Overview of the Mixed Script Information Retrieval (MSIR) at FIRE-2016

Somnath Banerjee
Jadavpur University, India
sb.cse.ju@gmail.com

Kunal Chakma
NIT Agartala, India
kchax4377@gmail.com

Sudip Kumar Naskar
Jadavpur University, India
sudip.naskar@cse.jdvu.ac.in

Amitava Das
IIIT Sriharikota, India
amitava.santu@gmail.com

Paolo Rosso
Technical University of Valencia, Spain
prosso@dsic.upv.es

Sivaji Bandyopadhyay
Jadavpur University, India
sbandyopadhyay@cse.jdvu.ac.in

Monojit Choudhury
Microsoft Research india
monojitc@microsoft.com

ABSTRACT

The shared task on Mixed Script Information Retrieval (MSIR) was organized for the fourth year in FIRE-2016. The track had two subtasks. Subtask-1 was on question classification where questions were in code mixed Bengali-English and Bengali was written in transliterated Roman script. Subtask-2 was on ad-hoc retrieval of Hindi film song lyrics, movie reviews and astrology documents, where both the queries and documents were in Hindi either written in Devanagari script or in Roman transliterated form. A total of 33 runs were submitted by 9 participating teams, of which 20 runs were for subtask-1 by 7 teams and 13 runs for subtask-2 by 7 teams. The overview presents a comprehensive report of the subtasks, datasets and performances of the submitted runs.

1. INTRODUCTION

A large number of languages, including Arabic, Russian, and most of the South and South East Asian languages like Bengali, Hindi etc., have their own indigenous scripts. However, the websites and the user generated content (such as tweets and blogs) in these languages are written using Roman script due to various socio-cultural and technological reasons[1]. This process of phonetically representing the words of a language in a non-native script is called transliteration. English being the most popular language of the web, transliteration, especially into the Roman script, is used abundantly on the Web not only for documents, but also for user queries that intend to search for these documents. This situation, where both documents and queries can be in more than one scripts, and the user expectation could be to retrieve documents across scripts is referred to as Mixed Script Information Retrieval.

The MSIR shared task was introduced in 2013 as "Transliterated Search" at FIRE-2013 [15]. Two pilot subtasks on transliterated search were introduced as a part of the FIRE-2013 shared task on MSIR. Subtask-1 was on language identification of the query words and subsequent back transliteration of the Indian language words. The subtask was conducted for three Indian languages - Hindi, Bengali and Gujarati. Subtask-2 was on ad hoc retrieval of Bollywood song lyrics - one of the most common forms of transliterated search that commercial search engines have to tackle. Five teams participated in the shared task.

In FIRE-2014, the scope of subtask-1 was extended to cover three more South Indian languages - Tamil, Kannada and Malayalam. In subtask-2, (a) queries in Devanagari script, and (b) more natural queries with splitting and joining of words, were introduced. More than 15 teams participated in the 2 subtasks [7].

Last year (FIRE-2015), the shared task was renamed from "Transliterated Search" to "Mixed Script Information Retrieval (MSIR)" for aligning it to the framework proposed by [8]. In FIRE-2015, three subtasks were conducted [17]. Subtask-1 was extended further by including more Indic languages, and transliterated text from all the languages were mixed. Subtask-2 was on searching movie dialogues and reviews along with song lyrics. Mixed script question answering (MSQA) was introduced as subtask-3. A total of 10 teams made 24 submissions for subtask-1 and subtask-2. In spite of a significant number of registrations, no run was received in subtask-3.

This year, we hosted two subtasks in the MSIR shared task. Subtask-1 was on classifying code-mixed cross-script question; this task was the continuation of last year's subtask-3. Here Bengali words were written in Roman transliterated Bengali. Here Bengali words were written in Roman transliterated Bengali. The subtask-2 was on information retrieval of Hindi-English code-mixed tweets. The objective of subtask-2 was to retrieve the top k tweets from a corpus [6] for a given query consisting of Hind-English terms where the Hindi terms are written in Roman transliterated form.

This paper provides the overview of the MSIR track in the Eighth Forum for Information Retrieval Conference 2016 (FIRE-2016). The track was coordinated jointly by Microsoft Research India, Jadavpur University, Technical University of Valencia, IIIT Sriharikota and NIT Agartala. Details of these tasks can also be found on the website https://msir2016.github.io/.

