Ensemble Classifier based approach for Code-Mixed Cross-Script Question Classification

[Team IINTU]

Debjyoti Bhattacharjee
School of Computer Science and Engineering
Nanyang Technological University
Singapore
debjyoti001@ntu.edu.sg

Paheli Bhattacharya
Dept. of Computer Science and Engineering
Indian Institute of Technology Kharagpur
West Bengal, India
paheli@iitkgp.ac.in

ABSTRACT

With an increasing popularity of social-media, people post updates that aid other users in finding answers to their questions. Most of the user-generated data on social-media are in code-mixed or multi-script form, where the words are represented phonetically in a non-native script. We address the problem of Question-Classfication on social-media data. We propose an ensemble classifier based approach towards question classification when the questions are written in mixed-script, specifically, the Roman script for the Bengali language. We separately train Random Forests, One-Vs-Rest and k-NN classifiers and then build an ensemble classifier that combines the best from the three worlds. We achieve an accuracy of 82% approximately, suggesting that the method works well in the task.

CCS Concepts

•Information systems → Question answering; •Computing methodologies → *Machine learning;* Cross-validation;

Keywords

Mixed Script Information Retrieval; Question Answering System; Question classification

1. INTRODUCTION

With the increase in popularity of the Web, users from all over the world now opt to write in their native language instead of English. A large number of South and South-East Asian languages are written in a transliterated form (phonetically representing words in a non-native script) using the Roman Script. These texts are said to be written in Mixed-Script. Since there are font-encoding issues in using the original script (for example, Devnagari for Hindi) people tend to transliterate or phonetically represent the words in the original language using the Roman script. To define Mixed Script Information Retrieval formally [6] we consider a set of natural languages $L = \{l_1, l_2, \cdots, l_n\}$ and a set of scripts $S = \{s_1, s_2, \cdots, s_n\}$ such that s_i is the native script for the language l_i. Given a word w, we represent it as two tuples $\langle l_i, s_j \rangle$ to imply w is in language l_i and written using script s_j. When $i = j$, we say that the word is written in its native script. Else, it has been transliterated into another script s_j. In practice, when textual content is a mixture of words from various languages or scripts or both, it is called Multi-Script (MS) or Code-Mixing. For instance, *"Kharagpur theke Howrah cab fare koto?"* (gloss : What is the cab fare from Kharagpur to Howrah?) has words from a single script (Roman) but from two different languages - English (cab , fare) and Bengali (*theke* , *koto*). The words in Bengali have been transliterated to the Roman Script. Intuitively this a very easy form of writing for people not as well-versed with English as for their native language tend to use them for conversing in social media. With the rise in popularity of social-media, people constantly post updates from their daily lives ranging from, but not limited to, sports and score updates, travel updates to food, hotel, transport and movie reviews, providing user feedbacks for Customer Support Systems through tweets and blogs. Although Question Answering (QA) is a well-addressed research problem with systems providing reasonable accuracy, QA on social-media text in mixed script is a challenging problem mainly due to the fact that there is no standardization of spellings for words written in non-native script. For instance, the Bengali word *"ekhon"* (meaning, now) may have multiple spellings − *"akhan"*, *"ekhon"*, *"ekhan"*, *"akon"* etc. Categorizing a question into a specific set of classes and then dealing with each class separately is an efficient method for QA systems. This is called Question Classification. Question classification aids in reducing the number of candidate answers and also can be used in effectively determining answer selection strategies [8]. In this paper, we deal with the problem of question classification for Multi-Script or code-mixed data. We have experimented with three machine learning based classifiers - Random Forests, One-vs-Rest and k-NN and then built an ensemble of these classifiers to achieve an higher accuracy.

