Classifier Combination Approach for Question Classification for Bengali Question Answering System

SOMNATH BANERJEE[1,*], SUDIP KUMAR NASKAR[1], PAOLO ROSSO[2] and SIVAJI BNDYOPADHYAY[1]

[1] Department of Computer Science and Engineering , Jadavpur University, Kolkata, India
[2] PRHLT Research Center, Universitat Politècnica de València, Spain

Abstract. Question classification (QC) is a prime constituent of automated question answering system. The work presented here demonstrates that the combination of multiple models achieve better classification performance than those obtained with existing individual models for the question classification task in Bengali. We have exploited state-of-the-art multiple model combination techniques, i.e., ensemble, stacking and voting, to increase QC accuracy. Lexical, syntactic and semantic features of Bengali questions are used for four well-known classifiers, namely Naïve Bayes, kernel Naïve Bayes, Rule Induction, and Decision Tree, which serve as our base learners. Single-layer question-class taxonomy with 8 coarse-grained classes is extended to two-layer taxonomy by adding 69 fine-grained classes. We carried out the experiments both on single-layer and two-layer taxonomies. Experimental results confirmed that classifier combination approaches outperform single classifier classification approaches by 4.02% for coarse-grained question classes. Overall, the stacking approach produces the best results for fine-grained classification and achieves 87.79% of accuracy. The approach presented here could be used in other Indo-Aryan or Indic languages to develop a question answering system.

MS received 17 November 2017; revised 25 August 2019; accepted 9 September 2019

Keywords. Bengali question classification; question classification; classifier combinations

1 Introduction

A Question Answering (QA) system is an automatic system capable of answering natural language questions in a human-like manner: with a concise, precise answer. With the explosion of information on the internet, research in QA is becoming increasingly important. QA is a research area that combines research from different, but related, fields like Information Retrieval (IR), Information Extraction (IE), Natural Language Processing (NLP), etc. QA is different from IR in the fact that the objective of QA is retrieving answers to questions while the goal of IR systems (i.e., search engines) is to just retrieve relevant documents. This implies that QA systems will possibly make the next generation search engines. According to [1], typically an automated QA system has three stages: question processing, passage retrieval and answer processing. In the question processing stage, the natural language question is analyzed to create a proper IR query and also the entity type of the answer is detected. The first task is called query reformation and the second is called QC. In this work, we focus on QC which is an important component of factoid question answering systems. Factoid questions (e.g. *Who founded Virgin Airlines?*) are questions that can be answered with simple facts expressed in short text answers [2]. QC plays an important role in the QA framework though different QA systems follow different architectures [3]. Furthermore, earlier studies [4–6] reported that QC has significant influence on the overall performance of the QA systems. The task of a QC module is to assign one or more class labels, depending on the classification strategy, to a given question written in a natural language. For example, for the question "*Which London street is the home of British journalism?*" the task of a QC component is to assign the label 'Location' to this question. Since it effectively predicts the type of the answer, QC is also often referred to as answer type prediction.

The two foremost motivations for QC are: locating the answer and choosing the search strategy. Knowing the question class not only reduces the search space to be explored for finding the answer but it also helps to find the true answer from a given set of candidate answers. On the other hand, the question class can also be used to choose the search strategy when the question is reformed to a query over IR engine. For example, consider the question "*What is a pyrotechnic display ?*". Identifying that the question class is of type 'Definition', the searching template for locating the answer may be for example "*pyrotechnic display is a ...*" or "*pyrotechnic displays are ...*", which are much more effective than simply searching by the question words.

Although the QA systems developed for European languages, particularly in English, have achieved reasonable accuracy, the situation for the Indian languages is completely different. Research on QA has not been initiated for most of the Indian languages. Like other Indian languages, Bengali

*For correspondence

(also known as 'Bangla') presents serious challenges for QA. Bengali is an Indo-Aryan language such as Hindi, Marathi, Gujrati etc. With about 193 million native and about 230 million total speakers, Bengali is one of the most spoken languages (ranked sixth) in the world and the second most commonly spoken language in India as per *2011 Census of India*[1]. Due to the rapid increase of contents in Bengali on the web, the research community have started to take notice and interest in Bengali. Unlike English, Bengali has many interrogatives [7]. Even in Bengali, the position of the interrogatives in the question text is not fixed due to relatively free phrase order of the language. Moreover, the language processing tools for Bengali are in the development phase.

One of the key issues of classification modeling is the enhancement of classification accuracy. In that regard, notable number of researchers have recently employed considerable attention to classifier combination methods. The idea is not to rely on a single decision making scheme. Instead, many single/individual classifiers are used for decision making by combining their individual opinions to arrive at a consensus decision.

The rest of the paper is structured as follows: we start with a discussion of the related work in Section 2. We discuss the Bengali question taxonomies in Section 3. The features for the classification task is described in Section 4. Section 5 discusses the detailed results. Finally, we conclude in Section 6.

2 Related Work

A considerable volume of research have been carried out on QC, question taxonomies and question features [8]. In the past decade, QC was enormously addressed. Broadly two different approaches are used to classify questions – rule-based [9, 10] and machine learning based [11, 12]. However, a number of researchers have also employed a few hybrid approaches which combine rule-based and machine learning based approaches [13, 14].

In rule-based approaches, manually handcrafted grammar rules are used to analyze a question in order to determine the answer type [9, 10]. Although handcrafted rules have been used successfully, however, designing these rules is expensive [15]. Li and Roth [15] stated that although rule-based approaches may perform well on a particular dataset but the classification performance may degrade on a new dataset and consequently it is difficult to scale them. Therefore, it is very much challenging to build a manual classifier with a limited number of rules. In contrast, machine learning based QC approaches are performed by extracting features from the questions, training a classifier and predicting the question class using the trained classifier. Many researchers employed machine learning techniques, e.g., maximum entropy [16], support vector machine [17], etc. by using different features, such as syntactic features [12], semantic features [11], etc. However, these works were primarily focused on the factoid questions of English and restricted to classify the questions into two categories (namely, *yes* and *no*) or a few predefined categories (e.g., 'what', 'how', 'why', 'when', 'where' and so on).

Many researchers have investigated the technique of combining the predictions of multiple classifiers to build a single classifier [18–21]. It has been observed that the resulting classifier is generally more accurate than any of the individual classifiers making up the ensemble. Both theoretical [22, 23] and empirical [24–26] studies were carried out on classifier combination. A number of studies were carried out on classifier combination methods for the QC task in the last decade. Xin *et al* [27] trained four SVM classifiers based on four different types of features and combined them with various strategies. They compared Adaboost [28], Neural Networks and Transition Based Learning (TBL) [29] combination methods on the trained classifiers. Their evaluation results on the TREC dataset revealed that the use of TBL combination method improved classification accuracy up to 1.6% compared to a single classifier trained on all features. Jia *et al* [30] proposed ensemble learning for Chinese question classification. They translated and modified the UIUC (University of Illinois, Urbana Champaign) and TREC (Text REtrieval Conference) dataset to Chinese. The proposed method achieved 87.6% precision for fine-grained question types. The ensemble method has also been employed for Chinese question classification [31]. The aforementioned experiments with Bagging [18] and AdaBoost.M1 [28] algorithms showed that such approaches can effectively utilize multiple classifiers to improve the accuracy rate of question classification than a single classifier.

