Overview of FIRE-2015 Shared Task on Mixed Script Information Retrieval

Royal Sequiera, Monojit Choudhury
Microsoft Research Lab India
{monojitc,a-rosequ}@microsoft.com

Parth Gupta, Paolo Rosso
Technical University of Valencia, Spain
{pgupta,prosso}@dsic.upv.es

Somnath Banerjee, Sudip Kumar Naskar, Sivaji Bandyopadhyay
Jadavpur University, Kolkata
{sb.cse.ju,sudip.naskar,sivaji_cse_ju}@gmail.com

Gokul Chittaranjan
QuaintScience, Bangalore
{gokulchittaranjan@gmail.com}

Amitava Das
IIIT, Sri City
amitava.das@iiits.in

Kunal Chakma
NIT, Agartala
kchax4377@gmail.com

ABSTRACT

The Transliterated Search track has been organized for the third year in FIRE . The track has three subtasks. Subtask I on language labeling of words in code-mixed text fragments was conducted for 8 Indian languages: Bangla, Gujarati, Hindi, Kannada, Malayalam, Marathi, Tamil, Telugu, mixed with English. In Subtask II on retrieval of Hindi film lyrics, along with transliterated queries in Roman script, this year we also had Devanagari queries. A total of 24 runs were submitted from 10 teams, of which 14 runs for subtask I and 10 runs for subtask II has been evaluated. As subtask III is a new task, it did not have any participation. Due to the change of the nature of task, the performance of the runs for Subtask I is lower comapared to that of last year's.

1. INTRODUCTION

A large number of languages, including Arabic, Russian, and most of the South and South East Asian languages, are written using indigenous scripts. However, often websites and 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* [2, 3]. A lack of standard keyboards, a large number of scripts, as well as familiarity with English and QWERTY keyboards has given rise to a number of transliteration schemes for generating Indian language text in Roman transliteration. Some of these are an attempt to standardise the mapping between the Indian language script and the Roman alphabet, e.g., ITRANS[1] but mostly the users define their own mappings that the readers can understand given their knowledge of the language. Transliteration, especially into Roman script, is used abundantly on the Web not only for documents, but also for user queries that intend to search for these documents.

A challenge that search engines face while processing transliterated queries and documents is that of extensive spelling variation. For instance, the word *dhanyavad* ("thank you" in Hindi and many other Indian languages) can be written in Roman script as *dhanyavaad*, *dhanyvad*, *danyavad*, *danyavaad*, *dhanyavada*, *dhanyabad*, and so on. The aim of this shared task is to systematically formalize several research problems that one must solve to tackle this unique situation prevalent in Web search for users of many languages around the world, develop related data sets, test benches and most importantly, build a research community around this important problem that has received very little attention till date.

In the shared task track, we have hosted, as in the previous years, a query labeling subtask, which is one of the first steps before one can tackle the bigger problem, an ad hoc retrieval subtask for Hindi film lyrics, which is one of the most searched items in India and a perfect and practical example of transliterated search and a third subtask introduced this year on mixed script question answering. In the first subtask, participants had to classify words in a query as English or a transliterated form of an Indian language word. In the latter case, they also had to provide the correct transliteration the native script. In the second subtask, participants had to retrieve the top few documents from a multilingual corpus with queries in Roman script. In the third subtask, the participants were required to provide answers to a set of factoid questions written in Romanized Bengali.

This paper provides the overview and datasets of the Mixed Script Information Retrieval track at the seventh Forum for Information Retrieval Conference 2015 (FIRE '15). The track was coordinated jointly by Microsoft Research India, Technical University of Valencia, and Jadavpur University and was supported by BMS College of Engineering, Bangalore. The track on mixed script IR consists of three subtasks. Therefore, the task descriptions, results, and analyses are divided into three parts in the rest of the paper. Details of these tasks can also be found on the website http://bit.ly/1G8bTvR. We have organized this shared task for the third year, and we have received we received participation from 10 teams. A total of 24 runs were submitted in total for subtask 1 and subtask 2.

This paper is organized as follows: First, we present subtask 1 in Sec. 1. Next, we describe subtask 2 in Sec. 2. Sec. 3 provides

[1] http://www.aczoom.com/itrans/

subtask 3 information. We conclude with a summary in Sec. 5.

