Code Mixed Cross Script Factoid Question Classification - A Deep Learning Approach

Somnath Banerjee [a], Sudip Naskar [a], Paolo Rosso [b], and Sivaji Bandyopadhyay [a]
[a] *Department of Computer Science and Engineering,*
Jadavpur University, India
E-mail: sb.cse.ju@gmail.com, {sudip.naskar, sbandyopadhyay}@cse.jdvu.ac.in
[b] *PRHLT Research Center,*
Universitat Politècnica de València, Spain
E-mail: prosso@dsic.upv.es

Abstract. Before the advent of the Internet era, code-mixing was mainly used in the spoken form. However, with the recent popular informal networking platforms such as Facebook, Twitter, Instagram, etc., in social media, code-mixing is being used more and more in written form. User-generated social media content is becoming an increasingly important resource in applied linguistics. Recent trends in social media usage have led to a proliferation of studies on social media content. Multilingual social media users often write native language content in non-native script (cross-script). Recently Banerjee et al. [9] introduced the code-mixed cross-script question answering research problem and reported that the ever increasing social media content could serve as a potential digital resource for less-computerized languages to build question answering systems. Question classification is a core task in question answering in which questions are assigned a class or a number of classes which denote the expected answer type(s). In this research work, we address the question classification task as part of the code-mixed cross-script question answering research problem. We combine deep learning framework with feature engineering to address the question classification task and enhance the state-of-the-art question classification accuracy by over 4% for code-mixed cross-script questions.

Keywords: Question Answering, code-mixing, cross-scripting, question classification, deep learning, social media content

1. Introduction

In the last decade, social media (SM) has experienced significant growth among the netigens[1] of all ages. As netigens are the publishers in SM, the user-generated content is ever increasing, multilingual, diverse, and may or may not be formal. Everyday netigens publish contents on diverse topics which deal with, and are often personal views or discussions on recent events, tourism, technology, products and services, etc. While posting in SM, the use of code-mixing is spreading widely in informal text communications such as newsgroups, tweets, blogs, and other SM platforms. Furthermore, SM users often phonetically use Roman script instead of using their own native scripts due to the various socio cultural reasons [1]. We refer to the phenomenon of phonetic use of a non-native script for writing native words as cross-script. Therefore, the multilingual aspect in the user generated content of informal SM text communication is reflected not only in words usage (i.e., code mixed) but also in writing script (i.e., use of non-native script). In this work, we deal with Bengali–English code-mixed cross-script content. In addition to the typical challenges in natural language processing, the different forms of user-generated noise present additional challenges for code-mixed cross-script (CMCS) SM content, as given below:

– Words are spelled differently by various speakers (examples: korchi (English gloss: 'am doing') - *korchee, krchi, krchee, krchii*; night - *n8, ngt, ni8*).

[1] Persons actively involved in Internet communication

- Words are contracted phonetically for the ease of writing and sometimes to fit the content within short length (e.g., twitter, sms), e.g. great → *grt*, tomorrow→*2morw*.
- Punctuations are often omitted from contracted words. (Examples: can't→*cant* , won't → *wont*).
- Often intentional misspelling (referred to as wordplay) occurs for emphasis e.g. 'i m veryyyy happy' ('I'm very happy').
- Asterisk (*) and numbers are used in vulgar words such as *f**k*, *b1tch*.
- Sometimes unintentional (i.e.,genuine) misspellings (typos) occur such as '*coulf*' - could.
- Use of common vocabulary words create language identification problem. E.g. 'take' is present in English as well as it is a valid transliteration of a Bengali word (English gloss: 'him/her').
- Capitalization rules are usually not followed. This makes the task of text analysis for SM text very challenging.

Question Answering (QA) systems are gaining great importance due to the increasing amount of web content and the high demand for digital information that regular information retrieval techniques can not satisfy. The research community in natural language processing (NLP) has started paying sincere attention to user generated content (UGC), due to its prevalence in electronic communication, particularly in the SM. Code-mixed cross-script content amounts to a large portion of this social media content (SMC). Recently Banerjee et al. [9] addressed the code-mixed cross-script QA research problem- "*Building a question answering system which takes cross-script code-mixed questions as information request, processes a cross-script code-mixed text corpus and provides an (or a list of) exact answer(s) as information response.*" They reported that the ever increasing code-mixed cross-script user generated SM data could serve as a potential source of digital content for less-computerized languages and towards the present need of addressing CMCS QA research. Banerjee et al. [9] reported the following motivating factors for this novel research problem.

- Multilingual non-native English speakers predominantly use the Roman script in SM platforms during their conversations even while the written communication takes place entirely in a native[2] language.
- To make the written communication more interesting and give it a global flavor, borrowing foreign words from different languages is a common phenomenon in SM communication and this is a growing trend.
- For research purpose in less-computerized languages, the ever growing posts could be used as a potential source of digital content.
- The research community needs to move towards the next generation search engine that boosts the necessity of developing QA system for less-resourced languages.

Providing an appropriate answer in response to the user's question is one of the practical challenges in NLP and information retrieval. The challenges become even more complex for CMCS QA [9]. Question processing, a vital task in QA, generally involves construction of question representation, derivation of category name specifying the type of the expected answer, and keyword extraction. The second task is referred to as question classification (QC). In this work, we address the QC task as part of the CMCS QA research problem. The major contributions of this work include the following:

- For the first time, deep learning framework with feature engineering for the CMCS QC research problem is proposed.
- Convolutional Neural Network (CNN) approach has been successfully deployed for the CMCS QC research task and four variants of CNN based models have been proposed.
- In spite of the small dataset, the deep architecture performs well due to combining feature engineering with deep learning framework.
- The proposed approach outperforms the previous approaches to CMCS QC. This study obtains 87.22% accuracy in CMCS QC and enhances the state-of-the-art QC accuracy by around 4%.