Presently, QA systems developed for European [32–34], Middle Eastern [35–37] and Asian languages [38–41] are capable of providing answers with reasonable accuracy. However, the scenario is different for Indian languages in which QA research is in a nascent stage. There are 22 official languages in India and for most of these Indian languages research in QA has not been started yet. A factoid Hindi QA system namely 'Prashnottar' was proposed in [42]. 'Prashnottar' uses handcrafted rules to identify question patterns for question classification. Recently, [43] developed a QA system for Hindi which uses Naïve Bayes technique for question classification. They reported that the classifier was tested on 75 questions. A few Hindi–English cross-lingual QA systems also have been reported in the literature. In 2003, [44] developed a cross-lingual QA system for Hindi and English which employed a rule-based approach for Question Classification. A multilingual restricted domain QA system was reported in [45] which also uses a rule-based approach for classifying questions. [46] provides a review of the state-of-the-art in Hindi QA and it criticized that [45] did not report the testset corpus statistics. In [47], a QA system in Hindi was reported. Based on the keywords, they applied rule-based approach for classifying the questions into six categories. For Telugu, [48] proposed a dialogue based QA system for the railway domain. They used rule-based approach to classify

[1] http://www.censusindia.gov.in/2011Census/Language-2011/Statement-1.pdf

questions. [49] reported a QA system developed for Punjabi. The work is based on a concept taken from physics: 'Point of Gravity'. However, they did not report any question classification approach. [50] reports a QA system for Malayalam which is based on named entity tagging and question classification. A Rule-Based Approach was adopted for Question Classification. It can be concluded from the related work that a few QA systems have been developed for Indian languages and mainly rule-based approach has been employed for classifying questions.

Like other Indian languages, research in Bengali QA has achieved very less attention than its western and middle-eastern counterparts. Although any QA system is not available in Bengali till date, however, a study [7] was carried out on the Bengali QC task in which suitable lexical, syntactic and semantic features and Bengali interrogatives were studied, a single-layer taxonomy of nine coarse-grained classes was proposed and 87.63% QC accuracy was reported. The proposed method used four classifiers independently, namely, Naïve Bayes, kernel Naïve Bayes, Rule Induction and Decision Tree. Both theoretical [22, 23] and empirical [24–26] studies confirm that the classifier combination approach is generally more accurate than any of the individual classifiers making up the ensemble. Furthermore, a number of studies [27, 30] were successfully carried out on classifier combination methods for the QC task which outperformed the individual classifiers. Therefore, we consider classifier combination for classifying Bengali questions. To the best of our knowledge, classifier combination methods have not been employed for the QC task for Indian languages, prior to the work reported in this paper. As discussed earlier, mainly rule-based approach was employed for the QC task along with individual classifier based approach. Furthermore, no research work can be found in the literature for fine-grained question classification in Bengali. The deep learning framework performs well when large datasets are available for training and the framework is less effective than traditional machine learning approaches when the training datasets are small in size. In this work, we deal with a dataset which has only 1,100 samples. Therefore, we prefer classifier combination approach over deep learning. Li and Roth [15] and Lee *et al* [51] proposed 50 and 62 fine grained classes for English and Chinese QC respectively. In our work, we proposed 69 fine grained question classes to develop a two-layer taxonomy for Bengali QC.

3 Proposed Question Taxonomies

The set of question categories is referred to as question taxonomy or question ontology. Since Bengali question classification is at an early stage of development, for simplicity initially a single-layer taxonomy for Bengali question types was proposed in [7] which consists of only eight coarse-grained classes and no fine-grained classes. No other investigation have been carried out for coarse-grained Bengali taxonomies till date. Later, based on the coarse-grained classes in [7], fine-grained question classes were proposed in [52]. Table 1 presents the Bengali Question taxonomy proposed in [7, 52].

Table 1. Two-layer Bengali Question Taxonomies

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

The taxonomy proposed by Li and Roth [15] contains 6 coarse-grained classes: *Abbreviation, Description, Entity, Human, Location, Numeric*. *Abbreviation* and *Description* classes of [15] are not present in Bengali taxonomy. Two coarse-grained classes of [15], namely, *Entity* and *Human* have resemblance with *Miscellaneous* and *Person* respectively in Bengali taxonomy. While *Location* and *Number* classes are present in both the taxonomies, *Organization* and *Method* classes are not present in [15]. In 2-layer Bengali taxonomy, 15 fine-grained classes of [15] are not present, namely, *abbreviation, expression, definition, description, manner, reason, event, letter, substance, title, description, state, code, distance, order*.

All the coarse-grained classes of Lee *et al* [51] are present in Bengali taxonomy. However, the *Method* class of Bengali taxonomy is not present in [51]. The *Artifact* class of [51] is similar to *Definition* and *Miscellaneous* of Bengali-taxonomy. In 2-layer Bengali taxonomy, 9 fine-grained classes of [51] are not included, namely, *firstperson, planet, province, political system, substance, range, number, range, order*.

The 5 fine-grained classes are introduced in Bengali taxonomy which are not present in [15] and [51]. The 5 classes are: *AGE, NATURAL, ARTIFICIAL, INSTRUMENTAL, NON-INSTRUMENTAL*. The *NATURAL* and *ARTIFICIAL* fine - grained classes belong to *Method* coarse-grained class which is not present in [15] and [51]. Similarly, *INSTRUMENTAL, NON-INSTRUMENTAL* fine-grained classes belong to the *Reason* coarse-grained class. Also, the *Reason* coarse-grained class is not present in [15] and [51]. The *AGE* fine-grained class belong to *Numerical* coarse class.

The taxonomies proposed in [15] and [51] did not deal with causal and procedural questions. The proposed Ben-

Table 2. Bengali question examples

Class	Example
Person (PER)	ke gOdZa prawiRTA karena ? (gloss: Who established Goura?)
Organization (ORG)	sinXu saByawAra safgeV koVna saByawAra mila KuzjeV pAoVyZA yAyZa ? (gloss: Which civilization has resemblance with the Indus Valley Civilization ?)
Location (LOC)	gOdZa koWAyZa abashiwa ? (gloss: Where is Goura situated ?)
Temporal (TEMP)	BAikiM-2 kawa baCara karmakRama Cila ? (gloss:For how many years Vaikin-2 was working ?)
Numerical (NUM)	sUrya WeVkeV Sani graheVra gadZa xUrawba kawa ? (gloss: What is the average distance of the planet Saturn from the Sun ?)
Method (METH)	AryasaByawA mahilArA kiBAbeV cula bAzXawa ? (gloss: How do the women braid hair in the Arya Civilization)
Reason (REA)	AryasaByawAkeV keVna bExika saByawA balA hayZa ? (gloss: Why the Arya Civilization is called the Vedic Civilization?)
Definition (DEF)	beVxa ki ? (gloss: What is Veda?)
Miscellaneous (MISC)	Arya samAjeV cArati barNa ki ki Cila ? (gloss: What are the four classes in the Arya Society?)

gali 2-layer taxonomy is based on the only available Bengali QA dataset [7] which contains causal and procedural questions. Therefore, the Bengali taxonomy contains question classes of causal and procedural questions. A few fine-grained classes of [15] and [51] are not included in the taxonomy because such questions are not present in the Bengali QA dataset. However, the proposed Bengali taxonomy is not final for Bengali QA task. Increasing the size of the said dataset is still in the process. Therefore, it is expected that the missing fine-grained classes will be incorporated in the taxonomy in future.

4 Features for Question Classification

In the task of machine learning based QC, deciding the optimal set of features to train the classifiers is crucial. The features used for the QC task can be broadly categorized into three different types: lexical, syntactic and semantic features [53]. In the present work, we also employed these three types of features suitable for the Bengali QC task.

Loni *et al* [53] represented questions for the QC task similar to document representation in the vector space model, i.e., a question is represented as a vector described by the words inside it. Therefore, a question Q_i can be represented as below:

$$Q_i = (W_{i1}, W_{i2}, W_{i3}, \ldots, W_{i(N-1)}, W_{iN})$$

where, W_{ik} = frequency of the term k in question Q_i, and N = total number of terms.

