An Evaluation of Classification Models for Question Topic Categorization

Bo Qu
Information School, Renmin University of China, China, 100872. E-mail: qb8542@ruc.edu.cn

Gao Cong
Blk N4, 50 Nanyang Avenue, Singapore 639798. E-mail: gaocong@ntu.edu.sg

Cuiping Li
Information School, Renmin University of China, China, 100872. E-mail: licuiping@ruc.edu.cn

Aixin Sun
Blk N4, 50 Nanyang Avenue, Singapore, 639798. E-mail: axsun@ntu.edu.sg

Hong Chen
Information School, Renmin University of China, China, 100872. E-mail: chong@ruc.edu.cn

We study the problem of question topic classification using a very large real-world Community Question Answering (CQA) dataset from Yahoo! Answers. The dataset comprises 3.9 million questions and these questions are organized into more than 1,000 categories in a hierarchy. To the best knowledge, this is the first systematic evaluation of the performance of different classification methods on question topic classification as well as short texts. Specifically, we empirically evaluate the following in classifying questions into CQA categories: (a) the usefulness of n-gram features and bag-of-word features; (b) the performance of three standard classification algorithms (naive Bayes, maximum entropy, and support vector machines); (c) the performance of the state-of-the-art hierarchical classification algorithms; (d) the effect of training data size on performance; and (e) the effectiveness of the different components of CQA data, including subject, content, asker, and the best answer. The experimental results show what aspects are important for question topic classification in terms of both effectiveness and efficiency. We believe that the experimental findings from this study will be useful in real-world classification problems.

Received July 5, 2011; revised November 23, 2011; accepted November 25, 2011

 DOI: 10.1002/asi.22611

Introduction

Community Question Answering (CQA) services are Internet services that enable users to ask and answer questions, as well as to browse and search through historical question-answer pairs. Examples of such community-driven knowledge services include Yahoo! Answers (answers.yahoo.com), Naver (www.naver.com), Baidu Zhidao (zhidao.baidu.com), and WikiAnswers (wiki.answers.com). These CQA services have developed rapidly and accumulated a large number of questions and answers since their launches. For example, Yahoo! answers had 86 million resolved questions as of June 22, 2010.

Questions in CQA services are organized into hierarchies of categories that often comprise thousands of leaf categories, where each category represents a topic. Figure 1 shows a small part of Yahoo! Answers hierarchy. The questions in the same category or subcategory are usually relevant to the same general topic. For example, the questions in the subcategory "Travel.Australia.Sydney" mainly are relevant to travel in the city of Sydney.

The category information in CQA is quite useful in at least the following three aspects: (a) The hierarchy of categories facilitates browsing questions and answers; (b) the category-based organization enables searching questions and answers (that is, to find similar questions for a given question) within a specific subcategory; and (c) the category information can be utilized to enhance the question search

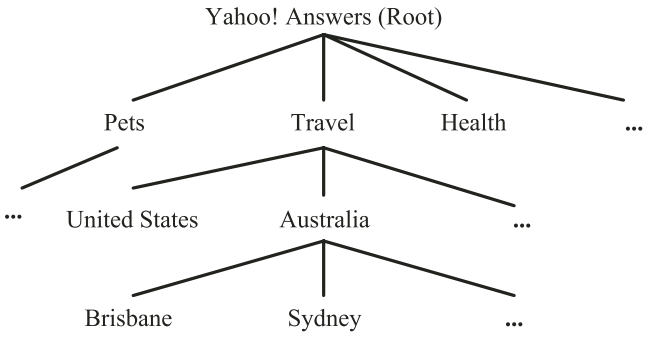

FIG. 1. Example category structure of Yahoo! Answers.

models, thus improving the performance of question search (Cao, Cong, Cui, Jensen, & Zhang, 2009).

In this article, we study a practical problem, namely, automatic question topic classification in CQA service. This problem is interesting because of the following aspects, in particular.

First, question topic classification will help to improve CQA services. When a user asks a new question in a CQA service, the question does not have a category, and thus most CQA services will request users to provide a category for the new question. In most of the current CQA services, *the user typically needs to manually choose a category label for a new question from a predefined hierarchy of categories*, which can easily comprise thousands of categories. Hence, it will make it easier for users if we can automatically suggest one or several categories for users to choose.

Second, we have access to a large number of QA pairs from Yahoo! Answers with 3.9 M questions organized in more than one thousand categories in a hierarchy. This dataset is much larger than those used in previous studies on text classification as shown in Table 1. The dataset enables us to systematically study the performance of different classification methods on short texts in the presence of a very large training data. This also enables us to study the effect of training data size on the classification performance. As each QA instance is typically much shorter than a normal document or web page, the experimental results will offer insight and guideline for other short text classification tasks, such as classifying twitter data and search query, and text-based video classification (Huang, Fu, & Chen, 2010).

Third, the QA data have some distinct features from document and web page data. A QA instance usually consists of a *subject* (i.e., question), an *asker*, *content*, one or more *answers* and among which a *best answer* is often identified through the voting of the community users or chosen by the asker. It is therefore interesting to evaluate the effectiveness of the different components of the CQA data in question topic classification task. Note that the question topic classification task is different from the traditional question classification defined in TREC QA (Li & Roth, 2002; Lin & Katz, 2006). We discuss this in the Related Work Section.

Contributions

With extensive experiments, this article makes three major contributions. First, we systematically study the problem of automatic question topic classification in Yahoo! Answers. To our best knowledge, this is the first work on question topic classification in CQA services. We study the effectiveness and efficiency of using two types of feature representation, bag-of-words and n-grams, for classification. We find that n-grams do not help for our classification task, which is different from the results reported in previous work (Cavnar & Trenkle, 1994) for text classification, where n-gram improves performance. We also study the usefulness of different components of QA pairs in question topic classification. Our experiments show that the features extracted from the component *subject* are more useful in classification than the features extracted from other components, while the combined features yield a better performance.

Second, we evaluate the effectiveness and efficiency of three standard classification methods, namely, naive Bayes (NB), maximum entropy (ME), and support vector machines (SVM) in question classification. Our work differs from previous evaluation work in two aspects: (a) We use short texts, while previous work consistently focused on normal text collection or web pages; and (b) we use a greater amount of data than the data used in previous work.

We have the following new findings for question classification. (a) In terms of micro-F score, NB is close to ME and SVM when the training data is very big while NB is still worse than SVM and ME in terms of macro-F score. (b) When we double the size of the training data, the performance gain for macro-F score is linear. However, the performance of classifiers has a linear increase when we use less than 0.6 M training data, but the improvement for micro-F score becomes smaller (sublinear) when the data size increases further. It is reported (Banko & Brill, 2001) that the accuracy of confusion sets problem (the problem of choosing the correct use of a word, e.g., {principle, principal}) increases linearly when the size of training data doubles, which is, however, a very different problem from text classification.

Third, considering that Yahoo! Answers has a hierarchy of categories, as do most other CQA services, we also employ state-of-the-art hierarchical classification methods, including single-path hierarchical (SPH) method (Koller & Sahami, 1997), multi-path hierarchical (MPH) method (Dumais & Chen, 2000), and refined hierarchical (RHC) method (Bennett & Nguyen, 2009). We combine the three methods with NB, ME, and SVM, respectively, and evaluate these combinations in terms of both effectiveness and efficiency.

To the best of our knowledge, no previous work has done a systematic evaluation of the different hierarchial classification methods, or their combination with different

DOI: 10.1002/asi

TABLE 1. Datasets used in previous studies and our work, including number of training/test/total instances.

Work	Dataset	#Training	#Test	#Total
[Bennett & Nguyen, 2009]	Open Directory Project	1.2 M	509 K	1.7 M
[Xue, Xing, Yang, & Yu, 2008]	Open Directory Project	1.2 M	130 K	1.3 M
[Cesa-Bianchi, Gentile, & Zaniboni, 2006]	*Reuters Corpus, Volume 1 (RCV1)*	40 K	16 K	56 K
[Liu et al., 2005]	Yahoo! Directory	0.5 M	275 K	0.8 M
[Wang & Zhou, 2001]	ACM Digital Library / IBM Patent	13 K/4 K	3 K/1 K	16 K/5 K
[Dumais & Chen, 2000]	LookSmart	50 K	10 K	60 K
[Labrou & Finin, 1999]	Yahoo! Dictionary	–	–	152 K
Our work	Yahoo! Answers	3.1 M	800 K	3.9 M

classification models. For example, Liu et al. (2005) compare SVM and hierarchical SVM, and Bennett and Nguyen (2009) compare NB and hierarchical NB. Our experimental results show that hierarchical classification methods perform similarly to their counterparts in terms of effectiveness, though they are more efficient. This finding is different from the experimental results reported in previous work on classifying normal documents or web pages, where hierarchical classification methods usually perform better. These results on short texts are complementary to previous work on evaluating classification methods, which clearly focused on either normal text collection or web pages.