The rest of the paper is structured as follows. We start with formally defining the QC problem in Section 2. The related works are described in Section 3. Section 4 presents the dataset. The taxonomy for CMCS questions is described in Section 5. Section 6

[2]Throughout this paper we refer to the non-English language of communication as the native language and English as the non-native language.

provides the details of the proposed framework. The feature engineering is discussed in Section 7. Section 8 details the experimental framework and the results. Finally, we draw conclusions in Section 9 and discuss future work.

2. Formal Definition of the Task

Adopting the formal definition of text categorization [34] to the problem of CMCS QC, the QC task can be defined as: a boolean value is assigned to each pair $(q_j, c_i) \in Q \times C$, where $Q = \{q_1, q_2, \ldots, q_k\}$ is the domain of questions and $C = \{c_1, c_2, \ldots, c_n\}$ is a set of predefined categories.

Assigning (q_j, c_i) to the value T ('True' or 1) indicates that q_j is judged to belong to the category c_i, while an assignment to the value F ('False' or 0) indicates that q_j is not judged as belonging to the category c_i. In a machine learning setting, the task is to make the unknown target function $\hat{\Phi} : Q \times C \rightarrow \{T, F\}$ approximate the ideal target function $\Phi : Q \times C \rightarrow \{T, F\}$, such that $\hat{\Phi}$ and Φ coincide as much as possible.

We can also present the QC task in another way. Let $F = \{f_1, f_2, \ldots, f_n\}$ be a set of factoid questions associated with domain D. Each factoid question $f : \langle w_1 w_2 w_3 \ldots w_p \rangle$ is a set of words where p denotes the total number of words in a question. The words $w_1, w_2, w_3, \ldots, w_p$ could be English words or transliterated from Bengali in the code mixed scenario. Let $T = \{t_1, t_2, \ldots, t_m\}$ be the set of question classes. Here n and m refer to the total number of questions and question classes respectively.

The objective of this task is to classify each given question $f_i \in F$ into one of the predefined coarse-grained classes $t_j \in T$. In the following example, the question should be classified to the class 'TEMPORAL'.

Example:

f_i : *last volvo bus kokhon chare?*

English gloss: When does the last volvo bus depart?

t_j : TEMPORAL

3. Related Work

A QA systems was developed since the development of Baseball system [17] in 1961. It was reported in [26] that QC task was addressed as an integral part of the QA system. However, mostly QC researches were based on non-CMCS data. Mainly two different approaches are used to classify questions: rule-based [21,30] and machine learning based [29,38]. However, a few hybrid approaches combine rule-based and machine learning based approaches [20,35]. Although enormous research works have been performed on European languages mostly in English, the scenario is not same for Indian languages. A few researches [3,4,5,8] on QA systems including QC were carried out for Bengali language.

Although the use of language mixing is becoming common in our written and spoken communication, no QA system has yet been developed on CMCS data. Recently, Banerjee et al.[9] proposed to develop QA system based on informal CMCS data. Also, a dataset [9] was developed for Bengali-English CMCS QA research. The QC task on a CMCS dataset was carried out on Mixed Script Information Retrieval (MSIR) shared task where 7 teams participated [6]. Rule based as well as machine learning based approaches were proposed for the Bengali-English CMCS QC task. The best performing team IINTU [12] approached the problem using ensemble learning which used three classifiers, namely, Random Forest (RF), One-vs-Rest and k-nearest neighbor. They achieved 83.33% of classification accuracy. Bharti Ganrsh et al. [18] employed vector space model (VSM) and achieved upto 80% accuracy. Saini et al. [33] used three machine learning classifiers, namely, Support Vector Machine (SVM), RF and Logistic Regression (LR) and the RF based model outperformed the other two classifiers. Only Majumber and Pakray [28] applied rule based (accuracy: 74.44%) as well as machine learning based (accuracy: 78.88%) approaches. They compiled a number of direct and dependent rules for each question class. In that work, Naïve Bayes was employed as machine learning based approach. Bhargava et al. [11] also employed three machine learning classifiers, namely Gaussian NB, LR and RF. The Gaussian NB classifier was found to outperform the other two classifiers. Anand Kumar and Sonman [27] developed two models. One was based on Bag-of-Words (BoW) whereas another was on Recurrent Neural Network (RNN). However, the BoW models (accuracy: 80.55%) outperformed the RNN models (accuracy: 73.88%) with large margin (almost 7%).

After the remarkable success in computer vision [24] and speech recognition [16], deep learning models have been successfully applied with significant success in recent times in various natural language processing tasks such as semantic analysis [36], machine translation [2], text summarization [32] and classifi-

cation problems [23]. Collobert et al. [15] proposed deep neural framework which can be applied to various natural language processing tasks including part-of-speech tagging, chunking, named entity recognition, and semantic role labeling. Deep learning framework has also been successfully applied to monolingual QA researches [22,31]. The deep learning framework performs well while the dataset is large. As the deep learning framework is less effective than other machine learning approaches while the dataset is small, the earlier attempt by Bharti Ganesh et al. [27] achieved only 73.88% of classification accuracy on CMCS question dataset which has only 330 data samples for training. The proposed deep learning framework additionally uses external feature engineering to overcome the scarcity of training dataset.