The rest of the paper is organized as follows. Section 2 and 3 describe the datasets, present and analyze the run submissions for the Subtask-1 and Subtask-2 respectively. We conclude with a summary in Section 4.

2. SUBTASK-1: CODE-MIXED CROSS-SCRIPT QUESTION ANSWERING

Being a classic application of natural language processing, question answering (QA) has practical applications in various domains such as education, health care, personal assistance, etc. QA is a retrieval task which is more challenging than the task of common search engines because the purpose of QA is to find accurate and concise answer to a question rather than just retrieving relevant documents containing the answer [11]. Recently, the code-mixed cross-script QA research problem was formally introduced in [2]. The first step of understanding a question is to perform question analysis. Question classification is an important task in question analysis which detects the answer type of the question. Question classification helps not only filter out a wide range of candidate answers but also determine answer selection strategies [11]. Furthermore, it has been observed that the performance of question classification has significant influence on the overall performance of a QA system.

Let, $Q = \{q_1, q_2, \ldots, q_n\}$ be a set of factoid questions associated with domain D. Each question $q : \langle w_1 w_2 w_3 \ldots w_p \rangle$, 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 $C = \{c_1, c_2, \ldots, c_m\}$ 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 subtask is to classify each given question $q_i \in Q$ into one of the predefined coarse-grained classes $c_j \in C$. For example, the question "*last volvo bus kokhon chare?*" (English gloss: "When does the last volvo bus depart?") should be classified to the class 'TEMPORAL'.

2.1 Datasets

We prepared the datasets for subtask-1 from the dataset described in [2] which is the only dataset available for code-mixed cross-script question answering research. The dataset described in [2] contains questions, messages and answers from the sports and tourism domains in code-mixed cross-script English–Bengali. The dataset contains a total of 20 documents from two domains, namely sports and tourism. There are 10 documents in the sports domain which consist of 116 informal posts and 192 questions, while the 10 documents in the tourism domain consist of 183 informal posts and 314 questions. We initially provided 330 labeled factoid questions as the development set to the participants after accepting the data usage agreement. The testset contains 180 unlabeled factoid questions. Table 1 and Table 2 provide statistics of the dataset. Question class specific distribution of the datasets is given in Figure 1.

Table 1: MSIR16 Subtask-1 Datasets

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

2.2 Submissions

A total of 15 research teams registered for subtask-1. However, only 7 teams submitted runs and a total of 20 runs were received. All the teams submitted 3 runs except AMRITA_CEN who submitted 2 runs.

Table 2: Subtask-1: 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

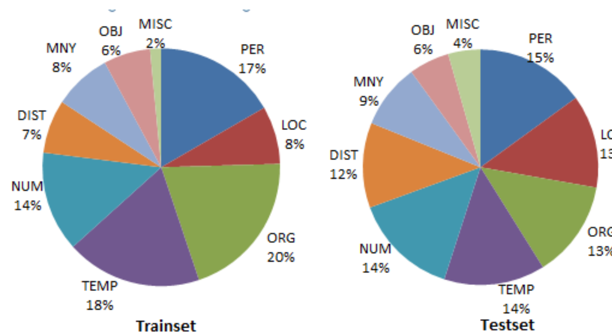

Figure 1: Classwise distribution of dataset

AMRITA_CEN [13] team submitted 2 runs. They used bag-of-words (BoW) model for the Run-1. The Run-2 was based on Recurrent Neural Network (RNN). The initial embedding vector was given to RNN and the output of RNN was fed to logistic regression for training. Overall, the BoW model outperformed the RNN model by almost 7% ons F1-measure.

AMRITA-CEN-NLP [10] team submitted 3 runs. They approached the problem using Vector Space Model (VSM). Weighted term based on the context was applied to overcome the shortcomings of VSM. The proposed approach achieved upto 80% accuracy in terms of F1-measure.

ANUJ [16] also submitted 3 runs. The author used *term frequency âĂŞ inverse document frequency* (TF-IDF) vector as feature. A number of machine learning algorithms, namely Support Vector Machines (SVM), Logistic Regression (LR), Random Forest (RF) and Gradient Boosting were applied using Grid Search to come up with the best parameters and model. The RF model performed the best among the 3 runs.

