Ensemble Approach for Fine-Grained Question Classification in Bengali

Somnath Banerjee
Department of Computer Science and Engineering
Jadavpur University, India
s.banerjee1980@gmail.com

Sivaji Bandyopadhyay
Department of Computer Science and Engineering
Jadavpur University, India
sivaji_cse_ju@yahoo.com

Abstract

This paper demonstrates that ensemble of multiple models achieves satisfactory classification performance in the task of question classification. Question classification plays a key role in automated question answering system by reducing the search space needed to find the answer. We have exploited state of the art ensemble techniques, i.e., bagging and boosting on lexical, syntactical and semantic features of Bengali questions for the question classification task. Naïve Bayes, kernel Naïve Bayes, Rule Induction and Decision Tree classifiers have been used as the base learners for bagging and boosting. The proposed work extends the single-layer Bengali question classification taxonomy to two-layer taxonomy by adding fine-grained classes for the coarse-grained classes. Sixty nine fine-grained classes have been proposed for nine coarse-grained classes. The experimental results show that boosting approach achieves slightly better accuracy than that of bagging approach in the task of Bengali question classification.

1 Introduction

Because of the huge size, high dynamics, and large diversity of the information on the Internet, researches on question answering (QA) are becoming very popular and challenging. QA systems focus on how to respond to users queries with exact answers. In recent years, many international question answering contests have been held at conferences and workshops, such as Text REtrieval Conference (TREC), Cross Language Evaluation Forum (CLEF) and NII Test Collection for IR Systems (NTCIR). Although Bengali is the sixth most spoken languages in the world, no QA contest in Bengali has been conducted so far.

Question Classification (QC) is an important component of QA System. The task of a QC is to assign one or more class labels, depending on classification strategy, to a given question written in natural language. For example, for the question What London street is the home of British journalism?, the task of question classification is to assign label "Location" to this question. Since we predict the type of the answer, QC is also referred as answer type prediction. The set of predefined categories which are considered as question classes are usually called question taxonomy or answer type taxonomy. QC has a key role in automated QA systems. Although different types of QA systems have different architectures, most of them follow a framework in which QC plays an important role (Voorhees, 2001). Furthermore, it has been shown that the performance of QC has significant influence on the overall performance of a QA system (Ittycheriah et al., 2001; Hovy et al., 2001; Moldovan et al., 2003).

Basically there are two main motivations for QC: locating the answer and choosing the search strategy. Knowing the question class not only reduces the search space needed to find the answer, it can also find the true answer in a given set of candidate answers. On the other hand, question class can also be used to choose the search strategy when the question is reformed to a query over information retrieval (IR) engine. For example, consider the question "What is a pyrotechnic display?". Identifying that the question class is "definition", the searching template for locating the answer can be for example "pyrotechnic display is a ..." or "pyrotechnic displays are ...", which are much better than simply searching by question words.

One of the main issues of classification modeling is the improvement of classification accuracy.

27th Pacific Asia Conference on Language, Information, and Computation pages 75—84

For that purpose, many researchers have recently placed considerable attention to the task of classifier combination methods. The idea is not to rely on a single decision making scheme. Instead, many single classifiers are used for decision making by combining their individual opinions to arrive at a consensus decision.

2 Related Work and Motivations

A lot of researches on question classification, question taxonomies, question features and question classifiers are being published continuously. Question classification in TREC QA has been intensively studied during the past decade. There are mainly two different approaches used to classify questions- one is rule based and another is machine learning based. However, a number of researchers have also used some hybrid approaches which combine rule-based and machine learning based approaches (Huang et al., 2008; Silva et al., 2011).

Rule based approaches use some manually handcrafted grammar rules to analyze the question to determine the answer type (Hull, 1999; Prager et al., 1999). Though handcrafted rules have been used successfully but suffer from the need to define too many rules to determine specific types (Li and Roth, 2004). (Li and Roth, 2004) also stated that though rule-based approaches may perform well on a particular dataset but they may have quite a poor performance on a new dataset and consequently it is difficult to scale them. So it is difficult to make a manual classifier with a limited amount of rules. On the other hand, machine learning-based approaches perform the QC by extracting some features from the questions, train a classifier and predicting the question class using the trained classifier. Many researchers have employed machine learning methods (e.g., maximum entropy and support vector machine) by using different features, such as syntactic features (Zhang et al., 2003) and semantic features (Moschitti et al., 2007). However, these methods mainly focused on English factoid questions and confined themselves to classify a question into two or a few predefined categories (e.g., ”what”,”how”, ”why”, ”when”, ”where” and so on).