4. Dataset

In this study, we used the dataset described in [9] which is the only CMCS dataset available for QA research. The shared task conducted in MSIR-16 [6] also used the same dataset. The dataset contains a total of 510 English–Bengali CMCS factoid questions from the sports and tourism domains. In the dataset, all CMCS questions are unambiguous, i.e., every question has only one class. The questions are written in Roman script irrespective of language, i.e., Bengali words are also written phonetically in Roman script. The sports domain dataset is on cricket, a popular outdoor game in the Indian subcontinent and many parts of the world. On the other hand, the tourism domain dataset is based on famous tourism spots of India. The training dataset contains 330 labeled factoid CMCS questions, whereas the testset contains 180 data samples. In the training set, the average length of a question is 5.321, while the average question length is 6.322 in the testset. The 'Organization' class dominates the training data (20%), whereas the majority class in the testset is the 'Person' class (15%). Table 1 provides the statistics of the experimental dataset. Question class specific distributions are given in Table 2.

Table 1
MSIR16 Datasets

Dataset	Questions(Q)	Total Words	Avg. Words/Q
Trainset	330	1776	5.321
Testset	180	1138	6.322

Table 2
MSIR16 dataset: question class statistics

Class	Training	Testing
Person (PER)	55	27
Location (LOC)	26	23
Organization (ORG)	67	24
Temporal(TEMP)	61	25
Numerical(NUM)	45	26
Distance(DIST)	24	21
Money(MNY)	26	16
Object(OBJ)	21	10
Miscellaneous(MISC)	5	8

5. Question Taxonomy

The set of question categories are referred to as the question taxonomy or question ontology. Categorizing a question enables the later components to apply appropriate knowledge extraction strategies in order to generate appropriate answers.

Table 3
Taxonomy and tagset for CMCS Questions

Tag	Name	Description
PER	Person	Persons name i.e., name of human beings
LOC	Location	Locations
ORG	Organization	Organizations
NUM	Quantity	Numerical e.g., statistical related questions
TEMP	Temporal	Temporal e.g., date, time, year i.e., time related
MONEY	Monitory	Money related questions
DIST	Distance	Measurable distance related questions
OBJ	Object	Object e.g. temples, cannon objects etc.
MISC	Miscellaneous	Questions which do not fall in other classes

In MSIR-16@FIRE [6], a question taxonomy for CMCS QA systems was proposed. The aforementioned taxonomy is the only available question taxonomy for CMCS QA till date. The proposed taxonomy includes 9 classes. The question taxonomy proposed by [3] for monolingual Bengali QA also has 9 coarse classes. However, three coarse classes (namely, method, reason and definition) are not present in the CMCS QA taxonomy as the proposed CMCS QA taxonomy in [6] was intended for factoid questions. The CMCS QA taxonomy is based on the domains: sports

Table 4
CMCS question examples

Class	Example
Person (PER)	ke Hazarduari toiri kore? (gloss: Who built Hazarduari)
Location (LOC)	airport theke kothai jabar bus nei? (gloss: To where there is no bus from Airport?)
Organization (ORG)	prepaid taxi counter naam ki? (gloss: what is the name of prepaid taxi counter?)
Numerical (NUM)	Hazarduari te koto dorja ache? (gloss: How many doors are there in Hazarduari?)
Temporal (TEMP)	Volvo bus howrah station jete koto time nei? (gloss:How much time a volvo bus take to reach Howrah Station?)
Monitory (MNY)	Airport theke Howrah Station volvo bus fare koto? (gloss: What is the fair of volvo bus from Airport to Howrah Station)
Distance (DIST)	airport theke howrah station distance koto? (gloss: What is the distance of Howrah Station from Airport?)
Object (OBJ)	Murshidabad kon nodir tire obosthito? (gloss: Which river is located beside Murshidabad?)
Miscellaneous (MISC)	early morning journey hole kon service valo? (gloss: Which service will be good in early morning journey?)

and tourism. In addition to the four well-known basic question classes, viz. person, location, organization and temporal expression, in [6] the authors proposed four domain specific classes, namely - monitory, object, distance and miscellaneous. The 'object' class represents questions whose answers are objects of type transport medium (e.g., *Calcutta Delhi Express, Volvo Bus, etc.*), tourism spot specific valuable objects which could be bought (e.g., *Baluchori Sharee, Teracota Horse, etc.*), seen (e.g., *palace, sea beach, mountain, etc.*) or experienced (e.g., *opera, mountaineering, etc.*), available entertainment activities (e.g. skiing, scuba diving, hiking, etc). Table 3 presents the Bengali-English CMCS question taxonomy proposed in [6]. Table 4 shows examples of 9 CMCS question classes.

6. CNN Based Question Classification Model

In this section, we introduce our Convolutional Neural Network (CNN) based architecture for CMCS QA. The architecture is shown in Figure 1.

In this regard, it is to be noted that we experimented with two variants of word-vectors: i) word-vector is kept static throughout the training, and ii) word-vector is fine-tuned via backpropagation.

Character quantization: The input to our CNN based question classification model is a sequence of encoded characters. We use *one-of-m* encoding in this model. For the CMCS QA corpus, initially we extract the character vocabulary which is of size S. We used this dictionary to quantize each character. Since we are dealing with only factoid questions, we ignore the character '?'. The maximum length of each sequence was set to L, and for short sequences the remaining part was replaced by zeros using zero-padding. Thus, we constructed a $S * L$ vector.