2. SUBTASK 1: QUERY WORD LABELING

Suppose that $q :< w_1 w_2 w_3 \ldots w_n >$, is a query is written in Roman script. The words, w_1, w_2, w_3, ..., w_n, The words, w1 w2 etc., could be standard English(**en**) words or transliterated from another language L={Bengali(**bn**), Gujarati(**gu**), Hindi(**hi**), Kannada(**kn**), Malayalam(**ml**), Marathi(**mr**), Tamil(**ta**), Telugu(**te**)}. The task is to label the words as en or L or *Named Entity* depending on whether it is an English word, or a transliterated L-language word [4], or a named-entity. Named Entities(**NE**) could be sub-categorized as *person*(**NE_P**), *location* (**NE_L**), *organization*(**NE_O**), *abbreviation*(**NE_PA**,**NE_LA**,**NE_OA**), *inflected named entities* and *other*. For instance, the word USA is tagged as **NE_LA** as the name entity is both a location and an abbreviation. Sometimes, the mixing of languages can occur at the word level. In other words, when two languages are mixed at word level, the root of the word in one language, say L_r, is inflected with a suffix that belongs to another language, say L_s. Such words should be tagged as **MIX**. A further granular annotation of the mixed tags can be done by identifying the languages L_r and L_s and thereby tagging the word as $MIX\_L_r - L_s$.

The subtask differs greatly from the previous years' query labeling task in several ways. While the previous years' subtask required one to identify the language of a query at the word level, when the language pairs are known a priori, this years subtask varies largely as the language pair of a query is not known in advance. In other words, given a document with queries belonging to several language pairs, a participant does a word level language identification of the query. Perhaps, this subtask is more appropriate than the previous year's subtask as such a problem relates more to several real life scenarios.

2.1 Datasets

This section describes the datasets that have been used for the subtasks this year. The following subsections discuss the dataset used for each of the subtasks.

The datasets were made available to the participants based on the e-mail requests. Information about these are available at the website `http://bit.ly/1G8bTvR`.

The data for various languages of subtask 1 was individually collected from various sources. The development set of the subtask was built by collating the previous year's training data set. In addition to the training set, we also gathered data for Kannada-English, Marathi-English, Tamil-English and Telugu-English language pairs.

In addition to the previous years' training data, newly annotated data from [5, 6] was used for Hindi-English language pair. Similarly, for Bangla-English language pairs, data was collected by combining previous year's data with data from [6]. For Gujarati-English language pair, the data from an external source was used in addition to the data from the previous year's shared task data. The data collection and annotation project for six language pairs viz. Gujarati-English, Kannada-English, Telugu-English, Marathi-English, Tamil-English and Malayalam-English was conducted at BMS College of Engineering under Microsoft Research grant. This year, we introduced two new language pairs viz. Marathi-English and Malayalam-English, the data for which was obtained from the aforementioned project.

To curtail the misuse of the data, we made it mandatory for the participants to sign an End User License Agreement (EULA), by which a participant could use the data only for the purposes of the shared task, could not share the data and would delete any copy of the data after they their participation in the shared task.

Lang2	Utterances	Tokens	L-tags	Old Data
		Development		
Bangla	388	9,680	3,551	21,119
Gujarati	149	937	890	937
Hindi	294	10,512	4,295	27,619
Kannada	276	2,746	1,622	0
Malayalam	150	2,111	1,159	2,111
Marathi	201	2,703	1,960	0
Tamil	342	6,000	3,153	0
Telugu	525	6,815	6,478	0
		Test Set		
Bangla	193	2,000	1,368	17,770
Gujarati	31	937	185	1,078
Hindi	190	2,000	1,601	32,200
Kannada	103	1,057	598	1,321
Malayalam	20	231	1,139	1,767
Marathi	29	627	454	0
Tamil	25	1,036	543	974
Telugu	80	1,066	524	0

Table 1: Number of sentences and tags provided for each language pair in the development set. . English was the one of the languages in all language pairs. Lang2 refers to the other language in the pair.

Number of teams who made a submission	9
Number of accepted teams (based on their output conforming to our output format and submitting a working note)	9
Number of runs received	14
Number of runs accepted	14

Table 2: Participation details for subtask 1; numbers in the table indicate the number of runs submitted.