There are also some notable studies that have used hybrid approach i.e., both rule-based and machine learning based approaches together. The most successful study (Silva et al., 2011) that works on question classification, first match the questions with some pre-defined rules and then use the matched rules as features in the machine learning-based classifier. The same approach is used in the work by (Huang et al., 2008). Machine learning-based and hybrid methods are the most successful approaches on question classification and most of the recent works are based on these approaches.

Many researchers have investigated the technique of combining the predictions of multiple classifiers to produce a single classifier (Breiman, 1996c; Clemen, 1989; Perrone, 1993;Wolpert, 1992).The resulting classifier is generally more accurate than any of the individual classifiers making up the ensemble. Both theoretical (Hansen and Salamon, 1990; Krogh and Vedelsby , 1995) and empirical (Hashem, 1997;Opitz and Shavlik, 1996a, 1996b) researches have been carried out successfully. Last decade a group of researchers focused on classifier combination methods in question classification task. Xin et al. (2005) trained four SVM classifiers based on four different types of features and combined them with various strategies. They compared Adaboost, (Schapire, 1999), Neural Networks and Transition-Based Learning (TBL) (Brill, 1995) combination methods on the trained classifiers. Their result on TREC dataset reveals that use of TBL combination method can improve classification accuracy up to 1.6% compared to a single classifier which is trained on all features. Later Xin et al. (2006) performed similar type of experiment and achieved improved accuracy on TREC dataset. Jia et al. (2007) proposed ensemble learning for Chinese question classification. They translated and modified UIUC and TREC dataset to Chinese language. The proposed method achieved 87.6% precision for fine grained question types. (Su et al., 2009) also employed ensemble method in Chinese question classification. The aforementioned experiments with Bagging and AdaBoost.M1 algorithms show that such approaches could effectively utilize multiple classifiers to improve the accuracy rate of question classification than single classifier.

Though no Question Answering System is available in Bengali language, but recently (Banerjee and Bandyopadhyay, 2012) have worked on Bengali question classification task. In their work suitable lexical, syntactic and semantic fea-

tures and Bengali interrogatives have been studied, single-layer taxonomy of nine coarse-grained classes has been proposed and 87.63% question classification accuracy has been achieved. But the proposed method use four classifiers (Naïve Bayes, kernel Naïve Bayes, Rule Induction and Decision Tree) independently. So far, classifier combination methods have not been used by any researcher in Bengali question classification task. Furthermore, no research works has been proposed for fine grained question classes in Bengali. (Li and Roth, 2004) and (Lee et al., 2005) have proposed 50 and 62 fine grained classes for English and Chinese QC respectively. We have proposed 69 fine grained question classes to develop two-layer taxonomy for Bengali QC.

3 Proposed Question Type Taxonomies

The set of question categories are referred as question taxonomies or question ontology. As Bengali question classification is at early stage of development, so for simplicity (Banerjee and Bandyopadhyay, 2012) have used single-layer taxonomy for Bengali question type which consists of only eight coarse-grained classes and no fine-grained classes. No other researches have been carried out for Bengali taxonomies so far.

In the present work, we have included the fine-grained classes to the Bengali question taxonomy keeping intact the coarse classes proposed by (Banerjee and Bandyopadhyay, 2012). Table-1 lists the proposed Bengali Question taxonomy.

The proposed fine-grained question classes are based on the coarse-grained classes proposed by (Banerjee and Bandyopadhyay, 2012). The fine-grained classes are proposed after investigating the corpus used by Banerjee and Bandyopadhyay (2012).

4 Features

In the task of QC, there is always an important problem to decide the optimal set of features to train the classifiers. Different studies have extracted various features with different approaches. The features in QC task can be categorized into three different types: lexical, syntactical and semantic features (Loni, 2011). We have also used three types of features for question classification.

Loni *et al.* (2011) also represented a question in the QC task similar to document representation in vector space model, i.e., a question is a vector which is described by the words inside it.

Therefore a question Q can be represented as:

$$Q = (W_1, W_2, W_3, ..., W_{N-1}, W_N)$$

Where, W_K= frequency of term K in question Q, and N= total number of Terms

Coarse-grained(9)	**Fine-grained(69)**
PER	GROUP, INDIVIDUAL, APPELLATION, INVENTOR/ DISCOVERER, POSITION, OTHER
ORG	BANK, COMPANY, SPORT-TEAM, UNIVERSITY, OTHER
LOC	CITY, CONTINENT, COUNTRY, ISLAND, LAKE, MOUNTAIN, OCEAN, ADDRESS, RIVER, OTHER
TEM	DATE, TIME, YEAR, MONTH, WEEK, DAY, OTHER
NUM	AGE, AREA, COUNT, LENGTH, FREQUENCY, MONEY, PERCENT, PHONE-NUMBER, SPEED, WEIGHT, TEMPERATURE, OTHER
METH	NATURAL, ARTIFICIAL
REA	INSTRUMENTAL, NON-INSTRUMENTAL
DEF	ANIMAL, BODY, CREATION, CURRENCY, FOOD, INSTRUMENT, OTHER, PLANT, PRODUCT, SPORT, SYMBOL, TECHNIQUE, TERM, WORD
MISC	COLOR, CURRENCY, ENTERTAINMENT, LANGUAGE, OTHER, VEHICLE, AFFAIR, DISEASE, PRESS, RELIGION

Table 1: Two-layer Bengali Question Taxonomies.