BITS_PILANI [4] submitted 3 runs. Instead of applying the classifiers on the code-mixed cross-script data, they convert the data into English. The translation was performed using *Google translation API*[1]. Then they applied three machine learning classifiers for each run, namely Gaussian Naïve Bayes, LR and RF Classifier. However, Gaussian Naive Bayes classifier outperformed the other two classifiers.

IINTU [5] was the best performing team. The team submitted 3 runs which were based on machine learning approaches. They trained three separate classifiers namely RF, One-vs-Rest and k-NN, followed by building an ensemble classifier using these 3 classifiers for the classification task. The ensemble classifier took the output label by each of the

[1]https://translate.google.com/

individual classifiers and selected the majority label as output. In case of tie any one label was chosen at random as output.

NLP-NITMZ [14] submitted 3 runs of which 2 runs were rule based - a first set of *direct* rules were applied for the Run-1 while a second set of *dependent* rules were used for the Run-3. A total of 39 rules were identified for the rule based runs. Naïve Bayes classifier was used in Run-2 whereas Naïve Bayes updateable classifier was used in Run-3.

IIT(ISM)D used three different machine learning based classification models - Sequential Minimal Optimization, Naïve Bayes Multimodel and Decision Tree FT to annotate the question text. This team submitted the runs after the deadline.

2.3 Results

In this section, we define the evaluation metrics used to evaluate the runs submitted to the subtask-1. Typically, the performance of a question classifier is measured by calculating the *accuracy* of that classifier on a particular test set [11]. We also used this metric for evaluating the code-mixed cross-script question classification performance.

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

In addition, we also computed the standard precision, recall and F1-measure to evaluate the class specific performances of the participating systems. The precision, recall and F1-measure of a classifier on a particular class c are defined as follows:

$$precision(P) = \frac{\text{number of samples correctly classified as } c}{\text{number of samples classified as } c}$$

$$recall(R) = \frac{\text{number of samples correctly classified as } c}{\text{total number of samples in class } c}$$

$$F1 - measure = \frac{2.P.R}{P + R}$$

Table 3 presents the performance of the submitted runs in terms of accuracy. Class specific performances are reported in Table 4. A baseline system was also developed for the sake of comparison using the BoW which obtained 79.444%

Table 3: Subtask-1: Teams Performance

Team	Run ID	Correct	Incorrect	Accuracy
Baseline	-	143	37	79.440
AmritaCEN	1	145	35	80.556
AmritaCEN	2	133	47	73.889
AMRITA-CEN-NLP	1	143	37	79.444
AMRITA-CEN-NLP	2	132	48	73.333
AMRITA-CEN-NLP	3	132	48	73.333
Anuj	1	139	41	77.222
Anuj	2	146	34	81.111
Anuj	3	141	39	78.333
BITS_PILANI	1	146	34	81.111
BITS_PILANI	2	144	36	80.000
BITS_PILANI	3	131	49	72.778
IINTU	1	147	33	81.667
IINTU	**2**	**150**	**30**	**83.333**
IINTU	3	146	34	81.111
NLP-NITMZ	1	134	46	74.444
NLP-NITMZ	2	134	46	74.444
NLP-NITMZ	3	142	38	78.889
*IIT(ISM)D	1	144	36	80.000
*IIT(ISM)D	2	142	38	78.889
*IIT(ISM)D	3	144	36	80.000

* denotes late submission

accuracy. It can be observed from Table 3 that the highest accuracy (83.333%) was achieved by the IINTU team. The classification performance on the temporal (TEMP) class was very high for almost all the teams. However, Table 4 and Figure 2 suggest that the miscellaneous (MISC) question class was very difficult to identify. Most of the teams could not identify the MISC class. The reason could be very low presence(2%) of MISC class in the training dataset.

3. SUBTASK-2: INFORMATION RETRIEVAL ON CODE-MIXED HINDI-ENGLISH TWEETS

This subtask is based on the concepts discussed in [8]. In this subtask, the objective was to retrieve Code-Mixed Hindi-English tweets from a corpus for code-mixed queries. The Hindi components in both the tweets and the queries are written in Roman transliterated form. This subtask did not consider cases where both Roman and Devanagari scripts are present. Therefore, the documents in this case are tweets consisting of code-mixed Hindi-English texts where the Hindi terms are in Roman transliterated form. Given a query consisting of Hindi and English terms written in Roman script, the system has to retrieve the top-k documents (i.e., tweets) from a corpus that contains Code-Mixed Hindi-English tweets. The expected output is a ranked list of the top twenty (k=20 here) tweets retrieved from the given corpus.

