An Effective Feature-Weighting Model for Question Classification

Peng Huang, JiaJun Bu*, Chun Chen, Guang Qiu
College of Computer Science, Zhejiang University, Huangzhou, China
{huangp, bjj, chenc, qiuguang}@zju.edu.cn

Abstract

Question classification is one of the most important subtasks in Question Answering systems. Now question taxonomy is getting larger and more fine-grained for better answer generation. Many approaches to question classification have been proposed and achieve reasonable results. However, all previous approaches use certain learning algorithm to learn a classifier from binary feature vectors, extracted from small size of labeled examples. In this paper we propose a feature-weighting model which assigns different weights to features instead of simple binary values. The main characteristic of this model is assigning more reasonable weight to features: these weights can be used to differentiate features each other according to their contribution to question classification. Furthermore, features are weighted depending on not only small labeled question collection but also large unlabeled question collection. Experimental results show that with this new feature-weighting model the SVM-based classifier outperforms the one without it to some extent.

1. Instructions

Question classification is the process by which a system can analyzes a question and labels the question based on its expected answer type. For example, the question "Who is the first one sent to the moon?" expects a person's name as an answer. The task of question classification is, given a finite set of possible question types, to learn a mapping from questions to one (or more) question type(s).

Question classification (QC) systems are primarily used as components of question answering (QA) systems. Since the start of the Question Answering Track at TREC-8 in 1999, Open-domain Question Answering (QA) systems have become an important direction in natural language processing domain. Different from traditional search engine, which returns a list of documents in response to a user query, question answering systems are designed to distill exact and concise answers to questions posed in natural language from a collection of documents, where an answer is generally a short fragment of text drawn from the corpus. Although there are many kinds of QA systems, most of them take a divide-and-conquer strategy and follow a general framework [16]. Given a question, most systems first analyze the question and use a QC system to determine the most likely expected answer type. Secondly, some form of document or passage-level retrieval is done to find top N paragraphs or documents from the corpus that match the result of question type. Finally, extraction of answer candidates is done based on second stage and then select most plausible answer from answer candidates. The most important subtask of above may be to determine the 'type' of the sought-after answer. In practice there are often several passages that contain an answer, so the failure in some passage(s) can be made up by success in others containing the correct answers. However, when the question is analyzed incorrectly, overall failure is much more likely. Without a question type, that is, the result of question classification, it would be much more difficult or even nearly infeasible to select correct answers from the possible answer candidates. The results in [9] show that about 36.4% of errors in QA systems are generated directly from QC module. Consequently, QC is proposed separately in many QA researches in recent years.

In recent years many approaches to question classification have been proposed, ranging from rule-based to machine learning. These approaches all have obtained pretty good results in their experiments respectively. However, most of them are trained with binary features obtained from labeled question collection through natural language processing tools. Intuitively, it is more reasonable to use different weights representing feature vector in learning algorithm. Moreover, the number of labeled question collection is usually so small that the features extracted from training examples have a low coverage over total feature space. In this paper we develop a question classifier based on a novel feature-weighting model, which weights each feature according to its contribution to question classification, to al-

*Corresponding Author

0-7695-3072-9/07 $25.00 © 2007 IEEE
DOI 10.1109/CIS.2007.12

IEEE computer society

leviate above problems to some extent. In general the number of labeled questions is limited, so features are evaluated on not only a small labeled question collection, but also a large amount of unlabeled questions.

The remainder of this paper is organized as follows. The next section briefly reviews previous work about question classification. In section 3 we first make a brief introduction to SVM leaning algorithm, and then describe the details of our feature-weighting model, together with some introduction to entropy theory and clustering techniques employed in the feature-weighting model. Section 4 shows the experimental results and some discussions. The last section concludes our paper and gives some ideas for future work.

2. Related Work

This section reviews some existing approaches to question classification. They are roughly divided into two groups: one is based on hand-crafted rules and another one is based on machine learning. For example, the system "SAIQA" [12] and Pasca et al. [10] used hand-crafted rules for question classification. Although pretty good results have been obtained in their experiments respectively, both of them suffer two common drawbacks in rule-based systems. It is tedious to create rules manually; these rules are usually domain-specific, and generated poor results when applied to another domain.