2. RELATED WORK

Jamatia et al. [7] experiments with code-mixed English-Hindi social-media text for Part-of-Speech tagging. They use both coarse and fine-grained tagsets for the task. Four machine learning algorithms − Conditional Random Fields, Sequential Minimal Optimization, Naïve Bayes and Random Forests, reporting highest accuracy with Random Forest based classifier. Information Retrieval on Multi-Script data has also been looked into [6]. Recent works on question classification include a machine learning based approach [8] towards question class. A hierarchial classifier is first used to classify the question into coarse-grained classes and then into fine-grained classes. The feature space consisted of primitive ones like pos tags, chunks, named entities and also complex features such as conjunctive n-gram features and relational features. Question-Answering corpus acquisition using social-media content and question acquisition with human involvement have been reported in [2]. In FIRE 2015, the Transliterated Search track introduced three subtasks — language labelling of words in code-mixed text fragments, ad-hoc retrieval of Hindi film lyrics, movie reviews and astrology documents and transliterated question answering where the documents as well as questions were in Bangla script or Roman transliterated Bangla [4].

3. TASK DESCRIPTION

Question Answering systems are a classic application of natural language processing, where the retrieval task has to find a concise and accurate answer to a given question. Question classification is one of the subtasks of QA system, required to determine the type of the answer corresponding to a question.

The Code-Mixed Cross-Script Question Classification task can be described as follows. Given a question Q written in Romanized Bengali, which can contain English words and phrases and a set $C = \{c_1, c_2, \ldots, c_n\}$ of question classes, the task is to classify the question Q into one of these predefined classes.

Example:

Question: airport theke howrah station distance koto ?

Question Class: DIST

3.1 Dataset description

The training dataset consists of 330 questions and each question is assigned to a single question class. There are 9 question classes in all and the number of questions in each class is shown in Table 1. The minimum and maximum number of words in a question is 2 and 11 respectively while each question on average has 5.3 words.

Table 1: Dataset classes and #Q per class

Class	#Q
DIST	24
LOC	26
MISC	5
MNY	26
NUM	45
OBJ	21
ORG	67
PER	55
TEMP	61
Total	330

4. PROPOSED APPROACH

To build a classifier to classify the questions into the specified classes, we created a vector representation of the each question which is used as input to the classifier. We considered the top 2000 most frequently occurring words in the supplied training dataset as features. Each question is represented as a 2000-element binary vector. Element $e_i = 1$, if the i^{th} most frequently word is present in the question, otherwise 0.

We used three separate classifiers namely Random Forests (RF), One-vs-Rest (OvR) classifier and k-Nearest Neighbour (k-NN) classifier, followed by building an ensemble classifier using these three classifiers for the classification task.

In k-NN classification, a sample is classified by a majority vote of its neighbours, with the object being assigned to the most common class among its k-nearest neighbours ($k > 0, k \in I$). k-NN classification is a lazy learning method which defers computation till the classification is performed. k-NN is one of the simplest classifiers.

One-vs–Rest strategy uses one classifier per class for fitting. Each classifier is trained against all the classes. The approach allows information regarding each class by inspecting the classifier trained for that class. In OvR, each classifier is trained with the entire data set while in RF, samples drawn from the original data set are used for training.

A Random Forest is a ensemble learning method which can be classification [3]. Random Forest fits a number of decision trees on various sub-samples of the dataset, with the samples drawn from the original dataset with or without replacement. Random Forests overcome the problem of overfitting of decision tree of their training set [5].

Using the above three classifiers, we built an ensemble classifier (EC). The ensemble classifier takes the output label by each of the individual classifiers and gives the majority label as output, otherwise any label is chosen at random as output. Each of the individual classifiers is trained on a subset of the original training dataset, by sampling with replacement.

In the following section, we describe the details of implementation of the classifiers and the obtained results.

5. EXPERIMENTAL SETUP AND RESULTS

We implemented the proposed approach using Python 3 and used the *scikit-learn* tool-kit for the classifiers. The following instantiations were used for the first three classifiers, which were available in *sckit-learn*. We implemented the ensemble classifier on our own.

```
rf = RandomForestClassifier(n_estimators=100)
ovr = OneVsRestClassifier(LinearSVC(random_state=0))
clf = neighbors.KNeighborsClassifier(30, weights=’uniform’)
```

We split the labelled data set into two parts — training set (90%) and validation set (10%). The RF classifier performed the best, followed by EC, OvR and k-NN in decreasing order of classification accuracy. Thereafter, we used these trained classifiers for classifying the test data set. During classification, we marked the samples for which all the 4-classifiers predicted the same label. We used these samples, in addition to the original labelled data set for retraining the classifiers.

