Overview and Datasets of FIRE 2013 Track on Transliterated Search

Rishiraj Saha Roy[1], Monojit Choudhury[2], Prasenjit Majumder[3] and Komal Agarwal[3]

[1] Indian Institute of Technology Kharagpur, Kharagpur - 721302, West Bengal, India.
rishiraj@cse.iitkgp.ernet.in
[2] Microsoft Research India, Bangalore - 560001, Karnataka, India.
monojitc@microsoft.com
[3] DA-IICT, Gandhinagar - 382007, Gujarat, India.
{prasenjit.majumder, komalagarwal07}@gmail.com

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 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* [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 that are used 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[4] 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.

This being the first year, we hosted a query labeling subtask, which is one of the first steps before one can tackle the bigger problem, and 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. In the first subtask, participants had to classify words in a query as English or a transliterated form of an Indian language

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

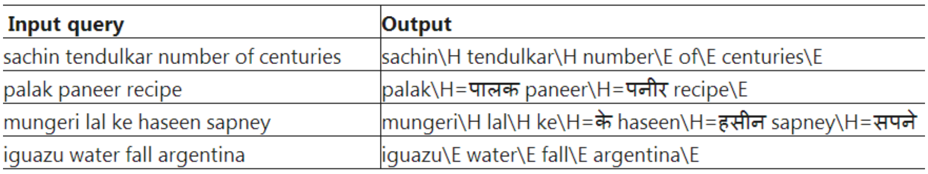
word. In the latter case, they also had to provide the correct transliteration the native script. In the second subtask, participants had to retrieve the top few documents from a multilingual corpus with queries in Roman script. In the coming years, we plan to expand these tasks to more languages and more domains; we also plan to host other sub-tasks related to transliterated search.

This paper provides the overview and datasets of the transliterated search track at the fifth Forum for Information Retrieval Conference 2013 (FIRE '13). The track was coordinated jointly by Microsoft Research India, IIT Kharagpur and DAIICT Gandhinagar. We are happy with the initial response that we have received and surely plan to continue this track in future FIRE conferences. Specifically, we received participation from five teams, three from India and two from abroad (Spain and Norway). A total of 25 runs were submitted in total for the two tasks. This paper is organized as follows. First, we present the shared tasks floated 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. These are described in separate sub-sections in this section, and similar parallelism has been maintained while discussing datasets, submissions and results in the rest of this paper. These details of the tasks can be found at the track website `http://bit.ly/1ewsOqv`.

2.1 Subtask 1: Query Word Labeling



Input query	Output
sachin tendulkar number of centuries	sachin\H tendulkar\H number\E of\E centuries\E
palak paneer recipe	palak\H=पालक paneer\H=पनीर recipe\E
mungeri lal ke haseen sapney	mungeri\H lal\H ke\H=के haseen\H=हसीन sapney\H=सपने
iguazu water fall argentina	iguazu\E water\E fall\E argentina\E

Fig. 1. Examples for subtask 1.

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, $\ldots$, w_n, could be standard English words or transliterated from another language L. The task is to label the words as E or L depending on whether it an English word, or a transliterated L-language word [4]. Then, for each transliterated word, the correct transliteration has to be provided in the native script (i.e., the script which is used for writing L).

Names of people and places in L should be considered as transliterated entries, whenever it is a native name. Thus, *Arundhati Roy* is a transliterated name, but *Ruskin Bond* is not (these may be labeled with a neutral tag

N). But transliterations of such names will not be evaluated. We floated this task for three language pairs – English-Hindi, English-Bangla, and English-Gujarati. Examples for the labeling task are provided in Figure 1. For these queries, L is Hindi and has to be labeled as H.

2.2 Subtask 2: Multi-script Ad hoc retrieval for Hindi Song Lyrics

Given a query in Roman script, the system has to retrieve the top-k documents from a corpus that has documents in mixed script (Roman and Devanagari). The input is a query written in Roman script, which is a transliterated form of a (possibly partial or incorrect) Hindi song title or some part of the lyrics. The output is a ranked list of ten ($k = 10$ here) songs both in Devanagari and Roman scripts, retrieved from a corpus of Hindi film lyrics, where some of the documents are in Devanagari and some in Roman transliterated form.

3 Datasets

In this section, we describe the datasets that have been released along with our tasks 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 track data 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 [5].

   (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 [6].

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[5]. 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 [6].
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 [7] 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.