Finally, we report detailed micro-F scores, macro-F scores, and runtime results of the state-of-the-art classification methods on a very large dataset. These results will offer guidelines for choosing appropriate classifiers (in terms of both effectiveness and efficiency) and sizes of training data for real applications, especially short text classification.

Paper Organization

The rest of this article is organized as follows. We begin detailing the different components of a QA pair and feature representation. We describe the classifiers we employed for question classification. Then we report the experimental study, followed by a review of the related work, and conclusions.

Problem Statement

In this article, we focus on the archive of Yahoo! Answers. In the sequence, we discuss the details of the question-answer instances in Yahoo! Answers. However, as most CQA services use similar structures as Yahoo! Answers, the results of this study are also applicable to other CQA services.

Each question in Yahoo! Answers encompasses six major components: *subject*, *content*, *asker*, *the best answer*, *all answers*, and *category*. We proceed to detail them. Figure 2 gives an example question, where asker is enclosed by a blue box, subject by a red box, and content by a green box.

Subject: Each question has a subject that is the title of a question, and is often an expression in a question. It usually contains the key information of a question.

Content: Content is an *optional* field. The purpose of adding content for a question is to provide more details on the subject of the question.

Asker: Asker is the person who asks a question.

Best Answer: The best answer to a question is determined by the asker or the community vote.

Other Answers: In addition to the best answer, a question may receive many other answers. We do not use this component in our work because it is less reliable compared with best answer.

Category: Each question belongs to a leaf category. Note that categories in Yahoo! Answers are organized as a tree, i.e., each category can belong to at most one parent category, which is called virtual category tree (Sun & Lim, 2001).

When a user asks a new question in Yahoo! Answers, the user needs to choose a category for the question. Obviously, at the time of asking a question, there is no answer to the question. Hence, we can only extract features from the other components (i.e., *subject*, *content*, and *asker*) at classification time.

We obtained the dataset used in this study from the Yahoo! Webscope datatset.[1] The dataset comprises 3.9 million English questions. The category hierarchy has three levels with 27 top-level categories, 305 second-level categories, and 1,096 leaf categories. Table 2 shows the number of questions that have *subject*, *content*, *asker*, and *best answer* components, respectively. We can see that only about half of the questions have *content* component. There are about 1 million distinct askers, i.e., each asker asks about 3.9 questions on average.

Table 3 gives the average number of questions in the categories in each level, the standard deviation of the number of questions, the maximum number of questions in the categories in each level, and the minimum number of questions in the categories in each level. Figure 3 shows the distribution of questions across the first-level categories. Figure 4 shows the number of child categories of the first-level categories.

[1]The Yahoo! Webscope dataset, available at http://research.yahoo.com/Academic_Relations.

DOI: 10.1002/asi

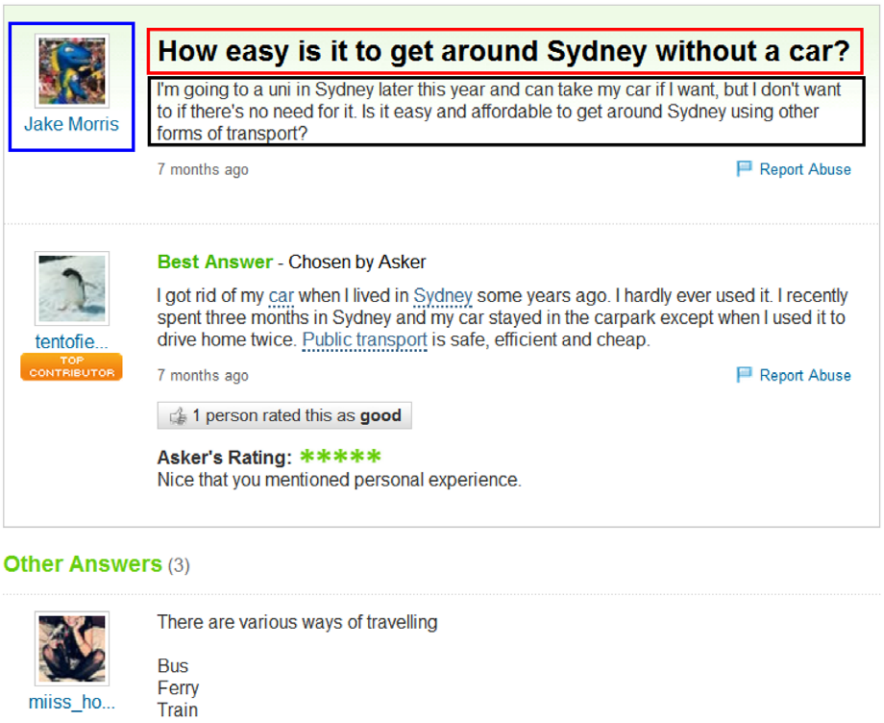

FIG. 2. An example of a QA pair. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

TABLE 2. Statistic information for different features.

Component	Questions	Subject	Content	Best answer	Asker
Number of instances	3.9 M	3.9 M	2.1 M	3.7 M	3.9 M

Note. M = million.

TABLE 3. Statistic on the number of questions in different levels of the category hierarchy.

Level	Average # questions	Standard deviation	Maximum # question	Minimum # question
1	142,700	125,315	512,659	277
2	12,632	23,469	237,104	1
3	3,515	13,353	237,104	1

Problem statement. Given a question with subject, content, and asker, we determine its leaf category in the category tree used in CQA archive.

Note that this does not mean that the best answer and other answers components cannot be used in building classifier. Like subject and content components, answers contain similar word features. As discussed earlier, one of the objectives of this article is to study the effectiveness of these components in classifying questions in CQA services.

Question Topic Classification

We present the methodology for question topic classification in Yahoo! Answers. Question topic classification involves a training phase and a test phase. During the training phase, a set of questions with known category labels is employed to train a classifier. During the test phase, the learned classifier is used to classify new questions.

Mapping Questions to Features

For each question, its components–subject, content, the best answer, and other answers–comprise words. We extract bags-of-words as features from these components. In addition, we also take into account word n-grams, i.e., sequences of n words. For example, the question subject "any good food" gives rise to two 2-grams "any good" and "good food." For the *asker* component, each *asker* becomes a feature.

Flat Question Classification Strategy

The flat strategy builds a flat classifier for the leaf categories without considering the nonleaf category in the hierarchical structure of categories. We explore three standard classification algorithms: NB, ME, and SVM.