Due to sparseness of the feature vector only non-zero valued features are kept in the feature vector. Therefore the size of samples is quite small despite the huge size of feature space. All lexical, syntactical and semantic features can be added to feature space to expand the above feature vector.

The next subsections describe the features used for Bengali question classification. We use the same features previously used by (Banerjee and Bandyopadhyay, 2012). In addition, we have considered one more feature, namely coarse-type as syntactical feature.

4.1 Lexical Features (f_L)

Lexical features of a question are generally extracted based on the context words of the question, i.e., the words which appear in a question. We have used five lexical features as below:

wh-word and ***wh-word position:*** Questions wh-word or interrogative is one of the important lexical features. Huang (Huang et al., 2008; Huang et al., 2009) has shown that considering question wh-words as a feature can improve the performance of classification for English. Because of the free-word-order nature of the Bengali language, the position of the wh-word has also been considered as another lexical feature. We considered the value of this feature according to the position first, middle, last in the given question.

wh-type: Unlike English language there are many interrogatives present in the Bengali language. We have considered Bengali interrogative type (wh-type) described by (Banerjee and Bandyopadhyay, 2012) as another lexical feature. They stated that Bengali interrogatives not only describe important information about expected answer but also indicate the Number representations (i.e., singular and plural) and classified wh-type in three categories-Simple Interrogative(SI) or Unit Interrogative(UI), Dual Interrogative(UI) and Compound/Composite Interrogative(CI).

question length: Blunsom *et al.* (2006) introduced the length of a question as an important lexical feature which is simply the number of words in a question. We have also considered this feature for Bengali question classification.

end marker: End marker plays an important role in Bengali question classification that is either '?' or '|' in Bengali. If the end marker is '|', then it has been observed from the experimental corpus that the given question is a definition question.

word shape: Word shapes refer to apparent properties of single words. Huang *et al.* (2008) introduced five categories for word shapes: all digits, lower case, upper case, mixed and other. Word shapes alone is not a good feature set for QC, but when they are combined with other kinds of features they usually improving the accuracy of QC (Huang et al., 2008; Loni et al., 2011). Capitalization feature is not present in Bengali; so we have considered the other three categories, i.e., all digit, mixed and other.

Example: *ke gOdZa prawiRTA karena* ?

Lexical features: wh-word: *ke* ; wh-word position: first ; wh-type: SSI; question length: 5; end-marker: ?

4.2 Syntactical Features (f_S)

Different works extracted several syntactical features with different approaches. The most common syntactical features are Part of Speech (POS) tags and head words (Loni et al., 2011).

***POS tags*:** This indicates the part-of-speech tag of each word in a question such as NN (Noun), NP (Noun Phrase), VP (Verb Phrase), JJ (adjective) etc. We have added all POS tags of question words in the feature vector. Similar approach has been successfully used for English (Li and Roth, 2004; Blunsom et al., 2006). This feature space is sometimes referred as the bag-of-pos tags. (Loni et al., 2011) introduced a feature namely tagged unigram which is simply the unigrams augmented with pos tags. Considering the tagged unigrams instead of normal unigrams can help the classifier to distinguish a word with different tags as two different features (Loni et al., 2011).

***Head words*:** A head word is usually defined as the most informative word in a question or as a word that specifies the object the question is looking for (Huang et al., 2008). Correctly identified headwords can significantly improve the classification accuracy since it is the most informative word in the question. For example, for the question What is the oldest city in Canada? the headword is city. The word city in this question can highly contribute to the classifier to classify this question as LOC:city.

Extracting questions headword is quite a challenging problem and no research has been conducted so far for Bengali. But, we have considered three cases based on the position of question-word or interrogative in the question-

Case I: if *question-word* (i.e., marked by WQ tag) appears at beginning, then the first NP chunk after the *question-word* will be considered as *head-word*. For example-

ke(/WQ) *gOdZa*(/NNP) *prawiRTA*(/NN) *karena*(/VM) ?(/SYM)

So, in the above example *gOdZa* is the head-word.

Case II: if the position of the *question-word* is in the middle of the question, then the immediate NP-chunk before the *question-word* will be considered as head-word. For example-
gOdZa(/NNP) *koWAyZa*(/WQ) *abashiwa*(/JJ) ?(/SYM)
So, in the above example *gOdZa* is the head-word.