The labeled data from all language pairs was collated into a single file to form a training data set. The training data set was composed of 2908 utterances and 51,513 tokens. The number of utterances, tokens for each language pair in the training set is given in the table 1. The training set was provided in two files viz. *input.txt* and *annotation.txt*. The *input.txt* file consisted of only the utterances where tokens are white space separated and each utterance was assigned a unique id. The *annotation.txt* file consisted of the annotations or labels of the tokens, exactly in the same order as the corresponding input utterance, which can be identified using the utterance id. Both the input and annotation files are XML formatted.

We used the unlabeled data set from the previous years' shared task in addition to the data that was procured from the Internet. Similar to the training set, the test set contained utterances belonging to different language pairs. The test set contained 792 utterances with 12,000 tokens. The number of utterances, tokens for each language pair in the training set is given in the table 1. Unlike the training set, only the *input.txt* was provided to the participants and the participants were asked submit *annotation.txt* file which was used for evaluation purposes.

2.2 Submissions

A total of 10 teams made 24 submissions for subtask 1 and subtask 2. Subtask 1 received 14 submissions from 9 unique teams. The details of the runs submitted are given in table 5. Majority of the teams in subtask 1 made single run submissions. Three teams viz. WISC, Watchdogs and JU_NLP submitted multiple runs. A

Team	Character n-grams	Token features	Rules	Dictionary	Context	Classifier
AmritaCEN	✓	✓	✓	-	✓	SVM
Hrothgar	✓	-	✓	✓	✓	Naive Bayes
IDRBTIR	✓	-	-	-	✓	SVM + Logistic Regression
ISMD	✓	-	-	-	-	MaxEnt
JU	✓	✓	-	✓	✓	CRF
JU_NLP	✓	✓	✓	-	✓	CRF
TeamZine	✓	✓	✓	✓	-	Linear SVM + Logistic Regression + Random Forest
Watchdogs	✓	✓	-	✓	✓	CRF
WISC	✓	-	-	✓	-	Linear Regression + Naive Bayes +Logistic Regression

Table 3: Description of systems for subtask-1.

Team	Run ID	F-score En	F-score IL	F-score MIX	F-score NE	F-score X	Token-Accuracy	Utterance-Accuracy	Average F-score	Weighted F-score
AmritaCEN	1	0.911	0.651	0.670	0.425	0.702	0.766	0.169	0.683	0.767
Hrothgar*	1	0.874	0.777	0.000	0.433	0.947	0.827	0.264	0.692	0.830
IDRBTIR	1	0.831	0.688	0.570	0.387	0.956	0.775	0.181	0.680	0.767
ISMD	1	0.905	0.603	0.400	0.462	0.961	0.771	0.173	0.615	0.769
JU	1	0.892	0.569	0.014	0.433	0.837	0.755	0.216	0.538	0.750
JU_NLP	1	0.747	0.573	0.670	0.432	0.929	0.715	0.129	0.610	0.700
JU_NLP	2	0.678	0.440	0.000	0.434	0.927	0.629	0.102	0.423	0.596
TeamZine	1	0.900	0.669	0.500	0.434	0.964	0.811	0.230	0.618	0.788
Watchdogs	1	0.698	0.644	0.000	0.410	0.967	0.689	0.858	0.576	0.701
Watchdogs	2	0.851	0.689	0.000	0.410	0.964	0.817	0.235	0.623	0.804
Watchdogs	3	0.840	0.561	0.000	0.397	0.963	0.756	0.197	0.525	0.734
WISC	1	0.721	0.356	0.000	0.249	0.824	0.548	0.240	0.387	0.568
WISC	2	0.721	0.408	0.000	0.249	0.824	0.548	0.240	0.387	0.568
WISC	3	0.722	0.408	0.000	0.249	0.822	0.548	0.240	0.387	0.568

Table 4: Subtask 1, language identification: Performance of submissions. * indicates the best performing team;IL =Indian Languages

Number of teams who made a submission	11
Number of accepted teams (based on their output conforming to our output format and submitting a working note)	10
Number of runs received	24
Number of runs accepted	24

Table 5: Participation details for all the subtasks; numbers in the table indicate the number of runs submitted.

total of 8 runs were submitted for subtask-2 by 4 teams. Details are given in table 5. Since subtask 3 is a newer subtask, it did not witness any participation.