NB. NB assumes conditional statistical independence of the individual features given the question category label. It computes the statistical distribution of each feature over the training set, which will be employed to find the category that is most likely to generate the observed feature vector in a given question. Specifically, we use the Weka (Hall, 2009) implementation of the multinomial distribution (Mccallum & Nigam, 1998). It has been shown that the NB with

DOI: 10.1002/asi

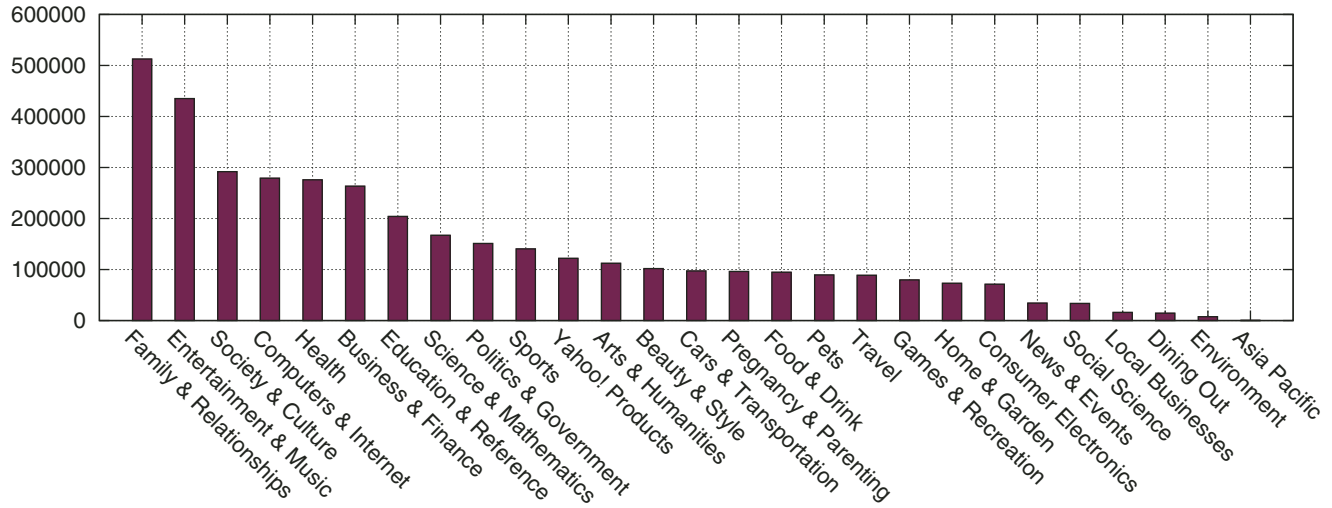

FIG. 3. Distribution of questions across the first-level categories. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

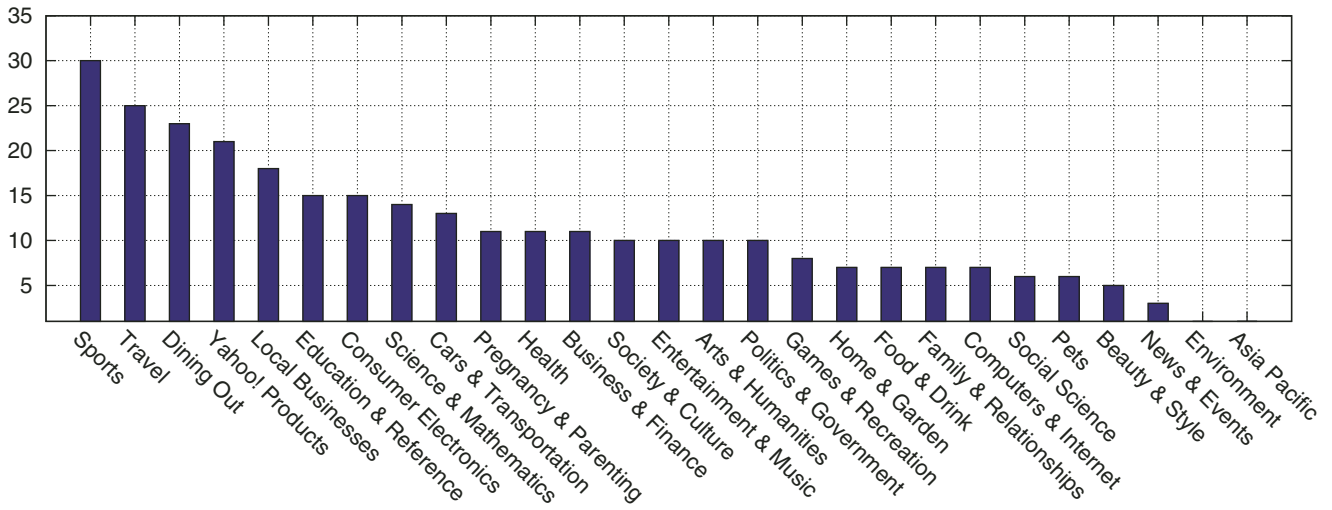

FIG. 4. The number of children categories of the first-level categories. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

multinomial distribution outperforms the NB using multivariate Bernoulli model (i.e., using binary word features; Mccallum & Nigam, 1998).

ME. ME computes the distribution of the observed features over the training set while maximizing the uncertainty of the distribution or the entropy. The optimization problem can be solved using iterative methods, iterative line search methods, gradient methods, iterative scaling methods, etc. Details about ME can be found in Manning and Klein (2003). Specifically, we use the Stanford implementation.[2]

SVM. SVM finds hyperplanes separating data of different categories. We use linear SVM because it is more effective and efficient for the applications with a huge number of instances as well as features (Keerthi, Sundararajan, Chang, Hsieh, & Lin, 2008). Specifically, we use the LIBLINEAR[3] implementation of multiclass linear SVM because it is suitable for solving large-scale problems (Keerthi et al., 2008), which is the winner of ICML 2008 large-scale learning challenge (linear SVM track).

Hierarchical Question Classification Strategy

The hierarchical strategy has been shown to be more efficient and sometimes more effective than the flat one in text classification (Koller & Sahami, 1997). We employ three hierarchical classification methods for question

[2]http://www-nlp.stanford.edu/software/classifier.shtml

[3]http://www.csie.ntu.edu.tw/~cjlin/libsvm/

DOI: 10.1002/asi

classification: single path hierarchical classifier (SPH; Koller & Sahami, 1997), multipath hierarchical classifier (MPH; Dumais & Chen, 2000), and refined hierarchical classifier (RHC; Bennett & Nguyen, 2009).

SPH. In the training phase of SPH (Koller & Sahami, 1997), for each internal node of the category tree, SPH trains a classifier using the questions belonging to its children nodes. In the testing phase, a test question is classified from top to bottom in the category tree along a single path. In other words, we use the classifier at the root node to classify a question into one of its children nodes, denoted by c_i, and then use the classifier at c_i to classify the question into one of c_i's children nodes. The process continues until a leaf node, i.e., the category of the test question, is reached.

MPH. The training phase of MPH (Dumais & Chen, 2000) is the same as that of SPH. However, MPH differs from SPH in the test phase in that SPH follows exactly a single path to reach the final category for a test question q, while MPH computes the classification probability of all paths and chooses the one with the highest combined probability to determine the category of the test data.

Specifically, the probability of a text question belonging to a leaf category is computed by $\prod_{i=1}^{h} P(L_i \mid q)$, where $P(\mathrm{L}_i \mid q)$ is the probability of question belonging to a category at level L_i. In our experiments, we find that the performance of MPH is very poor if we consider all the paths. Instead, we only choose the top-n children categories for each level rather than all the categories. Specifically, we set n at 5 in this paper.[4]

RHC. RHC follows the idea of the hierarchical classification method Refined Experts (Bennett & Nguyen, 2009). RHC encompasses a bottom-up phase and a top-down phase.

In the bottom-up phase, RHC trains binary classifiers at the lower level nodes using cross-validation over the training data, and then uses the membership predictions over the training data gathered during cross-validation as additional features available at the grandparent level node (to classify a question into the parent level categories). For example, in Figure 1, RHC trains binary classifier for each leaf node, e.g., at nodes "Brisbane" and "Sydney." For each training instance, its membership prediction (using cross-validation) at leaf nodes Brisbane and Sydney becomes additional features for the classifier built at node "Travel" (in the top-down phase) to classify a question into categories "United States," etc.

In the top-down phase, RHC builds a classifier at each node to classify a test into its children nodes using the enriched representation from the bottom-up phrase. Additionally, RHC employs the following optimization. RHC performs cross-validation over the training data and uses the predicted labels to filter the training data to a node, thus aligning the training distribution with what will likely occur during testing. Specifically, at a node, the model is trained using the actual training data at the node together with the training data that is misclassified to the node. For example, at node "Travel," if training data labeled with "Health" are classified to node Travel (known from cross-validation), then we augment the training data at Travel with those errors that form a new subcategory with label Health under Travel, to build classification model at Travel. At classification time, if a question is classified into a Health subcategory by the classifier at node Travel, then RHC will regard this as an error classification and move the question to the Health node.

[4]We tried different values at 5, 10, 15, 20, and 25. The performance difference is very small (in terms of micro-F score). The performance at 5 is the best, and hence is used in our experiments.