Case III: if *question-word* appears at last, i.e., just before end marker, then the immediate NP-chunk before the *question-word* will be considered as head-word. For example-
[bAMlAxeSe arWanIwi kaleja](/NNP) *kayZati* (/WQ) ?(/SYM)
So, in the above example *[bAMlAxeSe arWanIwi kaleja]* is the head-word.

Now, if we consider the following example-
ke gOdZa prawiRTA karena ?
Then, the syntactic features will be: [{WQ, 1}, {NNP, 1}, {NN, 1}, {VM, 1}]

4.3 Semantic Features (f_M)

Semantic features can be extracted based on the semantic meaning of the words in a question. We have used *related word* and *named entities* as semantic features.

Related word : In the absence of Bengali WordNet, a Bengali to Bengali dictionary[1] has been used to retrieve the related words. We have manually prepared three related word categories by analyzing the training data. *date:{ janmaxina, xina, xaSaka, GantA, sapwAha, mAsa, baCara...etc}*; *food:{ KAbAra, mACa, KAxya, mAKana, Pala, Alu, miRti, sbAxa...etc}*; *human_authority:{ narapawi, rAjA, praXAnamanwrI, bicArapawi, mahAparicAlaka, ceyZAramyAna, jenArela, sulawAna, samrAta, mahAXyakRa...etc}*; If a question word belongs to any of the above categories, then its category name will be added in the future vector.
ke gedZera sbAXIna narapawi [human_authority] Cilena ?

For the above example the semantic feature can be added to the feature vector as: [{*human-authority*, 1}]

***Named entities*:** Some studies (Li and Roth, 2004; Blunsom et al., 2006) have successfully used named entities as a semantic feature. To identify the Bengali named entities in the question text a Hidden Markov Model Based Named Entity Recognizer (NER) System (Ekbal et al., 2007) has been used as the Bengali NER system.
ke gOdZa[Location] *prawiRTA karena* ?

For the above example the semantic feature can be added to the feature vector as: [{Location, 1}]

[1] http://dsal.uchicago.edu/dictionaries/biswas-bangala/

5 Ensemble Learning for Question Classification

Two popular methods for creating accurate ensembles are *bagging* (Breiman, 1996c) and *boosting* (Freund and Schapire, 1996; Schapire, 1990). These methods rely on resampling" techniques to obtain different training sets for each of the classifiers.

5.1 Bagging

Bagging or bootstrap aggregation is a machine learning ensemble meta-algorithm technique proposed by Breiman (1996a, 1996b). It can be used to improve the classification in terms of stability and classification accuracy. It also reduces variance and helps to avoid "*overfitting*". Although it is usually applied to decision tree models, it can be used with any type of model. Bagging trains a number of base learners by bootstrap sampling to get an aggregated prediction. Each classifier's training set T_K is generated by randomly selecting with replacement from N examples, where N is the size of the original training set. In the resulting training set T_K, some of the original examples

Input: Training set T of N example, Learning Model M (e.g. Decision Tree, Nave Bayes etc.), Bagging size S

Output: ensemble model and predicted class

Training:

For each iteration s, $s = 1...S$
Randomly sample N examples with replacement from the training set T and generate training set T_K.
Apply base model M on training set T_K to generate model M_S

Classification:

For each of s model M_S
Predict class label via majority voting

Return the class label that has been predicted most often

Table 2: Bagging Method.

may be repeated. Thus each individual classifier B_L in the ensemble is generated with a different random sampling of the original training set T.

5.2 Boosting

Boosting is another well known machine learning ensemble meta-algorithm technique. This ensemble method produces a series of classifiers and the training set used by a classifier model is based on the performance of the earlier classifier model. In boosting, examples that are incorrectly predicted by previous classifiers in the series are chosen more often than the examples that were correctly predicted. Boosting attempts to produce new better classifiers by selecting incorrectly predicted samples than the correctly predicted samples from the training set used by the previous classifier. Freund and Schapire(1995) developed a famous boosting algorithm- AdaBoost. There are two versions of AdaBoost: AdaBoost.M1 and AdaBoost.M2. In the present work, AdaBoost.M1 has been used as the boosting method. The selected boosting method gives each training example equal weight. So, if the training size consists of N examples then each example get $1/N$ weight. A sampling distribution Dt has been defined, where $D_t(i)$ represent a probability that example i from the original training dataset is selected. A sample distribution D_t for building the j^{th} model is constructed by modifying the sampling distribution D_{t-1} from the $(j-1)^{th}$ step. Examples classified incorrectly in the previous step receive higher weights in the new data to cover misclassified samples.