To overcome these pitfalls of rule-based systems, machine learning technology is researched. Li and Roth [7] used SNoW learning algorithm with complicated features, such as POS, name entities, and semantic related word etc., for question classification. Zhang and Lee [17] applied SVMs, with syntactic structure features, to question classification. Chenung et al. [3] build a question classifier based on decision tree, with POS, keyword, noun phrase and head phrase features. The common of the above approaches is that they all focus on learning with more features extracted via deeper natural language processing tools without more consideration of the above problems. Considering that some natural language processing tools that may be not yet well developed, Solorio et al. [13] propose an approach to question classification with only surface text and simple retrieval results from Google search engine. Recently, David et al. [4] proposed an automatic feature extraction approach to question classification, which uses only statistical information from unlabeled corpus to extract features, without the help of natural language processing techniques.

3. Question Classification

3.1. Learner

Support Vector Machine (SVM) [14] is the core of much modern machine learning. Numerous theoretical and technical researches have showed its power in machine learning due to three novel ideas inside: maximizing minimum margin between classes, kernel function and support vector. Classifiers with large margin are known to have good generalization properties; when input feature space is not linear separable SVM can map, by using certain kernel function, the original input space to a high-dimensional feature space where the optimal separable hyperplane can be easily calculated; and the number of support vectors is usually in small proportion to total instances that make learning from huge instances possible. Recently, Zhang and Lee [17] applied some learning algorithms, including *k*-Nearest Neighbor (*k*NN) [1], Naïve Bayes [8], Decision tree [11], Sparse Network of Winnow, and SVM [14, 15], to question classification with surface text feature only, and results showed that SVM outperformed others. According to these results, we decide to use SVM as the learning algorithm in our question classifier.

3.2. Entropy in information theory

In information theory the concept of entropy is used as a measure of the uncertainty of a random variable. Let X be a discrete random variable with respect to alphabet A and $p(x) = Pr(X = x), x \in A$ be the probability function, then the entropy $H(X)$ of the discrete random variable X is defined as:

$$H(x) = -\sum_{x \in A} p(x) \log p(x) \qquad (1)$$

The larger the entropy $H(X)$ is, the more uncertain the random variable X is. In information retrieval many methods have been applied to evaluate term's relevance to documents, among which entropy-weighting, based on information theoretic ideas, is proved the most effective and sophisticated [5]. Let f_{it} be the frequency of word i in document t, n_i the total number of occurrences of word i in document collection, N the number of total documents in the collection, then the confusion (or entropy) of word i can be measured as follows:

$$H(i) = \sum_{t=1}^{N} \left[\frac{f_{it}}{n_i} \cdot \log \left(\frac{n_i}{f_{it}} \right) \right] \qquad (2)$$

The larger the confusion of a word is, the less important it is. The confusion achieves maximum value $\log(N)$ if the word is evenly distributed over all documents, and minimum value 0 if the word occurs in only one document.

3.3. Feature weighting model

The results in [17] showed that in the task of question classification approaches using larger *n*-gram features has little improvement compared with approaches using unigram features. Inversely, feature space has exponential increase. In addition, surface text feature can be obtained easily without other auxiliary tools, such as pos tagger, syntactic parser etc. Thus only unigram (i.e. word) feature is used in our feature model. In most previous approaches, a question is represented as a binary feature vector, that is to say, each element of feature vector is weighted as either 1 or 0, indicating the occurrence of word or otherwise. Intuitively, it is more reasonable if we substitute some specific value for 1 with respect to relevant word to quantify the feature's contribution to question classification. Inspired by the idea of entropy in information retrieval, in the proposed feature model we use the similar idea to evaluate feature importance. To calculate the entropy of a word, certain preprocessing is needed. Let C be the set of question types. Without loss of generality, it is denoted by $C = \{1, \ldots, N\}$. The total process of preprocessing is showed in figure 1. In figure 1 C_i is a set of words extracted from questions of type i, that is to say, C_i represents a word collection similar to documents. From the viewpoint of representation, each C_i is the same as a document because both of which are just a collection of words. Therefore we can also use the idea of entropy to evaluate word's importance. Let a_i be the weight of word i, f_{it} be the frequency of word i in C_t, n_i be the total number of occurrences of word i in all questions, then a_i is defined as:

$$a_i = \left(1 + \frac{1}{\log(N)} \sum_{t=1}^{N} \left[\frac{f_{it}}{n_i} \log\left(\frac{f_{it}}{n_i}\right)\right]\right) \tag{3}$$

Note that the most right $\log$ function is different from the one in formula 2, that is because the weight of word i is opposite to its entropy: the larger the entropy of word i is, the less important to question classification it is. In other words, the smaller weight is associated with word i. Consequently, a_i get the maximum value of 1 if word i occurs in only one set of question type, and the minimum value of 0 if the word is evenly distributed over all sets. Note that if a word occurs in only one set, for other sets f_{ik} is 0. We use the convention that $0 \log 0 = 0$, which is easily justified since $xlogx \rightarrow 0$ as $x \rightarrow 0$.

