Crosslingual Information Retrieval System Enhanced with Transliteration Generation and Mining

K Saravanan **Raghavendra Udupa** **A Kumaran**

Microsoft Research India
Bangalore, INDIA
{v-sarak,raghavu,kumarana}@microsoft.com

Abstract

This report documents the participation of Microsoft Research India (MSR India) in the Crosslingual Information Retrieval (CLIR) evaluation organized by the Forum for Information Retrieval Evaluation 2010 [FIRE 2010]. MSR India participated in two crosslingual evaluation tasks, namely the Hindi-English and Tamil-English crosslingual tasks, in addition to the English-English monolingual task. Our core CLIR engine employed a language modeling based approach using query likelihood based document ranking and a probabilistic translation lexicon learned from English-Hindi and English-Tamil parallel corpora. In addition, we employed two specific techniques to deal with out-of-vocabulary terms in the crosslingual runs: first, generating transliterations directly or transitively, and second, mining possible transliteration equivalents from the documents retrieved in the first-pass. We show experimentally that each of these techniques significantly improved the overall retrieval performance of our crosslingual IR system. Our system, using all of the topic-description-and-narrative information, achieved the peak retrieval performance of a MAP of 0.5133 in the monolingual English-English task; in crosslingual tasks, our systems achieved a peak performance of a MAP of 0.4977 in Hindi-English and 0.4145 in the Tamil-English. The post-task analyses indicate that the mining of appropriate transliterations from the top results of the first-pass retrieval achieved enhanced the crosslingual performance of our system overall, in addition to enhancing individual performance of more queries. Our Hindi-English crosslingual retrieval performance was nearly equal (~97%) to the English-English monolingual retrieval performance, indicating the effectiveness of our approaches to handle OOV's to enhance the baseline performance of our CLIR system.

1 Introduction

Evaluation of monolingual and crosslingual information retrieval systems has a long history, with successful campaigns of Cross-Language Evaluation Forum (CLEF) [CLEF] in European languages and NTCIR [NTCIR] in Chinese-Japanese-Korean languages. However, it is only in the recent past that evaluation included Indian languages. The 2006 edition of CLEF campaign is the first of such kind, and had Hindi-English and Tamil-English ad hoc crosslingual tracks [Peters, 2006]. The 2007 CLEF campaign introduced a special sub-task for Indian languages [Nardi and Peters, 2007], that included topics in 3 Indian languages, specifically, Hindi, Telugu, and Marathi, and a common English document collection. From 2008, the Forum for Information Retrieval Evaluation [FIRE], modeled after the highly successful CLEF and NTCIR campaigns, focused specifically on Indian languages and English. As a part of the FIRE initiative, document collections have been developed for some Indian languages, namely, Hindi, Marathi, Bangla, and English [Majumdar et al., 2008].

The Multilingual Systems group at Microsoft Research India participated in the CLEF 2007 campaign, in the Hindi-English track achieving a retrieval performance that was the second highest among all submitted runs [Jagadeesh and Kumaran, 2007]. In 2008, as a part of the FIRE campaign, we participated in English monolingual and Hindi-English crosslingual tracks [Udupa et al., 2008], achieving the best retrieval performance in each of these tracks. This year, FIRE organized several ad hoc monolingual and crosslingual retrieval tracks, and we participated in the English monolingual and crosslingual Hindi-English and Tamil-English ad hoc retrieval tracks. In this report, we detail our system and participation, and the performance of our official runs.

2 Retrieval System

2.1 Monolingual Retrieval Model

Our monolingual retrieval system is based on the well-known Language Modeling framework to information retrieval [Ponte and Croft, 1998; Zhai and Lafferty, 2004]. In this framework, the queries as well the documents are viewed as probability distributions. The similarity of a query (q) with a document (d) is measured in terms of the likelihood of the query under the document language model (or equivalently, as the Kullback-Leibler divergence of query and document unigram language models):

$$Score(q,d) = \sum_{w} p(w \mid q) \log p(w \mid d)$$

where w is the term in the lexicon. For a detailed description and discussion of the Language Modeling framework, please see [Ponte and Croft, 1998; Zhai and Lafferty, 2002; Zhai and Lafferty, 2004]. We smooth the document language model by interpolating with a corpus language model:

$$p_{sm}(w \mid d) = (1-\alpha) p_{mle}(w \mid d) + \alpha p(w \mid C)$$

2.2 Crosslingual Retrieval Model

In our CLIR model, the query in a source language (q_s) is translated into the target language – English – (q_t) using a probabilistic translation lexicon:

$$p(w_t \mid q_s) = \sum_{w_s} p(w_s \mid q_s) p(w_t \mid w_s)$$

Where, w_s is a source language term and w_t is a target language term. Note that the English translation (q_t) of the query need not have a surface realization. Nevertheless, the similarity of the translated query (q_t) with a document (d_t) is measured in terms of the Kullback-Leibler divergence of the query and the document language models, similar to the monolingual case:

$$Score(q_s, d_t) = \sum_{w_t} p(w_t \mid q_t) \log p(w_t \mid d_t)$$

$$= \sum_{w_t, w_s} p(w_s \mid q_s) p(w_t \mid w_s) \log p(w_t \mid d_t)$$

2.3 Handling Out-of-Vocabulary terms

Like any crosslingual system that makes use of a translation lexicon, we too faced the problem of out-of-vocabulary (OOV) query terms. Many of the OOV terms are named entities that can be transliterated to the target language.

To handle these OOV terms, we used two different approaches – Transliteration Generation and Transliteration Mining from the first-pass retrieval.

1. In Transliteration Generation, the transliterations of the OOV terms are generated using an automatic Machine Transliteration system – directly or transitively [Khapra et al., 2010].
2. In Transliteration Mining, the transliteration equivalents of the query OOV terms are mined from the top-retrieved documents from the first pass, which are subsequently used in the final retrieval [Udupa et al., 2009-a].

2.3.1 Transliteration Generation

In this section, we discuss two different methods of generating transliterations in a target language, for a given source language OOV term – Direct and Transitive. In direct transliteration, the OOV terms are directly transliterated using a transliteration system trained on source-target language parallel names corpora. In transitive transliteration, we use a two-stage transliteration system, transitioning through an intermediate language; such systems are useful, and perhaps the only possibility, when sufficient parallel data between source and target languages are not available, directly.

2.3.1.1 Direct Transliterations

The systematic comparison of the various transliteration systems in the NEWS-2009 workshop [Li et al., 2009] showed conclusively that orthography based discriminative models like Conditional Random Fields [Lafferty, 2001] performed well in a language-neutral manner. Hence, to design a CLIR platform that may scale well with languages, we developed a basic transliteration system based on Conditional Random Fields, with an optional language origin detection module. Such a system, trained on source-target language parallel names corpora was used for generating transliterations of OOV terms between source and target languages.

For word origin detection, we manually classified 3000 words from the training set into words of Indic origin and Western origin. Two n-gram language models were built, for each of the Indic origin and Western origin names, to classify all the name pairs in the training set as Indic or Western names. Manual verification showed that this method about 97% accurate, yielding good quality data that is used for training two distinct CRF-based modules for transliterating Indic and Western names.

Conditional Random Fields [Lafferty, 2001] are undirected graphical models used for labeling sequential data. Under this model, the conditional probability distribution of the target string given the source string is given by,

$$p(Y / X; \lambda) = \frac{1}{Z(X)} \cdot e^{\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k (Y_{t-1}, Y_t, X, t)}$$

where,

X = *source string*
Y = *target string*
T = *length of source string*
K = *number of features*
λ_k = *feature weights*
$Z(X)$ = *normalization constant*

CRF++[1], an open source implementation of CRF was used for training and further transliterating the names. We used the alignment model developed by [Udupa et al. 2009-a] to get the character level alignments for the parallel names in the training corpora. Under this alignment, each character in the source word is aligned to zero or more characters in the corresponding target word. A transliteration engine was trained, based on a rich feature set generated based on this character-aligned data; the feature set includes aligned characters in each direction within a small distance (typically, 2) and source and target bigrams and trigrams.

2.3.1.2 Transitive Transliterations

Transitive transliterations systems combine multiple direct transliterations systems serially to produce transliterations between source language to target language. Specifically, we assume that the parallel names corpora are available between the language pair, X and Y, and the language pair, Y and Z; we train two baseline CRF based transliteration systems (as outlined in the earlier section), between the language X and Y, and Y and Z. We trained each of these systems, using parallel names corpora. Each name in language X was provided as an input into X⇒Y transliteration system, and the top-10 candidate strings in language Y produced by the system were further given as input into Y⇒Z system. The output of this system were merged and re-ranked by their probability scores. Finally, the top-10 of the merged output was output as the compositional system output

2.3.2 Mining Transliteration Equivalents

The mining algorithm issues the translated query minus OOV terms to the information retrieval system and mines transliterations of OOV terms from the top results. Hence, in the first pass, each query-result pair is viewed as a "comparable" document pair. The mining algorithm hypothesizes a match between an OOV query term and a document term in the "comparable" document pair and employs a transliteration similarity model to decide whether the document term is a transliteration of the query term [Udupa et al., 2009-a; Udupa et al., 2009-b]. Transliterations mined in this manner are then used to retranslate the query and issued again, for the final retrieval.