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

In the present study, the features employed for classifying questions (cf. Table 1) are described in the following subsections. In addition to the features used for the coarse-grained classification, fine-grained classification uses an additional feature, namely coarse-class, i.e. label of the coarse-grained class.

4.1 *Lexical Features*

Lexical features (f_L) of a question are extracted from the words appearing in the question. Lexical features include interrogative-words, interrogative-word-positions, interrogative-type, question-length, end-marker and word-shape.

• *Interrogative-words and interrogative-word positions:* The interrogative-words (e.g., what, who, which etc.) of a question are important lexical features. They are often referred to as *wh*-words. Huang *et al* [13, 54] showed that considering question interrogative-word(s) as a feature can improve the performance of question classification task for English QA. Because of the relatively free word-ordering in Bengali, interrogative-words might not always appear at the beginning of the sentence, as in English. Therefore, the position of the interrogative (wh) words along with the interrogative words themselves have been considered as the lexical features. The position value is based on the appearance of the interrogative word in the question text and it can have any of the three values namely, first, middle and last.

• *Interrogative-type:* Unlike in English, there are many interrogatives present in the Bengali language. Twenty six Bengali interrogatives were reported in [7]. In the present work, the Bengali interrogative-type (wh-type) is considered as another lexical feature. In [7], the authors concluded that Bengali interrogatives not only provide important information about the expected answers but also indicate the number information (i.e., singular vs plural). In [7], wh-type was classified to three categories: Simple Interrogative (SI) or Unit Interrogative (UI), Dual Interrogative (UI) and Compound/Composite Interrogative (CI).

• *Question length:* Blunsom *et al* [55] introduced the length of a question as an important lexical feature which is simply the number of words in a question. We also considered this feature for the present study.

• *End marker:* The end marker plays an important role in Bengali QC task. Bengali question is end with either '?' or '|'. It has been observed from the experimental corpus that if the end marker is '|' (similar to dot (.) in English), then the given question is a definition question.

• *Word shape:* The word shape of each question word is considered as a feature. Word shapes refer to apparent properties of single words. Huang *et al* [13] introduced five categories for word shapes: all digits, lower case, upper case, mixed and other. Word shape alone is not a good feature for

QC, however, when it is combined with other kinds of features, it usually improves the accuracy of QC [13, 53]. Capitalization feature is not present in Bengali; so we have considered only the other three categories, i.e., all digits, mixed and other.

Example-1: *ke gOdZa prawiRTA karena ?*

Gloss: Who established Goura?

Lexical features: wh-word: ke; wh-word position: first; wh-type: SI; question length: 5; end marker: ? word shape: other

4.2 *Syntactic Features*

Although different works extracted several syntactic features (f_S), the most commonly used f_S are Part of Speech (POS) tags and head words [8].

- *POS tags:* In the present work, we used the POS tag of each word in a question such as NN (Noun), JJ (adjective), etc. POS of each question word is added to the feature vector. A similar approach was successfully used for English [15, 55]. This feature space is sometimes referred to as the bag-of-POS tags [53]. The Tagged-Unigram (TU) feature was formally introduced by [53]. TU feature is simply the unigrams augmented with POS tags. Loni *et al* [53] showed that considering the tagged-unigrams instead of normal unigrams can help the classifier to distinguish a word with different tags as two different features. For extracting the POS tags, the proposed classification work in Bengali uses a *Bengali Shallow Parser*[2] which produces POS tagged data as intermediate result.

- *Question head word:* Question head-word is the most informative word in a question as it specifies the object the question is looking for [13]. Correctly identifying head-words can significantly improve the classification accuracy. For example, in the question "*What is the oldest city in Canada?*" the headword is 'city'. The word 'city' in this question can highly contribute to classify this question as LOC: city.

Identifying the question's head-word is very challenging in Bengali because of its syntactic nature and no research has been conducted so far on this. Based on the position of the interrogative in the question, we use heuristics to identify the question head-words. According to the position of the interrogative, three cases are possible.

– *Position-I (at the beginning):* If the question-word (i.e., marked by WQ tag) appears at the beginning then the first NP chunk after the interrogative-word is considered as the head-word of the question. Let us consider the following question.

Example-2: *ke*(/WQ) *gOdZa*(/NNP) *prawiRTA*(/NN) *karena*(/VM) ?(/SYM)

English Gloss: Who established Goura ?

In the above example, gOdZa is the head-word.

– *Position-II (in between)*: If the position of the question-word is neither at the beginning or at the end then the immediate NP-chunk before the interrogative-word is considered as the head-word. Let us consider the following question.

Example-3: *gOdZa*(/NNP) *koWAyZa*(/WQ) *abashiwa*(/JJ) ?(/SYM)

English Gloss: Where is Goura situated ?

In the above example gOdZa is considered as the question head-word.

– *Position-III (at the end)*: If the question-word appears at the end (i.e., just before the end of sentence marker) then the immediate NP-chunk before the interrogative-word is considered as the question head-word. Therefore, a similar action is taken for Position II and III.

Example-4:[*bAMlAxeSe arWanIwi kaleja*](/NNP) *kayZati* (/WQ) ?(/SYM)

English Gloss: How many economics colleges are in Bangladesh?

Therefore, in the Example-4 [*bAMlAxeSe arWanIwi kaleja*] is the question head-word.

Now, if we consider the example "*ke gOdZa prawiRTA karena ?*" then the syntactic features will be: [{WQ, 1},{NNP, 1}, {NN, 1}, {VM, 1},{head-word,gOdZa}]. Here a feature is represented as {⟨ POS, frequency ⟩}.

4.3 *Semantic Features*

Semantic features (f_M) are extracted based on the semantics of the words in a question. In this study, related words and named entities are used as f_M.

- *Related word:* A Bengali synonym dictionary is used to retrieve the related words. Three lists of related words were manually prepared 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 three lists (namely date, food, human activity), then its category name is added to the feature vector. For instance, the question "*ke gedZera sbAXIna narapawi Cilena ?*" (gloss: who was the independent ruler of Goura ?) contains the word *narapawi* which belongs to the human authority list. For this example question the semantic feature is added to the feature vector as: [{human-authority, 1}].

- *Named entities:* We used named entities (NE) as a semantic feature which was also recommended in other works [15, 55] on other languages. To identify the Bengali named entities in the question text, a Margin Infused Relaxed Algorithm (MIRA) based Named Entity Recognizer (NER) [57] is used for the present study. For the Example-5 question, the NE semantic feature is added to the feature vector as: [Location, 1].

[1] All the Bengali examples in this paper are written in WX [56] notation which is a transliteration scheme for representing Indian languages in ASCII.

[2] http:// ltrc.iiit.ac.in/analyzer/bengali/

Example-5: *ke gOdZa*[Location] *prawiRTA karena*?
English Gloss: Who established Goura ?

5 Experiments and Results

Many supervised learning approaches [13,55,58] have been proposed for QC over the years. But these approaches primarily differ in the classifier they use and the features they train their classifier(s) on [8]. We assume that a Bengali question is unambiguous, i.e., a question belongs to only one class. Therefore, we considered multinomial classification which assigns the most likely class from the set of classes to a question. Recent studies [12–14] also considered one label per question.

We used state-of-the-art classifier combination approaches: ensemble, stacking and voting. We have used two contemporary methods for creating accurate ensembles, namely, bagging and boosting. We employed the Rapid Miner tool for all the experiments reported here. Each of the three classifier combination approaches was tested with Naïve Bayes (NB), Kernel Naïve Bayes (k-NB), Rule Induction (RI) and Decision Tree (DT) classifiers.