The character-set used in all of our models consists of 66 characters ($S = 66$) which includes 26 English lower-case letters (a to z), 26 English upper-case letters (A to Z), 10 digits (0 to 9) and 3 other characters excluding space (:, ', -). We ignore '?' since all the questions end with '?'.

Embedding Layer: In our model, the embedding layer is treated as a look-up table. Usually, the embedding layer accepts the charter sequence and converts it into an one dimensional vector of fixed length L using the alphabet vocabulary. The zero-padding (i.e., the missing part replaced by zeros) helps to keep the input vector to a fixed size L. Thus we can treat the embedding layer as a look-up table.

Features: The training data size is small and contains only 330 data samples. However, it is a well known fact that deep neural networks typically require and work well on large dataset while statistical approaches and machine learning based classifiers have been found to work better on small amount of training

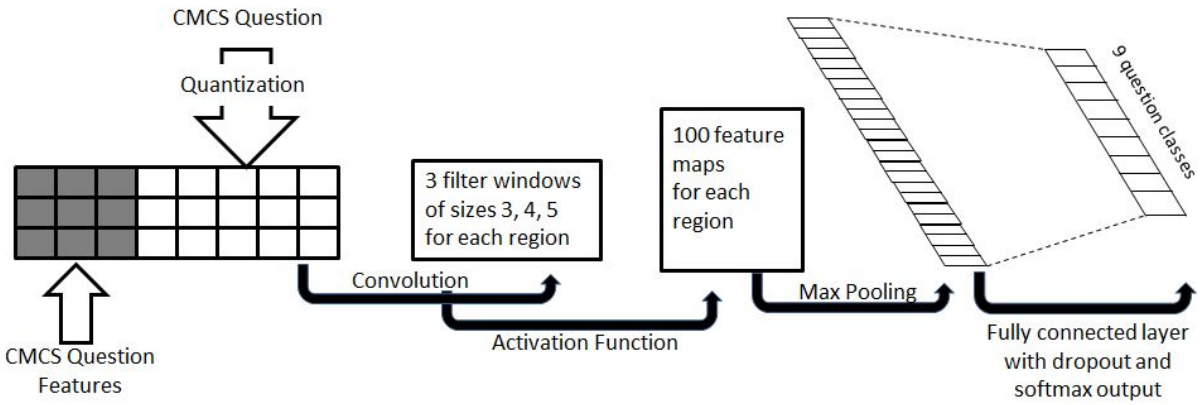

Fig. 1. Model architecture

data. Therefore, the deep learning approach has a high chance of not performing well on the small CMCS dataset.

To circumvent the problem of "too little data for deep learning", we tried integrating features with embeddings. We derived a feature set (cf. Section 7) from the CMCS dataset. We combined these features with the embedding output.

To maintain consistency, the output for each data sample is added after the feature values for each data sample. To understand the effectiveness of coupling feature engineering with deep learning framework, we experimented with two setups: one with feature engineering and another without it.

Convolutional layer: Let $q_i \in \mathbb{R}^k$ be the k-dimensional vector corresponding to the i-th word in the CMCS question. A CMCS question is represented as $q_{1:n} = q_1 \oplus q_2 \oplus ... \oplus q_n$, where, the CMCS question contains the words $q_1, q_2, \ldots, q_n$ and $\oplus$ is the concatenation operator.

Also, let $qf_{1:m} = qf_1 \oplus qf_2 \oplus ... \oplus qf_m$ be the feature set for the CMCS question $q_{1:n}$. After combining the feature set $qf_{1:m}$ with the vector representation of the question $q_{1:n}$, the resulting vector is $l_{1:m+n} = qf_{1:m} \oplus q_{1:n}$. Therefore, $l_{1:m+n} = l_1 \oplus l_2 \oplus ... \oplus l_{m+n}$, where either $l_i \in qf_{1:m}$ or $l_i \in q_{1:n}$.

Let $l_{i:i+j}$ refer to the concatenation of $l_i, l_{i+1}, \ldots, l_{i+j}$. In the convolution operation, the filter $w \in \mathbb{R}^{hk}$ is applied to a window of h words to produce new features such as feature s_i is generated from a window of words $l_{i:i+h1}$ by $s_i = f(w.l_{i:i+h-1} + b)$, where, $b \in \mathbb{R}$ is a bias term and f is a non-linear function. A feature map $s = [s_1, s_2, \ldots, s_{nh+1}]$ (where, $s \in \mathbb{R}^{n-h+1}$) is produced by applying the aforesaid filter to each possible window of h words (i.e., $\{l_{1:h}, l_{2:h+1}, \ldots, l_{nh+1:n}\}$) in the question. Max-pooling operation is applied to the feature map s to obtain the maximum value $s' = \max\{s\}$ for the particular filter. The objective of the max pooling is to capture the most important feature with the highest value for each feature map. Thus, one feature is extracted from one filter. However, the proposed architecture uses multiple filters with varying window sizes to obtain multiple features. Then, these features are passed to the next layer, i.e., a fully-connected layer.

Fully-connected layer: The fully-connected layer is also known as the dense layer. In this layer, all the resulting features, which are extracted using the max-pooling operation, are combined. Hence, the fully-connected layer combines the most of the useful features and constructs a hierarchical representation for the final stage, i.e., the output layer.