Experiments and Results

We design our experiments to evaluate the following:

- The usefulness of bag-of-words features and n-gram features
- The performance of the classifiers NB, ME, SVM, and the three hierarchical methods, in terms of both effectiveness and efficiency
- The effect of varying training data size on both effectiveness and efficiency
- The usefulness of the components of QA data, including subject, content, asker and best answer

Experimental Settings

Data preparation. We randomly select 20% of QA data from each leaf category of the whole dataset as the test data that nearly has 0.8 million QA instances. For the training data, we randomly select 20% of QA data in a similar manner to be the default training data when there is no need to evaluate the affect of varying training data size. Additionally, we also generate another six training datasets by randomly selecting 1%, 5%, 10%, 40%, 60%, and 80% of the whole dataset, respectively. These six datasets are used to evaluate the effect of varying training data. Note that there is no overlap between the training and test data instances, in all settings. We use bag-of-words features extracted from the subject component of the questions to represent questions when the feature representation or usefulness of question component is not evaluated.

Performance metric. To evaluate the performance of classifiers, we employ two popularly used measures, namely, micro-averaged and macro-averaged F_1 scores (Yang & Liu, 1999), denoted by micro-F score and macro-F score, respectively. All experiments are conducted on a server with 2.0 GHz Intel 2-Core CPU and 8 GB memory.

Evaluating Bag-of-Words and n-Gram Features

The first set of experiments is to evaluate the usefulness of n-gram features in question topic classification in terms of both the effectiveness and efficiency for both hierarchical

DOI: 10.1002/asi

TABLE 4. The effectiveness of n-gram features.

Classifier	Features (n-grams)	Flat		Hierarchical (SPH)	
		Micro-F score	Macro-F score	Micro-F score	Macro-F score
NB	1-gram	0.359	0.068	0.358	0.098
	1+2-grams	0.355	0.075	0.354	0.101
	1+2+3-grams	0.340	0.073	0.340	0.091
	1+2+3+4-grams	0.326	0.072	0.327	0.072
ME	1-gram	0.375	0.171	0.372	0.188
	1+2-grams	–	–	0.370	0.177
	1+2+3-grams	–	–	0.367	0.173
	1+2+3+4-grams	–	–	0.365	0.171
SVM	1-gram	0.391	0.183	0.382	0.201
	1+2-grams	0.390	0.181	0.381	0.190
	1+2+3-grams	0.389	0.181	0.380	0.190
	1+2+3+4-grams	0.387	0.180	0.380	0.189

Note. NB = naive Bayes; ME = maximum entropy; SVM = support vector machines; SPH = single-path hierarchical.

and flat classifiers. We consider four sets of features generated from subjects of questions: bag-of-words (i.e., 1-gram), 1+2-grams, 1+2+3-grams, and 1+2+3+4-grams, where 1+2-grams means the union of bag-of-word features and 2-gram features, and the others follow. We use the flat models presented in the Question Topic Classification Section and their combinations with the hierarchical model SPH in this experiment.

Table 4 shows the effectiveness of classifiers using different sets of features in terms of micro-F score and macro-F score. We can see that bag-of-words features (i.e., 1-gram) perform the best in terms of micro-F score, and the performance in terms of micro-F score drops slightly when we include larger n-grams as features. Note that ME ran out of memory when we used 1+2-grams.

We also experiment using 2-grams, 3-grams, or 4-grams alone as features. However the performance is much worse than when they are used together with 1-gram. Although it is reported that using n-gram features usually improves classification performance for document classification (Cavnar & Trenkle, 1994), it really does not help for question topic classification. The reason could be that questions are much shorter than normal documents, and thus the n-gram features are usually very sparse, especially when we increase the value of n.

Performance of Different Classifiers

This experiment is to compare the effectiveness and efficiency of the 12 different classification methods, including three flat classification models, and their combinations with the three hierarchical methods. Similar to the first set of experiments, the question features are extracted from the subject component using bag-of-words representation.

Figure 5(a) shows the effectiveness of different classification models in terms of micro-F score and macro-F score. Figure 5(b) reports the efficiency of different classification models. For consistency, classification methods based on the same model (e.g., NB) are plotted using the same color.[5]

We make the following observations:

- For the three hierarchical classification methods, in terms of both micro-F score and macro-F score, SPH performs similarly as does RHC, and both outperform MPH, when combined with either of the three standard classification models, NB, ME, and SVM. All the differences are statistically significant using t test, p-value $\ll 0.01$. It is a bit to our surprise that the simple SPH method achieves similar performance as does the state-of-the-art approach RHC. We are aware that in the application domain (Dumais & Chen, 2000) of RHC, an instance may belong to multiple categories, while in our problem, each question belongs to a single category. And we did not build the 1,096 binary classifiers on the leaf nodes in the bottom-up process of RHC because of the limited memory and the inconspicuous improvement with the binary classifiers on the higher level nodes. We are also aware that MPH is usually used in the case that an instance may belong to multiple categories in previous work.
- For the three flat models, in terms of micro-F score, SVM (0.391) outperforms NB (0.359) by 8.9% and ME (0.375) by 4.3% (See Table 4 and Figure 5(a)). In terms of macro-F score, SVM (0.183) outperforms NB (0.068) by 169% and ME (0.171) by 7%. SVM and ME greatly outperform NB in terms of macro-F score. All the improvements are statistically significant using t test, with p-value $\ll 0.01$.
- By comparing SPH method with its flat counterpart, we find that the hierarchical method really cannot improve flat classification model in terms of micro-F score (SPH+NB is slightly better than NB while SPH+ME and SPH+SVM are slightly worse than ME and SVM, respectively). However, SPH improves the standard flat classification models in term of macro-F score, e.g., SPH+NB over NB by 44%, SPH+ME over ME by 9.9% and SPH+SVM over SVM by 9.8%. Again, all the differences are statistically significant using a t test, with p-value $\ll 0.01$.

[5]For clarity, we recommend viewing all diagrams presented in this section directly from the color PDF, or from a color print copy.

DOI: 10.1002/asi

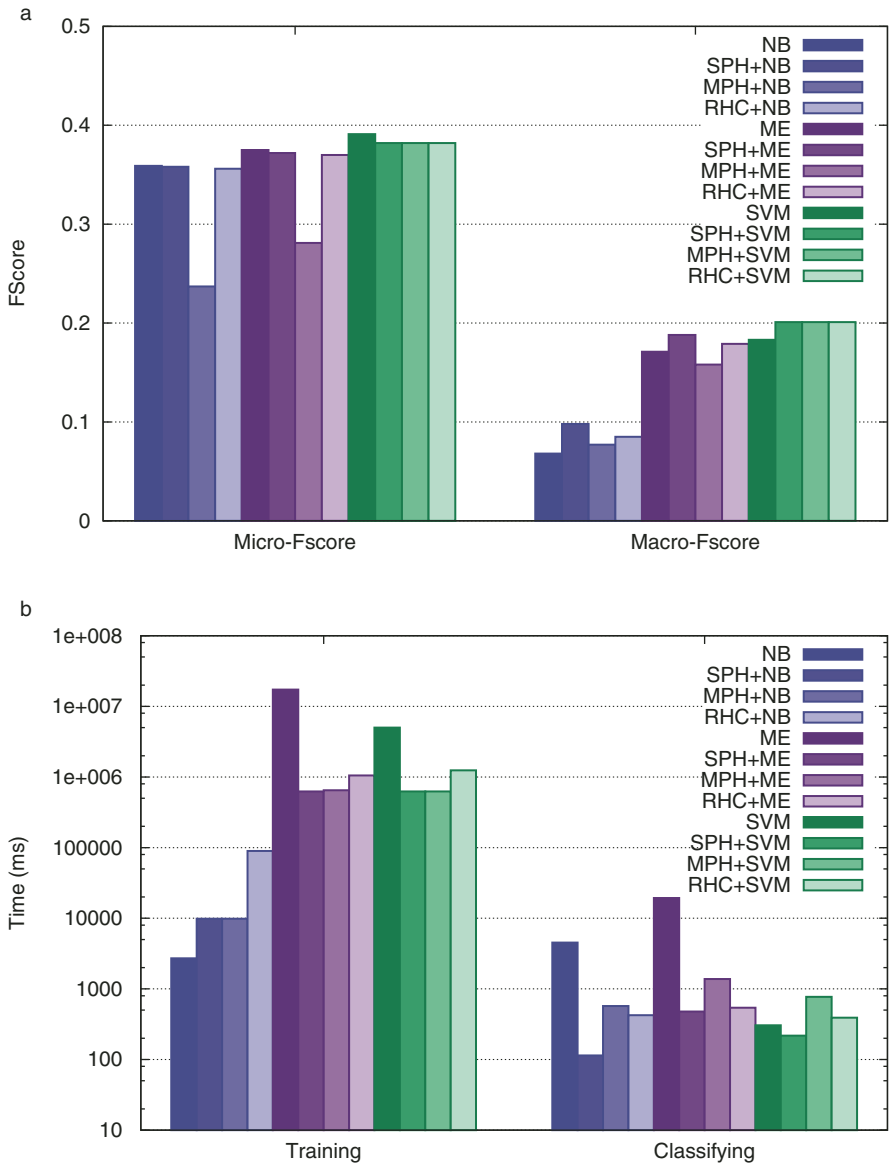

FIG. 5. The performance comparison of different classifiers. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

