Overview of FIRE 2014 Track on Transliterated Search

Monojit Choudhury, Gokul Chittaranjan
Microsoft Research Lab India
{monojitc,t-gochit}@microsoft.com

Parth Gupta
Technical University of Valencia, Spain
pgupta@dsic.upv.es

Amitava Das
Univ. of North Texas, USA
amitava.das@unt.edu

ABSTRACT

The Transliterated Search track has been organized for the second year in FIRE. The track has two subtasks. Subtask 1 on language labeling of words in code-mixed text fragments was conducted for 6 Indian languages: Bangla, Gujarati, Hindi, Malayalam, Tamil, Telugu, mixed with English. In Subtask 2 on retrieval of Hindi film lyrics, along with transliterated queries in Roman script, this year we also had Devanagari queries. A total of 54 runs were submitted from 18 teams, of which 35 runs for subtask 1 and 7 runs for subtask 2 has been evaluated. Performance of the runs is comparable to that of last year's.

1. INTRODUCTION

The shared task on transliterated search was introduced last year in FIRE 2013. There were two subtasks on labeling of the query words and ad hoc retrieval for transliterated lyrics queries [1]. This year, we hosted the same two subtasks, but with additional features and language pairs. A large number of teams participated in the shared task.

We provide here an overview of the transliterated search track at the sixth Forum for Information Retrieval Conference 2014 (FIRE '14). First, we present a description of the shared tasks in Sec. 2. Next, we describe the datasets associated with these tasks in Sec. 3. Sec. 4 records task participation information. We discuss results in Sec. 5 and conclude with a summary in Sec. 6.

2. TASKS

Our track on transliterated search contains two major sub-tasks. Therefore, in the rest of the paper, we divide our descriptions, results, and analyses into two parts, one for each of these subtasks. Details of these tasks can also be found on the website `http://bit.ly/1q6rb6h`.

2.1 Subtask 1: Language Identification and Transliteration

Suppose that $s :< w_1 w_2 w_3 \ldots w_n >$, is a sentence is written in Roman script. The words, w_1, w_2, w_3, ..., w_n, could be English words or transliterated from another language L. The task is to label the words as E or L or *NE* depending on whether it an English word, or a transliterated L-language word [2], or a named-entity. Named entities are further typed as *person*, *location*, *organization*, and *acronym*. Further, for some language pairs, words of E can be inflected with suffix of L or vice versa. In such cases, they have to be tagged as *MIX*. We also introduced a tag *O=others* for words which cannot be classified as L, E, NE or MIX. These could be punctuations, numbers, emoticons or foreign language words.

For each transliterated word (i.e. words with tag L), the correct transliteration has to be provided in the native script (i.e., the script which is used for writing L).

We added three language pairs to this subtask this year, amounting to a total of six language pairs – English-Bangla, English-Gujarati, English-Hindi, English-Kannada, English-Malayalam and English-Tamil. Furthermore, last year the labeling task was restricted to queries or very short text fragments. This year, most of our sentences were acquired from social media posts (public) and blogs. With a large number of spelling variations and contractions happening over social media, we believe the task this year was more challenging than last year's.

2.2 Subtask 2: Mixed-Script Ad hoc Retrieval for Hindi Song Lyrics

Given a query in Roman or Devanagari script, the system has to retrieve the top-k documents that are either in Devnagari or in Roman transliterated form or in both the scripts (mixed-script documents). Like last year, we used the Bollywood song lyrics corpus and song queries as our dataset, but two new concepts were introduced this year. First, the queries could also be in Devanagari. Second, Roman queries could have splitting or joining of words. For instance, "`main pal do palka shayar hun`" (where the words `pal` and `ka` has been *joined* incorrectly), or "`madhu ban ki sugandh`" (where the word `madhuban` has been incorrectly *split* incorrectly). We observed that in the Bing logs, a large number of lyrics queries featured this kind of noise, which is probably due to lack of formal education in Hindi. Indeed, Bollywood's popularity stretch far beyond the Hindi speakers. Hence, we chose to introduce these types of challenging queries this year.

3. DATASETS

In this section, we describe the datasets that have been released for the tasks described in the previous section and those that could be generally useful for solving transliteration tasks. While the former have been carefully constructed by us using manual and automated techniques and have been made available to participants through email requests, the latter are external resources freely available online. Information about these are available at the website `http://bit.ly/194bOTT`.