Output layer: The final layer (i.e., the output layer) consists of 9 neurons because we have 9 target question classes. The output layer uses 'softmax' as the nonlinear activation function.

7. Feature Engineering

Usually, in deep learning architecture, the underlined system derives the features for the task. However, in this proposed architecture for code-mixed cross-script question classification, we provide a feature set along with data for training. Our experimental results confirm that feeding explicit linguistic feature set separately enhances the CMCS question classification accuracy. This section describes all the features em-

ployed for this study. We employed lexical and semantic features in this study. 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. On the other hand, semantic features can be extracted based on the semantics of the words in a question. In this study, language of the token, related word and named entities were used as semantic features.

Interrogative word: In question classification study on formal monolingual data, one of the important lexical features is the interrogative word in the question (often referred to as *wh*-word). For example, 'who' is the wh-word of the question "who is the president of US?". In the formal QC task, experimental studies [20,19] confirmed that considering questions' wh-words as a feature improves the performance of the QC task. Moreover, wh-words provide an important cue to determine the question type, e.g., 'Person' question class has a high chance to contain the wh-word 'who'. Similarly, questions of type 'Location' contain 'where' as the wh-word. Even though the study in [3] reported 26 interrogatives in Bengali, we found 12 interrogatives in the CMCS corpus. It is worth mentioning that we did not find any dual interrogative and compound interrogative in the CMCS corpus. One of the main reasons behind this is that the CMCS questions are less complex and short in nature compared to formal questions.

Wh-word position: Usually, the interrogative word or wh-word of a question in English appears at the beginning of the question text. But in case of Indian languages, the position of the wh-word is not fixed and can appear in all three potential positions - i.e., at beginning, in the middle and at end. This is because of the relatively free word-order nature of the Indian languages. Therefore, we considered the position of the wh-word as another lexical feature. The value of this feature is set according to the position of the interrogative in the given question - namely first, middle, last. Examples are given below:
Case-1: *Koto* travel tax pore India border e?
Case-2: Semi-official guide *koto* taka charge nei?
Case-3: Kolkata theke bishnupur er distance *koto*?

Language of the token: In the CMCS content scenario, language of the word is a key feature for text processing task. We employed automatic language identifier described in [7] which achieved the best identification accuracy for Bengali-English CMCS data in the shared task on Transliterated Search [14]. The language identifier proposed in [7] reported an accuracy of 92.88% and tagged the words with the tagset {English, Bengali, Others}.

Alphanumeric: In social media content, users often express legitimate vocabulary words in alphanumeric form for saving typing effort or to express their style. Examples include abbreviated words like 'gr8' ('great'), 'b4' ('before'), etc. If a token is alphanumeric then the feature value is set to 1, and 0 otherwise.

Capitalization: We classified this feature into four specialized cases: entire word is in uppercase, first character of the word is in capital, any intermediate character is in capital, and otherwise. The value of the respective flag is set to 1 if it satisfies the case.

Named Entities: In a good number of (non-CMCS) QC studies [25,13], named entities (NE) were used successfully as a semantic feature. In this study, we employed the NE recognition (NER) system of [10] which was also developed for CMCS dataset and is the only existing NER system for Bengali-English CMCS data. The NER system considers 12 named entity classes to classify the NEs into - namely Person, Location, Temporal, Organization, Quantity, Sports terms, Tourism Event, Transport, Artifact, Distance, Monitory and Miscellaneous. We considered 12 binary valued flags, one flag for each NE. The value of a flag is set to 1, if the respective NE type is present in the given question.

Related word cues: This feature is used as a semantic feature and plays a crucial role to recognize question classes. We identified ten clues after analysing the corpus for each question class. All the clues are used as binary features. These cues are good indicators for recognizing question classes. For example, the words 'distance', 'theke' (English gloss: 'from'), 'dure' (English gloss: 'far'), etc. have high chance of appearing in the question of type distance. Similarly, the currency words 'taka' (English gloss: 'money'), 'npr' (Nepalese currency), 'rs' (Indian currency), etc. often appear in the monitory question type. Also food related words such as 'biriyani', 'coconut', 'fish', 'egg', etc. and transaction related words such as 'fees', 'rate', 'fare', 'price', etc. appear in the monitory question type. While preparing the word-lists, initially a list of the most frequent ten words was prepared for each question class by studying the training corpus. After carefully observing the corpus, we manually prepared a stop word list (for both English and Bengali). Af-

ter removing the stop words from the word list, the frequency of the words were calculated. Words having frequency ≥ 3 are considered as semantic features. It was observed from these word-lists that the word lists are overlapping. For example, the word 'theke' (English gloss: 'from') appears in the distance word-list (frequency 19) as well the location word-list (frequency 9). We keep the overlapping words across the word lists which are considered as semantic features.

8. Experiments and Results

In this section, we discuss the various experiments carried out and present the corresponding evaluation results.

Baseline: We consider the best performing system IINTU [12] of the MSIR shared task on Code-Mixed Cross-Script Question Classification [6] as the baseline. The IINTU system is based on ensemble learning which used three classifiers, namely, Random Forest, One-vs-Rest and k-nearest neighbours and the system achieved 83.33% classification accuracy on the shared task testset.