2.3.2.1 Transliteration Similarity Model

Our transliteration similarity model is an extension of W-HMM word alignment model presented in [He, 2007] and requires no language-specific knowledge. It is a character-level hidden alignment model that makes use of a richer local context in both the transition and emission models compared to the classic HMM model [Och and Ney, 2002]. The transition probability depends on both the jump width and the previous source character as in the W-HMM model. The emission probability depends on the current source character and the previous target character unlike the W-HMM model. The transition and emission models are not affected by data scarcity unlike Machine Translation as the character lexicon of a language is typically several orders smaller than its word lexicon. Instead of using any single alignment of characters in the pair (w_S, w_T), we marginalize over all possible alignments:

$$p(t_1^m \mid s_1^n) = \sum_{A} \prod_{j=1}^{m} p(a_j \mid a_{j-1}, s_{a_{j-1}}) p(t_j \mid s_{a_j}, t_{j-1})$$

[1] CRF++ http://crfpp.sourceforge.net/.

Here, t_j (respectively, s_i) denotes the j^{th} (respectively, i^{th}) character in target word w_T (respectively, source word w_S) and $A \equiv a_1^m$ is the hidden alignment between w_T and w_S where t_j is aligned to s_{a_j}, $j = 1,...,m$. We estimate the parameters of the model by learning over a training set of transliteration pairs. We use the EM algorithm to iteratively estimate the model parameters. The transliteration similarity score of a pair (w_S, w_T) is $log\ p(w_T|w_S)$ appropriately transformed.

3 Experimental Setup

3.1 Data (FIRE data)

The English document collection provided by FIRE organizers [FIRE] was used in all our runs. The English document collection consists of ~124,000 news articles from "The Telegraph India" from 2004-07. All the English documents were stemmed using the Porter stemmer [Porter, 1980]. We ignored the stop words in the documents as well as the queries. We did not stem the query terms.

There are 50 queries in each language, each having a topic, description and narrative, successively expanding the scope of the query.

3.2 Bilingual Dictionaries

We used statistical dictionaries for both Hindi-English and Tamil-English crosslingual retrieval, using the dictionaries generated by training statistical word alignment models on Hindi-English parallel corpora (~100K parallel sentences) and Tamil-English parallel corpora (~50 K parallel sentences) using the GIZA++ tool [Och and Ney, 2002]. We used 5 iterations of IBM Model 1 and 5 iterations of HMM [Och and Ney, 2002]. The Hindi-English dictionary has ~59K Hindi words and ~63K English words. The Tamil-English dictionary has ~107K Tamil words and ~45K English words. We used only top 4 translations for every source word.

3.3 Crosslingual System

We used the crosslingual system described in section 2.2.

3.4 Transliteration Systems

Training Direct Transliteration Systems: The direct transliteration systems were trained with about 15K parallel names in Hindi and English and Tamil and English. We observed that the quality of a transliteration system trained with 15K corpora is asymptotically close to that of a system trained with much larger corpora.

Training Transitive Transliteration Systems: The transitive transliteration systems chains two distinct transliteration systems, each trained with about 15K of appropriate parallel names corpora [Khapra et al., 2010]. In our case, we used Kannada as the intermediate language, and trained two systems: one between Hindi and Kannada, and another between Kannada and English. Kannada was chosen as the intermediate language as it has a near superset of phoneme inventory of Hindi and English. The transitive transliteration methodology was used only for Hindi-English crosslingual runs. We used top 5 results from transliteration generation for query translation.

3.5 Similarity model for Transliteration Mining

We trained Hindi-English and Tamil-English transliteration similarity models on 16k parallel single word names in Hindi-English and Tamil-English language pairs respectively, and ran 15 iterations of EM. For each query, we considered top-100 documents returned by the crosslingual system for the purpose of mining. We refer to [Udupa et al., 2009-a] for details of the mining methodology. A transliteration similarity threshold value of 1.0 was used to filter the output.