Classification accuracy is used to evaluate the results of our experiments. Accuracy is the widely used evaluation metric to determine the class discrimination ability of classifiers, and is calculated using the following equation:

$$\text{accuracy} = \frac{\text{number of correctly classified samples}}{\text{total number of tested samples}}$$

5.1 *Corpus Annotation and Statistics*

We carried out our experiments on the dataset described in [7]. The questions in this dataset are acquired from different domains, e.g., education, geography, history, science, etc. We hired two native language (i.e., Bengali) specialists for annotating the corpus. Another senior native language expert was hired to support the two language specialists. The annotators were instructed to consult the senior native language expert in case of any confusion. In order to minimize disagreement, two language specialists gathered to discuss the question taxonomy in detail before initiating the annotation task. We set a constraint that a question will be annotated such that it is unambiguous, i.e., only a question class will be assigned to a question. We measured the inter-annotator agreement using non-weighted kappa coefficients [59]. The kappa coefficient for the annotation task was 0.85 which represents very high agreement. In case of or disagreement, the senior language specialist took the final decision.

The class-specific distribution of questions in the corpus is given in Table 3. It can be observed from Table 3 that the most frequent question class in the dataset is ‘Person’. The dataset contains a total of 1,100 questions. We divided the question corpus into 7:3 ratio for experimentation. The experimental dataset consists of 1100 Bengali questions of which 70% are used for training and the rest (331 questions, 30%) for testing the classification models.

Table 3. Corpus statistics

Class	Train	Test	Overall
Person	172	90	262
Organization	74	30	104
Location	76	30	106
Temporal	81	35	116
Numerical	71	30	101
Methodical	75	29	104
Reason	73	26	99
Definition	78	38	116
Miscellaneous	69	23	92
Total	769	331	1100

5.2 *Coarse-Grained Classification*

The empirical study of state-of-the-art classifier combination approaches (i.e., ensemble, stacking, and voting) was performed on the said dataset using four classifiers - namely, NB, k-NB, RI and DT. Each experiment can be thought of as a combination of three experiments since each classifier model was tested on $\{f_L\}$, $\{f_L, f_S\}$ and $\{f_L, f_S, f_M\}$ feature sets separately. Overall thirteen experiments were performed for coarse-grained classification and the evaluation results are reported in Table 4.

5.2.1 *Ensemble Bagging*

The bagging approach was applied separately to four classifiers (i.e., NB, k-NB, RI and DT) and the obtained accuracies are summarized in Table 4. Initially, the size (i.e., number of iterations) of the base learner was set to 2. Subsequently, experiments were performed with gradually increasing size ($size > 2$). The classification accuracy enhanced with increase in size. However, after a certain size, the accuracy was almost stable. At $size = 2$ and feature set $\{f_L, f_S, f_M\}$, the NB classifier achieved 82.23% accuracy and at $size \geq 9$, it became stable with 83.25% accuracy. At $size = 2$ and feature set $\{f_L, f_S, f_M\}$,the k-NB classifier achieved 83.87% accuracy and at $size \geq 15$, it became stable with 84.22% accuracy. At $size = 2$ and feature set $\{f_L, f_S, f_M\}$, the RI classifier achieved 85.97% accuracy and at $size \geq 8$, it be-

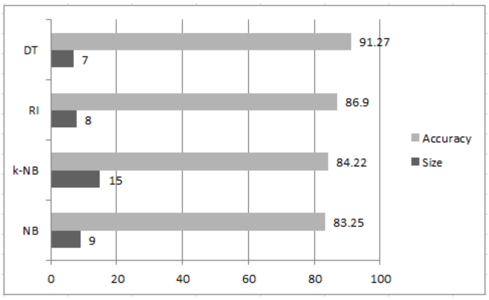

Figure 1. Size and Accuracy variation in Bagging with $\{f_L, f_S, f_M\}$

Table 4. Classifier combination results for coarse-grained classification

Approach	Base-Learner	Model-Learner	f_L	f_L+ f_S	f_L+f_S+f_M
Bagging	NB	x	81.53	82.77	83.25
	k-NB	x	82.09	83.37	84.22
	RI	x	83.96	85.61	86.90
	DT	x	85.23	86.41	**91.27**
Boosting	NB	x	81.74	82.71	83.51
	k-NB	x	83.86	85.63	86.87
	RI	x	83.55	85.59	86.27
	DT	x	85.21	86.58	**91.13**
Stacking	k-NB, RI, DT	NB	81.76	82.79	83.64
	NB, RI, DT	k-NB	83.86	85.54	86.75
	NB, k-NB, DT	RI	85.55	87.69	**91.32**
	NB, k-NB, RI,	DT	85.07	86.73	89.13
Voting	NB, k-NB, RI, DT	x	86.59	88.43	**91.65**

came stable with 86.90% accuracy. At *size* = 2 and feature set $\{f_L, f_S, f_M\}$,the DT classifier achieved 88.09% accuracy and at *size* $\geq$ 7, it became stable with 91.27% accuracy. It was observed from the experiments that with bagging the DT classifier performs best on any feature set for any size. For the experiments with the f_L features, the bagging size of NB, k-NB, RI and DT are 12, 19, 11 and 10 respectively after which classification accuracy becomes stable. Similarly, for experiments with $\{f_L, f_S\}$ feature set, the optimal bagging sizes are 10, 17, 9 and 8 for NB, k-NB, RI and DT respectively after which the corresponding classification accuracies converge. The Figure 1 shows the variation in size and accuracy for the best feature set.

5.2.2 *Ensemble Boosting*

Like bagging, boosting (AdaBoost.M1) was also applied separately to the four base classifiers. Table 4 tabulates the accuracies obtained with the boosting approach with the four classifiers. Here, we empirically fixed the iterations of boosting for the four classifiers to 12, 16, 10 and 8 respectively for the feature set $\{f_L, f_S, f_M\}$, since the corresponding weight of $\frac{1}{\beta_t}$ becomes less than 1 beyond those values. If $\frac{1}{\beta_t}$ is less than 1, then the weight of the classifier model in boosting may be less than zero for that iteration. The Figure 2 shows the variation in size and accuracy for the best feature set.

Similarly, for the feature sets $\{f_L, f_S\}$ and $\{f_L\}$ the iterations are set to 13, 18, 12, 9 and 14, 19, 14, 11 respectively for the four classifiers. Overall the DT classifier performs the best. However, unlike in bagging, k-NB performs better than RI with boosting.

5.2.3 *Stacking*

In stacking, three out of the four classifiers are used as the base learners (BL) and the remaining classifier is used as the model learner (ML). Therefore, four experiments were conducted separately for each of the four classifiers as the ML. The obtained accuracies are summarized in Table 4.

Experimental results revealed that with RI as the model learner and NB, k-NB, DT as the base learners, the classifier achieves the best classification accuracy.

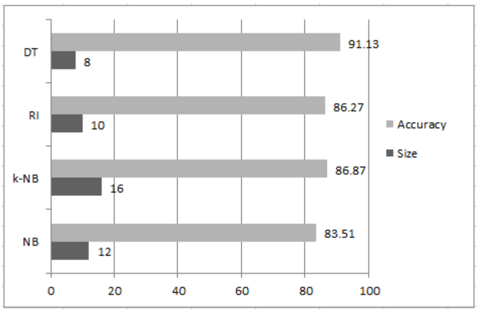

Figure 2. Size and Accuracy variation in Boosting with $\{f_L, f_S, f_M\}$

5.2.4 *Voting*

In voting, four classifiers altogether were used as the base learners and majority vote was used as voting approach. The evaluation results of the voting approach are presented in Table 4.

5.3 *Result Analysis of Coarse-Grained Classification*

Classifier combination is an established research known under different names in the literature: committees of learners, mixtures of experts, classifier ensembles, multiple classifier systems, etc. A number of research [18,19,22,24] established that classifier combination could produce better results than single classifier. Generally, the key to the success of classifier combination approach is that it builds a set of diverse classifiers where each classifier is based on different subsets of the training data. Therefore, our objective is to verify the impact of the classifier combination approaches over the individual classifier approaches on Bengali QC task.