3.1 General

These datasets can generally useful for a variety of transliteration tasks. These include word frequency lists, word transliteration pairs, miscellaneous tools and corpora for various languages.

1. **English**
   (a) **English word frequency list** This dataset is available in a plain tab-separated text format. It contains the standard dictionary of English words followed by their frequencies computed from a large corpus. It contains some noise (very low frequency entries) as it is constructed from a news corpora.
2. **Hindi**
   (a) **Hindi word frequency list** This dataset is available in a plain tab-separated text format. It contains Hindi words (in Devanagari script) followed by their frequency computed from a large Leipzig corpus (see below).
   (b) **Hindi word transliteration pairs 1** This is available in a plain tab-separated text format. It contains $30,823$ transliterated Hindi words (Roman script) followed by the same word in Devanagari. It contains Roman spelling variations for the same Hindi word (transliteration pairs found using alignment of Bollywood song lyrics). It does not contain frequency of occurrence of a particular word transliteration pair [3].
   (c) **Hindi word transliteration pairs 2** This dataset contains annotations (Hindi word transliteration pairs) collected from different users in mulitple setups – chat, dictation and other scenarios. These may be collated into a single resource file if desired; it also provides the frequency of occurrence of a particular word transliteration pair [4].
3. **Bangla**
   (a) **Bangla word frequency list** This is available in a plain tab-separated text format. It contains Bangla words (Roman script, ITRANS format) followed by their frequency computed from a large Anandabazar Patrika corpus[1]. The ITRANS to UTF-8 converter below can be used for obtaining the words in Bangla script.
   (b) **Bangla word transliteration pairs** This dataset contains annotations (Bangla word transliteration pairs) collected from different users in mulitple setups – chat, dictation and other scenarios. These may be collated into a single resource file if desired; it will also provide the frequency of occurrence of a particular word transliteration pair [4].
4. **Gujarati**
   (a) **Gujarati word frequency list** This is available in a plain tab-separated text format. It contains Gujarati words (in Gujarati script) followed by their frequency computed from a large Leipzig corpus (see below).
   (b) **Gujarati word transliteration pairs** This is available in a plain tab-separated text format. It contains transliterated Gujarati words (Roman script) followed by the same word in Gujarati script. Due to the poor availability of Gujarati resources, this is a small list of 546 entries created from our training data.
5. **General**
   (a) **Leipzig corpora collection** This dataset has several large corpora for multiple languages. The word frequency lists for English, Hindi and Gujarati have been constructed from Leipzig corpora. Please cite the paper mentioned on the site [5] in your working notes.
   (b) **ITRANS to UTF-8 converter for Bangla** This tool has been developed by IIT Kharagpur. Look for "Windows $->$ Stand-alone Application Available Modules". One can register on the site for free and download the application.

[1] `http://www.anandabazar.com/`

3.2 Subtask 1

We split the data we collected for all the 6 language pairs from various publicly available sources. For the Hindi-English langauge pair, data was procured from the last year's shared task and newly annotated data from our more recent work [6, 7]. For the Bangla-English language pair, similarly, data from last year's shared task was combined with data from [7]. Gujarati-English pair data remained the same from the previous year and lastly, the 3 other language pairs have been introduced this year and it is based out of publicly available data from the Internet. Details of the sources from which the data was collected is given in table 3. We also made it mandatory for the participants to sign a data usage agreement to ensure that all participants used the data only for the purposes of this shared task, do not share it, and would delete all copies of it after they make their submissions.

The labeled data, from all language pairs, except Kannada-English and Tamil-English, were split into development and test sets. Since we had few labeled data samples from Kannada-English and Tamil-English pairs, we used them entirely for as our test set.

The number of sentences, tokens of each kind and transliterations available for each language pair for the development set is given in table 1. Since the amount of data was moderate, we did not recommend its use for training algorithms, but rather as a development set for tuning model parameters and to understand and analyze word transliteration pairs. This data was provided as UTF-8 encoded text files. For the Tamil and Kannada language pairs, as mentioned before, because we had less than 150 labeled sentences, we released 2-3 labeled examples for participants to familiarize themselves with the annotation format.

For the test set, we combined the labeled data we had procured with unlabeled data using publicly available content from the Internet. This procedure has enabled us to obtain labels for a large quantity of unlabeled data from the submissions made to the shared task this year. We intend to use this to easily obtain ground truth for a larger development/test set for shared tasks to be conducted in the future. Details about the number of sentences in the test set and the size of the subset of sentences that were labeled for each language pair are given in table 2.