All the submissions made by the teams for subtask 1 have used supervised machine learning techniques with character n-grams and character features to identify the language of the tokens. However, WISC and ISMD teams have not used any character features to train the classifier. TeamZine has used word normalization has one of the features, Watchdogs converted the words into vectors using Word2Vec techniques, clustering the vectors using k-means algorithm and then using cluster IDs as the features. Three teams, Watchdogs, JU and JU_NLP have gone beyond using token and character level features, by using contextual information or a sequence tagger. A brief summary of all the systems is given in table 3.

2.3 Results

In this section, we define the metric used to evaluate the runs submitted to the subtasks. For subtask 1, we used the standard precision, recall and f-measure values for evaluation. In addition, we also used the average f-measure and weighted f-measure metrics to compare the performance of teams. As there were some discrepancy in the training data with respect to the X tag, two separate versions of the aforementioned metrics were released: one considering the X tags liberally and the other version where X tags were considered strictly.

2.3.1 Evaluation metrics

We used the following metrics for evaluating Subtask 1. For each category, we compute the precision, recall and F-score as shown below.

$$\text{Precision (P)} = \frac{\#(\text{Correct category})}{\#(\text{Generated category})} \quad (1)$$

$$\text{Rrecall (R)} = \frac{\#(\text{Correct category})}{\#(\text{Reference category})} \quad (2)$$

$$\text{F–score (F)} = \frac{2 \times P \times R}{P + R} \quad (3)$$

Metric	Aggregate Mean	Aggregate Std. Deviation	Aggregate Median	Max Score	#tokens
En F-score	0.807	0.084	0.836	0.911	17948
Ta F-Score	0.726	0.126	0.749	0.891	3153
Bn F-Score	0.707	0.134	0.755	0.854	3551
Hi F-Score	0.617	0.159	0.641	0.813	4295
Mr F-Score	0.599	0.178	0.647	0.831	1960
Kn F-Score	0.575	0.175	0.606	0.871	1622
Ml F-Score	0.476	0.179	0.476	0.745	1159
Te F-Score	0.465	0.134	0.482	0.776	6478
Gu F-Score	0.134	0.103	0.133	0.348	890

Table 6: Subtask 1, correlation between the aggregate mean for each language and the number of tokens present in the training data for that language

Metric	Aggregate Mean	Aggregate Std. Deviation	Aggregate Median	Max Score
IL F-Score	0.538	0.148	0.561	0.766
En F-score	0.807	0.084	0.836	0.911
X F-Score	0.899	0.079	0.938	0.967
NE F-Score	0.371	0.075	0.410	0.462
MIX F-Score	0.278	0.391	0.000	0.670

Table 7: Summary of the subtask 1 evaluation

	en	hi	bn	ml	mr	kn	te	gu
ta	0.052	0.012	0.007	**0.090**	0.010	0.045	0.040	0.021
gu	0.081	**0.226**	0.062	0.010	0.066	0.020	0.024	-
te	**0.121**	0.044	0.030	0.040	0.028	0.048	-	-
kn	**0.099**	0.021	0.025	0.028	0.014	-	-	-
mr	**0.070**	0.065	0.038	0.008	-	-	-	-
ml	**0.112**	0.009	0.019	-	-	-	-	-
bn	**0.074**	0.063	-	-	-	-	-	-
hi	0.107	-	-	-	-	-	-	-

Table 8: Confusing language pairs

2.3.2 Discussion

Table 4 shows the results for all the subtask 1 submissions. Table 6 summarizes the overall performance of the submissions for each language category and table 7 illustrates the performance of the systems with respect to all the categories. The approaches followed and error analyses for each of these submission can be found in the individual working notes of the participants.

Table 6 presents the correlation between the aggregate mean score for each language and corresponding number of tokens provided in the training file. It can be seen that the mean score decreases as the number of the tokens available for that language decreases. However, the score for **te** has been considerably low in spite of having a large number of tokens in the training file. This discrepancy can be attributed to the fact that the **te** data provided in the training file was not naturally generated. As some of the teams have used additional data sets, which might have affected their performance, such a correlation does not exist between the Max Score and number of scores.