The automated Bengali QC system by [7] is based on four classifiers, namely NB, k-NB, RI and DT, which were used separately.

Table 5. Experimental results of [7]

Classifier	f_L	f_L+f_S	f_L+f_S+f_M
NB	80.65	81.34	81.89
k-NB	81.09	82.37	83.21
RI	83.31	84.23	85.57
DT	84.19	85.69	87.63

The experimental results obtained by [7] are shown in Table 5. In that work, NB was used as the baseline and the DT classifier achieved the highest accuracy of 87.63% (cf. Table 5). A comparison of the results in Table 4 and Table 5 reveals that each classifier combination model performs better than the single classifier models in terms of classification accuracy. The prime reason is that classifier combination approaches reduce model bias and variance more effectively than individual classifiers.

In comparison to the earlier experiments reported in [7], with the bagging approach, classification accuracy of each classifier increases notably with bagging. The classification accuracy on the $\{f_L\}$, $\{f_L, f_S\}$ and $\{f_L, f_S, f_M\}$ feature sets increases by 1.04%, 0.72% and 3.64% for best performing DT classifier. Similarly, with the boosting approach, the classification accuracy for the best performing DT classifiers notably increases by 1.02%, 0.89% and 3.50% on $\{f_L\}$, $\{f_L, f_S\}$ and $\{f_L, f_S, f_M\}$ feature set. The stacking approach increases the accuracy on the $\{f_L, f_S\}$ feature set than the bagging and boosting approaches. This approach increases the classification accuracy by 1.36%, 2.74% and 0.69% on the $\{f_L\}$, $\{f_L, f_S\}$ and $\{f_L, f_S, f_M\}$ feature sets respectively. The voting approach not only increases the classification accuracy, but also provides the maximum accuracy for all the feature sets than the other combined approaches. The voting approach increases the classification accuracy on the $\{f_L\}$, $\{f_L, f_S\}$ and $\{f_L, f_S, f_M\}$ feature sets by 2.40%, 2.40% and 4.02% respectively. Therefore, overall the voting approach with majority voting performed the best among the four classifier combination approaches.

Bagging approach helps to avoid over fitting by reducing variance [18]. However, after certain iteration, it cannot reduce variance. Hence, after certain iteration, it does not improve the performance of the model. Therefore, we observed that after size (i.e., number of iterations), it was unable to enhance the accuracy.

On the other hand, boosting approach enhance the performance of the model by primarily reducing the bias [60]. However, after certain iteration (size) it cannot be improved. Because after certain iterations, the corresponding weight of $\frac{1}{\beta_t}$ becomes less than 1. If $\frac{1}{\beta_t}$ is less than 1, then the weight of the classifier model in boosting may be less than zero for that iteration. Therefore, we were not able to improve the accuracy after specific boosting size.

In stacking, the model learner is trained on the outputs of the base learners that are trained based on a complete training set [21]. Out experiment reveals that RI as model learner and NB, k-NB, DT as the base learners outperforms the other models.

In the context of Bengali question classification task, we conclude from the experimental results that although classifier combination approach outperforms the individual classifier approach, the impact of different classifier combination approaches is almost same for the Bengali course classes. Because, we obtained almost similar accuracy for different classifier combination approaches, namely, ensemble, stacking and voting.

Table 6. Fine-grained classification using individual classifiers

Classifier	Class	f_L	f_L+f_S	f_L+f_S+f_M
NB	F_{PER}	74.07	75.54	77.07
	F_{ORG}	75.33	76.55	77.70
	F_{LOC}	76.15	77.02	77.87
	F_{TEM}	75.74	77.16	77.97
	F_{NUM}	74.61	75.45	76.55
	F_{METH}	76.35	77.42	78.50
	F_{REA}	76.19	77.20	78.02
	F_{DEF}	76.30	77.45	78.56
	F_{MISC}	75.80	76.95	77.40
k-NB	F_{PER}	75.72	77.33	78.41
	F_{ORG}	76.76	77.97	79.28
	F_{LOC}	77.52	78.55	79.40
	F_{TEM}	77.22	78.73	79.57
	F_{NUM}	76.09	76.94	78.05
	F_{METH}	77.92	79.14	80.24
	F_{REA}	77.82	79.36	80.33
	F_{DEF}	77.99	79.40	80.43
	F_{MISC}	77.37	78.74	79.60
RI	F_{PER}	77.96	79.04	80.12
	F_{ORG}	78.29	79.56	80.75
	F_{LOC}	77.67	78.36	79.18
	F_{TEM}	79.17	80.76	81.73
	F_{NUM}	78.04	79.03	80.42
	F_{METH}	79.87	81.00	82.12
	F_{REA}	79.62	80.93	82.06
	F_{DEF}	78.98	80.28	81.28
	F_{MISC}	78.59	79.91	80.90
DT	F_{PER}	80.37	82.06	83.61
	F_{ORG}	78.78	80.26	81.68
	F_{LOC}	78.51	79.63	80.94
	F_{TEM}	80.58	82.03	83.50
	F_{NUM}	79.00	80.50	81.85
	F_{METH}	80.62	82.55	84.47
	F_{REA}	80.51	82.49	84.42
	F_{DEF}	79.89	81.07	82.49
	F_{MISC}	79.74	81.72	84.07

5.4 *Fine-Grained Classification*

Initially, we applied NB, k-NB, RI and DT classifiers separately. Each classifier was trained with $\{f_L\}$, $\{f_L, f_S\}$ and $\{f_L, f_S, f_M\}$ feature sets. The performance of the classifiers increases gradually with incorporation of syntactic and semantic features (i.e., $\{f_L\} \rightarrow \{f_L, f_S\} \rightarrow \{f_L, f_S, f_M\}$). The NB classifiers achieved around 77% of accuracy while the k-

Table 7. Ensemble results of fine-grained classification

		Bagging			Boosting		
		f_L	f_L+f_S	f_L+f_S+f_M	f_L	f_L+f_S	f_L+f_S+f_M
NB	F_{PER}	79.65	81.23	82.87	79.89	81.41	82.95
	F_{ORG}	81.01	82.32	83.55	81.65	82.73	83.98
	F_{LOC}	81.89	82.82	83.73	82.28	83.85	85.04
	F_{TEM}	81.45	82.97	83.84	81.89	83.01	83.97
	F_{NUM}	80.23	81.13	82.31	81.02	81.92	83.03
	F_{METH}	82.10	83.25	84.41	82.25	83.37	84.53
	F_{REA}	81.93	83.02	84.17	82.06	83.11	84.23
	F_{DEF}	82.05	83.29	84.47	82.09	83.32	84.56
	F_{MISC}	81.51	82.75	83.23	81.62	82.79	83.75
k-NB	F_{PER}	80.13	81.83	82.97	80.17	81.91	83.02
	F_{ORG}	81.23	82.51	83.89	81.29	82.63	83.91
	F_{LOC}	82.03	83.12	84.02	82.10	83.17	84.09
	F_{TEM}	81.71	83.31	84.20	81.79	83.39	84.28
	F_{NUM}	80.52	81.42	82.59	80.63	81.58	82.69
	F_{METH}	82.45	83.75	84.91	82.48	83.79	84.98
	F_{REA}	82.35	83.98	85.01	82.41	84.02	85.09
	F_{DEF}	82.53	84.02	85.11	82.61	84.12	85.13
	F_{MISC}	81.87	83.32	84.23	81.91	83.39	84.28
RI	F_{PER}	81.85	82.98	84.12	81.92	83.06	84.22
	F_{ORG}	82.19	83.53	84.78	82.25	83.61	84.85
	F_{LOC}	81.54	82.27	83.13	81.55	82.26	83.15
	F_{TEM}	83.12	84.79	85.81	83.18	84.85	85.93
	F_{NUM}	81.93	82.97	84.43	82.01	83.03	84.49
	F_{METH}	83.85	85.04	86.22	83.91	85.06	86.31
	F_{REA}	83.59	84.97	86.15	83.68	85.11	86.33
	F_{DEF}	82.92	84.28	85.33	82.95	84.32	85.41
	F_{MISC}	82.51	83.89	84.93	82.57	83.93	84.98
DT	F_{PER}	84.79	86.57	88.21	84.81	86.63	88.53
	F_{ORG}	83.11	84.67	86.17	83.14	84.73	86.23
	F_{LOC}	82.83	84.01	85.39	82.87	84.13	85.52
	F_{TEM}	85.01	86.54	88.09	85.03	86.58	88.15
	F_{NUM}	83.34	84.92	86.35	83.38	84.97	86.44
	F_{METH}	85.05	87.09	89.11	85.09	87.14	89.12
	F_{REA}	84.93	87.02	89.06	84.96	87.11	89.09
	F_{DEF}	84.28	85.53	87.02	84.29	85.55	87.05
	F_{MISC}	84.12	86.21	88.69	84.15	86.23	88.73