Now the weight a_i, calculated from labeled question collection, can be used to evaluate the importance of word i to question discrimination. Usually the size of labeled question collection is small, so the calculated weight in terms of formula 3 can not reveal the real certain word's contribution to question classification. In this work unlabeled questions are used to alleviate this problem. Unlike labeled questions, unlabeled questions can be obtained easily from Internet.

```
Initialization:
  $C_t = \varnothing,\ t = 1,...,N$
for each question $i$ of type $j$
  for each word $k$ in question $i$
    $C_j = C_j \cup \{k\}$
  end for
end for
```

Figure 1. The pseudo code of question preprocessing

We extract question from web documents according to two heuristic rules which make a promise that extracted questions are 'real' questions and well-formed. The following is the details of the rules:

- Sentence must contain certain interrogative, such as 'where', 'how' etc.
- Sentence must end with question mark '?'

Among the above two rules, the former is clear, while the latter is used to exclude sentences that contain interrogative but are not 'real' interrogatory. For example, a sentence, "I am sure this is just the place where he lived in his childhood", is in fact not a question, although it contains interrogative 'where'.

After finishing the construction of unlabeled question collection, the idea of entropy described previously, however, can not be directly applied to unlabeled questions because these questions have not yet been classified. Thus clustering technique can be used naturally. In machine learning, clustering analysis often refers to the process of grouping a set of unlabeled physical or abstract objects into classes of similar objects, in contrast to supervised learning. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. A cluster of questions extracted from web pages can be treated collectively as one group in question classification. In this work a variation of *K*-means algorithm [6], which use a statistical test to automatically disclose the appropriate number *k* in *K*-means, is used for question clustering. Consequently, unlabeled question collection is divided into a certain number of groups. Questions in one cluster are 'close' to each other and are different from questions in other clusters. So questions in a same cluster can roughly be thought of having a same type, except no specific class labels. Without loss of generality, let M be the final number of generated clusters, we can easily calculate the weight of word i, denoted by b_i, in terms of formula 3 by substituting M for N. Finally, we calculate

Table 1. The taxonomy of question types

Coarse	Fine-grained
ABBR	abbreviation expansion
DESC	definition description manner reason
ENTY	animal body color creation currency disease/medical event food instrument language letter other plant product religion sport substance symbol technique term vehicle word
HUM	description group individual title
LOC	city country mountain other state
NUM	code count date distance money order other percent period speed temperature size weight

the final weight of word i as combination of a_i and b_i as follows:

$$b_i = \left(1 + \frac{1}{\log(M)} \sum_{t=1}^{k} \left[\frac{f_{it}}{n_t} \log\left(\frac{f_{it}}{n_t}\right) \right] \right)$$

$$w_i = \frac{n_{i1}}{n_{i1} + n_{i2}} a_i + \frac{n_{i2}}{n_{i1} + n_{i2}} b_i \qquad (4)$$

where n_{i1} is the total number of occurrences of word i in labeled question collection, n_{i2} is the total number of occurrences of word i in unlabeled question collection.

4. Experimental Study

4.1. Data

Feature weight is calculated according to two question collections: labeled and unlabeled. The former comes from UIUC [7] consisting of a training dataset of size 5500 and a test dataset of size 500. These data have been labeled manually by the providers, according to the question taxonomy described in [7], including 6 coarse and 50 fine-grained question types respectively as shown in table 1.

In addition, we download one million pages from Internet and save them locally as normal documents. Then we extract questions according to rules in subsection 3.3 from documents. Finally we get an unlabeled question collection of size 25,000.

4.2. Experiment Results and Analysis

We used the public available implementation of SVM provided by Chang [2], which use a SMO-type decomposition method for working set selection to train the support vector. The kernel function used for mapping the input space was default, i.e. RBF kernel function. In the experiments we used 10-fold cross validation.