3.3 Subtask 2

We first released a development (tuning) data for the IR system – 25 queries, associated relevance judgments (*qrels*) and the corpus. The queries were Bollywood song lyrics. The corpus consisted of $62,894$ documents which contained song titles and lyrics in Roman (ITRANS or plain format), Devanagari and mixed scripts. The test set consisted of thirty five queries in either Roman or Devanagari script. On an average, there were 65.48 qrels per query with average relevant documents per query to be 7.37 and cross-script[2]

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

Lang2	Sentences	Tokens	E-tags	L-tags	MIX	O	NEs	Translits
Bangla	800	20,648	8,786	7,617	0	3,783	462	364
Gujarati	150	937	47	890	0	0	0	890
Hindi	1,230	27,614	11,486	11,989	0	3,371	768	2,420
Malayalam	150	1,914	326	1,139	65	292	92	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.

Lang2	Test-set Size	Subset with labels	Tokens	E-tags	L-tags	MIX	O	NEs	Translits
Bangla	1,000	739	17,305	7,215	6,392	0	3,236	462	397
Gujarati	1,000	150	1,078	12	1,050	0	0	16	1,064
Hindi	1,273	1,270	32,111	12,434	13,676	0	4,815	1,186	2,542
Kannada	1,000	119	1,271	280	812	3	138	38	815
Malayalam	1,000	120	1,473	243	885	37	233	75	885
Tamil	1,000	49	974	460	399	0	115	0	0

Table 2: Number of sentences and tags on which language pairs was evaluated on, in the test set. English was the one of the languages in all language pairs. Lang2 refers to the other language in the pair.

Lang2	Data Sources
Bangla	http://www.facebook.com/JuConfessions, http://www.gutenberg.org/ebookx/18581
Gujarati	http://www.gutenberg.org/ebookx/18581, http://songslyricserver.blogspot.com/p/blog-page_19289.html
Hindi	https://www.facebook.com/Confessions.IITB, https://www.facebook.com/DUConfess1, Some manually curated data from Facebook public pages
Kannada	http://kannadalyric.blogspot.in, http://www.gutenberg.org/ebookx/18581
Malayalam	https://www.facebook.com/keralatourismofficial, https://www.facebook.com/mathrubhumicom, https://www.facebook.com/asianetnews, http:///www.facebook.com/AsianetNews, http://filmsonglyrics.wordpress.com, http://www.gutenberg.org/ebookx/18581
Tamil	Various public blogs with Tamil content, http://www.gutenberg.org/ebookx/18581

Table 3: Sources from which data was obtained for the test and development sets, across different language pairs.

Number of teams who made a submission	18
Number of accepted teams (based on their output conforming to our output format and submitting a working note)	16
Number of runs received	54
Number of runs accepted	39

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

relevant documents to be 3.26. The mean query length was 4.57 words. The song lyrics documents were created by crawling several popular domains like *dhingana*, *musicmaza* and *hindilyrix*.

4. SUBMISSIONS OVERVIEW

A total of 18 teams made 54 submissions for both subtasks. Of these, 39 runs were declared valid as they conformed to the output format. These runs came from 16 unique teams. These details are given in table 5. Of the accepted runs, in subtask-1, Hindi-English was the most common language pair (constituting 17 runs), followed by Bangla-English (8 runs). Gujarati-English and Kannada-English had 3 runs and Malaylam-English and Tamil-English had 2 runs, respectively. Half of teams submitted a single run for a language pair. IITP-TS, JU-NLP-LAB, PESIT-CS-FIRE-IR, L1, and BITS-Lipyantran submitted multiple runs. A total of 7 runs were submitted for subtask-2 by 4 teams. Details are given in table 6.

Most of the submissions made by the teams for subtask-1 have utilized character based n-gram features with additional token-level features along with a supervised classifier for language identification and dictionaries along with rules to obtain transliteration. Two submissions (Asterisk and BITS-Lipyantran) has used a readily available API (Google transliteration API) for the transliteration task. Four teams, BITS-Lipyantran, IIITH, IITP-TS, 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 4.