NB and RI classifiers achieved around 80% of accuracy for the fine-grained question classes. Only the DT classifier obtained more than 80% accuracy for all the question classes. The detailed evaluation results of the fine-grained question classification task using individual classifier are given in Table 6. The subsequent sections describe the experiments with classifier combination approaches.

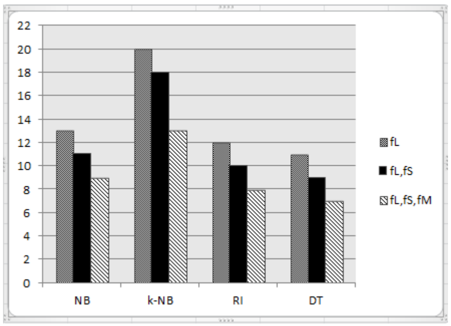

Figure 3. Size variation in Bagging

5.4.1 *Ensemble Bagging*

In this approach, we use four classifiers as base learners individually: NB, k-NB, RI and DT. Initially, the base learners are trained using the lexical features (f_L). Then semantic and syntactic features are added gradually for classification model generation. Therefore, three classification models were generated for each base learner. Thus, altogether 12 models were prepared for bagging. Like coarse-grained classification, initially the size (number of iteration) of the base learner was set to 2. Subsequently experiments were performed with gradually increasing sizes (*size* > 2). The classification accuracy increased with higher values of size. However, after certain iterations the accuracy was almost stable. For the fine-grained classes of PER coarse-class (i.e.,

F_{PER}), with $\{f_L, f_S, f_M\}$) feature set at *size* = 2 , the NB classifier achieved 81.98% classification accuracy and at *size* $\geq$ 9, it became stable with 82.87% accuracy. Similarly, with $\{f_L, f_S, f_M\}$ feature set the k-NB, RI and DT classifiers achieved stable accuracies at *size* equal to 13, 8 and 7 respectively. For the lexical feature set, the bagging size of NB, k-NB, RI and DT were 13, 20, 12 and 11 respectively after which the classification accuracy became stable. For the combined lexical and syntactic features, the recorded bagging size of NB, k-NB, RI and DT were 11, 18, 10 and 9 respectively. Figure 3 depicts the iteration size for the bagging approach.

5.4.2 *Ensemble Boosting*

Like the ensemble bagging approach, we applied boosting (i.e., AdaBoost.M1) separately to the four classifiers. Experimental results confirm that performances of the four base classifiers improve slightly using AdaBoost.M1. Table 7 presents the results of the boosting experiments and shows that altogether DT outperforms the other classifiers in the ensemble approach, i.e., bagging and boosting.

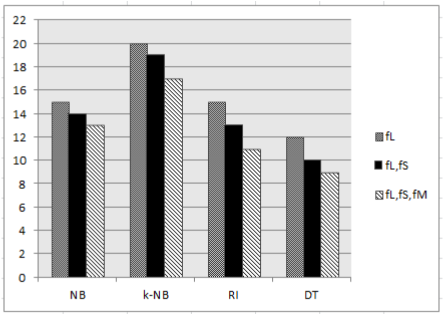

Figure 4. Size variation in Boosting

In the boosting approach, the number of iterations depends on $\frac{1}{\beta_t}$. When the value of $\frac{1}{\beta_t}$ becomes less than 1, then for that iteration the weight of the boosting classification may be less than zero. Hence, we empirically fixed the iterations of AdaBoost.M1 for the four classifiers (i.e., NB, k-NB, RI and DT) to 13, 17, 11 and 9 respectively for the feature set $\{f_L, f_S, f_M\}$ since the weight of $\frac{1}{\beta_t}$ becomes less than 1 after those values. Similarly, for feature set $\{f_L, f_S\}$ and $\{f_L\}$the iterations were 14, 19, 13, 10 and 15, 20, 15, 12 respectively for the four base classifiers. Figure 4 depicts the iteration sizes of the four classifiers in the boosting approach.

5.4.3 *Stacking*

As discussed in Section 5.2.3, in stacking one classifier plays the role of ML while the remaining classifiers act as BLs. Therefore, with four classifiers four experiments were conducted separately. The obtained accuracies are reported in Table 8. From the experimental results it was observed that the model trained with DT as the model learner and NB, k-NB, RI as the base learners achieved the best classification accuracy.

Table 8. Results of fine-grained classification with stacking

Base Learner	Model Learner	Class	f_L	f_L+f_S	f_L+f_S+f_M
k-NB,RI,DT	NB	F_{PER}	79.81	81.67	82.86
		F_{ORG}	81.79	83.02	84.02
		F_{LOC}	81.97	83.74	84.91
		F_{TEM}	81.45	82.81	83.73
		F_{NUM}	81.83	82.07	83.54
		F_{METH}	82.15	83.13	84.09
		F_{REA}	82.24	83.36	84.42
		F_{DEF}	81.76	83.05	84.23
		F_{MISC}	80.21	82.33	83.21
NB,RI,DT	k-NB	F_{PER}	79.93	81.79	83.03
		F_{ORG}	81.86	83.16	84.13
		F_{LOC}	82.08	83.82	85.06
		F_{TEM}	81.52	83.01	83.87
		F_{NUM}	81.97	82.18	83.71
		F_{METH}	82.28	83.20	84.18
		F_{REA}	82.31	83.43	84.45
		F_{DEF}	81.82	83.21	84.31
		F_{MISC}	80.29	82.42	83.35
NB,k-NB,DT	RI	F_{PER}	80.56	83.06	84.22
		F_{ORG}	82.86	83.98	85.03
		F_{LOC}	80.23	81.49	82.95
		F_{TEM}	83.21	84.78	85.97
		F_{NUM}	82.37	83.42	84.77
		F_{METH}	83.54	84.93	86.27
		F_{REA}	84.03	85.75	86.73
		F_{DEF}	80.01	82.33	84.21
		F_{MISC}	82.45	83.86	84.87
NB,k-NB,RI	DT	F_{PER}	84.97	86.69	88.71
		F_{ORG}	83.32	85.06	87.43
		F_{LOC}	82.93	84.21	85.71
		F_{TEM}	84.84	86.13	87.95
		F_{NUM}	83.57	85.17	87.49
		F_{METH}	84.85	86.91	88.56
		F_{REA}	84.69	86.78	88.29
		F_{DEF}	84.38	85.65	87.51
		F_{MISC}	84.02	86.11	88.42

5.4.4 *Voting*

Unlike the ensemble approach, in the voting approach all the classifiers were applied at the same time to predict the question class. Table 9 tabulates the the accuracies obtained with this approach.