3.2 Subtask 1

We initially provided 500, 100 and 150 labelled queries for English, Bangla and Gujarati respectively. These contain 1056, 298 and 546 distinct word transliteration pairs respectively. Due to the small size of the data, we did not recommend the use of these for training participant algorithms, but rather as a development set for tuning model parameters and understanding and analyzing word transliteration pairs. These were provided as text files that must be opened in UTF-8 encoding to properly view contents. We advised the use of Notepad++ `http://notepad-plus-plus.org/` (Encoding –> Encode in UTF-8). We provided a further 500, 100 and 150 unlabelled queries as test data for English, Bangla and Gujarati respectively.

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

Team name	Team name (short)	No. of runs (Subtask 1)	No. of runs (Subtask 2)
TU Valencia, Spain	TUVal	3 (Hindi)	3
Microsoft Research India	MSRI	3 (Hindi), 3 (Bangla), 3 (Gujarati)	No participation
NTNU Norway	NTNU	1 (Hindi), 1 (Bangla)	3
Gujarat University	GU	2 (Hindi)	2
ISM Dhanbad	ISM	1 (Hindi)	No participation

Table 1. Team and run details for both subtasks.

Even though the entries to be transliterated have been termed as “queries” (*munshi premchand ebook collection*), they can also be natural language (NL) sentences (*mera ghar rubbery factory ke pass hai*) or text fragments (*main jab school se waapas aa rahaa tha*). The entries contain words like *jab*, *hum*, *main*, *tan*, and *man*, which have interpretations both in English and in some other language (Hindi in this particular case). The base language can be English (*munshi premchand ebook collection*) or Hindi (*main jab school se waapas aa rahaa tha*). The idea was to have text fragments on which standard parsers and POS taggers will not work accurately. Hence, participants will be driven to use empirical methods like context modeling. Even though the cases are philosophically different, to a typical algorithm these cases will not make any difference. We note that out-of-vocabulary (OOV) words could possibly be easily identified from unnatural POS patterns had the entries followed a complete NL sentential form (for English or another language). The domains of these entries were diverse, like technology (*internet explorer baar baar crash ho raha hai*), Bollywood movies (*jab we met full cast, hum tum movie review*), and travel (*new york mein ghoomne ki jagah, uttar pradesh tourism guide*). Participants were allowed to use any external resource. The data creators comprised of a mix of undergraduate, post-graduate and graduate students in the age group of $22 - 25$ years, and were native speakers of the language for which they provided data.

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 $62,888$ documents which contained song titles and lyrics in Roman (ITRANS or plain format), Devanagari and mixed scripts. The test set also consisted of twenty five queries. On an average, there were 28.38 qrels per query. The mean query length was 4.5 words. The song lyrics documents were created by crawling several popular domains like *dhingana*, *musicmaza* and *hindilyrix*.

4 Submissions overview

One team each from five institutes participated in our shared tasks – TU Valencia (Spain), Microsoft Research India (MSRI), NTNU Norway, Gujarat University and Indian School of Mines (ISM) Dhanbad. We note that MSRI is a participating but non-competing team as MSRI is also one of the organizers of this track. Out of these five teams, three teams (TU Valencia, NTNU Norway and Gujarat University) participated in both subtasks 1 and 2. The other two teams (Microsoft Research India and ISM Dhanbad) participated only in subtask 1. The teams, their short names used subsequently in the results section (Sec. 5), and the number of runs submitted for each subtask are shown in Table 1. Language specific numbers for subtask 1 are also reported. The top three constributors in terms of total runs submitted are MSRI, TU-Valencia, and NTNU Norway. Initially, after floating the task, we received a show of interest from eighteen teams, from fifteen institutes, which is quite encouraging for a track in its first year. However, teams dropped out at various stages of task. Analyzing the reasons and creating better awareness about the track will remain our focus for the coming year.

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

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.

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

$$\text{Exact transliteration pair match (ETPM)} = \frac{\#(\text{Pairs for which translits. match exactly})}{\#(\text{Pairs for which both o/p and reference labels are L})} \tag{2}$$

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 [2] 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.

$$\text{Transliteration precision (TP)} = \frac{\#(\text{Correct transliterations})}{\#(\text{Generated transliterations})} \tag{3}$$

$$\text{Transliteration recall (TR)} = \frac{\#(\text{Correct transliterations})}{\#(\text{Reference transliterations})} \tag{4}$$

$$\text{Transliteration F–score (TF)} = \frac{2 \times TP \times TR}{TP + TR} \tag{5}$$

$$\text{Labelling accuracy (LA)} = \frac{\#(\text{Correct label pairs})}{\#(\text{Correct label pairs}) + \#(\text{Incorrect label pairs})} \tag{6}$$