We also infer that the most confusing language pairs from the confusion matrices of the individual submissions. For a language pair $L1$-$L2$, we calculate the number of times $L1$ is confused with $L2$ and also the number of times $L2$ is confused with $L1$. We average both the counts over an average of all the submissions. Table 8 illustrates the results obtained. It was found that the **gu-hi** is the most confusing language pair. We also observe that apart from **gu-hi** and **ta-ml** language pairs all the other Indian languages are mostly confused with **en**. This may not be surprising, given the presence of large amount of **en** tokens in the training set.

3. SUBTASK 2: MIXED-SCRIPT AD HOC RETRIEVAL FOR HINDI

This subtask uses the terminology and concepts defined in [7]. In this subtask, the goal was to retrieve mixed-script documents from a corpus for a given mixed-script query. This year, the documents and queries were written in Hindi language but using either Roman or Devanagari script. Given a query in Roman or Devanagari script, the system has to retrieve the top-k documents from a corpus that contains mixed script (Roman and Devanagari). The input is a query written in Roman (transliterated) or Devanagari script. The output is a ranked list of ten ($k = 10$ here) documents both in Devanagari and Roman scripts, retrieved from a corpus. This year there were three different genres or documents: *i)* Hindi songs lyrics, *ii)* movie reviews, and *iii)* astrology. Total 25 queries were used to prepare the test collection for various information needs. Queries related to lyrics documents were expressing the need to retrieve relevant song lyric while queries related to movie reviews and astrology were informational in nature.

3.1 Datasets

We first released a development (tuning) data for the IR system – 15 queries, associated relevance judgments (*qrels*) and the corpus. The queries were related to three genres: *i)* Hindi songs lyrics, *ii)* movie reviews, and *iii)* astrology. The corpus consisted of $63,334$ documents in Roman (ITRANS or plain format), Devanagari and mixed scripts. The test set consisted of 25 queries in either Roman or Devanagari script. On an average, there were 47.52 qrels per query with average relevant documents per query to be 5.00 and cross-script[2] relevant documents to be 3.04. The mean query length was 4.04 words. The song lyrics documents were created by crawling several popular domains like *dhingana*, *musicmaza* and *hindilyrix*. The movie reviews data was crawled from `xyz` while astrology data was crawled from `http://astrology.raftaar.in/`.

3.2 Submissions

Total 5 teams submitted 12 runs. Most of the submitted runs handled the mixed-script aspect using some type of transliteration approach and then different matching techniques were used to retrieve documents.

BIT-M system consisted of two modules, the transliteration module, and the searching module. The transliteration module was trained using transliteration pairs data provided. The module was a statistical model which used the relative frequency of letter group mappings. The search module used the transliteration module to treat everything in devanagari script. LCS based similarity was used to resolve erroneous and ambiguous transliterations. Documents were preprocessed and indexed on the content bigrams. Parts of queries were expanded centred around high idf terms.

[2]Those documents which contain duplicate content in both the scripts are ignored.

Watchdogs used Google transliterator to transliterate every Roman script word in the documents and queries to Devanagari word. They submitted 4 runs with these settings: *1.* Indexed the individual words using simple analyser in lucene and then fired the query, *2.* Indexed using word level 2 to 6 grams and then fired a query, *3.* Removed all the vowel signs and spaces from the documents and queries and indexed the character level 2-6 grams of the documents, and *4.* Removed the spaces and replaced vowel signs with actual characters in the documents and queries and indexed the character level 2-6 grams of the documents.

ISMD also submitted four runs. First two runs were using simple indexing, with and without query expansion. Third and fourth runs were using block indexing, with and without query expansion.

The other teams did not share their approaches.

3.3 Results

The test collection for Subtask 2 contained 25 queries in Roman and Devanagari script. The queries were of different difficulty levels: world level joining and splitting, ambiguous short queries, different script queries and inclusion of different language keywords. We received total 12 runs from 5 different teams and the performance evaluation of the runs is presented in Table 3.2. We also present performance of systems in cross-script setting: where query and relevant documents are strictly in different scripts. Cross-script results are reported in Table 3.2.