For subtask 2, the teams followed two different strategies; some teams use a single operating script, either Devanagari or Roman, and then transliterate the documents and queries which are in the other script to the operating script. Other teams have generated

Team	Character n-grams	Token features	Rules	Dictionary	Context	Classifier	Transliteration
Asterisk	-	-	✓	✓	-	-	Google transliteration API
BTS-Lipyantran	✓	-	✓	✓	✓	SVM + Naive Bayes	Google Transliteration API
BMS-Brainz	✓	-	✓	✓	-	-	Rule-based
DA-IR	✓	-	-	-	-	Path-matching	Rule-based with Hindi as the base language
I1	-	✓	-	✓	-	Naive Bayes	-
IIITH	✓	✓	-	✓-	✓	SVM + Linear Kernel	ID3 classifier + Indic-converter
IITP-TS	✓	✓	-	-	✓	SVM, Random forests, decision tree	Rules along with probability distributions
ISI	-	-	✓	✓	-	-	Uses a dictionary and rule-based
ISM-D	✓	-	-	-	-	MaxEnt	Character mapping rules and dictionary
JU-NLP	✓	✓	-	-	-	CRF	Phrase-based statistical transliteration tool
PESIT-CS-FIRE	✓	-	-	-	-	SVM and Naive Bayes	Uses a dictionary and rule-based
Salazar	-	-	✓	✓	-	-	Uses a dictionary and rule-based
Sparkplug	-	-	✓	✓	-	-	Uses a dictionary and rule-based

Table 4: Description of systems for subtask-1.

Team Name	Subtask-1						Subtask-2
	bn	hi	gu	ml	kn	ta	
asterisk	-	1	-	-	-	-	-
BIT	-	-	-	-	-	-	2
BITS-Lipyantran	-	2	-	-	-	-	2
BMS-Brainz	1	-	1	1	1	1	-
DA-IR	-	1	1	-	-	-	-
DCU	-	-	-	-	-	-	2
I1	-	2	-	-	1	-	-
IIITH	1	1	1	1	1	1	1
IITP-TS	3	3	-	-	-	-	-
ISI	1	-	-	-	-	-	-
ISMD	-	3	-	-	-	-	-
JU-NLP-LAB	2	-	-	-	-	-	-
PESIT-CS-FIRE-IR	-	2	-	-	-	-	-
Salazar	-	1	-	-	-	-	-
Sparkplug	-	1	-	-	-	-	-
Total	**8**	**17**	**3**	**2**	**3**	**2**	**7**

Table 6: Participation details for all the subtasks; numbers in the table indicate the number of runs submitted.

cross-script equivalents for indexing the documents as well as matching the queries. Here is a brief overview of the approaches used by the teams for subtask 2.

BITS-LIPYANTARAN back-transliterated the queries and documents to Devanagari using Google Transliteration engine. They removed vowels as part of the normalising step and indexed character n-grams as tokens with n∈{3,4,5,6}. They also supplied some hand-tailored rules for consonants mappings.

DCU generated a dictionary of cross-script equivalents from the documents in the corpus which contained the song in both scripts. For the out-of-vocabulary (OOV) terms, some hand-tailored rules and an Transliteration engine was used. They divided documents in fields like title and body and indexed them according to the script separately. For equivalents, they used edit-distance based term matching with some threshold. The tokens of index were word uni and bi-grams.

BIT prepared an initial word bi-gram query to contain terms from both scripts using Google Transliterate. They retrieved a first document and enriched query from the word n-grams from this first retrieved document. This expanded query is then used for retrieval.

IIIT-H considered Roman as the operating script i.e. all the documents and query were converted in Roman script using a transliteration strategy. They applied some normalisation rules like repetition of the same character was replaced by single occurence. The documents were divided based on fields like title, first line, first stanza, artist name, body etc during indexing. The term variation in Roman script was handled using edit-distance with some pruning.

5. RESULTS

The ideal way to measure the effectiveness of an algorithm output on subtask 1 is not an obvious choice. We try to be as thorough as possible to reward or penalize in all the different aspects of the labeling task, and try to adapt traditional metrics wherever applicable. Subtask 2, on the other hand, can be easily evaluated using standard IR metrics. In this section, we first precisely define the metrics used for evaluating the runs submitted to subtasks one and two. We then tabulate the performance of all the participating teams.

