Code Mixed Cross Script Question Classification

Anuj Saini
Sapient Global Markets
Gurgaon, Haryana
asaini13@sapient.com

ABSTRACT

With the growth in our society, one of the most affected aspect of our routine life is language. We tend to mix our conversations in more than one language, often mixing up regional language with English language is a lot more common practice. This mixing of languages is referred as code mixing, where we mix different linguistic constituents such as phrases, proper nouns, morphemes etc. to come up code mixed script. With exponential growth of social media, we are using more and more code mixed cross script for our conversation on Facebook, WhatsApp, or Twitter. On the other hand, the language should be understood by the automated question answering system which is one of the most import application of AI. And now the trend is code mixed languages but current work is around a single language. At FIRE 2016, as a part of Shared Task1 CMCS (Code Mixed Cross Script Question Classification), we have worked on the problem of classify a code mixed question into 9 given classes. Shared Task is focused on Indian regional languages, wherein we worked on Bengali-English code mixed cross script questions classification. As scripting used in training data is English only, so all Bengali text was also written using English script only. We have used Machine Learning for question classification and used ensemble based Random Forest algorithm. As it's a code-mixed script, so traditional NLP components may not work well, so worked on a custom solution using own set of features for Classification.

CCS Concepts

• Theory of computation →Random Forest
• Computing methodologies→Natural language Processing

Keywords

Question Answering; Machine Learning; Code-Mixing; Code switching; Classification; Random Forest; TFIDF; Stemming; Natural Language Processing.

1. INTRODUCTION

According to Census of India of 2001, India has a total of 122 major languages and 1599 other languages [1]. And with the advancement of technologies and social media, languages are now getting mixed with other languages. A big chunk of population of India is bilingual or even trilingual wherein Hindi and English along with Dravidian Languages are most spoken languages. Code mixing is defined as mixing of more than one language syntactically, wherein linguistic constituents such as syntax, morphology, words/phrases etc. are mixed [2] [3].

Social media such as WhatsApp, Facebook[8] etc. are widely used by most of the audience using mobile phone or smart devices where communication is more often mix of English and regional language. Even though autocorrect provide by most of these devices is in English only. So it's very common as well as convenient for users to communicate in code mixed which is easy and fast to use. Eventually English along with regional language has becomes an integral part of communication for most of social media users. There are many words which people use in English only *aapka mobile number kya hai*, where no one actually knows what is corresponding Hindi word to mobile. Even google also supports code mixed queries and is able to translate and return results.

But one of the next generation systems, automated question answering systems, are still mostly catering one language such as English or French. One of the first building blocks for any question answering system is to understand and classify question. A question could be of many types such as When, Where, What, Why etc. So it's important to classify question what kind of answer is a question looking for. This shared task is about classifying question in one of these classes. This paper describes our approach for the subtask 1 in the shared task on Mixed Script Information Retrieval [9] in FIRE-2016. We have used basic text preprocessing such Tokenizer, n-grmas, along with and CountVector and TFIDF vector to generate features and then fed training data of vectors into Machine Learning algorithms such as Naïve Bayes, SVM, and Random Forest etc.

2. RELATED WORK

A lot of similar work has been done in the past on mixed languages and question classification on various sources. Khyati and Manoj [5] did classification of Hindi-English mixed languages questions. They used SVM based classification with text preprocessing using transliteration and translation and then using text features. Language Identification [6] is also another related preprocessing part of this system where one needs to know language of individual word for better results.

3. DATA SET DESCRIPTION

There were a total of 330 data points in the training set containing code mixed text of Bengali and English with 9 question classes to predict from. Each class describes type of answer it is expecting for subTask1. A detail distribution of classes and its count for CMCS task is mentioned in Table1.

Table 1. Classes & its count for SubTask1

Class	**Count**
MNY	26
TEMP	61
DIST	24
LOC	26

ORG	67
MISC	5
OBJ	21
NUM	45
PER	55

Each data point contains a code mixed question and a corresponding class in training data. Organizers have also provided a test dataset for evaluation purpose which had only code mixed questions for which our system have to predict class having a total of 181 questions.

4. FEATURE GENERATION

As our solution is machine learning based, so we need to convert out text into some vectors to train our model. Before converting our text into vectors, we have applied some basic preprocessing on the raw code mixed text using NLP based custom pipeline.

4.1 Text Preprocessing

We have used NLTK tokenizer on all questions to tokenize sentences into tokens.

Further, we have also generated bigrams of tokens, as bigrams are important to identify some important phrases to be used for feature generation such as *koto run*.

Also we analyzed that proper nouns entities such as person names, locations names were causing a lot of noise, so we have identified proper nouns and replaced them with XX.

We did not apply any stemming and stop words removal knowingly, as we analyzed and tested that stop words like koto, where, and who etc. are very important features for this classification task. So we kept all the terms in our vocabulary along with new set of bigram words.