4. SUBTASK 3: MIXED-SCRIPT QUESTION ANSWERING

Nowadays, among many other things in social networks, people share their travel experiences gathered during their visits to popular tourist spots. Often social media users seek suggestions as guidance from their social networks before traveling, such as mode of communication, travel fair, places to visit, accommodation, foods, etc. Similarly, sports events are among the mostly discussed topics in social media. People post live updates on scores, results and fixtures of ongoing sports events such as Football leagues (e.g. Champions League, Indian Super League, English Premier League etc.), Cricket Series (e.g. ODI, T20, Test), Olympic games, Tennis tournaments (e.g. Wimbledon, US Open, etc.), etc. Though question answering (QA) is a well addressed research problem and several QA systems are available with reasonable accuracy, there has been hardly any research on QA on social media text mainly due to the challenges social media content presents to NLP research. Here we introduce a subtask on mixed-script QA considering the information need of the bilingual speakers particularly on the tourist and sports domains.

4.1 Task

Let, $Q = \{q_1, q_2, \ldots, q_n\}$, be a set of factoid questions associated with a document corpus C in domain D and topic T, written in Romanized Bengali. The document corpus C consists of a set of Romanized Bengali social media messages which could be code-mixed with English (i.e., it also contains English words and phrases). The task is to build a QA system which can output the exact answer, along with the message/posts identification number (msg_ID) and message segment (S_ans) that contains the exact answer. An example is given in Table 11. This task deals with factoid questions only. For this subtask,

domain D = {Sports, Tourism}

Mixed Language pair = {Bengali-English}

4.2 Dataset

Table 11: Example

Domain	Tourism
msg ID	T818
Message/Post	Howrah station theke roj 3 train diyeche ekhon Digha jabar jonno...just chill chill!!
QID	TQ8008
Question	Howrah station theke Digha jabar koiti train ache?
Exact Answer	3
S_Ans	Howrah station theke roj 3 train diyeche ekhon Digha jabar jonno
M_ans	T818

Being the most likely potential source of code-mixed cross-script data, we procured all the data from social media, e.g., Facebook, Twitter, blogs, forums, etc. Initially, we released a small dataset which was made available to all the registered participants after signing the agreement analogous to subtask-1. The code-mixed messages/posts related to ten popular tourist spots in India were selected for tourism domain. For sports domain, posts/messages related to recently held ten exciting cricket matches were selected. We split the corpus in two sets namely, development and test sets. The distribution of public posts/messages and questions in the corpus for the two different domains, namely Sports and Tourism, are given in Table 12.

Table 12: Corpus statistics (D=Document, M= Message, Q= Question)

Training					
Domain	**D**	**M**	**Q**	**Avg. M/D**	**Avg. Q/D**
Sports	5	53	89	10.6	17.8
Tourism	5	93	161	18.6	32.2
Overall	10	146	250	14.6	25.0
Test					
Sports	5	63	103	12.6	20.6
Tourism	5	90	153	18.0	30.6
Overall	10	153	256	15.3	25.6

4.3 Submission Overview

A total of X teams registered for subtask-3. However, no runs were submitted by the registered participants. In this scenario, we produced a baseline system which is presented in the following section.

4.4 Baseline

A baseline system has been developed to confront the challenges of this task. At first, a language identifier [8] was applied to identify the languages in the dataset, i.e., Bengali and English. Then interrogatives were identified using an interrogative list. Word level translation was applied to English words using an in-house English to Bengali dictionary which is prepared as part of the ELIMT project. The detected Bengali words are transliterated using a phrase-based statistical machine transliteration system. Then a named entity (NE) system [9] is applied only to the transliterated Bengali words to identify the NEs. The aforementioned steps are applied both to the messages and questions. Then separate procedures are applied for code-mixed messages and questions. For each question, heuristically based on the interrogative an expected NE of answer type is assigned and a query is formed after removing the stop

words. The highest ranked message is selected by measuring the semantic similarity to the query words. The exact answer is extracted from the highest ranked messages using the suggested NE type of the answer in query formulation step. In case of missing the expected NE type, the system chooses the NOA option. This baseline only can output exact answer with message identification number. i.e., partial supported answer. Baseline results are reported in Table 13.

Table 13: Baseline Results

	Domain	N	N_r	N_u	C@1	Acc	ASP
Train	Sports	89	43	9	0.5371	0.4894	0.3670
	Tourism	161	84	6	0.5284	0.5148	0.3861
Test	Sports	103	46	10	0.4891	0.4457	0.3343
	Tourism	157	88	5	0.5747	0.5588	0.4199

4.5 Evaluation Matrices

In this task, an answer is basically structured as [*Answer String* (AS), *Message Segment* (MS), *Message ID* (MId)] triplet, where-

– AS is the one of the exact answers (EA) and must be an NE in this case,

– MS is the supported text segment for the extracted answer, and

– MId is the unique identifier of the message that justifies the answer.