5.1 Evaluation metrics

5.1.1 Subtask 1

We used the following metrics for evaluating Subtask 1. Our metrics reflect various degrees of strictness, including the strictest (Exact Query Match Fraction) to the most lenient (Labeling Accuracy) metrics.

$$\text{Exact query match fraction (EQMF)} = \frac{\#(\text{Quer. for which lang. labels and translits. match exactly})}{\#(\text{All queries})} \quad (1)$$

$$\text{Exact query match fraction LI only (EQMF}_2) = \frac{\#(\text{Quer. for which lang. labels match exactly})}{\#(\text{All queries})} \quad (2)$$

$$\text{Exact transliteration pair match (ETPM)} = \frac{\#(\text{Pairs for which translits. match exactly})}{\#(\text{Pairs for which both o/p and reference labels are L})} \quad (3)$$

The value of this ratio can be treated as a measure of transliteration precision, but the absolute values of the numerator and denominator are also important. For example, when there are 2000 true L words in the reference annotations, it is possible that a method can detect 5 of these and produce the correct transliterations for each, and have a ratio value of 1.0. Another method can detect 200 of these, and produce correct transliterations for 150, and obtain a value of 0.75. We treat the second method as a better one. We note that, as Knight and Graehl [8] point out, back-transliteration is *less forgiving* than forward transliteration for there may be many ways to transliterate a word in another script (forward transliteration) but there is only one way in which a transliterated word can be rendered back in its native form (back-transliteration). Our task thus requires the algorithm to only perform back-transliteration and thus there is only one correct transliteration answer for a word in a given context. Along these lines, we also compute the transliteration precision, recall and F-score as below.

$$\text{Transliteration precision (TP)} = \frac{\#(\text{Correct transliterations})}{\#(\text{Generated transliterations})} \quad (4)$$

$$\text{Transliteration recall (TR)} = \frac{\#(\text{Correct transliterations})}{\#(\text{Reference transliterations})} \quad (5)$$

$$\text{Transliteration F–score (TF)} = \frac{2 \times TP \times TR}{TP + TR} \quad (6)$$

$$\text{Labelling accuracy (LA)} = \frac{\#(\text{Correct label pairs})}{\text{Total}\#\text{pairs}} \quad (7)$$

$$\text{English precision (EP)} = \frac{\#(\text{E}-\text{E pairs})}{\#(\text{E}-\text{X pairs})} \quad (8)$$

$$\text{English recall (ER)} = \frac{\#(\text{E}-\text{E pairs})}{\#(\text{X}-\text{E pairs})} \quad (9)$$

$$\text{EnglishF}-\text{Score(EF)} = \frac{2 \times EP \times ER}{EP + ER} \quad (10)$$

Here, an A-B pair refers to a word that is labeled by the system as A, whereas the actual label (i.e., the ground truth) is B. X is a wildcard that stands for any category label. Thus, E-E pair is a word that is of English and also labeled by the system as E, whereas E-X pair consists of all those words which are labeled as English by the system irrespective of the ground truth. Like E-precision etc., we have L-precision, L-recall, and $L-$F-Score for L where L. We also measured precision, recall and F-score of the other classes, i.e., *NE*, *MIX* and O, but they have not been reported here.

In our transliteration evaluation strategy we relaxed certain constraints for string matching. We handle certain cases of unicode normalization, and do not penalize mistakes made on homorganic nasal – *chandrabindu* replaced by *bindu* and the non-obligatory use of the *nukta*.

5.1.2 Subtask 2

For evaluating subtask 2, we used the well-established IR metrics of normalized Discounted Cumulative Gain (nDCG) [9], Mean Average Precision (MAP) [10] and Mean Reciprocal Rank (MRR) [11].

We used the following process for computing nDCG. The formula used for DCG@p was as follows

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2 i} \quad (11)$$

where p is the rank at which we are computing DCG and rel_i is the graded relevance of the document at rank i. For IDCG@p, we sort the RJs for a particular query in the pool in descending order and take the top$-p$ from the pool, and compute DCG@p for that list (since that is the best possible (ideal) ranking for that query). Then, as usual, we have

$$nDCG@p = \frac{DCG@p}{IDCG@p} \quad (12)$$

nDCG was computed after looking at the first five and the first ten retrieved documents (nDCG@5 and nDCG@10).

For computing MAP, we first compute average precision $AveP$ for every query, where AveP is given by

$$AveP = \frac{\sum_{k=1}^{n}(P(k) \times rel(k))}{\text{No. of relevant documents}} \quad (13)$$

where where k is the rank in the sequence of retrieved documents, n is the number of retrieved documents, $P(k)$ is the precision at cut-off k in the list and rel_k is an indicator function equaling 1 if the item at rank k is a relevant document, zero otherwise. Then,

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \quad (14)$$

where Q is the number of queries. In our case, we consider relevance judgments 1 and 2 as non-relevant, and 3, 4, and 5 as relevant. MAP was computed after looking at the first ten retrieved documents.

The reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer ($rank_i$). MRR is the average of the reciprocal ranks of results for a sample of queries Q

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \quad (15)$$

In our case, we consider relevance judgments 1 and 2 as incorrect answers, and 3, 4 and 5 as correct answers. MRR was computed after looking at the first ten retrieved documents. We observed that minor changes in these conventions do not alter the general trends of the results.

We also introduced two new metrics this year. First is *Recall@10* or R@10, defined as follows:

$$R@10(q_i) = \frac{\#(\text{relevant doc retrieved for } q_i \text{ in top 10 ranks})}{\max(\#(\text{relevant doc for } q_i), 10)} \quad (16)$$

$$R@10 = \frac{1}{|Q|} \sum_{i=1}^{|Q|} R@10(q_i) \quad (17)$$

The second metric, *Cross-Script Recall* or *csR@10* measures the *R@10* while considering only those documents which are in a different script than that of the query. Mixed-script documents are

ignored while computing *csR@10*. Thus, if the query is in Roman script, we compute the recall on a subset of relevant documents for the query which are only in Devanagari, and vice versa. *csR@10* indicates the cross-script retrieval power of a system, which is not explicitly captured by any of the other aforementioned metrics.

5.2 Subtask 1

Tables 7 and 8 list out the results of all the submissions for subtask-1 that were well formatted. We had several other runs which could not be evaluated because of formatting issues[3]. One team, though had well formatted runs and corresponding evaluation scores, decided to withdrew their submission and hence is not included in this report.

We observe that there is no single run which has the best score across all the metrics. Therefore, we decided to use the average of 5 different metrics: LF, EF, TF, LA and ETPM (normalized by the max. ETPM for all submissions for that language pair) to rank the runs. The best performing run for each language pair according to this score is marked with an asterisk in Table 7. Note that some teams did not generate the transliterations, and their ETPM score was assumed to be 0.

Interestingly, five different teams have topped in the different language pairs. Some teams, such as JU-NLP-LAB, DA-IR and IITP-TS, participated in one or two language pairs. They seem to have fine-tuned their system for those languages and performed very well in the respective language tracks. These three teams respectively topped in the Bangla-English, Gujarati-English and Hindi-English tracks. Only two teams, IIITH and BMS-Brainz, participated in all the language pairs. Some salient observations about the performance of the different systems are described below.

- Two teams (Asterish and BITS-Lipyantaran) used Google transliteration API for Hindi, and they have the highest TF scores. IITP-TS comes close with their indigenous transliteration system.

- The teams which used machine learning on token based and n-gram features have higher labeling accuracy than the teams which only relied on dictionaries and rules. Team Salazar is a notable exception, though. Their LA is comparable to, if not better than, most of the other runs that use machine learning.

- It is difficult to estimate the inherent hardness associated with the languages, if any, for this task because the datasets are not uniform and the amount of data released is also different. Moreover, most of the teams submitted in only one or two languages and a fair comparison across languages cannot be done. Nevertheless, we see very high LA scores for the three Dravidian languages in spite of the fact that no training data was released for these languages. On the other hand, Hindi seemed to be the hardest language to tackle for this task. This might be because Dravidian languages have a rich morphology and large number inflections which can help one to detect the words easily. Bangla and Gujarati has fewer inflections and Hindi has still fewer.

- Transliteration seems to be a hard problem in general. The best TF score, which is for Hindi, is only 0.304. There is a large room for improvement in back-transliteration for Indic languages.

[3]The participants were intimated about the formatting errors and allowed to resubmit several times till the automatic evaluation script could handle the submission. Nevertheless, some teams did not or could not submit the run in the required format, and had to be discarded.

5.3 Subtask 2

The test collection for Subtask 2 contained 35 queries in Roman and Devanagari scripts. The Devenagari queries were intentionally kept simple and relatively unambiguous, whereas the Roman queries were of varying difficulty. They featured world level joining and splitting as well as highly ambiguous short queries. Table 5.3 presents the results of the 7 runs received. We observe that the two runs from BITS-Lipyantran performs best across all the metrics. Using Devanagari as the working script, and mapping both the queries and documents to Devanagari must have helped them because in the native script their is usually one single correct spelling. Moreover, use of Google Transliteration API and word level n-grams for indexing and matching must have helped in improving the precision. For all the systems, csR@10 has the lowest absolute value, which implies that there is a scope for improving the cross-script retrieval. Systems are performing reasonably well when the scripts of the query and the document are the same. Probably errors introduced during transliteration or cross-script word mapping accumulates and eventually brings down the cross-script recall.