- In terms of classification time, the SPH method is able to speed up the standard flat classification models significantly: SPH+NB is an order of magnitude faster than NB, so do SPH+ME and ME; SPH+SVM saves 1/3 classification time of SVM. The hierarchical classifiers need less classification time in that the categories they consider are fewer than those of flat ones. For example, consider a training set with 64 categories that are organized as a four-level tree and each nonleaf node has four children nodes. The flat classifiers need to consider 64 categories to classify a test question, while the SPH only needs to do three times of classification, each of which SPH considers four categories. In addition, among the three hierarchical methods, MPH takes the longest classification time in that it needs to compute the combined probabilities of a number of paths for a test question.
- In terms of training time, SPH method can significantly reduce the training time of ME and SVM by orders of magnitude, while SPH+NB takes longer than NB. In addition, RHC is the most expensive in terms of training time, as expected.

By comparing the training time and classification time of the three standard classification models, we can see that ME is the most expensive in terms of both training time and classification time, SVM is also slow at training, but is the most efficient in classification, and NB is efficient at training, though it is slower than SVM at classification time. However, we would tame this finding because we use public tools for NB, ME, and SVM, which could be implemented with different levels of optimization.

DOI: 10.1002/asi

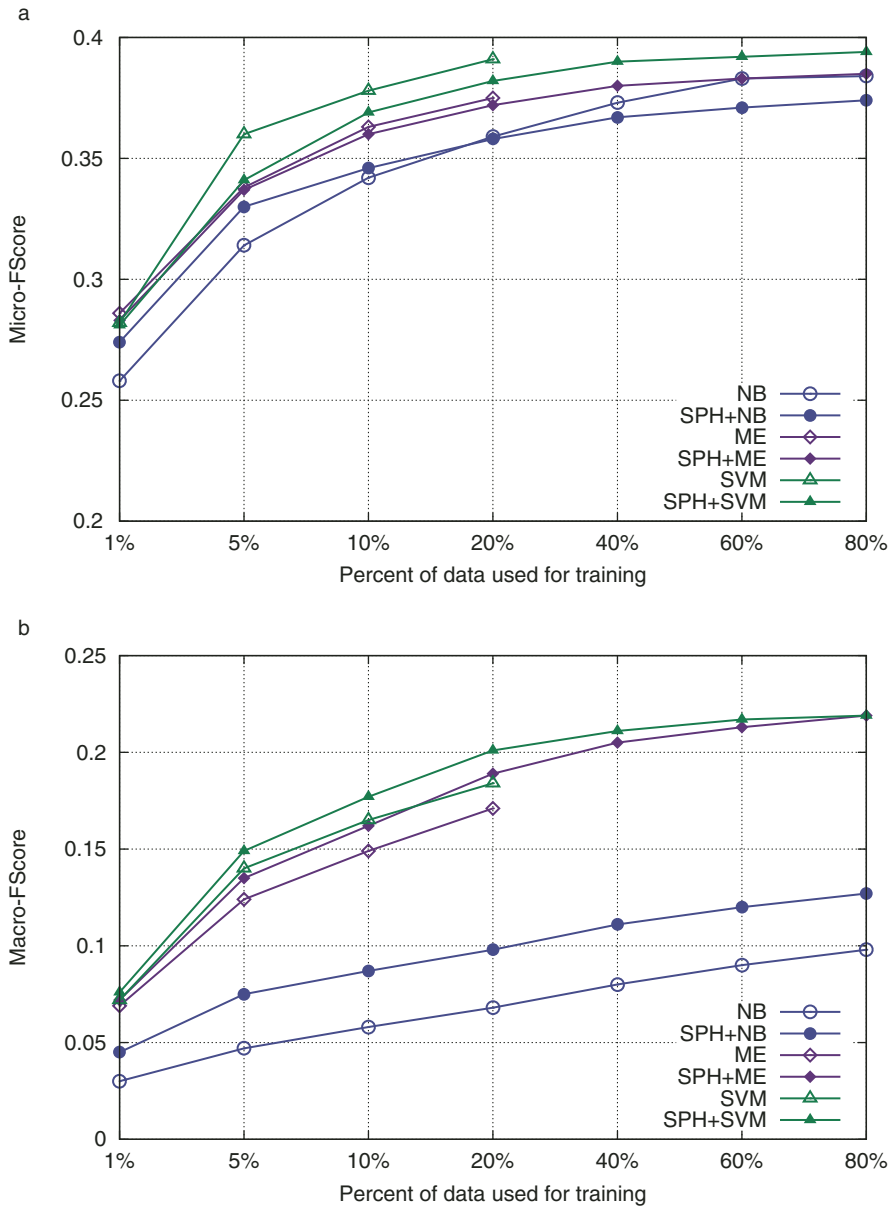

FIG. 6. The effectiveness of different classifiers when varying the size of training data. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Varying the Size of Training Data

This experiment is to study the performance of different classifiers by varying the size of training data. Specifically, we evaluate the effectiveness and efficiency of various flat classifiers and their combinations with SPH, the most effective hierarchial classification method as shown in the last set of experiments, when the size of the training data varies from 1% to 80% of the whole dataset. Note that 1% of the data corresponds to 0.039 million questions and 80% of the data corresponds to 3.1 M questions. Again, we extract bag-of-words features from the subject component of the questions to build classifiers.

Figure 6 shows the effectiveness of various classifiers on training data of different sizes. Note that flat ME and flat SVM ran out of memory when we used 40% of the data as the training data. For clarity and consistency, flat classification algorithms (NB, ME, and SVM) are plotted using empty symbols, while hierarchical algorithms are plotted using solid symbols. A flat classification model and its corresponding hierarchical algorithm are plotted using the same color.

The Flat approach focuses on the categories with more questions, while the Hierarchical approach focuses on the ones with less questions when the training data is large enough. This can be seen as the micro-F score (reflecting the performance of the categories with more questions) of the Flat approach is slightly higher than that of the hierarchical approach, while the macro-F score (reflecting the performance of less popular question categories) of the Flat approach is lower than that of the hierarchical approach.