Table 9. Results of fine-grained classification with voting

Base Learner	Class	f_L	f_L+f_S	f_L+f_S+f_M
NB, k-NB,RI,DT	F_{PER}	79.81	81.67	82.86
	F_{ORG}	81.79	83.02	84.02
	F_{LOC}	81.97	83.74	84.91
	F_{TEM}	81.45	82.81	83.73
	F_{NUM}	81.83	82.07	83.54
	F_{METH}	82.15	83.13	84.09
	F_{REA}	82.24	83.36	84.42
	F_{DEF}	81.76	83.05	84.23
	F_{MISC}	80.21	82.33	83.21

5.5 *Result Analysis of Fine-Grained Classification*

As research studies [18,19,22,24] argued that classifier combination approaches provide better prediction results over individual classifier approach, our motivation is to verify the impact of the classifier combination approaches on Bengali QC task.

Initially, we carried out our experiment with individual classifier approach and applied NB, k-NB, RI and DT classifiers separately. Table 6 presents the results obtained using individual classifier approach. In fine-grained classification task, we used the identical features that were also used in coarse-grained classification. Inevitably, the obtained accuracies for fine-grained classification is less than the coarse-grained classification using the same feature sets.

Then, we applied the state-of-the-art classifier combination techniques on the lexical, syntactic and semantic feature sets. Figure 3 depicts the bagging size (i.e., number of iterations) for fine-grained classification. Breiman [18] stated that bagging approach improves the performance of a prediction model by reducing the variance. However, after certain iteration, it cannot reduce variance and the model becomes stable. Hence, after certain iteration, we were not able to improve the performance of the models. We noticed that the bagging approach requires more iteration to stable in fine-grained classification in comparison to the coarse-grained classification. In contrast, boosting approach enhance the performance of the model by primarily reducing the bias [60]. After certain iterations, the boosting approach cannot reduce the bias because the corresponding weight of $\frac{1}{\beta_t}$ becomes less than 1. If $\frac{1}{\beta_t}$ is less than 1, then the weight of the classifier model in boosting may be less than zero for that iteration. Hence, in Figure 4, we can see that the boosting size is stable after certain iterations. Table 7 shows that the boosting approach achieves slightly better performance than the bagging. In stacking approach, one classifier plays the role of ML and a set of classifiers act as BLs. In the stacking approaches, the setup with NB, k-NB, RI as BLs and DT as ML outperforms other setup combinations. The stacking approach outperforms the voting approach with slight margin. However, the boosting approach with the base classifier DT achieves the best. It was noticed from the fine-grained question classification that all the classifier combination approaches beat the individual classifier approaches with a notable margin.

5.6 *Error Analysis*

We checked the dataset and the system output to analyze the errors. We observed the following as the major sources of errors in the proposed system.

- Questions belonging to different question classes have the same content words which make the classifiers confuse and wrongly classify the questions into same class. For example, both the questions “koVna saByawAkeV bExika saByawA balA hayZa ?” (gloss: which civilization is called the Vedic Civilization?), “Arya saByawAkeV keVna bExika saByawA balA hayZa ?” (gloss: why the Arya Civilization is called the Vedic Civilization?) have the same content words: *saByawAkeV, bExika, saByawA, hayZa.*

- In Bengli, the dual interrogatives consist of two single interrogatives. Thus, classifiers get confused by encountering two interrogative words. Therefore, classifiers often misclassify such questions.

- The classifiers wrongly classified the Bengali questions which are long and complex. For example, ‘keVna AXunika yugeVra paNdiweVrA maneV kareVna yeV, sinXu saByawA xrAbidZa jAwIra xbArA sqRti hayZeVCila ? (gloss: why the modern scholars think that the Indus Valley Civilization is created by the Aryans?).

6 Conclusions

Although QA research in other languages (such as English) has progressed significantly, for majority of Indian languages it is at the early stage of development. In this study, we addressed the QC task for Bengali, one of the most spoken languages in the world and the second most spoken language in India. We reported experiments for coarse-grained and fine-grained question classification. We employed lexical, syntactic and semantic features. We applied classifiers individually as well as combination approaches. The automated Bengali question classification system obtains up to 91.65% accuracy for coarse-grained classes and 87.79% for fine-grained classes using classifier combination approaches based on four classifiers, namely NB, k-NB, RI and DT. The contributions of this work are listed below.

- This work successfully deploys state-of-the-art classifier combination approaches for the question classification task in Bengali.

- We have empirically established the efficacy of the classifier combination approach over individual classifier approach for coarse-grained question classification as well as fine-grained question classification.

- We have extended the single layered (coarse-grained) taxonomy into two layered (coarse-grained and fine-grained) taxonomy by incorporating 69 fine-grained classes to the question classification taxonomy.

- This work improves QC accuracy which in turns enhances the Bengali QA system performance.

In coarse-grained question classification, overall the voting approach with majority voting technique performs best among the four classifier combination approaches, namely bagging, boosting, stacking, and voting. However, the stacking approach produces the best results for fine-grained classification.

The only available QA dataset for Bengali contains only 1,100 questions. In future, we would like to contribute to enlarge the dataset. One of the future directions of this study is employing the state-of-the-art neural network techniques. Also, we would like to apply the approaches used in this study to other less investigated languages.

7 Acknowledgments

Somnath Banerjee and Sudip Kumar Naskar are supported by Digital India Corporation (formerly Media Lab Asia), MeitY, Government of India, under the Visvesvaraya PhD Scheme for Electronics & IT.

The work of Paolo Rosso was partially funded by the Spanish MICINN under the research project PGC2018-096212-B-C31.

References

[1] Dan Jurafsky and James H Martin. *Speech and language processing*. Pearson, 2014.

[2] James H Martin and Daniel Jurafsky. Speech and language processing. *International Edition*, 710, 2000.

[3] Ellen M Voorhees. Overview of the trec 2001 question answering track. *NIST special publication*, pages 42–51, 2002.

[4] Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. Toward semantics-based answer pinpointing. In *Human language technology research*, pp. 1–7. ACL, 2001.

[5] Abraham Ittycheriah, Martin Franz, Wei-Jing Zhu, Adwait Ratnaparkhi, and Richard J Mammone. Ibm's statistical question answering system. In *TREC*, 2000.

[6] Dan Moldovan, Marius Paşca, Sanda Harabagiu, and Mihai Surdeanu. Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems (TOIS)*, 21(2):133–154, 2003.

[7] Somnath Banerjee and Sivaji Bandyopadhyay. Bengali question classification: Towards developing QA system. In *Proceedings of the 3rd Workshop on South and Sotheast Asian Language Processing (SANLP), COLING*, pages 25–40, 2012.

[8] Babak Loni. A survey of state-of-the-art methods on question classification. *Delft University of Technology, Tech. Rep*, 2011.

[9] David A Hull. Xerox trec-8 question answering track report. In *TREC*, 1999.

[10] John Prager, Dragomir Radev, Eric Brown, Anni Coden, and Valerie Samn. The use of predictive annotation for question answering in trec8. *Information Retrieval*, 1(3):4, 1999.

[11] Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. Exploiting syntactic and shallow semantic kernels for question answer classification. In *Annual meeting-association for computational linguistics*, (45), pp. 776, 2007.

[12] Dell Zhang and Wee Sun Lee. Question classification using support vector machines. In *Proceedings of Research and development in informaion retrieval*, pages 26–32. ACM, 2003.