Model Variants: As discussed in Section 6, we performed experimentation based on two criteria: word vector tuning and feature feeding. For word-vector tuning variants, the word-vector either remains static throughout training or is tuned using backpropagation. For the feature feeding criterion, either the features are

Table 5
Experimental models

		Word Vector	
		Static	Fine-tuned (i.e., non-static)
Feature Feed	No	CNN-S	CNN-nS
	Yes	CNN-FS	CNN-FnS

fed to the model along with the dataset or only the dataset is given as input to the model. Therefore, based on these two criteria, four models were developed (cf. Table 5) which are described below:

- CNN static (CNN-S): CNN based model where the word-vector is kept static throughout training.
- CNN static with feature feed (CNN-FS): CNN combined with feature engineering based model where the word-vector is kept static throughout training.
- CNN non-static (CNN-nS): CNN based model where the word-vector is fine-tuned via backpropagation.
- CNN non-static with feature feed (CNN-FnS): CNN combined with feature engineering based model where the word-vector is fine-tuned via backpropagation.

Hyper-parameters and Training: For all the experiments, we used rectified linear units, filter windows (h) of 3, 4, 5 with 100 feature maps each and mini-batch size of 10. For regularization, we employed dropout on the penultimate layer with a constraint on l2-norms with dropout rate (p) of 0.5 and l2 constraint of 3. These values were chosen via a grid search on the training data set. We do not otherwise perform any dataset specific tuning other than early stopping on development set. We randomly selected 10% of the training data as the development set. Training is done through stochastic gradient descent over shuffled mini-batches with the Adadelta update rule [37].

Results: As discussed in the dataset section, we evaluated the models on the MSIR test set which contains 180 data samples.

The overall CMCS QC performance was measured in terms of accuracy (cf. Table 6) while the class specific performances were measured using precision, recall and F-measure (cf. Table 7). Figure 2 presents a comparison of the class specific performances of the proposed models against the best performing system in the MSIR shared task. It is evident from Table 6 that CNN-S and CNN-nS models perform well behind the majority of the system submissions in the MSIR shared task. However, both the feature feed based CNN models (CNN-FS and CNN-FnS) outperform the best

Table 6
Performance of different approaches

Approach	**Correct**	**Incorrect**	**Accuracy(%)**
NLP-NITMZ	142	38	78.89
AMRITA-CEN-NLP	143	37	79.44
IIT(ISM)D	144	36	80.00
AmritaCEN	145	35	80.56
Anuj	146	34	81.11
BITS_PILANI	146	34	81.11
IINTU	*150*	*30*	*83.33*
CNN-S	*140*	*40*	*77.78*
CNN-nS	*143*	*37*	*79.44*
CNN-FS	*152*	*28*	*84.44*
CNN-FnS	*157*	*23*	*87.22*

Table 7
Class specific model performance (NA: no identification of a class)

		Baseline	CNN-S	CNN-nS	CNN-FS	CNN-FnS
PER	Precision	84.00%	86.36%	76.67%	92.00%	91.67%
	Recall	77.78%	70.37%	85.19%	85.19%	81.48%
	F-Score	80.77%	77.55%	80.70%	88.46%	86.27%
LOC	Precision	84.62%	85.00%	87.50%	84.00%	84.00%
	Recall	95.65%	73.91%	91.30%	91.30%	91.30%
	F-Score	89.80%	79.07%	89.36%	87.50%	87.50%
ORG	Precision	55.88%	44.19%	62.07%	51.52%	65.38%
	Recall	79.17%	79.17%	75.00%	70.83%	70.83%
	F-Score	65.52%	56.72%	67.92%	59.65%	68.00%
NUM	Precision	89.66%	92.86%	74.29%	92.86%	100.00%
	Recall	100.00%	100.00%	100.00%	100.00%	100.00%
	F-Score	94.55%	96.30%	85.25%	96.30%	100.00%
TEMP	Precision	100.00%	96.15%	85.71%	100.00%	89.29%
	Recall	100.00%	100.00%	96.00%	100.00%	100.00%
	F-Score	100.00%	98.04%	90.57%	100.00%	94.34%
MONEY	Precision	81.25%	75.00%	100.00%	93.75%	93.75%
	Recall	81.25%	93.75%	75.00%	93.75%	93.75%
	F-Score	81.25%	83.33%	85.71%	93.75%	93.75%
DIST	Precision	100.00%	100.00%	100.00%	100.00%	95.24%
	Recall	95.24%	71.43%	66.67%	90.48%	95.24%
	F-Score	97.56%	83.33%	80.00%	95.00%	95.24%
OBJ	Precision	80.00%	100.00%	62.50%	66.67%	70.00%
	Recall	40.00%	30.00%	50.00%	60.00%	70.00%
	F-Score	53.33%	46.15%	55.56%	63.16%	70.00%
MISC	Precision	NA	33.33%	NA	NA	100.00%
	Recall	NA	12.50%	NA	NA	50.00%
	F-Score	NA	18.18%	NA	NA	66.67%

performing system in the shared task, IINTU. Earlier AmritaCEN also applied a deep approach on the same dataset and achieved the same accuracies that we obtained with the CNN-S and CNN-nS models. This implies that feature feeding with training data overcomes the loophole of deep learning based models with small dataset. Our approach even outperforms the earlier machine learning based approaches which were applied on the MSIR dataset. The CNN-FnS model correctly identifies the question classes for 157 out of 180 questions in total whereas the IINTU systems classified 150 questions correctly. The F-score for all the classes are above 85% except for OBG, ORG and MISC. This phenomenon can be attributed to the fact that there are many instances in the MSIR dataset where the 'OBG' class is overlapping with the 'ORG' class, which creats a confusion between these two classes.