DOI: 10.1002/asi

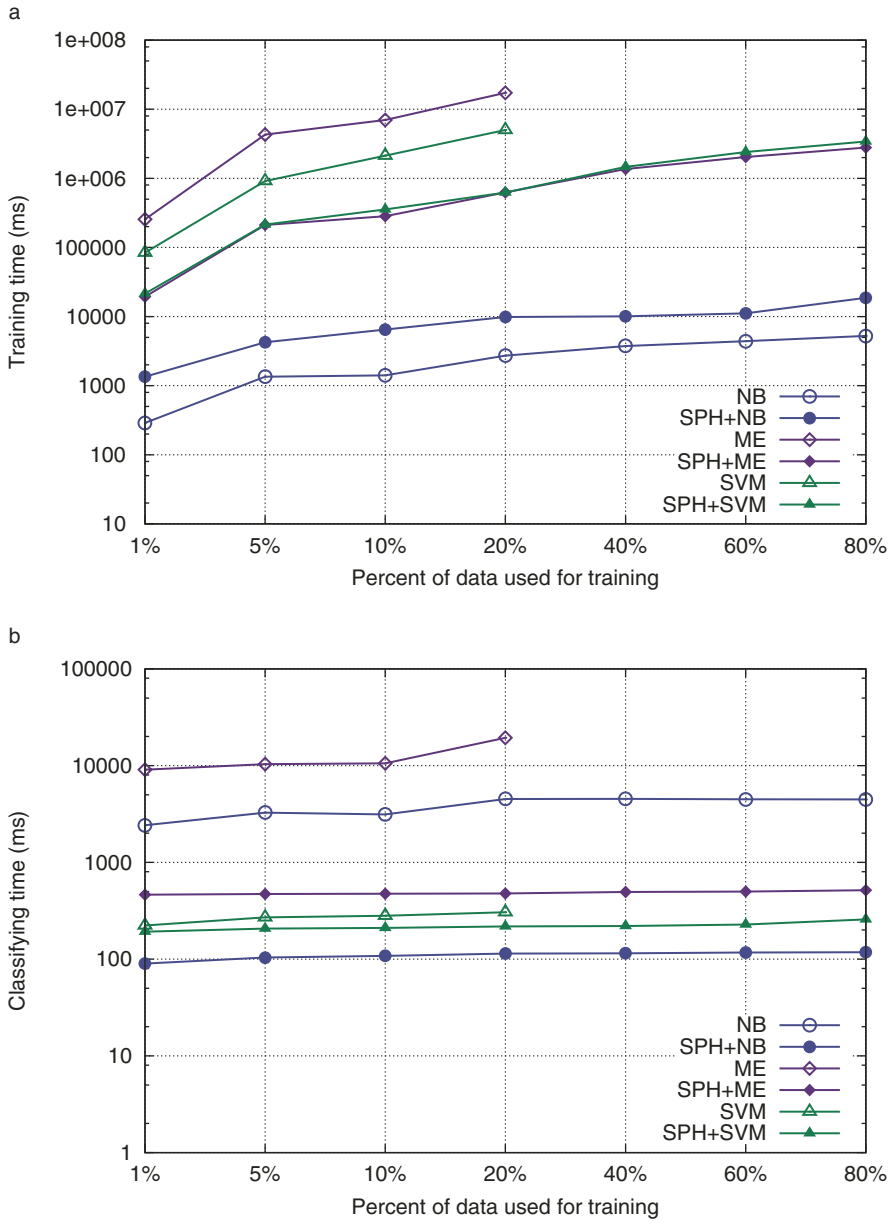

FIG. 7. The efficiency of different classifiers when varying the size of training data. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

We make the following two observations:

- As we increase the size of the training data, the performance of all methods improves in terms of both micro-F score and macro-F score. The amount of performance improvement is significant when the size of the training data is relatively small. However, when the size of the training data is relatively large, the performance becomes relatively stable and less sensitive to the size of training data. When the size of the training data is larger than 40% of all data, the absolute improvement in terms of both micro-F score and macro-F score caused by increasing the size of training data is less than 0.005 for all classifiers. For example, when we double the size of the training data from 5% to 10%, the performance of SPH+SVM changes from 0.341 to 0.369 (which represents a 8.2% improvement); however when we double the training data from 40% to 80%, the performance of SPH+SVM increases only by 1%.
- In general, the SVM classifiers, including the flat SVM and SPH+SVM, achieve the best performance, followed by ME classifiers, and NB classifiers, in turn, on training data of various sizes. However, when the size of training data is large, the difference among the different classifiers in terms of micro-F score is very small. When the size of the training data is relatively small (1% of all data), the ME classifiers perform better than others in terms of micro-F score.

Figure 7 shows the efficiency of the different classifiers with various sizes of training data. As expected, the training time increases with the increase of the size of the training

DOI: 10.1002/asi

TABLE 5. The effectiveness of different question components.

		Flat		Hierarchical (SPH)	
Classifier	Components	Micro-F score	Macro-F score	Micro-F score	Macro-F score
NB	Subject	0.359	0.068	0.358	0.098
	Content	0.344	0.062	0.338	0.079
	Asker	0.098	0.004	0.105	0.009
	Best answer	0.246	0.068	0.234	0.084
ME	Subject	0.375	0.171	0.372	0.188
	Content	0.329	0.151	0.310	0.120
	Asker	0.058	0.016	0.059	0.017
	Best answer	–	–	0.267	0.107
SVM	Subject	0.391	0.183	0.382	0.201
	Content	0.346	0.120	0.326	0.120
	Asker	0.108	0.017	0.108	0.017
	Best answer	0.299	0.105	0.248	0.072

Note. NB = naive Bayes; ME = maximum entropy; SVM = support vector machines; SPH = single-path hierarchical.

TABLE 6. The effectiveness of different components for NB classifier.

	Flat NB		Hierarchial SPH+NB	
Components	Micro-F score	Macro-F score	Micro-F score	Macro-F score
All components	0.398	0.078	0.384	0.099
No subject	0.358	0.072	0.346	0.088
No best answer	0.387	0.070	0.382	0.089
No content	0.376	0.073	0.366	0.094
No asker	0.393	0.086	0.376	0.100

Note. SPH = single-path hierarchical; NB = naive Bayes.

data for all methods. We also observe that the runtime of NB and SPH+NB increases mildly with the size of training data, while the runtime of methods based on ME and SVM increases more dramatically.

Evaluating the Usefulness of Different Question Components

Sole component. This experiment is to study the usefulness of the different components in question topic classification when they are used alone. Specifically, we use features extracted from a single component, i.e., subject, content, asker, or best answer, to build classifiers. We use each asker as a feature and use bag-of-words features for other components. At classification time, features extracted from subject, content, and asker are available, but not from answer component. However, we find that using content features nearly cannot improve the effectiveness of classifiers, while it will take longer time. Hence, we only use features extracted from subject and asker at classification time.

Table 5 shows the effectiveness of different components. We can see that classifiers using the subject component have better performance than others. All the differences are statistically significant using a t test, p-value $\ll 0.01$. It is expected that subject component is effective because it usually contains the core information of the questions. In contrast, the component content is not always given by the asker. The component best answer might contain both the useful information for classification and some noise. The features extracted from asker component are very sparse and the performance of classifiers using asker component is poor as expected.

Ablation experiment. We present two experiments to study the usefulness of question components in question topic classification. This experiment is to explore which component plays a more important role in question topic classification. We remove a component from the four components each time to explore the effect of the removal on the effectiveness of NB and SPH+NB, which run much faster than SVM and ME. Table 6 reports the result. Without the subject component, both the macro-F score and micro-F score drop most severely. In contrast, the asker component appears to have less effect. These results are in accord with the experimental results reported in the last experiment (sole component)–they show that the subject component is the most important one.

Summary and Discussion

The main findings in our experiments are summarized as follows:

- The features extracted from subjects are more useful in classification than features extracted from other components, while the combined features are the most effective.

DOI: 10.1002/asi

- N-gram features nearly do not help in question classification.
- SVM outperforms NB and ME in terms of effectiveness, while NB takes the shortest training time and SVM is the most efficient at classification time. The performance of NB is close to SVM and ME in terms of micro-F score when the size of the training data is very large, while NB is still worse than SVM and ME in terms of macro-F score.
- SPH is more efficient than the other two hierarchial classification methods and achieves better or similar performance in terms of effectiveness.
- SPH cannot improve the standard flat classification models in terms of effectiveness, while it improves efficiency greatly.
- When we double the size of the training data, the performance gain for macro-F score is linear. In terms of micro-F score, the performance of classifiers has a linear increase when we use less than 0.6 million (20%) QA instances as the training data; however the improvement becomes smaller (sublinear) as the data size increases.

These findings are new for CQA data (or short texts; see the Introduction section for the comparison with the findings in previous work on text classification). There findings can provide detailed practical guidelines for choosing an algorithm and the size of training data for question classification.

Although there are publications on evaluating the methods of document classification, previous evaluation work has not been conducted on QA data or other short texts. The performance of text classification methods on question classification is a research challenge. Even if we consider the work for the traditional text classification, our work still differs significantly from the existing work in that no work evaluates the state-of-the-art hierarchal classification methods and their combinations with different classification models (NB, SVM, and ME) as we do.

Related Work

We proceed to discuss related work.

Question Classification in TREC QA

No published work exists for classifying "questions" into the category hierarchies of CQA, where each category represents a topic.