3.6 Performance Measures

The standard measures for evaluating our tasks were used, specifically, Mean Average Precision (MAP) and Precision at top-10 (P@10).

4 Results and Analysis

In this section, we present our experimental results and our analysis. In addition to the whole query (title, description and narrative), we ran our experiments with short queries with just title or with title and description. Table 1 shows the notation that we used in our description.

T	Title
TD	Title and Description
TDN	Title, Description and Narration
M	Transliteration Mining
G_D	Transliteration Generation - Direct
G_T	Transliteration Generation - Transitive

Table 1: Notations used

Table 2 shows the MAP and precision of our monolingual as well as crosslingual official runs. The format of the run ids in the results table is '*Source-Target-Data-Methodology*', where '*Data'* indicates the data used for query, and is one of {T, TD, TDN} and '*Methodology'* indicates the methodology and from the set {M, G_D, G_T, M+G_D, M+G_T}. The '+' refers to the combination of more than one approach. The symbols star (*) and plus ($^+$) indicate statistically significant differences with 95% and 90% confidence respectively according to the paired t-test.

Run	**MAP**	**P@10**
English-English-T	0.3653	0.344
English-English-TD	0.4571	0.406
English-English-TDN	**0.5133**	**0.462**
Hindi-English-T	0.2931	0.26
Hindi-English-T[G_D]	0.3168*	0.282
Hindi-English-T[G_T]	0.314*	0.276
Hindi-English-T[M]	**0.339***	**0.304**
Hindi-English-T[M+G_D]	0.3388*	0.302
Hindi-English-T[M+G_T]	0.3388*	0.302
Hindi-English-TD	0.4042	0.356
Hindi-English-TD[G_D]	0.4336*	0.386
Hindi-English-TD[G_T]	0.4369*	0.382
Hindi-English-TD[M]	0.4376*	**0.388**
Hindi-English-TD[M+G_D]	**0.4378***	0.386
Hindi-English-TD[M+G_T]	0.4375*	0.386
Hindi-English-TDN	0.4748	0.424
Hindi-English-TDN[G_D]	0.4942*	0.434
Hindi-English-TDN[G_T]	0.497*	0.438
Hindi-English-TDN[M]	**0.4977***	0.442
Hindi-English-TDN[M+G_D]	0.4971*	**0.444**
Hindi-English-TDN[M+G_T]	0.4965*	**0.444**
Tamil-English-T	0.271	0.258
Tamil-English-T[G_D]	**0.2891**$^+$	**0.268**
Tamil-English-T[M]	0.2815*	0.258
Tamil-English-T[M+G_D]	0.2816$^+$	0.268
Tamil-English-TD	0.3439	0.346
Tamil-English-TD[G_D]	0.3548$^+$	0.35
Tamil-English-TD[M]	**0.3621***	0.346
Tamil-English-TD[M+G_D]	0.3617*	**0.362**
Tamil-English-TDN	0.3912	0.368
Tamil-English-TDN[G_D]	0.4068*	0.378
Tamil-English-TDN[M]	**0.4145***	0.368
Tamil-English-TDN[M+G_D]	0.4139*	**0.394**

Table 2: Monolingual and Crosslingual Retrieval Performance

4.1 Monolingual Retrieval

We submitted 3 official English monolingual runs, as presented in the first three rows of table 2. With the full query (TDN), our system achieved a peak MAP score 0.5133. Generally this performance is thought to be the upper bound for crosslingual performance.

4.2 Crosslingual Retrieval : Hindi-English

We submitted 18 official runs on Hindi-English crosslingual track, as shown in Table 2. The three runs under 'T', 'TD' and 'TDN' were run without handling the OOV terms, and hence provide a baseline for measuring the incremental performance due to transliteration generation or mining.

For the following analysis, we consider only the queries using all of topic-description-and narrative portions of the queries. Our basic Hindi-English crosslingual run 'Hindi-English-TDN' (without transliteration generation or mining), achieved the MAP score 0.4748, and our best crosslingual run 'Hindi-English-TDN[M]' with mining achieved a MAP score of 0.4977. Significantly, our basic run achieves 92% of the monolingual performance, and the crosslingual run enhanced with transliteration mining, 97% of the monolingual retrieval performance.

4.3 Crosslingual Retrieval : Tamil-English

We submitted 12 official Tamil-English crosslingual runs, as shown in Table 2. As in Hindi-English runs, the three runs under 'T', 'TD' and 'TDN' were run without handling the OOV terms, and hence provide a baseline for measuring the incremental performance due to transliteration generation or mining.