Input: Training set T of N example, Learning Model M (e.g. Decision Tree, Nave Bayes etc.), Number of steps S

Output: ensemble model and predicted class

Training:

$D_t(1) = 1/N$

For each iteration t, $t = 1...S$

Take K samples from the training set according to D_t and Train a classifier h_t on the samples calculate the error ε_t of h_t :

$$\varepsilon_t = \sum_{i:h_t(x_i)\neq y_i} D_t$$

Calculate weight β_t of h_t: β_t=$\varepsilon_t/(1-\varepsilon_t)$

Calculate new sampling distribution

$$D_{t+1}(i)=\frac{D_t(i)}{Z_t} \times \begin{cases} \beta_i & h_t(x_i) = y_i \\ 1 & \textit{Otherwise} \end{cases}$$

Weight w_t of classifier h_t: w_t= $\log(1/\beta_t)$

Classification:

Classify according to the weighted majority of classifiers

Return the class that correspond to the maximal sum of weights

Table 3: Boosting Method.

6 Experiments

This section describes our empirical study of Ensemble: bagging and boosting approaches. Each of these two approaches has been tested with Naïve Bayes (NB), Kernel Naïve Bayes (k-NB), Rule Induction (RI) and Decision Tree (DT).The previous work on Bengali question classification task used these four classifiers. So in the present work, we have used those classifiers to establish the effect of combining models.

6.1 Dataset

The present research work adopts the same corpus used by (Banerjee and Bandyopadhyay, 2012). The corpus consists of 1100 Bengali questions of different domains, e.g., education, geography, history, science etc. Two highly qualified human annotator annotate the questions with an agreement at kappa statistics of 93.48%. We have used 770 questions (70%) for training and rest 330 questions (30%) to test the classification models.

6.2 Results

Four different experiments have been performed for each bagging and boosting. So, altogether eight different experiments have been performed for the ensemble approach. A classifier model has been tested on $f_L + f_S + f_M$ features. The outcome of the experiments have been tabulated and described in the next sub-sections.

In our study, *classification accuracy* has been used to evaluate the results of the experiments. *accuracy* is the widely used evaluation metric to determine the class discrimination ability of classifiers, and is calculated using the following equation:

$$accuracy(\%) = \frac{T_P+T_N}{P+N}$$

where, T_P = true positive samples; T_N = true negative samples; P = positive samples; N = negative

samples.

It is a primary metric in evaluating classifier performances and it is defined as the percentage of test samples that are correctly classified by the algorithm.

6.2.1 Results based on Bagging

Bagging approach has been applied separately to four classifiers (i.e., NB, k-NB, RI and DT) and Table-4 tabulates the detailed information of the accuracy obtained. F_X represents fine-grained classes of coarse-grained class X. Initially the $size$ (number of iteration) of the base learner is set to 2. Then experiments have been performed with gradually increased size ($size > 2$).The classification accuracy has been increased with increase in $size$. But after a certain $size$ value, the accuracy has been almost stable.

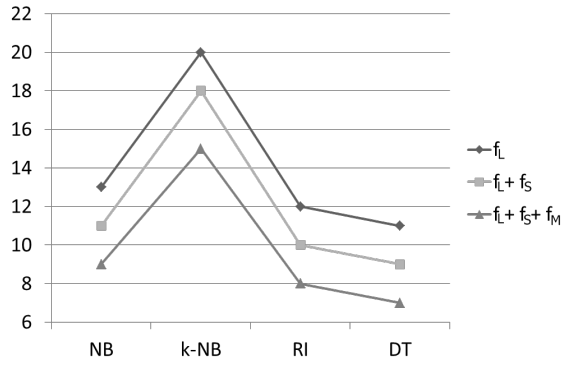

Figure 1: Size variation in Bagging

For the fine-grained classes of PER coarse class, i.e., F_{PER}, at $size$=2 and feature = f_L+ f_S+f_M, the NB classifier achieves 81.98% accuracy and at $size$>=9, it becomes stable with 82.87% accuracy. Similarly, at $size$=2 and feature=f_L+f_S+f_M, the k-NB classifier achieves 82.36% accuracy and at $size$>=15, it becomes stable with 82.97% accuracy. At $size$=2 and feature= f_L+f_S+f_M , the RI classifier achieves 83.89% accuracy and at $size$>=8, it becomes stable with 84.12% accuracy. At $size$=2 and feature= f_L+f_S+f_M, the DT classifier achieves 87.76% accuracy and at $size$>=7, it becomes stable with 88.21% accuracy. It has been observed from the experiments that at each case *bagging* with DT requires less size, i.e., less iteration then the other used classifiers for the fine grained-classes.

For experiment with f_L features, the *bagging size* of NB, k-NB, RI and DT are 13, 20, 12 and 11 respectively after which classification accuracy becomes stable for the fine-grained classes.