Most existing proposals e.g., (Zhang & Lee, 2003; Moschitti, Quarteroni, Basili, & Manandhar, 2007; Blunsom, Kocik, & Curran, 2006; Suzuki, Taira, Sasaki, & Maeda, 2003) are on classifying questions based on the expected answer types (also called targets, e.g., location, time, human, numerical value) as in TREC QA track. These proposals usually use thousands of human annotated questions as training data and classify questions into tens of categories (e.g., 56 categories in Li and Roth, 2002). Such question classification can be utilized to find answers from document corpus but is very different from the question topic classification that classify questions into category hierarchies (that represent the topics of questions, e.g., "Travel.China") as studied in this work.

In the TREC QA track, the existing proposals usually exploit some natural language features, such as the POS tags of words and syntactic structure of questions, and some words and their locations (e.g., questions beginning with "who" are likely to belong to "human' ' category.) In contrast, the categories that we use are the topics of questions, rather than the targets (or types) of questions. Our classification taxonomy is like those studies in text classification and web page classification, where bag-of-words are used as features predominantly.

The POS tag and syntactic structure features used in question classification of TREC QA would not work for question topic classification of CQA because one category (topic) of CQA taxonomy may contain all types of questions in TREC QA. A specific example is that "who" questions often indicate "person" type of TREC question classification, and "which country" indicates "location" type of TREC questions. Such features could be captured by POS tags and syntactic structure features. However, who questions can belong to any topic in the question topic classification task, and so do which country questions. Additionally, unlike TREC questions, the questions in CQA are often informal and it is more challenging for parsers to parse their syntactic structures (Wang, Ming, & Chua, 2009). Hence, the POS tag and syntactic structure features used in TREC QA will not be distinguishable in topic classification of CQA. We also notice a recent work (Harper, Weinberg, Logie, & Konstan, 2010) that classifies questions in CQA into the types of TREC QA.

Text Classification

Standard flat classification algorithms, such as SVM, NB, k-nearest neighbor (kNN), are evaluated for document classification (Sebastiani, 2002).

Sun and Lim (2001) divide the existing work on hierarchical text classification into two categories: the big-bang approach and the top-down approach. In the big-bang approach, a single classifier is trained and used to assign one or more internal or leaf categories of the category tree to a test document. The big-bang approach has been developed using a Rocchio-like classifier (Labrou & Finin, 1999), association rules-based classifier (Wang & Zhou, 2001), and SVM (Cai & Hofmann, 2004). In our problem where all questions belong to the leaf level, there is almost no difference between the big-bang approach and the traditional flat classifier.

In the top-down approach, one or more flat classifiers are constructed at each level of the category tree in the training phase, while in the classification phase, each test document is classified from the higher levels to lower ones until it reaches a final category (Sun & Lim, 2001). The top-down level-based approach can be implemented by different strategies (Silla & Freitas, 2011), namely, single path strategy, multi-path strategy, and two stage strategy. Koller and Sahami (1997) implement the top-down approach with a single path strategy based on multiple Bayesian classifiers. Dumais and Chen (2000) propose a multipath method based on SVM.

DOI: 10.1002/asi

Liu et al. (2005) evaluate the performance of flat SVM and multipath SVM method on a web page set from Yahoo! Directory. As reported in their work, the hierarchical classifier is more effective than the flat one.

Xue, Xing, Yang, and Yu (2008) propose a two-stage approach for hierarchical text classification in which the first search stage is to prune the large-scale hierarchy to a set of category candidates for each document. The second stage is to train a classification model based on this pruned category tree. Very recently, Bennett and Nguyen (2009) proposed another two-stage approach, RHC.

We evaluate the performance of three flat classification methods, and three hierarchical classification methods including the recent approach (Bennett & Nguyen, 2009). As discussed in the Introduction, compared with the previous work, we use a much larger dataset as shown in Table 3 and questions in our problem are shorter than normal documents or webpages.

Query Classification

The 2005 KDD Cup has motivated interests on the topical classification of web queries. The KDD Cup dataset comprises only 111 training queries and 800 test queries, and the task is to classify queries into 67 categories. Because of the lack of substantial training data and the sparseness of query features, research efforts focus on enlarging the training data and enriching the feature representation of queries, and different approaches have been proposed.

Many proposals on the topical classification of queries aim to enrich the feature representation of queries. The winning solution of 2005 KDD Cup (Shen et al., 2006) expands each query by its search engine results to derive features and builds document classifiers based on a document taxonomy, e.g., Open Directory Project (www.dmoz.org), and then classifications in the document taxonomy are mapped to those in the target taxonomy of queries. This solution addresses the feature sparseness problem of short queries by enriching query with features extracted from search engine results, and it addresses the problem of lack of training data by building document classifier using other existing taxonomies containing more training data. Beitzel et al. (Beitzel, Jensen, Chowdhury, & Frieder, 2007) employ a larger training data and test data, containing 6,666 and 10,000 queries, respectively, and classify queries into 18 categories. They find that a simple method of using the snippets of the retrieval documents of 6,666 queries to build classifiers performs 40% better than the strategy of bridging the classification results of an external taxonomy to the target taxonomy for the query classification. Broder et al. (2007) transform the problem of query classification into that of document classification. They also enrich the queries with their search engine results without using an external taxonomy for bridging.

Several studies have employed a semisupervised method to enlarge training data. Li et al. (Li, Wang, Shen, & Acero, 2010) proposed exploiting user click-through data to increase the amount of training data. Beitzel et al. (2005) propose an approach to mine the vast amount of unlabeled data in web query logs to improve automatic topical web query classification. Human-machine interaction (Schumaker, & Chen, 2010) is another approach to address the feature sparseness of short queries. Liu and Lin (2003) construct and maintain a profile for each information category, and propose an effective method of interacting with the users to map users' information needs expressed by a short query to suitable categories.

Discussion. The existing techniques for query classification are developed for a very different setting from our work. Those techniques would not be practical or useful for the question classification task at a large scale.

Most of research on query classification is motivated by the 2005 KDD Cup. The main challenge there is that queries do not come with labels and need to be manually labeled to create training/test data. The size of the available training data is very small (only 111 training queries). The techniques developed are to address such issues.

The winning solution of 2005 KDD Cup and some subsequent research use external document taxonomy containing a larger number of training data to build classifiers, and then classifications in the external document taxonomy are mapped to those in the target taxonomy of queries. However, lack of training data is not a problem for CQA classification– each QA has a label in CQA services and the size of training data is huge (Yahoo! Answers had 98 million questions as of August 2010). We use 3.9 million QA data elements in our experiments. It is reported by Beitzel et al. (2007) that with a larger size of training data and test data of queries, containing 6,666 and 10,000 queries, respectively, mapping to an external taxonomy does not help.

Furthermore, some studies expand each query with the snippets of the retrieval documents of the query by posing each query to a search engine. This will not be applicable to our problem due to two reasons: (a) questions are much longer than queries and search engines usually cannot return relevant results for questions (it is known that search engines cannot handle long queries well; Bendersky & Croft, 2008). We tested 10 questions, but most of the top-10 returned snippets appear to be noise rather than relevant to questions. For example, for a real Yahoo! answer question "Wat is the best way to talk my mom into letting me get a snake???", only the webpage from Yahoo! Answers is related (as of October 11, 2011). (b) It is too time consuming to process 3 million questions using such a method. Moreover, it takes time to pose a user's question to a search engine and then extract the top snippets to suggest the category of a question for the user in CQA services. Such a strategy is impractical for a CQA service. In fact, questions are normally much longer than the Web queries, and thus feature expansion for question would be less necessary.

Additionally, none of existing work on query classification has conducted a systematic evaluation of different

DOI: 10.1002/asi

classification models on a large amount of training data for query classification.

Finally, we note that other work on CQA includes question search e.g., (Cao, Duan, Lin, & Yu, 2011; Wang et al., 2009; Cao et al., 2009; Cao, Cong, Cui, & Jensen, 2010), surveys on CQA services (e.g., Agichtein, Castillo, Donato, Gionis, & Mishne, 2008; Gazan, 2011), and CQA integration (Wei, Cong, Li, Ng, & Li, 2011).

Conclusion

Question topic classification is a real problem for the CQA services. Performance of text classification methods on question classification (and short text classification) is a research challenge. This article is the first work on question topic classification. We apply a variety of flat classifiers, NB, ME, and SVM and combine them with state-of-the-art hierarchical models for question classification. We also study the usefulness of several components of questions in classification and different feature representations. Experiments are conducted on a very large dataset from Yahoo! Answers to evaluate the performance of different classifiers, the usefulness of different features, and the effect of training data size. We have reported some new findings from our experimental study. The results reported in this article could have immediate and practical affect on CQA services and probably other short text classification, such as classifying tweets, which is attracting the attention of the research community.