The results on the test data set for the classifiers RF, EC and OvR were submitted as final run and is shown summarily in Table 2 and Table 3. The classification results of kNN classifier were not submitted as run and hence accuracy of the results is not available.

6. CONCLUSION AND FUTURE WORK

In this paper we have addressed the problem of question classification for Bengali-English Code-Mixed social-media data. We have experimented with three machine learning based classifiers - Random Forests, One-vs-Rest and k-NN and then built an ensemble of these classifiers to achieve the best results. The method is scalable to other Code-Mixed languages mainly because it does not perform any language

Table 2: Results of individual classes

Classifier	Class	I	IC	P	R	F-1
EC	PER	24	20	0.83	0.74	0.78
RF		25	21	0.84	0.77	0.80
OvR		23	19	0.82	0.70	0.76
EC	LOC	26	21	0.80	0.91	0.85
RF		26	22	0.84	0.95	0.89
OvR		26	21	0.80	0.91	0.85
EC	ORG	36	19	0.52	0.79	0.63
RF		34	19	0.55	0.79	0.65
OvR		40	19	0.47	0.79	0.59
EC	NUM	30	26	0.86	1	0.92
RF		29	26	0.89	1	0.94
OvR		29	26	0.89	1	0.94
EC	TEMP	25	25	1	1	1
RF		25	25	1	1	1
OvR		25	25	1	1	1
EC	MONEY	16	13	0.81	0.81	0.81
RF		16	13	0.81	0.81	0.81
OvR		12	12	1	0.75	0.85
EC	DIST	20	20	1	0.95	0.97
RF		20	20	1	0.95	0.97
OvR		22	21	0.95	1	0.97
EC	OBJ	3	3	1	0.3	0.46
RF		5	4	0.8	0.4	0.53
OvR		3	3	1	0.3	0.46
EC	MISC	0	0	NA	NA	NA
RF		0	0	NA	NA	NA
OvR		0	0	NA	NA	NA

Table 3: Overall Results

Classifier	Correct	Incorrect	Accuracy
EC	147	33	81.66
RF	150	30	83.33
OvR	146	34	81.11

or script-based feature engineering.

We would like to experiment with other multi-script data where more than two languages have been mixed. We aim to apply other machine learning algorithms with more linguistic and syntactic features.

7. REFERENCES

[1] S. Banerjee, K. Chakma, S. K. Naskar, A. Das, P. Rosso, S. Bandyopadhyay, and M. 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.

[2] S. Banerjee, S. K. Naskar, P. Rosso, and S. Bandyopadhyay. The first cross-script code-mixed question answering corpus. In *Modelling, Learning and mining for Cross/Multilinguality Workshop, 38th European Conference on Information Retrieval (ECIR)*, pages 56–65, 2016.

[3] Y. Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[4] M. Choudhury, P. Gupta, P. Rosso, S. Kumar, S. Banerjee, S. K. Naskar, S. Bandyopadhyay, G. Chittaranjan, A. Das, and K. Chakma. Overview of fire-2015 shared task on mixed script information retrieval.

[5] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[6] P. Gupta, K. Bali, R. E. Banchs, M. Choudhury, and P. Rosso. Query expansion for mixed-script information retrieval. In *The 37th Annual ACM SIGIR Conference*, pages 677–686, 2014.

[7] A. Jamatia, B. Gambäck, and A. Das. Part-of-speech tagging for code-mixed english-hindi twitter and facebook chat messages. In *10th Recent Advances of Natural Language Processing (RANLP)*, pages 239–248, 2015.

[8] X. Li and D. Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.