Correct label pairs imply $E{-}E$ and $L{-}L$, while incorrect label pairs include $E{-}L$ and $L{-}E$, where E is for English and L stands for the language initial (H, B and G for Hindi, Bangla and Gujarati respectively). Note that the notation $E{-}L$ implies “Output: E, Reference: L”, and likewise for the other pairs.

$$\text{English precision (EP)} = \frac{\#(\text{E}-\text{E pairs})}{\#(\text{E}-\text{L pairs}) + \#(\text{E}-\text{E pairs})} \tag{7}$$

$$\text{English recall (ER)} = \frac{\#(\text{E}-\text{E pairs})}{\#(\text{L}-\text{E pairs}) + \#(\text{E}-\text{E pairs})} \tag{8}$$

$$\text{EnglishF}-\text{Score(EF)} = \frac{2 \times EP \times ER}{EP + ER} \tag{9}$$

Similarly, we have L-precision, L-recall, and $L-$F-Score for L where L is Hindi, Bangla and Gujarati. We note that all names with ambiguous L-labelling and transliterations were excluded from the analysis.

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 case, *chandrabindu* replaced by *bindu* and the non-obligatory use of the *nukta*.

Subtask 2 For evaluating subtask 2, we used the well-established IR metrics of normalized Discounted Cumulative Gain (nDCG) [8], Mean Average Precision (MAP) [9] and Mean Reciprocal Rank (MRR) [10].

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

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2 i} \tag{10}$$

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

$$nDCG@p = \frac{DCG@p}{IDCG@p} \tag{11}$$

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

$$AveP = \frac{\sum_{k=1}^{n}(P(k) \times rel(k))}{\text{No. of relevant documents}} \tag{12}$$

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,

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \tag{13}$$

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

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{14}$$

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.

5.2 Subtask 1

Results for subtask 1 for Hindi are presented in Table 2, for metrics defined earlier. MSRI (non-competing team) has performed the best for Hindi, but several other runs have very good performance on many metrics. The numbers of true $\backslash H$, $\backslash E$ and $\backslash N$ tags were 2444, 777 and 232 respectively. Among the MSRI runs, run-2 had the best metrics. Since MSRI is non-competing, TU-Valencia is declared the winner in this task for Hindi as its runs have the second best performance on five of the twelve metrics (all of its runs have the highest LA, EP, EF, LR and LF). ISM Dhanbad comes a close second with best performance on four of the twelve metrics ($EQMF$, TP, TR and TF) and moderately good performance on the others. Gujarat University scored the

Metric	ISM	NTNU	GU-1	GU-2	TUVal-1	TUVal-2	TUVal-3	MSRI-1	MSRI-2	MSRI-3	Max	Median
EQMF	0.086	0.000	0.036	0.002	0.022	0.020	0.006	0.194	**0.198**	0.186	0.198	0.029
ETPM	1584/ 2117	540/ 1829	880/ 1853	316/ 1851	1038/ 2392	1063/ 2392	936/ 2392	1985/ 2414	1985/ 2417	1979/ 2415	N. A.	N. A.
TP	0.725	0.292	0.470	0.169	0.417	0.427	0.376	0.813	**0.814**	0.810	0.814	0.449
TR	0.648	0.221	0.360	0.129	0.425	0.435	0.383	**0.813**	**0.813**	0.810	0.813	0.430
TF	0.685	0.252	0.408	0.147	0.421	0.431	0.380	**0.813**	**0.813**	0.810	0.813	0.426
LA	0.878	0.803	0.811	0.810	0.954	0.954	0.954	0.982	**0.985**	0.983	0.985	0.954
EP	0.685	0.552	0.562	0.562	0.930	0.930	0.930	0.963	**0.967**	0.964	0.967	0.930
ER	0.914	0.972	**0.976**	**0.976**	0.875	0.875	0.875	0.964	0.970	0.964	0.976	0.964
EF	0.783	0.704	0.713	0.713	0.902	0.902	0.902	0.963	**0.969**	0.964	0.969	0.902
LP	0.969	0.988	0.990	0.990	0.961	0.961	0.961	0.989	**0.991**	0.989	0.991	0.988
LR	0.867	0.749	0.759	0.758	0.979	0.979	0.979	0.988	**0.989**	**0.989**	0.989	0.979
LF	0.915	0.852	0.859	0.858	0.970	0.970	0.970	0.988	**0.990**	0.989	0.990	0.970

The highest value(s) in a row (among submitted runs) is marked in **boldface**.