3.1 Datasets

Initially we released 6,133 code-mixed Hindi-English tweets with 23 queries as the training dataset. Later we released a document collection containing 2,796 code-mixed tweets along with with 12 code-mixed queries as the testset. Query terms are mostly *named entities* with Roman transliterated Hindi words. The average length of the queries in the training set is 3.43 words and in the testset it is 3.25 words. The tweets in the training set cover 10 topics whereas the testset cover 3 topics.

3.2 Submissions

This year total 7 teams have submitted 13 runs. The submitted runs for the retrieval task of Code-Mixed tweets mostly adopted preprocessing of the data and then applying different techniques for retrieving the desired tweets. Team Amrita_CEN [9] removed some Hindi/English stop words to declutter useless words. After that, they have tokenized all the tweets. The cosine distance was used to score the relevance of tweets to the query. After that, the top 20 tweets based on the scores were retrieved. Team CEN@Amrita[18] used a Vector Space Model based approach. Team UB [12] has adopted three different techniques for the retrieval task. First, they have used *Named Entity boosts* where the purpose was to boost the documents based on their NE matches from the query, i.e., the query was parsed to extract NEs and each document (tweet) that matched the given NE was provided a small numeric boost. At the second level of boosting, phrase matching was done , i.e., documents that more closely matched the input query phrase were ranked higher than those that did not. The UB team used *Synonym Expansion* and *Narrative based weighting* as the second and third techniques. Team NITA_NITMZ[3] performed stop words removal followed by query segmentation and finally merging. Team IIT(ISM)_D considered every tweet as a document

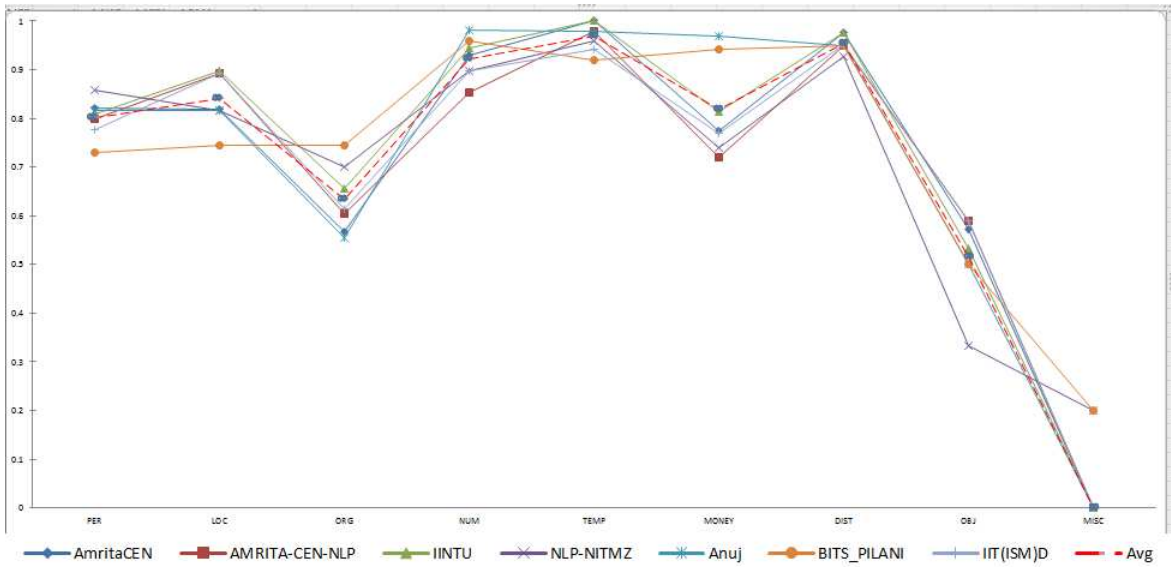

Figure 2: Subtask-1: F-Measure of different teams for classes (Best run)

Table 4: Subtask-1: Class specific performances (NA denotes no identification of a class)