For example, "*bengal r sobcheye boro mosjid ki?*" (English gloss: Which is the largest mosque of Bengal?) and "*Hazarduari er opposit e kon masjid ache?*" (English gloss: Which mosque is located at the opposite of Hazarduari?). The earlier best system also achieved only 65% F-score for 'ORG' class. Our system performs slightly better on the 'ORG' class and achieves 68% F-score. However, it outperforms the IINTU system on the 'OBG' class with a good margin (17%). In the MSIR training dataset, only 5 are of type 'MISC' class out of 330 data samples. Therefore, earlier systems struggled to deal with the 'MISC' class and most of the systems were unable to detect it. The

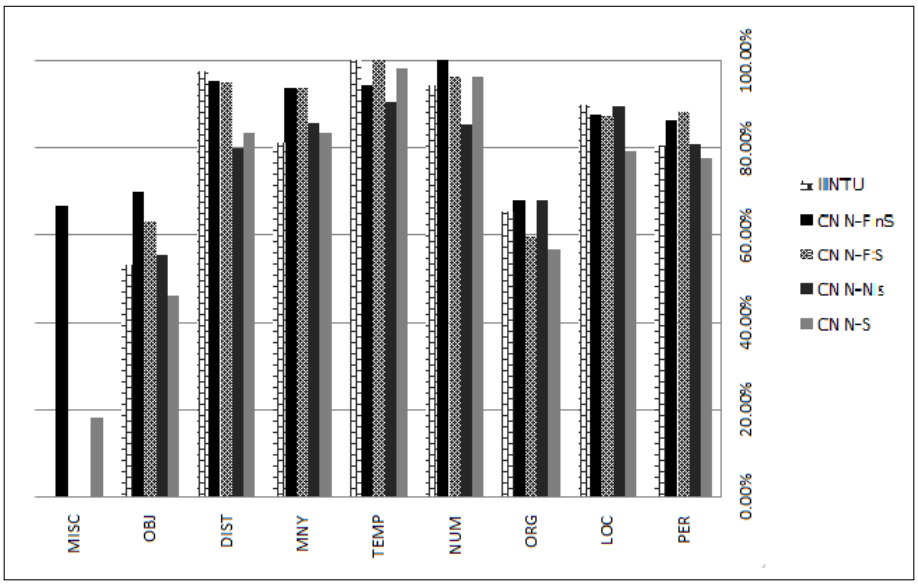

Fig. 2. Class specific performance

CNN-FnS model correctly identified 4 MISC instances out of 8 in the test set while the IINTU system failed to detect any of the 'MISC' instances correctly. Our system outperforms the IINTU system with respect to all the question classes except for the 'LOC' class. The 'LOC' class comprises only 8% of the training data. A comparison of the four models reveals that our features did not work well for questions of type 'LOC'.

9. Conclusions

With the recent emergence of social media, netigens are publishing contents on diverse topics more than ever before. Over the last few years, NLP research has witnessed a lot of emphasis on the use of user-generated SM content. One such initiative is [9] who proposed QA on code-mised cross-script user-generated SMC. In this paper, we studied the question classification task on Bengali-English CMCS data by employing a CNN based deep learning framework. Although deep learning frameworks generally do not rely on feature engineering, we combined together feature engineering and deep learning framework. We conducted experiments on the MSIR-16@FIRE shared task dataset which is the only CMCS QA dataset available. Our experimental results demonstrate that the proposed models outperform the best results reported so far on this dataset and provide the new state-of-the-art results. Collectively, our studies outlines the critical role of feature engineering in deep learning framework.

As future work, we would like to further evaluate the proposed models on other QC datasets available. We would also like to investigate the efficacy of the proposed feature engineering in deep learning approach on other NLP tasks. With respect to the structural perspective, we plan to employ RNN based mechanism along with feature engineering. The only available CMCS dataset for QA research contains only 510 data samples. In future, we would also like to contribute towards building a large CMCS QA dataset.

Acknowledgments

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

The work of the third author was partially supported by the SomEMBED TIN2015-71147-C2-1-P MINECO research project.

References

[1] Umair Z Ahmed, Kalika Bali, Monojit Choudhury, and Sowmya VB. Challenges in Designing Input Method Editors for Indian languages: The Role of Word-origin and Context. *Advances in Text Input Methods (WTIM 2011)*, pages 1–9, 2011.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] 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.

[4] Somnath Banerjee and Sivaji Bandyopadhyay. An Empirical Study of Combining Multiple Models in Bengali Question Classification. In *Proceedings of International Joint Conference on Natural Language Processing (IJCNLP), Japan*, pages 892–896, 2013.

[5] Somnath Banerjee and Sivaji Bandyopadhyay. Ensemble Approach for Fine-Grained Question Classification in Bengali. In *Proceedings of the 27th Pacific Asia Conference on Language, Information and Computation*. Department of English, National Chengchi University, 2013.

[6] Somnath Banerjee, Kunal Chakma, Sudip Kumar Naskar, Amitava Das, Paolo Rosso, Sivaji Bandyopadhyay, and Monojit Choudhury. Overview of the Mixed Script Information Retrieval (MSIR) at FIRE. In *Working Notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, CEUR Workshop Proceedings. CEUR-WS.org, 2016.

[7] Somnath Banerjee, Alapan Kuila, Aniruddha Roy, Sudip Kumar Naskar, Paolo Rosso, and Sivaji Bandyopadhyay. A Hybrid Approach for Transliterated Word-level Language Identification: CRF with Post-processing Heuristics. In *Proceedings of the Forum for Information Retrieval Evaluation*, pages 54–59. ACM, 2014.

