY20120613125401420 and JCYJ20120613151940045) and Key Basic Research Foundation of Shenzhen (JC201005260118A).

## References

1. Hu, H., Ren, F., Kuroiwa, S., Zhang, S.: A question answering system on special domain and the implementation of speech interface. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 458–469. Springer, Heidelberg (2006)
2. Mollá, D., Vicedo, J.L.: Question answering in restricted domains: An overview. Computational Linguistics **33**(1), 41–61 (2007)
3. Lu, S., Chiang, D., Keh, H., et al.: Chinese text classification by the Naïve Bayes Classifier and the associative classifier with multiple confidence threshold values. Knowledge-based Systems **23**, 598–604 (2010)
4. Zhang, D., Lee, W.S.: Question classification using support vector machines. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, Canada, July 28-August 1, pp. 26–32 (2003)
5. Pilászy, I.: Text categorization and support vector machine. In: The Proceedings of the 6th International Symposium of Hungarian Researchers on Computational Intelligence (2005)
6. He, X., Zhu, C., Zhao, T.: Research on short text classification for web forum. In: 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Shanghai, China, July 26-28, pp. 1052–1056 (2011)
7. Li, X., Roth, D.: Learning question classifiers: the role of semantic information. Natural Language Engineering **12**(3), 229–249 (2006)
8. Bu, F., Zhu, X., Hao, Y., et al.: Function-based question classification for general QA. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, Massachusetts, USA, October 09-11, pp. 1119–1128 (2010)
9. Fan, X., Hu, H.: A new model for chinese short-text classification considering feature extension. In: 2010 International Conference on Artificial Intelligence and Computational Intelligence (AICI), Sanya, China, October 23-24, pp. 7–11 (2010)