.

Table 2. Preprocessing on Code Mixed Script

PreProcess	Input	Preprocessed
Tokenization	prepaid taxi kokhon chalu hoi	[prepaid, taxi, kokhon, chalu, hoi]
bigrams	prepaid taxi kokhon chalu hoi	[prepaid taxi, taxi kokhon, kokhon chalu, chalu hoi]
Proper Nouns Replacements	Hazarduari te koto dorja ache	XX te koto dorja ache

4.2 Text Vectors (Features)

A set of text features has been generated on word and bigrams level using text to vector conversions. Each feature is assigned a corresponding numeric value to train model.

4.2.1 Count Vectorizer

Count vector simply counts the frequency of each token (unigrams and bigrams here) on Code Mixed questions corpus. This vector produces a sparse representation of the counts and number of features produced will be equal to the vocabulary size provided. As we did not filter out any word except replacement of proper nouns, so in total, 1030 features have been generated by this vector. Out of which as many as 667 terms had a frequency of 1, so we discarded those features and hence we are left with 363 features to be used. This is also called Term frequency and is denoted as below

$$\mathbf{tf}(\boldsymbol{t},\boldsymbol{d}) = \boldsymbol{f}_{t,d}$$

Where t is term in document d.

A snapshot of features generated by this vectorizer is mentioned in Figure 1.

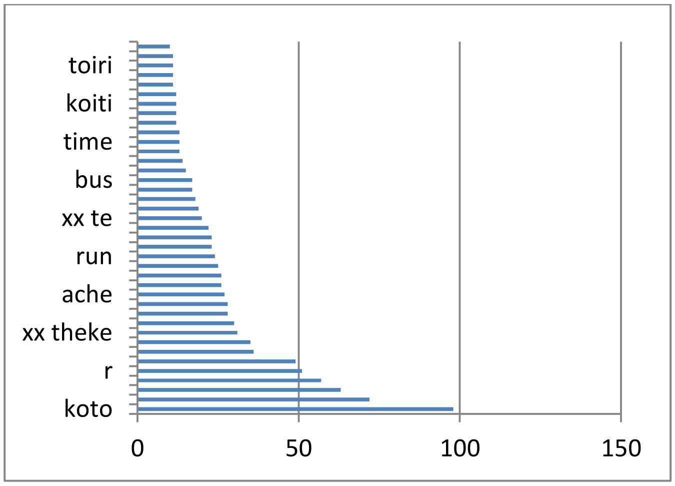

Figure 1. Term Features by Count Vectorizer

4.2.2 TFIDF Vectorizer

This vectorizer takes in all code mixed raw text and generate tf-idf of all the terms to get importance of all the words and bigrams. Tfidf is very useful feature generation model in text applications and is denoted as follows.

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t.$$

Where idf is inverse document frequency and is denoted as.

$$\text{idf}_t = \log \frac{N}{\text{df}_t}.$$

We have only used terms having term frequency more than 1 and hence finally produced 163 features by this vectorizer having tf-idf values for each term. A snapshot of features generated by tfidf vectorizer is mentioned in Figure 2.

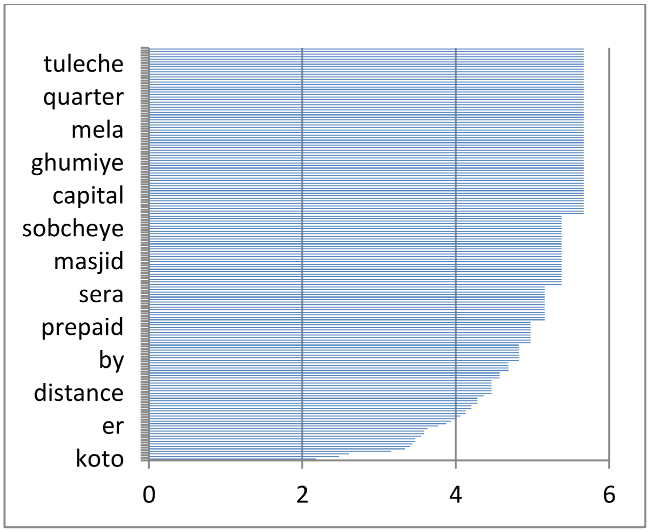

Figure 2. Term Features by TFIDF Vectorizer

Overall we have generated ~500 features using both these vectors without removing any term as we found stop words relevant for this task.

5. CLASSIFICATION

Text vectors as generated in section 3.2 has been used as training data to train classifiers using Python Scikit. A number of different algorithms with different parameters have been tested before coming up with the best algorithm and its parameters. Support Vector Machines (SVM), Logistic Regression, Random Forest, Gradient Boosting have been tested using Grid Search to come up with best parameters and model. Finally, Random forest with 100 n_estimators, max_depth of 10, min_samples_leaf of 4 and min_samples_split of 4, were identified as the best set of parameters to be used with Random Forest. Overall training time for model is less than 1 second on quad-core Machine with 8GB of RAM.