And For experiment with f_L+f_S features, the *bagging size* of NB, k-NB, RI and DT are 11, 18, 10 and 9 respectively after which classification accuracy becomes stable for the fine-grained classes.

		f_L	f_L+f_S	f_L+f_S+f_M
Naïve Bayes	F_{PER}	79.65%	81.23%	82.87%
	F_{ORG}	81.01%	82.32%	83.55%
	F_{LOC}	81.89%	82.82%	83.73%
	F_{TEM}	81.45%	82.97%	83.84%
	F_{NUM}	80.23%	81.13%	82.31%
	F_{METH}	82.10%	83.25%	84.41%
	F_{REA}	81.93%	83.02%	84.17%
	F_{DEF}	82.05%	83.29%	84.47%
	F_{MISC}	81.51%	82.75%	83.23%
Kernel Naïve Bayes	F_{PER}	80.13%	81.83%	82.97%
	F_{ORG}	81.23%	82.51%	83.89%
	F_{LOC}	82.03%	83.12%	84.02%
	F_{TEM}	81.71%	83.31%	84.20%
	F_{NUM}	80.52%	81.42%	82.59%
	F_{METH}	82.45%	83.75%	84.91%
	F_{REA}	82.35%	83.98%	85.01%
	F_{DEF}	82.53%	84.02%	85.11%
	F_{MISC}	81.87%	83.32%	84.23%
Rule Induction	F_{PER}	81.85%	82.98%	84.12%
	F_{ORG}	82.19%	83.53%	84.78%
	F_{LOC}	81.54%	82.27%	83.13%
	F_{TEM}	83.12%	84.79%	85.81%
	F_{NUM}	81.93%	82.97%	84.43%
	F_{METH}	83.85%	85.04%	86.22%
	F_{REA}	83.59%	84.97%	86.15%
	F_{DEF}	82.92%	84.28%	85.33%
	F_{MISC}	82.51%	83.89%	84.93%
Decision Tree	F_{PER}	84.79%	86.57%	88.21%
	F_{ORG}	83.11%	84.67%	86.17%
	F_{LOC}	82.83%	84.01%	85.39%
	F_{TEM}	85.01%	86.54%	88.09%
	F_{NUM}	83.34%	84.92%	86.35%
	F_{METH}	85.05%	87.09%	89.11%
	F_{REA}	84.93%	87.02%	89.06%
	F_{DEF}	84.28%	85.53%	87.02%
	F_{MISC}	84.12%	86.21%	88.69%

Table 4: Experimental results of Bagging.

It has been also noted that the fine-grained classes of DEF coarse class, i.e., F_{DEF} have achieved highest accuracy of 84.47% and 85.11%

for NB and k-NB classifiers respectively. And, the fine-grained classes of METH coarse class, i.e., F_{METH} have achieved highest accuracy of 86.22% and 89.11% for RI and DT classifiers respectively.

Table-4 shows that k-NB classifiers increases slight accuracy performance over NB classifier, but accuracy has been drastically improved by RI and DT classifiers. Overall, DT classifier has outperformed other classifiers to classify all fine-grained classes of the coarse-grained classes.

6.2.2 Results based on AdaBoost.M1

Like *bagging*, AdaBoost.M1 has also been applied separately to the four classifiers (i.e., NB, k-NB, RI and DT).

Table-5 tabulates the detailed information of the accuracy obtained for fine-grained question classes. The experiment results show that the performances of the four classifiers have been improved slightly using AdaBoost.M1. And, overall DT outperforms other classifiers in *ensemble* approach i.e., *bagging* and *boosting*.

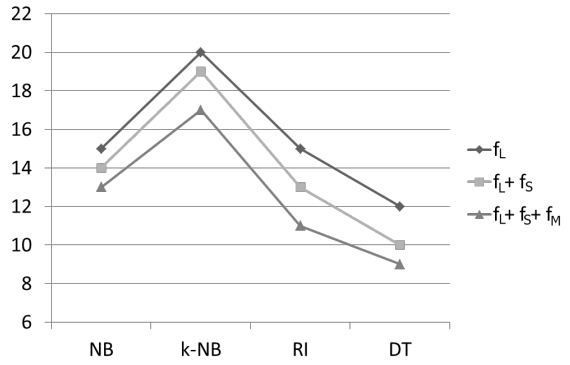

Figure 2: Size variation in Boosting

Here, we empirically fix the iterations of AdaBoost.M1 for four classifiers (i.e., NB, k-NB, RI and DT) to 13, 17, 11 and 9 respectively for features= f_L+f_S+f_M because the weight of $(1/\beta_t)$ is less than 1 after those values. If $(1/\beta_t)$ is less than 1, then the weight of classifier model in boosting may be less than zero for that iteration. Similarly, for features=f_L+f_S and features=f_L the iterations are 14, 19, 13, 10 and 15, 20, 15, 12 respectively for four classifiers correspondingly. Figure-2 depicts the iterations size of four classifiers (i.e., NB, k-NB, RI, DT) in *boosting* approach.