[8] Somnath Banerjee, Sudip Kumar Naskar, and Sivaji Bandyopadhyay. BFQA: A Bengali Factoid Question Answering System. In *International Conference on Text, Speech and Dialogue (TSD)*, pages 217–224. Springer, 2014.

[9] Somnath Banerjee, Sudip Kumar Naskar, Paolo Rosso, and Sivaji Bandyopadhyay. The First Cross-Script Code-Mixed Question Answering Corpus. *Proceedings of the workshop on Modeling, Learning and Mining for Cross/Multilinguality (MultiLingMine 2016), co-located with The 38th European Conference on Information Retrieval (ECIR)*, 2016.

[10] Somnath Banerjee, Sudip Kumar Naskar, Paolo Rosso, and Sivaji Bandyopadhyay. Named Entity Recognition on Code-Mixed Cross-Script Social Media Content. In *The 18th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing), Budapest, Hungary.*, 2017.

[11] Rupal Bhargava, Shubham Khandelwal, Akshit Bhatia, and Yashvardhan Sharmai. Modeling Classifier for Code Mixed Cross Script Questions. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, CEUR Workshop Proceedings. CEUR-WS.org, 2016.

[12] Debjyoti Bhattacharjee and Paheli Bhattacharya. Ensemble Classifier Based Approach for Code-Mixed Cross-Script Question Classification. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, CEUR Workshop Proceedings. CEUR-WS.org, 2016.

[13] Phil Blunsom, Krystle Kocik, and James R Curran. Question Classification with Log-linear Models. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 615–616. ACM, 2006.

[14] Monojit Choudhury, Gokul Chittaranjan Gupta, Parth Gupta, and Amitava Das. Overview of FIRE 2014 Track on Translitterated Search. In *Forum for Information Retrieval Evaluation (FIRE)*, 2014.

[15] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural Language Processing (almost) from Scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[16] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech Recognition with Deep Recurrent Neural Networks. In *Acoustics, Speech and Signal Processing (ICASSP)*, pages 6645–6649. IEEE, 2013.

[17] Bert F Green Jr, Alice K Wolf, Carol Chomsky, and Kenneth Laughery. Baseball: an Automatic Question-Answerer. In *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference*, pages 219–224. ACM, 1961.

[18] Barathi Ganesh HB, Anand Kumar M, and Soman KP. Distributional Semantic Representation for Text Classification. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, CEUR Workshop Proceedings. CEUR-WS.org, 2016.

[19] Zhiheng Huang, Marcus Thint, and Asli Celikyilmaz. Investigation of Question Classifier in Question Answering. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP): Volume 2-Volume 2*, pages 543–550. Association for Computational Linguistics, 2009.

[20] Zhiheng Huang, Marcus Thint, and Zengchang Qin. Question Classification using Head Words and their Hypernyms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 927–936. Association for Computational Linguistics, 2008.

[21] David A Hull. Xerox TREC-8 Question Answering Track Report. In *TREC*, 1999.

[22] Mohit Iyyer, Jordan L Boyd-Graber, Leonardo Max Batista Claudino, Richard Socher, and Hal Daumé III. A Neural Network for Factoid Question Answering over Paragraphs. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 633–644, 2014.

[23] Rie Johnson and Tong Zhang. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. In *Association for Computational Linguistics (ACL)*, 2015.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[25] Xin Li and Dan Roth. Learning Question Classifiers: the Role of Semantic Information. *Natural Language Engineering*, 12(03):229–249, 2006.

[26] Babak Loni. A Survey of State-of-the-art Methods on Question Classification. *Delft University of Technology, Tech. Rep*, 2011.

[27] Anand Kumar M and Soman K P. Amrita-CEN@MSIR-FIRE2016: Code-Mixed Question Classification using BoWs and RNN Embeddings. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, CEUR Workshop Proceedings. CEUR-WS.org, 2016.

[28] Goutam Majumder and Partha Pakray. NLP-NITMZ @ MSIR 2016 System for Code-Mixed Cross-Script Question Classi-

fication. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, CEUR Workshop Proceedings. CEUR-WS.org, 2016.

[29] Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. Exploiting Syntactic and Shallow Semantic Kernels for Question Answer Classification. In *Association For Computational Linguistics*, volume 45, page 776, 2007.

[30] 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.

[31] Xipeng Qiu and Xuanjing Huang. Convolutional Neural Tensor Network Architecture for Community-Based Question Answering. In *IJCAI*, pages 1305–1311, 2015.

[32] Alexander M Rush, SEAS Harvard, Sumit Chopra, and Jason Weston. A Neural Attention Model for Sentence Summarization. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[33] Anuj Saini. Code Mixed Cross Script Question Classification. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, CEUR Workshop Proceedings. CEUR-WS.org, 2016.

[34] Fabrizio Sebastiani. Machine Learning in Automated Text Categorization. *ACM Computing Surveys (CSUR)*, 34(1):1–47, 2002.

[35] 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.

[36] Duyu Tang, Bing Qin, and Ting Liu. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1422–1432, 2015.

[37] Matthew D Zeiler. ADADELTA: an Adaptive Learning Rate Method. *arXiv preprint arXiv:1212.5701*, 2012.

[38] Dell Zhang and Wee Sun Lee. Question Classification using Support Vector Machines. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 26–32. ACM, 2003.