Table 2. Subtask 1 results for Hindi.

best on two metrics (ER and LP, run-1). We will devise a more sophisticated ranking scheme for the teams from next year based on the weighted average of performance on each of the metrics, rather than by simple majority of numbers of peak metric attainments. It is quite possible that a run does not emerge the best in any metric yet achieve very good performance on all the metrics, while another run can be the best in some and the worst on some other metrics. The former algorithm would be the one to resort to in a practical setup. However, such an evaluation is meaningful only when the number of submitted runs is somewhat higher than what we had this year, and we expect that to be the case in future versions of this track.

We have the following two evaluation variants for Bangla. There are many cases where there occurs *morpheme level mixing*. For example, *cinemar, torrenter* and *bulbtar*. The ideal labeling for such cases would be *cinema*$\backslash E$ *ar*$\backslash B$, along with the transliteration for *ar*; similarly for *torrent\E er\B* and *bulb\E tar\B*. However, since none of the submitted runs accounted for such cases, we performed two analyses – one where such cases are excluded from analysis (Table 3 left half), and one where we penalized them (Table 3 right half). Expectedly, the relevant metric values (TP, TF, LA, LP and LF) in the former case are slightly higher. The numbers of true $\backslash B$, $\backslash E$ and $\backslash N$ tags were 387, 120 and 59 respectively. MSRI and NTNU participated for Bangla, and among the two MSRI (non-competing team) performed much better (runs 2 and 3). NTNU had the best performance for ER and LP. The metric values for these teams being lower for Bangla with respect to their performances for Hindi is explained by the fact that the Bangla dataset, though much smaller (200 queries) than Hindi (1000 queries), was much more carefully crafted with several Bangla words which were English dictionary entries. Thus, performing well on the Bangla dataset is expected to be harder. Since MSRI is non-competing, NTNU is declared the winner in this task for Bangla as its run has the second best performance on all the metrics.

Metrics	NTNU	MSRI-1	MSRI-2	MSRI-3	Max	Median	NTNU	MSRI-1	MSRI-2	MSRI-3	Max	Median
EQMF	0.000	**0.010**	**0.010**	**0.010**	0.010	0.010	0.000	**0.010**	**0.010**	**0.010**	0.010	0.010
ETPM	59/ 242	186/ 360	193/ 371	197/ 370	N. A.	N. A.	59/ 242	186/ 360	193/ 371	197/ 370	N. A.	N. A.
TP	0.239	0.503	0.508	**0.518**	0.518	0.505	0.232	0.495	0.500	**0.510**	0.510	0.497
TR	0.153	0.481	0.499	**0.509**	0.509	0.490	0.153	0.481	0.499	**0.509**	0.509	0.490
TF	0.186	0.491	0.503	**0.514**	0.514	0.497	0.184	0.488	0.499	**0.510**	0.510	0.494
LA	0.699	0.926	**0.950**	0.946	0.950	0.936	0.690	0.915	**0.939**	0.935	0.939	0.925
EP	0.425	0.791	**0.866**	0.857	0.866	0.824	0.425	0.792	**0.867**	0.858	0.867	0.825
ER	**0.955**	0.911	0.920	0.911	0.955	0.915	**0.899**	0.866	0.874	0.866	0.899	0.870
EF	0.588	0.847	**0.892**	0.883	0.892	0.865	0.577	0.827	**0.870**	0.862	0.870	0.845
LP	**0.980**	0.973	0.976	0.974	0.980	0.975	0.953	0.957	**0.961**	0.959	0.961	0.958
LR	0.625	0.930	**0.959**	0.956	0.959	0.943	0.625	0.930	**0.959**	0.956	0.959	0.943
LF	0.763	0.951	**0.967**	0.965	0.967	0.958	0.755	0.944	**0.960**	0.957	0.960	0.951

The highest value(s) in a row (among submitted runs) is marked in **boldface**.

Table 3. Subtask 1 results for Bangla (left half: morpheme-level mixing errors ignored; right half: morpheme-level mixing error penalized).

Subtask 1 results for Gujarati are presented in Table 4. The numbers of true $\backslash G$, $\backslash E$ and $\backslash N$ tags were 1046, 12 and 16 respectively. The very high metric values for LP, LR and LF are a consequence of the fact that 97% of all tokens in the dataset were Gujarati. We will work towards improving the current skewness in the tags next year. Only MSRI (non-competing team) submitted runs for Gujarati, and among their runs, run-2 performed the best on majority of the metrics. However, runs-1 and 3 were close behind. Since MSRI is non-competing, there is no winner in this task for Gujarati.

The methodologies and error analyses for each of these runs (all languages) can be understood through the individual participant working notes for this track.