Team	Run ID	PER	LOC	ORG	NUM	TEMP	MONEY	DIST	OBJ	MISC
AmritaCEN	1	0.8214	0.8182	0.5667	0.9286	**1.0000**	0.7742	0.9756	0.5714	NA
AmritaCEN	2	0.7541	0.8095	0.6667	0.8125	**1.0000**	0.4615	0.8649	NA	NA
AMRITA-CEN-NLP	1	0.8000	0.8936	0.6032	0.8525	0.9796	0.7200	0.9500	**0.5882**	NA
AMRITA-CEN-NLP	2	0.7500	0.7273	0.5507	0.8387	0.9434	0.5833	0.9756	0.1818	NA
AMRITA-CEN-NLP	3	0.6939	0.8936	0.5455	0.8125	0.9804	0.6154	0.8333	0.3077	NA
IINTU	1	0.7843	0.8571	0.6333	0.9286	**1.0000**	0.8125	0.9756	0.4615	NA
IINTU	2	0.8077	**0.8980**	0.6552	0.9455	**1.0000**	0.8125	0.9756	0.5333	NA
IINTU	3	0.7600	0.8571	0.5938	0.9455	**1.0000**	0.8571	**0.9767**	0.4615	NA
NLP-NITMZ	1	0.7347	0.8444	0.5667	0.8387	0.9796	0.6154	0.9268	0.2857	0.1429
NLP-NITMZ	2	0.6190	0.8444	0.5667	0.9630	0.8000	0.7333	0.9756	0.4286	0.1429
NLP-NITMZ	3	**0.8571**	0.8163	0.7000	0.8966	0.9583	0.7407	0.9268	0.3333	**0.2000**
Anuj	1	0.7600	0.8936	0.6032	0.8125	0.9804	0.7200	0.8649	0.5333	NA
Anuj	2	0.8163	0.8163	0.5538	**0.9811**	0.9796	0.9677	0.9500	0.5000	NA
Anuj	3	0.8163	0.8936	0.5846	0.8254	**1.0000**	0.7200	0.8947	0.5333	NA
BITS_PILANI	1	0.7297	0.7442	**0.7442**	0.9600	0.9200	0.9412	0.9500	0.5000	**0.2000**
BITS_PILANI	2	0.6753	0.7805	0.7273	0.9455	0.9600	**1.0000**	0.8947	0.4286	NA
BITS_PILANI	3	0.6190	0.7805	0.7179	0.8125	0.8936	0.9333	0.6452	0.5333	NA
*IIT(ISM)D	1	0.7755	0.8936	0.6129	0.8966	0.9412	0.7692	0.9524	**0.5882**	NA
*IIT(ISM)D	2	0.8400	0.8750	0.6780	0.8525	0.9091	0.6667	0.9500	0.1667	NA
*IIT(ISM)D	3	0.8000	0.8936	0.6207	0.8667	**1.0000**	0.6923	0.9500	0.5333	NA
Avg		0.7607	0.8415	0.6245	0.8858	0.9613	0.7568	0.9204	0.4458	NA

and indexed using uniword indexing on Terrier implementation. Query terms were expanded using soundex coding scheme. Terms with identical soundex code were selected as candidate query and included in final queries to retrieve the relevent tweets (documents). Further, they have used three different retrieval models BM25, DFR and TF-IDF to measure the similarity. However, this team submitted the runs after the deadline.

3.3 Results

The retrieval task requires that the retrieved documents at higher ranks be more important than the retrieved documents at lower ranks for a given query and we want our measures to account for that. Therefore, set based evaluation metrics such as Precision, Recall and F-measure are not suitable for this task. Therefore, we used Mean Average Precision (MAP) as the performance evaluation metric for subtask-2. MAP is also referred to as "*average precision* at seen relevant documents". The idea is that, first, *average precision* is computed for each query and subsequently the average precisions are averaged over the queries. MAP is represented as

$$MAP = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{Q_j} \sum_{i=1}^{Q_j} P(doc_i)$$

where Q_j refers to the number of relevant documents for query j, N indicates the number of queries and $P(doc_i)$ represents precision at the i^{th} relevant document.

The evaluation results of the submitted runs are reported in Table 5. The highest MAP (0.0377) was achieved by team Amrita@CEN which is still very low. The significantly low MAP values in Table 5 suggest that the task of retrieving Code-Mixed tweets against query terms comprising code-mixed Hindi and English words is a difficult task and the techniques proposed by the teams do not produce satisfactory results. Therefore, the problem of retrieving relevant code-mixed tweets requires better techniques and methodologies to be developed for improving system performance.