For the following analysis, we consider only the full query setup that use all of the topic-description-and-narrative (TDN) portions of the query. Our basic Tamil-English crosslingual run (without generation and mining), achieved the MAP score 0.3912. Our best crosslingual run 'Tamil-English-TDN[M]' that uses transliteration mining achieves a MAP of 0.4145. Note that this score is ~81% of our monolingual English retrieval performance.

4.4 Handing OOV terms and its Effect on CLIR performance

We observe that in each of the above runs, handling OOV terms (by Transliteration generation

or mining) significantly boosts the CLIR performance. In subsequent sections, we analyze the effect of handling OOV terms in Hindi-English and Tamil-English CLIR runs.

4.4.1 OOV terms in Hindi-English CLIR

In FIRE2010 Hindi queries there were totally 73 OOV terms and 31 of them are terms that are proper names that may be transliterated.

Mining transliterations from a set of documents in the first-pass retrieval provided the correct transliteration equivalents for 24 OOV terms, and hence directly influenced the retrieval performance in the second-pass.

We also observed that the transliteration generation alone also improves the performance of CLIR system, though marginally lower than that of mining. Performance of the transitive transliteration generation is similar to that of the direct transliteration generation. The MAP differences between direct and transitive approaches are marginal, in TDN setup 0.0028, in TD setup 0.0033 and in T setup 0.0028, with direct generation performing marginally better. As in mining, transliteration generation approaches boosted the performance of the CLIR system to ~96% of our monolingual IR performance.

4.4.2 OOV terms in Tamil-English CLIR

In FIRE2010 Tamil queries there were totally 129 OOV terms and 61 of them are names that may be transliterated. Mining actually got transliteration equivalents for 24 OOV terms. However, as Tamil being a Dravidian language, is highly agglutinative, we find that the OOV terms in Tamil need not be in their root form. Our error analysis shows that about 26% of them are inflected or agglutinated. Our mining algorithm was able to mine some of them but with a relaxed setting for the transliteration similarity score for mining, which potentially introduces noisy terms. We believe that the use of a good stemmer for inflectional languages like Tamil may help our mining algorithm and, transitively, the crosslingual retrieval performance.

Similar to the Hindi-English system, transliteration generation also improves the performance of Tamil-English crosslingual retrieval performance, though marginally lower than mining.

4.4.3 Transliteration Mining Vs Generation

The results our CLIR experiments, augmented with transliteration generation and mining, indicate that both approaches help in CLIR performance.

For example, consider the query number 112 in Hindi shown in table 3. The OOV terms of Hindi query are shown in bold, and those OOV terms that have an English transliteration equivalent are highlighted.

Type	Query
Title	गुटखा मालिकों का **अन्डरवर्ल्ड** के साथ **उलझाव**
Description	प्रसिद्ध गुटखा कम्पनी (**माणिकचन्द** और गोवा)के साथ दाऊद इब्राहिम के सम्बन्ध
Narration	प्रासंगिक **प्रलेख** में **माणिकचन्द** गुटखा और गोवा गुटखा मालिकों का **अन्डरवर्ल्ड डॅन** दाऊद इब्राहिम के साथ सम्बन्ध, से सम्बन्धित सूचनाएँ यहाँ होनी चाहिये। अन्य कम्पनियों के साथ दाऊद इब्राहिम के सम्बन्ध यहाँ अप्रासंगिक हैं।

Table 3: Hindi query no. 112

The Hindi query has five OOV terms (‘अन्डरवर्ल्ड’, ‘उलझाव’, ‘माणिकचन्द’, ‘प्रलेख’ and ‘अन्डरवर्ल्ड’), out of which two of them (‘अन्डरवर्ल्ड’ and ‘माणिकचन्द’) are names that may be transliterated. Mining was able to identify the valid English equivalents for these two (‘underworld’ and ‘manikchand’), whereas generation produced only one English equivalent (for ‘manikchand’) correctly. In addition, as we consider top 5 results for all the OOV terms for transliteration generation, many more noise terms were also generated, affecting the retrieval performance. Similar trends were observed in the Tamil-English CLIR system as well.

The Tables 4 & 5 show some examples of OOV terms and the corresponding generated and mined transliterations.