References

Agichtein, E., Castillo, C., Donato, D., Gionis, A., & Mishne, G. (2008). Finding high-quality content in social media. In Proceedings of the International Conference on Web Search and Web Data Mining (WSDM) (pp. 183–194). New York: ACM Press.

Banko, M., & Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation. In Proceedings of the 39th Annual Meeting on Association for Computational Linguistics (ACL) (pp. 26–33). Stroudsburg, PA: Association for Computational Linguistics.

Beitzel, S. M., Jensen, E. C., Chowdhury, A., & Frieder, O. (2007). Varying approaches to topical web query classification. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 783–784). New York: ACM Press.

Beitzel, S. M., Jensen, E. C., Frieder, O., Lewis, D. D., Chowdhury, A., & Kolcz, A. (2005). Improving automatic query classification via semi-supervised learning. In Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM) (pp. 42–49). Menlo Park, CA: IEEE Computer Society.

Bendersky, M., & Croft, W. B. (2008). Discovering key concepts in verbose queries. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 491–498). New York: ACM Press.

Bennett, P. N., & Nguyen, N. (2009). Refined experts: Improving classification in large taxonomies. In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 11–18). New York: ACM Press

Blunsom, P., Kocik, K., & Curran, J. R. (2006). Question classification with log-linear models. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 615–616). New York: ACM Press.

Broder, A. Z., Fontoura, M., Gabrilovich, E., Joshi, A., Josifovski, V., & Zhang, T. (2007). Robust classification of rare queries using web knowledge. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 231–238). New York: ACM Press.

Cai, L., & Hofmann, T. (2004). Hierarchical document categorization with support vector machines. In Proceedings of the 13th ACM International Conference on Information and Knowledge Management (CIKM) (pp. 78–87). New York: ACM Press.

Cao, X., Cong, G., Cui, B., & Jensen, C. S. (2010). A generalized framework of exploring category information for question retrieval in community question answer archives. A generalized framework of exploring category information for question retrieval in community question answer archives. In Proceedings of the 19th International Conference on World Wide Web (WWW) (pp. 201–210). New York: ACM Press.

Cao, X., Cong, G., Cui, B., Jensen, C. S., & Zhang, C. (2009). The use of categorization information in language models for question retrieval. In Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM) (pp. 265–274). New York: ACM Press.

Cao, Y., Duan, H., Lin, C. Y., & Yu, Y. (2011). Re-ranking question search results by clustering questions. Journal of the American Society for Information Science and Technology, 62, 1177–1187.

Cavnar, W. B., & Trenkle, J. M. (1994). N-gram-based text categorization. In Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval (pp. 161–175). Hingham, MA: Kluwer Academic Publishers.

Cesa-Bianchi, N., Gentile, C., & Zaniboni, L. (2006). Hierarchical classification: Combining Bayes with SVM. In Proceedings of the 23rd international conference on Machine learning (ICML) (pp. 177–184). New York: ACM Press.

Dumais, S., & Chen, H. (2000). Hierarchical classification of Web content. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 256–263). New York: ACM Press.

Gazan, R. (2011). Social Q&A. Journal of the American Society for Information Science and Technology, 62(12), 2301–2312.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I H. (2009). The WEKA data mining software: An update. SIGKDD Explorations, 11(1), 10–18.

Harper, F. M., Weinberg, J., Logie, J,. & Konstan, J. A. (2010). Question types in social Q&A sites. First Monday, 15(7).

Huang, C., Fu, T., & Chen, H. (2010). Text-based video content classification for online video-sharing sites. Journal of the American Society for Information Science and Technology, 61, 891–906.

Keerthi, S. S., Sundararajan, S., Chang, K. W., Hsieh, C. J., & Lin, C. J. (2008). A sequential dual method for large scale multi-class linear svmsA sequential dual method for large scale multi-class linear svms. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 408–416). New York: ACM Press.

Koller, D., & Sahami, M. (1997). Hierarchically classifying documents using very few words. In Proceedings of the 14th International Conference on Machine Learning (ICML) (pp. 170–178). San Francisco: Morgan Kaufmann Publishers Inc.

Labrou, Y., & Finin, T. (1999). Yahoo! as an ontology: Using Yahoo! categories to describe documents. In Proceedings of the Eighth International Conference on Information and Knowledge Management (CIKM) (pp. 180–187). New York: ACM Press.

Li, X., & Roth, D. (2002). Learning question classifiersLearning question classifiers. In Proceedings of the 19th International Conference on Computational Linguistics (COLING) (Vol. 1, pp. 1–7). Stroudsburg, PA: Association for Computational Linguistics.

Li, X., Wang, Y. Y., Shen, D., & Acero, A. (2010). Learning with click graph for query intent classification. ACM Transactions on Information Systems, 28, 12:1–12:20.

Lin, J., & Katz, B. (2006). Building a reusable test collection for question answering. Journal of the American Society for Information Science and Technology, 57, 851–861.

DOI: 10.1002/asi

Liu, R. L. (2010). Context-based term frequency assessment for text classification. Journal of the American Society for Information Science and Technology, 61(2), 300–309.

Liu, R. L., & Lin, W. J. (2003). Mining for interactive identification of users' information needs. Information Systems, 28, 815–833.

Liu, T. Y., Yang, Y., Wan, H., Zeng, H. J., Chen, Z., & Ma, W. Y. (2005). Support vector machines classification with a very large-scale taxonomy. SIGKDD Explorations, 71, 36–43.

Manning, C., & Klein, D. (2003). Optimization, maxent models, and conditional estimation without magic. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL): Tutorials (Vol. 5, pp. 8). Stroudsburg, PA: Association for Computational Linguistics.

Mccallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification. In Proceedings of the Association for the Advancement of Artificial Intelligence Workshop on Learning for Text Categorization (AAAI '98) (pp. 41–48). Palo Alto, CA: Association for the Advancement of Artificial Intelligence.

Moschitti, A., Quarteroni, S., Basili, R., & Manandhar, S. (2007). Exploiting syntactic and shallow semantic kernels for question answer classification. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL). Prague, Czech Republic. Stroudsburg, PA: Association for Computational Linguistics.

Schumaker, R. P., & Chen, H. (2010). Interaction analysis of the alice chatterbot: A two-study investigation of dialog and domain questioning. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 40(1) 40–51.

Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Survey, 34(1), 1–47.

Shen, D., Pan, R., Sun, J. T., Pan, J. J., Wu, K., & Yin, J. (2006). Query enrichment for web-query classification. ACM Transactions on Information Systems, 24, 320–352.

Silla, C. N., & Freitas, A. A. (2011). A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery, 22(1-2), 31–72.

Sun, A., & Lim, E. P. (2001). Hierarchical text classification and evaluation. In Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM) (pp. 521–528). Menlo Park, CA: IEEE Computer Society.

Suzuki, J., Taira, H., Sasaki, Y., & Maeda, E. (2003). Question classification using HDAG kernel. In Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering (MultiSumQA) (Vol. 12, pp. 61–68). Stroudsburg, PA: Association for Computational Linguistics.

Wang, K., Ming, Z., & Chua, T. S. (2009). A syntactic tree matching approach to finding similar questions in community-based QA services. In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 187–194). New York: ACM Press.

Wang, K., & Zhou, S. (2001). Hierarchical classification of real life documents. In Proceedings of the First SIAM International Conference on Data Mining (SDM), Chicago, IL.

Wei, W., Cong, G., Li, X., Ng, S. K., & Li, G. (2011). Integrating community question and answer archives. In Proceedings of the 25th Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence (AAAI) (pp. 1255–1260). Palo Alto, CA: Association for the Advancement of Artificial Intelligence.

Xue, G. R., Xing, D., Yang, Q., & Yu, Y. (2008). Deep classification in large-scale text hierarchies. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 619–626). New York: ACM Press.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 42–49). New York: ACM Press.

Zhang, D., & Lee, W S. (2003). Question classification using support vector machines. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 26–32). New York: ACM Press.

DOI: 10.1002/asi