6. RESULTS

We have used 10 fold cross validation to compute overall accuracy for the system. For the subtask 1 CMCS we have got overall accuracy of 0.8495 with F1 score of 0.84. Detailed performance matrix of the model is given as below in Table 3 for all the classes.

Table 3. Subtask 1 Scores Summary on Train Data

Class	Precision	Recall	F1-Score	Support
DIST	0.87	0.83	0.85	24
LOC	0.95	0.87	0.91	23
MISC	0	0	0	5
MNY	0.82	0.88	0.85	26
NUM	0.93	0.89	0.91	45
OBJ	0.58	0.71	0.64	21
ORG	0.76	0.72	0.74	61
PER	0.85	0.91	0.88	55
TEMP	0.93	0.97	0.95	59
avg / total	0.83	0.84	0.84	319

Location, temporal and numeric questions were best classified classes and model failed to predict miscellaneous classes.

We have submitted three submissions on test data and best accuracy score we have received is 81.11111 which is 2nd amongst all teams. Detailed performance matrix on test data is given in Table 4.

Table 4. Subtask 1 Scores Summary on Test Data

Class	Precision	Recall	F1-Score	Support
DIST	0.87	0.83	0.85	24
LOC	0.95	0.87	0.91	23
MISC	0	0	0	5
MNY	0.82	0.88	0.85	26
NUM	0.93	0.89	0.91	45
OBJ	0.58	0.71	0.64	21
ORG	0.76	0.72	0.74	61
PER	0.85	0.91	0.88	55
TEMP	0.93	0.97	0.95	59
avg / total	0.83	0.84	0.84	319

7. CONCLUSION AND FUTURE SCOPE

In this paper, we have presented our approach for question classification on code mixing. Question classification is a first step towards building a question answering system. Moreover work which has been done in code mixed languages is mostly in Hindi-English [5] language in India. And as most of the current work in this domain is around single language which is unrealistic in countries like India where most of communication on social media is code mixed. This shared task is a milestone step towards building such realistic applications for future. Future scope of Information Retrieval [7] systems is going to be question answering systems where system could understand question. We have used very basic set of features here by simply calculating importance of words/phrases within corpus itself. Using part of speech tag which is un-touched area in our solution could definitely add a lot more to the solution. Also using regional lexical dictionaries like WordNet, e.g. Hindi WordNet for Hindi code mixed problems will surely help a lot to build more sophisticated solution.

8. ACKNOWLEDGMENTS

We would like to thank organizers for conducting this shared task and also building the training data. We also would like to thank Sapient Corporation for giving us an opportunity to work and explore the world of text analytics.

9. REFERENCES

[1] Vijayanunni, M. (26–29 August 1998). "Planning for the 2001 Census of India based on the 1991 Census"(PDF). 18th Population Census Conference. Honolulu, Hawaii, USA: Association of National Census and Statistics Directors of America, Asia, and the Pacific. Archived from the original (PDF) on 19 November 2008. Retrieved 17 December 2014.

[2] Muysken, Pieter. 2000. Bilingual Speech: A Typology of Code-mixing. Cambridge University Press. ISBN 0-521-77168-4

[3] Bokamba, Eyamba G. 1989. Are there syntactic constraints on code-mixing? World Englishes, 8(3), 277-292.

[4] Somnath Banerjee, Sudip Kumar Naskar, Paolo Rosso, and Sivaji Bandyopadhyay. 2016. The First Cross-Script Code-Mixed Question Answering Corpus. In: Modeling, Learning and Mining for Cross/Multilinguality Workshop, 38th European Conference on Information Retrieval (ECIR), pp.56-65.

[5] Khyathi Chandu Raghavi, Manoj Kumar Chinnakotla, and Manish Shrivastava. Answer ka type kya he?: Learning to classify questions in code-mixed language. In Proceedings of the 24th International Conference on World Wide Web Companion, pages 853–858. International World Wide Web Conferences Steering Committee, 2015. 6.

[6] G. Chittaranjan and Y. Vyas. Word-level Language Identification using CRF: Code-switching Shared Task Report of MSR India System. In EMNLP 2014, page 73.

[7] P. Gupta, K. Bali, R. E. Banchs, M. Choudhury, and P. Rosso. Query Expansion for Mixed-Script Information Retrieval. In SIGIR ’14, pages 677–686, ACM, 2014.

[8] T. Hidayat. An analysis of Code Switching used by Facebookers. 2012.

[9] S. Banerjee, K. Chakma, S. K. Naskar, A. Das, P. Rosso, S. Bandyopadhyay, and M. Choudhury. Overview of the Mixed Script Information Retrieval 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.