5.3 Subtask 2

Subtask 2 results are presented in Table 5. For subtask 2, NTNU, GU and TU-Val submitted three, two and three runs respectively. Out of these, run-2 of TU-Val performed the best on all the four metrics. Thus, TU-Valencia is decalred the winner in this track. The second best performance was also achieved by TU-Valencia, whose run-1 performed the second best on three of the four metrics (nDCG@5, nDCG@10 and MAP). The third best performance was again by the run-3 of TU-Valencia. So TU-Valencia clearly dominated in subtask 2. The performance of the GU runs and NTNU-2 and 3 were comparable, while NTNU-1 resulted in the poorest performance.

6 Summary

The first edition of the transliterated search task has been a success. The participation has been encouraging and we plan to continue the task in subsequent FIRE conferences. We received a total of 25 run submissions from five different teams across the

Metric	MSRI-1	MSRI-2	MSRI-3	Max	Median
EQMF	**0.080**	0.073	0.067	0.080	0.073
ETPM	485/ 1009	499/ 1026	490/ 1014	N. A.	N. A.
TP	0.481	**0.485**	0.483	0.485	0.483
TR	0.462	**0.475**	0.467	0.475	0.467
TF	0.471	**0.480**	0.475	0.480	0.475
LA	0.961	**0.976**	0.966	0.976	0.966
EP	0.226	**0.294**	0.250	0.294	0.250
ER	**1.000**	0.833	**1.000**	1.000	1.000
EF	0.369	**0.435**	0.400	0.435	0.400
LP	**1.000**	0.998	**1.000**	1.000	1.000
LR	0.961	**0.977**	0.966	0.977	0.966
LF	0.980	**0.988**	0.983	0.988	0.983

The highest value(s) in a row (among submitted runs) is marked in **boldface**.

Table 4. Subtask 1 results for Gujarati.

Metric	NTNU-1	NTNU-2	NTNU-3	GU-1	GU-2	TUVal-1	TUVal-2	TUVal-3	Max	Median
nDCG@5	0.205	0.523	0.561	0.563	0.526	0.767	**0.805**	0.758	0.805	0.562
nDCG@10	0.207	0.520	0.560	0.562	0.523	0.764	**0.800**	0.753	0.800	0.561
MAP	0.003	0.152	0.197	0.255	0.216	0.421	**0.424**	0.356	0.424	0.234
MRR	0.018	0.555	0.593	0.584	0.573	0.775	**0.844**	0.777	0.844	0.588

The highest value in a row (among submitted runs) is marked in **boldface**.

Table 5. Subtask 2 results.

world (three from India and two from abroad). All teams chose to participate in the language labeling task (five teams, twenty runs) but there was significant interest in the cross lingual retrieval task as well (three teams, eight runs). The winners of the two tasks are summarized here (Table 6). The peak metric values for TF, a representative metric for subtask 1, are 0.813, 0.510 and 0.480 for Hindi, Bangla and Gujarati respectively. Highest nDCG@10 for subtask 2 was reported at 0.8. Multi-lingual information retrieval is still a challenging problem and as seen from the results, there is a lot of scope for improvement in the techniques used for this task. We plan to enrich and expand our datasets with interactive feedback with the partipants. We wish to improve the quality of our reference annotations by engaging more coders and striving for high inter-annotator agreement through multiple rounds of training. We will also make the result analysis more comprehensive in upcoming editions, where we expect a significantly higher number of submissions. Overall, we look forward in excitement to organizing this task again at FIRE 2014 with increased enthusiasm and awareness.

Subtask	**Winner**	**Comments**
Subtask 1 (Hindi)	TU-Valencia	Winning run best in 5/12 metrics
Subtask 1 (Bangla)	NTNU-Norway	Winning run best in 12/12 metrics
Subtask 1 (Gujarati)	N. A.	N. A.
Subtask 2	TU-Valencia	Winning run best in 4/4 metrics

Table 6. Winners of subtasks.

Acknowledgments

The first author was supported by Microsoft Corporation and Microsoft Research India under the Microsoft Research India PhD Fellowship Award. We would like to thank Rohan Ramanath from Carnegie Mellon University and Ranita Biswas, Swadhin Pradhan, M. Dastagiri Reddy and Yogarshi Vyas from IIT Kharagpur for helping us with data creation. 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 oppotunity and platform for conducting this new track smoothly.

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. Gupta, K., Choudhury, M., Bali, K.: Mining hindi-english transliteration pairs from online hindi lyrics. In: LREC. (2012) 2459–2465
6. 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)
7. 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
8. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst. **20** (October 2002) 422–446
9. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc. (1986)
10. Voorhees, E.M., Tice, D.M.: The trec-8 question answering track evaluation. In: TREC-8. (1999) 83–105