While answering the questions, one has to consider the following:

i) The QA system has the provision of not answering, i.e., no answer option (NAO).

ii) The answer is the exact answer to the question.

iii) The exact answer must be a Named Entity.

iv) The system has to return a single exact answer. In case there exists more than one correct answer to a question, the system needs to provide only one of the correct answers.

While evaluating, the primary focus remains on "responsiveness" and "usefulness" of each answer. Each answer is manually judged by native speaking assessors. Each [AS, MS, MId] triplet is assessed in a five-valued scale (Table 14) and marked with exactly one of the following judgments:

- **Incorrect:** The AS does not contain EA (i.e., not responsive)
- **Unsupported:** The AS contains correct EA, but MS and MId do not support the EA (i.e., missing usefulness)
- **Partial-supported:** The AS contains the correct EA with correct Mid, but MS does not support EA
- **Correct:** The AS provides the correct EA with correctly supporting MS and MId (i.e., "responsive" as well as "useful").
- **Inexact:** The supporting MS and MId are correct, but the AS is wrong.

In order to maintain consistency with previous QA shared tasks, we have chosen accuracy and c@1 as evaluation metrics. MRR has not been considered since a QA system is supposed to return an exact answer, i.e. not list. Just as in the past ResPubliQA[3] campaigns, systems are allowed to have the option of withholding the answer to a question because they are not sufficiently confident that it is correct (i.e., NAO). As per ResPubliQA, inclusion of NAO improves the system performance by reducing the number of incorrect answers.

[3]http://nlp.uned.es/clef-qa/repository/resPubliQA.php

Table 14: Judgment Scale

Judgment	AS	MS	MId	Score
Incorrect (W)	X	X	X	0.00
Inexact(I)	X	✓	✓	0.25
Unsupported (U)	✓	X	X	0.50
Partial-supported (P)	✓	X	✓	0.75
Correct (C)	✓	✓	✓	1.00

Now, C@1 = $\frac{1}{N}(N_r + N_u . \frac{N_r}{N_u})$

Accuracy = $\frac{N_r}{N}$

C@1 = Accuracy; if $N_u = 0$

Where, N_r = number of right answers.

N_u = number of unanswered questions

N = total questions

Correct, Partially-supported and Unsupported answers provide the exact answers only.

Therefore, $N_r = (\#C + \#U + \#P)$

Considering the importance of supporting segment, we introduce a new metric "Answer-Support performance" (ASP) which measures the answer correctness.

$ASP = \frac{1}{N}(c \times 1.0 + p \times 0.75 + i \times 0.25)$

where, c, p and i denote total number of correct, partially-supported and inexact answers.

4.6 Discussion

In spite of a significant number of registrations in subtask-3, no run was received. Personal communication with registered participants revealed that the time provided for this subtask was not sufficient to develop the required mixed-script QA system. Next year we could simplify the task load by asking participants to solve various subtasks of the said QA system, such as question classification, question focus identification, etc. Introducing more Indian-English language pairs could encourage this subtask across other Indian languages speakers.

5. SUMMARY

In this overview, we elaborated on the various subtasks of the Mixed Script Information Retrieval track at the seventh Forum for Information Retrieval Conference (FIRE'15). The overview is divided into three major parts one for each subtask, where the dataset, evaluations metric and results are discussed in detail.

There were a total of 24 submissions from a total of 10 teams. In subtask 1, we noted that **gu-hi** was the most confused language pair. It was also found that the performance of the system for a category is positively correlated to the number of tokens for that category.

Acknowledgments

We would like to thank Prof Shambhavi Pradeep, BMS College of Engineering for data creation for subtask 1 in 6 Indian languages, Dnyaneshwar Patil, ISI Kolkata for contributing Marathi data, Shubham Kumar, IIT Patna, for contributing towards subtask 2 data. We are also grateful to Shruti Rijhwani and Kalika Bali, Microsoft Research Lab India, for helping with reviewing the working notes of this shared task.