We used prediction accuracies on labeled test data to evaluate the effectiveness of our feature-weighting model.

Table 2. Question classification accuracies for coarse and fine-grained categories.

Question Type	M_1	M_2	M_3
coarse-grained	87%	86.8%	87.2%
fine-grained	79.6%	81.8%	85.2%

To compare results we constructed a baseline experiment which adopted a widely-used binary-weighting model (i.e. features are weighted either 0 or 1). In our experiments there were total three kinds of feature model were evaluated. The first was the *binary*-weighting model; the second was the a_i-weighting model which uses formula 3 to weight features; the last was the proposed w_i-weighting model which weight features according to formula 4. We carried out the experiments for both of the coarse-grained and fine-grained question categories and all results were listed in table 2, where M_1, M_2 and M_3 indicated the *binary*-weighting, a_i-weighting and w_i-weighting model respectively.

From the results we can see that our new feature-weighting model M_3, comparing to M_1, improved the accuracies of classification for fine-grained question categories. However, in the case that M_3 was applied to coarse-grained question categories the improvement is trivial, and the reason may be the number of coarse-grained categories is small and not fit for the real distribution of unlabeled question. Similarly, for coarse-grained question categories, results obtained in M_1 and M_2 are very close. We think the main reason, that M_1 and M_2 obtained the similar results, is that both the number of coarse categories and the size of training set are so small that the calculated entropy of word is not pretty effective and can not disclose its real importance.

5. Conclusions

In this paper we have proposed an effective feature-weighting model for question classification. The main idea aims at improving question classification accuracy by using the more reasonable weighted feature vector to represent a question. To alleviate the problem of limited labeled collection to some extent, we make use of both labeled and unlabeled question collections to calculate the weight of word. The experimental results demonstrate the usefulness of our new feature-weighting model in the task of question classification, comparing to traditional binary-value feature representation.

References

[1] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[2] C. C. Chang and C. J. Lin. Libsvm: a library for support vector machines. 80:604–611, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[3] Z. Cheung, K. L. Phan, A. Mahidadia, and A. Hoffmann. Feature extraction for learning to classify questions. *Proceedings of Advances in Artificial Intelligence (AI 2004)*, pages 1069–1075, 2004.

[4] T. David, J. L. Vicedo, B. Empar, and M. Lidia. Automatic feature extraction for question classification based on dissimilarity of probability distributions. *Advances in Natural Language Processing*, 2006.

[5] S. Dumais. Improving the retrieval information from external sources. *Behaviour Research Methods, Instruments and Computers*, 23:229–236, 1991.

[6] G. Hamerly and C. Elkan. Learning the k in k-means. *Advances in Neural Information Processing Systems*, 17, 2003.

[7] S. J. Li, J. Zhang, X. Huang, S. Bai, and Q. Liu. Semantic computation in a chinese question-answering system. *Journal of Computer Science and Technology*, 17(6):933–939, Nov 2002. 622GL Times Cited:1 Cited References Count:7.

[8] T. M. Mitchell. *Machine Learning*. McGraw-Hill Higher Education, 1997.

[9] D. Moldovan. Question-answering systems in knowledge management. *Ieee Intelligent Systems*, 16(6):90–92, Nov-Dec 2001. 498TR Times Cited:0 Cited References Count:2.

[10] M. Pasca and S. M. Harabagiu. High performance question/answering. *Proceedings of SIGIR*, pages 366–374, 2001.

[11] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[12] Y. Sasaki, H. Isozaki, T. Hirao, K. Kokuryou, and E. Maeda. Ntt's qa systems for ntcir qac-1. *Working Notes of the Third NTCIR Workshop Meeting*, pages 63–70, 2002.

[13] T. Solorio, M. Perez-Coutino, M. Montes-y Gomez, L. Villasenor-Pineda, and A. Lopez-Lopez. A language independent method for question classification. *Proc. of the 20th Int. Conf. on Computational Linguistics (COLING-04). Geneva, Switzerland*, 2004.

[14] V. N. Vapnik. *Statistical learning theory*. Wiley, 1998.

[15] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2000.

[16] E. M. Voorhees. Overview of the trec 2003 question answering track. *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*, 2003.

[17] D. Zhang and W. S. Lee. Question classification using support vector machines. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 26–32, 2003.