		f_L	f_L+f_S	f_L+f_S+f_M
Naïve Bayes	F_{PER}	79.89%	81.41%	82.95%
	F_{ORG}	81.65%	82.73%	83.98%
	F_{LOC}	82.28%	83.85%	85.04%
	F_{TEM}	81.89%	83.01%	83.97%
	F_{NUM}	81.02%	81.92%	83.03%
	F_{METH}	82.25%	83.37%	84.53%
	F_{REA}	82.06%	83.11%	84.23%
	F_{DEF}	82.09%	83.32%	84.56%
	F_{MISC}	81.62%	82.79%	83.75%
Kernel Naïve Bayes	F_{PER}	80.17%	81.91%	83.02%
	F_{ORG}	81.29%	82.63%	83.91%
	F_{LOC}	82.10%	83.17%	84.09%
	F_{TEM}	81.79%	83.39%	84.28%
	F_{NUM}	80.63%	81.58%	82.69%
	F_{METH}	82.48%	83.79%	84.98%
	F_{REA}	82.41%	84.02%	85.09%
	F_{DEF}	82.61%	84.12%	85.13%
	F_{MISC}	81.91%	83.39%	84.28%
Rule Induction	F_{PER}	81.92%	83.06%	84.22%
	F_{ORG}	82.25%	83.61%	84.85%
	F_{LOC}	81.55%	82.26%	83.15%
	F_{TEM}	83.18%	84.85%	85.93%
	F_{NUM}	82.01%	83.03%	84.49%
	F_{METH}	83.91%	85.06%	86.31%
	F_{REA}	83.68%	85.11%	86.33%
	F_{DEF}	82.95%	84.32%	85.41%
	F_{MISC}	82.57%	83.93%	84.98%
Decision Tree	F_{PER}	84.81%	86.63%	88.53%
	F_{ORG}	83.14%	84.73%	86.23%
	F_{LOC}	82.87%	84.13%	85.52%
	F_{TEM}	85.03%	86.58%	88.15%
	F_{NUM}	83.38%	84.97%	86.44%
	F_{METH}	85.09%	87.14%	89.12%
	F_{REA}	84.96%	87.11%	89.09%
	F_{DEF}	84.29%	85.55%	87.05%
	F_{MISC}	84.15%	86.23%	88.73%

Table 5: Experimental results of AdaBoost.M1.

7 Conclusions and Perspectives

The automated Bengali question classification system by (Banerjee and Bandyopadhyay, 2012) is based on four classifiers namely Naïve Bayes, Kernel Naïve Bayes, Rule Induction and Decision Tree. In that work, single-layer taxonomy has been proposed and verified. But, no fine-grained

classes have been proposed or experimented so far. The main contributions of this paper are as follows-

i) This work extends Bengali question classification taxonomy by adding fine-grained classes for coarse-grained classes.

ii) This work introduces a new method for Bengali question classification. Ensemble approach has not been used in Bengali QC so far. This work successfully deploys the *ensemble* approach for classifying fine-grained question class in Bengali question classification.

iii) Sixty nine Fine-grained question classes have been proposed and experimented.

It has been observed from the experiment results that overall DT classifier with *boosting* approach has performed best. Experimental results show that *ensemble* approach performs well and achieves satisfactory performance in terms of accuracy.

The main future direction of our research is to exploit other lexical, semantic and syntactic features for Bengali question classification. In future an investigation can be performed on including new classifiers, e.g., k-nearest neighbor etc. It is also worth investigating other classifier combination approaches i.e., voting, stacking for fine-grained question classes in Bengali. In the current work, we have only investigated the Bengali questions. But, this work can be applied to other languages having low resources.

Acknowledgments

We acknowledge the support of the Department of Electronics and Information Technology (DeitY), Ministry of Communications and Information Technology (MCIT), Government of India funded project "*Development of Cross Lingual Information Access (CLIA) System Phase II*".

References

Abraham Ittycheriah , Franz Martin, Zhu Wei-Jing, Adwait Ratnaparkhi, and Richard J. Mammone. 2001. *IBMs statistical question answering system*. In Proceedings of the 9th Text Retrieval Conference, NIST.

Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. 2007. *Exploiting syntactic and shallow semantic kernels for question answer classification*. ACL, vol. 45, no. 1, pp. 776.

Anders Krogh and Jesper Vedelsby. 1995. *Neural network ensembles, cross validation, and active learning*. Advances in neural information processing systems, Vol. 7, pp. 231-238 Cambridge, MA. MIT Press.

Asif Ekbal and Sivaji Bandyopadhyay. 2007. *A Hidden Markov Model Based Named Entity Recognition System: Bengali and Hindi as Case Studies*. PReMI 2007: 545-552.