Table 5: Results for Subtask-2 showing Mean Average Precision

Team	Run ID	MAP
UB	1	0.0217
UB	2	0.016
UB	3	0.0152
Anuj	1	0.0209
Anuj	2	0.0199
Amrita_CEN	1	**0.0377**
NLP-NITMZ	1	0.0203
NITA_NITMZ	1	0.0047
CEN@Amrita	1	0.0315
CEn@Amrita	2	0.0168
IIT(ISM)D	1	0.0021
IIT(ISM)D	2	0.0083
IIT(ISM)D	3	0.0021

4. SUMMARY

In this overview, we elaborated on the two subtasks of the MSIR-2016 at FIRE-2016. The overview is divided into two major parts one for each subtask, where the dataset, evaluations metric and results are discussed in detail. A total of 33 runs were submitted from 9 unique teams.

In subtask-1, 20 runs were received from 7 teams. The best performing team achieved 83.333% accuracy. The average question classification performance obtained in terms of accuracy was 78.19% which was quite satisfactory considering this new research problem. The subtask-1 deals with code-mixed Bengali-English language. In the coming years, we would like to include more Indian languages. The participation was encouraging and we plan to continue the subtask-1 in subsequent FIRE conferences.

Subtask-2 received a total of 13 run submissions from 7 teams out of which one team submitted after the deadline. The best MAP value achieved was 0.0377 which is considerably low. From the results of the run submissions it can be inferred that information retrieval of code-mixed informal micro blog texts such as tweets is a very challenging task. Therefore, the stated problem opens and calls for new avenues of research for developing better techniques and methodologies.

5. ACKNOWLEDGMENTS

Somnath Banerjee, Sudip Kumar Naskar and Sivaji Bandyopadhyay acknowledge the support of the Ministry of Electronics and Information Technology (MeitY), Government of India, through the project "CLIA System Phase II".

The work of Paolo Rosso has been partially funded by SomEMBED MINECO TIN2015-71147-C2-1-P research project and by the Generalitat Valenciana under the grant ALMAMATER (PrometeoII/2014/030).

We would also like to thank everybody who helped spread awareness about this track in their respective capacities and the entire FIRE team for giving us the opportunity and platform for conducting this new track smoothly.

6. REFERENCES

[1] U. Z. Ahmed, K. Bali, M. Choudhury, and S. 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] S. Banerjee, S. K. Naskar, P. Rosso, and S. 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.

[3] P. Bhardwaj, P. Pakray, V. Bajpeyee, and A. Taneja. Information Retrieval on Code-Mixed Hindi-English Tweets. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, CEUR Workshop Proceedings, December 2016.

[4] R. Bhargava, S. Khandelwal, A. Bhatia, and Y. 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.

[5] D. Bhattacharjee and P. 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.

[6] K. Chakma and A. Das. CMIR: A Corpus for Evaluation of Code Mixed Information Retrieval of Hindi-English Tweets. In *In the 17th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING)*, April 2016.

[7] M. Choudhury, G. C. Gupta, P. Gupta, and A. Das. Overview of FIRE 2014 Track on Transliterated Search, 2014.

[8] P. Gupta, K. Bali, R. E. Banchs, M. Choudhury, and P. Rosso. Query expansion for mixed-script information retrieval. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 677–686. ACM, 2014.

[9] B. G. HB, A. K. M, and S. KP. Distributional Semantic Representation for Information Retrieval. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, CEUR Workshop Proceedings, December 2016.

[10] B. G. HB, A. K. M, and S. 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.

[11] 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.

[12] N. Londhe and R. K. Srihari. Exploiting Named Entity Mentions Towards Code Mixed IR : Working Notes for

the UB system submission for MSIR@FIREâĂŹ16.

[13] A. K. M and S. 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.

[14] G. Majumder and P. Pakray. NLP-NITMZ @ MSIR 2016 System 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.

[15] R. S. Roy, M. Choudhury, P. Majumder, and K. Agarwal. Overview and datasets of FIRE 2013 track on transliterated search. In *Fifth Forum for Information Retrieval Evaluation*, 2013.

[16] A. 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.

[17] R. Sequiera, 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.

[18] S. Singh, A. K. M, and S. KP. CEN@Amrita: Information Retrieval on CodeMixed Hindi-English Tweets Using Vector Space Models. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, CEUR Workshop Proceedings, December 2016.