6. SUMMARY

The transliterated search shared task was introduced last year in FIRE 2013; we had received 17 runs from 5 teams in subtask 1 and 8 runs from 3 teams for subtask 2. This year, we had nearly a 3 fold increase in the teams and runs for subtask 1. In subtask 2, we had similar number of runs and teams. This clearly shows that the track is gaining popularity and has been successful in building a research community.

Subtask 1 is a very fundamental task and has applications much wider than transliterated search. While language labeling seems like an easy and solved task, the performance of the team in the shared task shows that for some languages like Hindi and Bangla, the best systems are only 90% accurate, which leaves a lot of room for research and improvement. Even for the other languages, where the accuracy of some systems have gone up to 98%, we believe this is due to the nature of the datasets rather than inherent simplicity of the problem. In the coming years, we would definitely like to create a bigger repository of annotated data and include more Indic languages and if possible, a few other languages. Transliteration is far from a solved problem and we need more awareness and data around it.

In subtask 2, this year we introduced Devanagari queries as well as more challenging Roman transliterated queries with realistic errors commonly seen in Web search queries. Here also we see that cross-script retrieval performance is still lagging behind the same-script retrieval performance. Furthermore, native script retrieval seems to be easier (e.g., average NDCG@5 over all the runs for the Devanagari queries 0.722, and for the transliterated Roman queries is 0.511). Among the Roman transliterated queries, the average NDCG@5 for queries with splitting or joining of words is 0.286, whereas for other transliterated queries the average is 0.617. Thus, developping a practical search engine for lyrics or transliterated search in general is very challenging, and there is a lot of scope for research and innovation.

As a final remark, we would like to mention that it is not possible to freely distribute the data collected for subtask 1 because it has been taken from various social media websites and blogs with various privacy policies. Right now the data is only available to those who participate in this shared task, and can be used only for this

Team	Run	NDCG@1	NDCG@5	NDCG@10	MAP	MRR	R@10	csR@10
BIT	1	0.5024	0.3967	0.3612	0.2698	0.5243	0.4343	0.2193
BIT	2	0.6452	0.4918	0.4572	0.3415	0.6271	0.4822	0.1898
BITS-Lipyantran	1	0.7500	0.7817	0.6822	0.6263	0.7929	0.6818	0.4144
BITS-Lipyantran*	2	0.7708	0.7954	0.6977	0.6421	0.8171	0.6918	0.4430
DCU	1	0.5786	0.5924	0.5626	0.4112	0.6269	0.4943	0.3483
DCU	2	0.4143	0.3933	0.3710	0.2063	0.3979	0.2807	0.3035
IIITH	1	0.6429	0.5262	0.5105	0.4120	0.6730	0.5806	0.3407

Table 9: Results for subtask II. The highest scoring team has been marked *.

shared task. We are trying our best to take the required permissions to make this data freely available for the community.

Acknowledgments

We would like to than Rishiraj Saha Roy, Adobe Research Lab Bangalore, for valuable suggestions and help in conducting the workshop on the shared task. We are also grateful to all the people who have voluntarily contributed to the datasets: Yesha Shah, Swati Jhawar, Ria Gupta, Prof Dinesh Babu J, Kumaresh Krishnan, P. S. Srinivasan, Rekha Vaidyanathan, Dr.Shambhavi B R, Dr.Sagar B M, Sandesh, Shwetha Kulkarni, and Abhishek J.

7. REFERENCES

[1] Roy, R.S., Choudhury, M., Majumder, P., Agarwal, K.: Overview and datasets of fire 2013 shared task on transliterated search. In: Working notes of FIRE. (2013)

[2] 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

[3] Gupta, K., Choudhury, M., Bali, K.: Mining hindi-english transliteration pairs from online hindi lyrics. In: LREC. (2012) 2459–2465

[4] Sowmya, V., Choudhury, M., Bali, K., Dasgupta, T., Basu, A.: Resource creation for training and testing of transliteration systems for indian languages. In: LREC. (2010)

[5] Quasthoff, U., Richter, M., Biemann, C.: Corpus portal for search in monolingual corpora. In: Proceedings of the fifth international conference on language resources and evaluation. (2006) 1799–1802

[6] 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

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

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

[9] Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst. **20** (October 2002) 422–446