Babak Loni. 2011. *A survey of state-of-the-art methods on question classification*. Delft University of Technology, Tech. Rep (2011): 1-40.

Babak Loni, Gijs van Tulder, Pascal Wiggers, Marco Loog, and David Tax. 2011. *Question classification with weighted combination of lexical, syntactical and semantic features*. TSD, pages 243-250.

Dan Moldovan, Marius Pasca, SandaHarabagiu, and MihaiSurdeanu. 2003. *Performance issues and error analysis in an open-domain question answering system*. ACM Trans. Inf. Syst., 21:133-154.

David A. Hull. 1999. *Xerox TREC-8 question answering track report*. In Voorhees and Harman.

David H. Wolpert. 1992. *Stacked generalization*. Neural Networks, 5, 241-259.

David W. Opitz and Jude W. Shavlik. 1996a. *Actively searching for an effective neural network ensemble*. Connection Science, 8(3/4): 337-354.

David W. Opitz and Jude W. Shavlik. 1996b. *Generating accurate and diverse members of a neural network ensemble*. Advances in Neural Information Processing Systems, Vol. 8, pp. 535-541 Cambridge, MA. MIT Press.

Dell Zhang and Wee Sun Lee. 2003. *Question classification using support vector machines*. ACM SIGIR, pages 26-32,New York,USA, ACM.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin yew Lin, and Deepak Ravichandran. 2001. *Toward semantics-based answer pinpointing*.

Ellen M. Voorhees. 2001. *Overview of the TREC 2001 question answering track*. TREC, pp. 42-51.

Eric Brill. 1995. *Transformation-based error driven learning and natural language processing: A case study in part of speech tagging*. Computational linguistics. 21(4), page: 543-565.

Joao Silva, Luisa Coheur, Ana Mendes, and Andreas Wichert. 2011. *From symbolic to sub-symbolic information in question classification*. Artificial Intelligence Review, 35(2):137-154.

John Prager, Dragomir Radev, Eric Brown, and Anni Coden. 1999. *The use of predictive annotation for question answering in trec8*. TREC-8,pp.399-411.NIST.

Keliang Jia, Kang Chen, Xiaozhong Fan, Yu Zhang. 2007. *Chinese Question Classification Based on Ensemble Learning*. ACIS. pp. 342-347.

Lars Kai Hansen, and Peter Salamon. 1990. *Neural network ensembles*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12, 993-1001.

Lei Su, Hongzhi Liao, Zhengtao Yu, Quan Zhao. 2009. *Ensemble Learning for Question Classification*. ICIS 2009. pp. 501-505.

Leo Breiman. 1996a. *Bagging predictors*. Machine Learning, 24(2), 123140.

Leo Breiman. 1996b. *Bias, variance, and arcing classifiers*. Tech. rep. 460, UC-Berkeley,Berkeley, CA.

Leo Breiman. 1996c. *Stacked regressions*. Machine Learning, 24(1), 49-64.

LI Xin, Xuan-Jing HUANG, and Li-de WU. 2006. *Question Classification by Ensemble Learning*. IJCSNS, 6(3), page : 147.

Michael Peter Perrone. 1993. *Improving Regression Estimation: Averaging Methods for Variance Reduction with Extension to General Convex Measure Optimization*. Ph.D. thesis, Brown University, Providence, RI.

Phil Blunsom, Krystle Kocik, and James R. Curran. 2006. *Question classification with log-linear models*. ACM SIGIR,pp. 615616.

Robert E. Schapire. 1990. *The strength of weak learnability*. Machine Learning, 5(2), page:197-227.

Robert T. Clemen. 1989. *Combining forecasts: A review and annotated bibliography*. International Journal of Forecasting 5, no. 4: 559-583.

Sherif Hashem. 1997. *Optimal linear combinations of neural networks*. Neural Networks, 10 (4), pp:599-614.

Somnath Banerjee and Sivaji Bandyopadhyay. 2012. *Bengali Question Classification: Towards Developing QA System*. In Proceedings of WSSANLP-COLING, pages 25-40, Mumbai, India.

Somnath Banerjee and Sivaji Bandyopadhyay. 2012a. *Question Classification and Answering from Procedural Text in English*. In Proceedings of QACD-COLING, pages 11-26, Mumbai, India.

Xin Li and Dan Roth. 2004. *Learning question classifiers: The role of semantic information*. COLING, pp. 556-562.

Zhiheng Huang, Marcus Thint, and Asli Celikyilmaz. 2009. *Investigation of question classifier in question answering*. EMNLP , pp. 543550.

Zhiheng Huang, Marcus Thint, and Zengchang Qin. 2008. *Question classification using head words and their hypernyms*. EMNLP, pp. 927-936.