[13] Zhiheng Huang, Marcus Thint, and Zengchang Qin. Question classification using head words and their hypernyms. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 927–936. ACL, 2008.

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

[15] Xin Li and Dan Roth. Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(03):229–249, 2006.

[16] Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. Maximum entropy markov models for information extraction and segmentation. In *International Conference on Machine Learning (ICML)*, volume 17, pages 591–598, 2000.

[17] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[18] Leo Breiman. Bagging predictors. *Machine learning*, 24(2): 123–140, 1996.

[19] Robert T Clemen. Combining forecasts: A review and annotated bibliography. *International journal of forecasting*, 5(4):559–583, 1989.

[20] Michael Peter Perrone. *Improving regression estimation: Averaging methods for variance reduction with extensions to general convex measure optimization*. PhD thesis, Brown University, 1993.

[21] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[22] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12:993–1001, 1990.

[23] Anders Krogh, Jesper Vedelsby, et al. Neural network ensembles, cross validation, and active learning. *Advances in neural information processing systems*, 7:231–238, 1995.

[24] Sherif Hashem. Optimal linear combinations of neural networks. *Neural networks*, 10(4):599–614, 1997.

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

[26] Jude W Shavlik. Generating accurate and diverse members of a neural-network ensemble. 1996.

[27] Xin Li, Xuan-Jing Huang, and de WU Li. Question classification by ensemble learning. *IJCSNS*, 6(3):147, 2006.

[28] Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

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

[30] Keliang Jia, Kang Chen, Xiaozhong Fan, and Yu Zhang. Chinese question classification based on ensemble learning. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD 2007. ACIS*, volume 3, pages 342–347. IEEE, 2007.

[31] Lei Su, Hongzhi Liao, Zhengtao Yu, and Quan Zhao. Ensemble learning for question classification. In *Intelligent Computing and Intelligent Systems,ICIS*,pages 501–505.IEEE, 2009.

[32] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, and others Building Watson: An overview of the DeepQA project. In *AI magazine*, 31(3), pages 59–79, 2010.

[33] Manuel Alberto Pérez-Coutiño, Manuel Montes-y-Gómez, Aurelio López-López, and Luis Villaseñor Pineda. Experiments for Tuning the Values of Lexical Features in Question Answering for Spanish. In *CLEF (Working Notes)*, 2005.

[34] Günter Neumann and Bogdan Sacaleanu. A Cross–Language Question/Answering–System for German and English. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 559–571, Springer, 2003.

[35] FA Mohammed, Khaled Nasser, and HM Harb. Question classification with log-linear models. In *ACM SIGART Bulletin*, pages 21–30. ACM, 1993.

[36] Paolo Rosso, Yassine Benajiba, and Abdelouahi Lyhyaoui. In *Proc. 4th Conf. on Scientific Research Outlook & Technology Development in the Arab world*, pages 11–14, 2006.

[37] Lahsen Abouenour, Karim Bouzoubaa, and Paolo Rosso. IDRAAQ: New Arabic question answering system based on query expansion and passage retrieval. In *CELCT*, 2012.

[38] Tetsuya Sakai, Yoshimi Saito, Yumi Ichimura, Makoto Koyama, Tomoharu Kokubu, and Toshihiko Manabe. ASKMi: A Japanese question answering system based on semantic role analysis, In *Coupling approaches, coupling media and coupling languages for information retrieval.* pp. 215–231, 2004.

[39] Hideki Isozaki, Katsuhito Sudoh, and Hajime Tsukada NTT's Japanese-English Cross-Language Question Answering System. In *NTCIR*, 2005.

[40] Zhang Yongkui, Zhao Zheqian, Bai Lijun, and Chen Xinqing. Internet-based Chinese question-answering system. In *Computer Engineering*, volume 15, pages 34, 2003.

[41] Ang Sun, Minghu Jiang, Yifan He, Lin Chen, and Baozong Yuan. Chinese question answering based on syntax analysis and answer classification, In *Acta Electronica Sinica*, volume 36, number 5, pages 833–839, 2008.

[42] Shriya Sahu, Nandkishor Vasnik, and Devshri Roy. Prashnottar: a hindi question answering system. In *International Journal of Computer Science & Information Technology*, 4(2):149, 2012.

[43] Garima Nanda, Mohit Dua, and Krishma Singla. A hindi question answering system using machine learning approach. In *Computational Techniques in Information and Communication Technologies (ICCTICT)*, pages 311–314. IEEE, 2016.

[44] Satoshi Sekine and Ralph Grishman. Hindi-English cross-lingual question-answering system. In *ACM Transactions on Asian Language Information Processing (TALIP)*, 2(3): 181–192, 2003.

[45] Pushpraj Shukla, Amitabha Mukherjee, and Achla Raina. Towards a language independent encoding of documents. In *NLUCS 2004*, page 116, 2004.

[46] Santosh Kumar Ray, Amir Ahmad, and Khaled Shaalan. A review of the state of the art in hindi question answering systems. In *Intelligent Natural Language Processing: Trends and Applications*, pages 265–292. Springer, 2018.

[47] Praveen Kumar, Shrikant Kashyap, Ankush Mittal, and Sumit Gupta. A query answering system for e-learning Hindi documents. In *South Asian Language Review*, 13(1&2):69–81, 2003.

[48] Rami Reddy Nandi Reddy and Sivaji Bandyopadhyay. Dialogue based question answering system in Telugu. In *Proceedings of the Workshop on Multilingual Question Answering*, pages 53–60, 2006.

[49] Gursharan Singh Dhanjal, Sukhwinder Sharma, and Paramjot Kaur Sarao. Gravity based punjabi question answering system. In *International Journal of Computer Applications*, 147(3), 2016.

[50] MS Bindu and Idicula Sumam Mary. Design And Development Of A Named Entity Based Question Answering System For Malayalam Language. *PhD thesis, Cochin University Of Science And Technology*, 2012.

[51] Cheng-Wei Lee *et al*, Asqa: Academia sinica question answering system for NTCIR-5 CLQA. In *NTCIR-5 Workshop,Japan*, pages 202–208, 2005.

[52] Somnath Banerjee and Sivaji Bandyopadhyay. Ensemble approach for fine-grained question classification in Bengali. In the proceedings of the 27th Pacific Asia Conference on Language,Information, and Computation (PACLIC-27), pp. 75–84 2013.

[53] Babak Loni, Gijs Van Tulder, Pascal Wiggers, David MJ Tax, and Marco Loog. Question classification by weighted combination of lexical, syntactic and semantic features. In *TSD*, pages 243–250. Springer, 2011.

[54] Zhiheng Huang, Marcus Thint, and Asli Celikyilmaz. Investigation of question classifier in question answering. In *Proceedings of Empirical Methods in Natural Language Processing: Volume 2*, pages 543–550. ACL, 2009.

[55] Phil Blunsom, Krystle Kocik, and James R Curran. Question classification with log-linear models. In *Proceedings of ACM SIGIR conference on Research and development in information retrieval*, pages 615–616. ACM, 2006.

[56] Sapan Diwakar, Pulkit Goyal, and Rohit Gupta. Transliteration among indian languages using wx notation. In *Conference on Natural Language Processing*, number EPFL-CONF-168805, pp. 147–150. Saarland University Press, 2010.

[57] Somnath Banerjee, Sudip Kumar Naskar, and Sivaji Bandyopadhyay. Bengali named entity recognition using margin infused relaxed algorithm. In *International Conference on Text, Speech, and Dialogue*, pages 125–132. Springer, 2014.

[58] Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. ACL, 2002.

[59] Jacob Cohen. A coefficient of agreement for nominal scales.. In *Educational and psychological measurement* 20(1), pages 37–46, 1960.

[60] Robert E Schapire The strength of weak learnability *Machine learning*, 5(2):197–227,1990