6. REFERENCES

[1] Ahmed, U.Z., Bali, K., Choudhury, M., B., S.V.: Challenges in designing input method editors for indian languages: The role of word-origin and context. Advances in Text Input Methods (WTIM 2011) (2011) 1–9

[2] Knight, K., Graehl, J.: Machine transliteration. Computational Linguistics **24**(4) (1998) 599–612

[3] Antony, P., Soman, K.: Machine transliteration for indian languages: A literature survey. International Journal of Scientific & Engineering Research, IJSER **2** (2011) 1–8

[4] King, B., Abney, S.: Labeling the languages of words in mixed-language documents using weakly supervised methods. In: Proceedings of NAACL-HLT. (2013) 1110–1119

[5] Vyas, Y., Gella, S., Sharma, J., Bali, K., Choudhury, M.: Pos tagging of english-hindi code-mixed social media content. In: EMNLP'14. (2014) 974–979

[6] Barman, U., Das, A., Wagner, J., Foster, J.: Code mixing: A challenge for language identification in the language of social media. (2014) 13–23

[7] Gupta, P., Bali, K., Banchs, R.E., Choudhury, M., Rosso, P.: Query expansion for mixed-script information retrieval. In: The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014. (2014) 677–686

[8] Banerjee, S., Kuila, A., Roy, A., Naskar, S.K., Rosso, P., Bandyopadhyay, S.: A hybrid approach for transliterated word-level language identification: Crf with post-processing heuristics. In: FIRE, ACM Digital Publishing (2014)

[9] Banerjee, S., Naskar, S., Bandyopadhyay, S.: Bengali named entity recognition using margin infused relaxed algorithm. In: TSD, Springer International Publishing (2014) 125–132

Team	NDCG@1	NDCG@5	NDCG@10	MAP	MRR	R@10
AmritaCEN	0.2300	0.2386	0.1913	0.0986	0.2067	0.1308
BIT-M	0.7567	0.6837	0.6790	0.3922	0.5890	0.4735
Watchdogs-1	0.6700	0.5922	0.6057	0.3173	0.4964	0.3962
Watchdogs-2	0.5267	0.5424	0.5631	0.2922	0.3790	0.4435
Watchdogs-3	0.6967	0.6991	0.7160	0.3814	0.5613	0.4921
Watchdogs-4	0.5633	0.5124	0.5173	0.2360	0.3944	0.2932
ISMD-1	0.4133	0.4268	0.4335	0.0928	0.2440	0.1361
ISMD-2	0.4933	0.5277	0.5328	0.1444	0.3180	0.2051
ISMD-3	0.3867	0.4422	0.4489	0.0954	0.2207	0.1418
ISMD-4	0.4967	0.5375	0.5369	0.1507	0.3397	0.2438
QAIIITH-1	0.3433	0.3481	0.3532	0.0705	0.2100	0.1020
QAIIITH-2	0.3767	0.3275	0.3477	0.0561	0.2017	0.1042

Table 9: Results for subtask II averaged over test queries.

Team	NDCG@1	NDCG@5	NDCG@10	MAP	MRR	R@10
AmritaCEN	0.1367	0.1182	0.1106	0.0898	0.1533	0.1280
BIT-M	0.3400	0.3350	0.3678	0.2960	0.3904	0.4551
Watchdogs-1	0.4233	0.3264	0.3721	0.2804	0.4164	0.3774
Watchdogs-2	0.1833	0.2681	0.3315	0.2168	0.2757	0.4356
Watchdogs-3	0.3333	0.3964	0.4358	0.3060	0.4233	0.5058
Watchdogs-4	0.2900	0.2684	0.2997	0.2047	0.3244	0.2914
ISMD-1	0.0600	0.0949	0.1048	0.0452	0.0714	0.0721
ISMD-2	0.1767	0.2688	0.2824	0.1335	0.1987	0.2156
ISMD-3	0.0600	0.1098	0.1191	0.0563	0.0848	0.0988
ISMD-4	0.2267	0.3242	0.3375	0.1522	0.2253	0.2769
QAIIITH-1	0.0600	0.0626	0.0689	0.0313	0.0907	0.0582
QAIIITH-2	0.0200	0.0539	0.0673	0.0234	0.0567	0.0661

Table 10: Results for subtask II averaged over test queries in cross-script setting.