[10] Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc. (1986)

[11] Voorhees, E.M., Tice, D.M.: The TREC-8 Question Answering Track Evaluation. In: TREC-8. (1999) 83–105

Team	Run ID	LF	EF	LA	$EQMF_2$
Bangla-English					
BMS-Brainz	1	0.701	0.781	0.776	0.29
IIITH	1	0.833	0.861	0.85	0.383
IITP-TS	1	0.88	0.907	0.886	0.411
IITP-TS	2	0.881	0.907	0.886	0.41
IITP-TS	3	0.861	0.888	0.87	0.379
ISI	1	0.835	0.882	0.862	0.378
JU-NLP-LAB*	1	0.899	0.92	0.905	0.444
JU-NLP-LAB	2	0.899	0.92	0.905	0.444
Gujarati-English					
BMS-Brainz	1	0.856	0.071	0.746	0.173
DA-IR*	1	0.981	0.2	0.963	0.847
IIITH	1	0.923	0.145	0.856	0.387
Hindi-English					
asterisk	1	0.782	0.803	0.654	0.126
BITS-Lipyantaran	1	0.835	0.827	0.838	0.205
BITS-Lipyantaran	2	0.82	0.813	0.826	0.177
DA-IR	1	0.778	0.75	0.771	0.153
I1	1	0.806	0.797	0.807	0.195
I1	2	0.756	0.664	0.738	0.165
IIITH	1	0.787	0.794	0.792	0.143
IITP-TS*	1	0.908	0.899	0.879	0.269
IITP-TS	2	0.907	0.899	0.878	0.265
IITP-TS	3	0.885	0.873	0.857	0.209
ISMD	1	0.895	0.878	0.872	0.269
ISMD	2	0.911	0.901	0.886	0.276
ISMD	3	0.911	0.901	0.886	0.276
PESIT-CS-FIRE-IR	1	0.81	0.782	0.654	0.157
PESIT-CS-FIRE-IR	2	0.812	0.782	0.656	0.158
Salazar	1	0.883	0.857	0.855	0.231
Sparkplug	1	0.693	0.641	0.599	0.053
Kannada-English					
BMS-Brainz*	1	0.894	0.681	0.836	0.218
I1	1	0.892	0.757	0.848	0.269
IIITH	1	0.932	0.854	0.9	0.429
Malayalam-English					
BMS-Brainz	1	0.851	0.588	0.785	0.217
IIITH*	1	0.928	0.86	0.891	0.383
Tamil-English					
BMS-Brainz	1	0.705	0.816	0.799	0.122
IIITH*	1	0.985	0.986	0.986	0.714

Table 7: Subtask 1, language identification: Performance of submissions. * indicates the best performing team for each language pair.

Team	Run ID	TF	EQMF	ETPM
Bangla-English				
IIITH	1	0.021	0.004	72/288
IITP-TS	1	0.073	0.005	228/337
IITP-TS	2	0.073	0.005	228/337
IITP-TS	3	0.071	0.005	231/344
ISI	1	0.053	0.004	174/309
JU-NLP-LAB	1	0.062	0.005	227/364
JU-NLP-LAB	2	0.037	0.004	134/364
Gujarati-English				
DA-IR	1	0.463	0.02	492/1035
IIITH	1	0.261	0.007	259/911
Hindi-English				
Asterisk	1	0.304	0.002	1605/1936
BITS-Lipyantaran	1	0.258	0.005	1923/2156
BITS-Lipyantaran	2	0.252	0.004	1876/2109
DA-IR	1	0.163	0.001	1330/2153
IIITH	1	0.122	0.001	907/2004
IITP-TS	1	0.244	0.005	1933/2306
IITP-TS	2	0.244	0.004	1931/2301
IITP-TS	3	0.24	0.004	1871/2226
ISMD	1	0.217	0.001	1616/2203
ISMD	2	0.118	0.001	924/2251
ISMD	3	0.204	0	1596/2251
PESIT-CS-FIRE-IR	1	0.112	0	895/2157
PESIT-CS-FIRE-IR	2	0.152	0.001	1238/2158
Salazar	1	0.15	0	1086/2235
Sparkplug	1	0.208	0.001	1214/1333
Kannada-English				
BMS-Brainz	1	0.525	0	433/732
IIITH	1	0	0	0/751
Malayalam-English				
IIITH	1	0.098	0.008	90/852

Table 8: Subtask 1, transliteration: Performance of submissions