Hindi /Tamil OOV	Generation – Direct	Generation – Transitive
आँध्र	aandhra, andhra, aandra, aanara, aandhara	aandhra, andhra, aandhrar, andhrar, aandhar
इस्राइली	israili, israeli, israili, israilli, istraili,	israily, isralie, israly, isrily, israly
मसजिद	masjid, masajid, masjaid, msajid, masjed	masjid, masajid, maszid, maszid, masajid

என்செபாலிடிஸ்	encebalidis, encebalydis, encepalidis, encebalitis, ensebalidis	-NA-
மகராஷ்டிர	makrashtir, makrashir, makrashtira, magrashtir, makrashira	-NA-

Table 4: OOV terms and generated equivalents

Hindi / Tamil OOV terms	Mined English words
आँध्र	andhra
इस्राइली	israel, israeli, israelis
मसजिद	masjid, masjids
என்செபாலிடிஸ்	encephalitis
மகராஷ்டிர	maharashtra, maharashtras

Table 5: OOV terms and mined equivalents

4.5 Hybrid approach: Mining with Transliteration Generation

In addition to generation and mining of transliteration equivalents, we did experiments that employed a combination of mining and generation. In this methodology, we first mine the transliteration equivalents using mining, and employed transliteration generation for those terms for which mining produced no results. In Table 2, the run ids with '$M+G_D$' and '$M+G_T$', refer that they are combination of mining and generation. We observe that the hybrid approaches had minimal impact on the overall crosslingual performance, in both Hindi-English and Tamil-English.

5 Conclusion

We detailed the system participation of Microsoft Research India in the FIRE-2010 campaign [FIRE]. We participated in English monolingual track and two crosslingual tracks - Hindi-English and Tamil-English. We presented system and analyzed our results. We show that our basic CLIR system is improved significantly by the two methodologies for handling OOV words – transliteration generation and mining. Significantly, we show that our crosslingual retrieval performance (that is enhanced with transliteration generation or mining) is nearly equal to that of our monolingual performance, validating our methodologies for handling OOV terms in the crosslingual retrieval.

Acknowledgments

We thank Abhijit Bhole for his work on transliteration similarity model, Mitesh Khapra for his contributions to transliteration generation (both direct and transitive) and Jagadeesh Jagarlamudi for his work on the language modeling based information retrieval system.

References

Brown, P.E., Della Pietra, V.J., Della Pietra, S.A. and Mercer, R.L. 1993. *The mathematics of statistical machine translation: parameter estimation*. Computation Linguistics.

The Cross-Language Evaluation Forum (CLEF). http://clef-campaign.org.

Forum for Information Retrieval Evaluation. http://www.isical.ac.in/~fire/.

He. X. 2007. *Using word dependent transition models in HMM based word alignment for statistical machine translation*. In Proceedings of 2nd ACL Workshop on Statistical Machine Translation (2007).

Jagarlamudi, J. and Kumaran, A. 2007. *Cross-Lingual Information Retrieval System for Indian Languages*. Working Notes for the CLEF 2007 Workshop.

Khapra, M., Kumaran, A. and Bhattacharyya, P. 2010. *Everybody loves a rich cousin: An empirical study of transliteration through bridge languages.* In proceedings of NAACL 2010.

Lafferty, J., McCallum, A. and Pereira, F. 2001. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. Proceedings of the International Conference on Machine Learning, 2001.

Li, H., Kumaran, A., Pervouchine, V. and Zhang, M. 2009. *Report of NEWS 2009 Machine Transliteration Shared Task*. Proceedings of the ACL 2009 Workshop on Named Entities (NEWS 2009), Association for Computational Linguistics, August 2009.

Majumder, P., Mitra, M., Pal, D., Bandyopadhyay, A., Maiti, S., Mitra, S., Sen, A. and Pal, S. 2008. *Text collections for FIRE.Proceedings of SIGIR 2008*.

Nardi, A. and Peters, C. 2006. *Working Notes for the CLEF 2007 Workshop.*

NTCIR: http://research.nii.ac.jp/ntcir/.

Och, F. and Ney, H. 2002. *A systematic comparison of various statistical alignment models.* Computation Linguistics.

Peters, C. 2006. *Working Notes for the CLEF 2006 Workshop.*

Ponte, J. M. and Croft, W.B. 1998. *A language modeling approach to information retrieval.* Proceedings of SIGIR 1998.

Porter, M.F. 1980. *An algorithm for suffix stripping*. Program, 14(3):130–137.

Udupa, R., Jagarlamudi, J. and Saravanan, K. 2008. *Microsoft Research India at FIRE2008: Hindi-English Cross-Language Information Retrieval.* Working notes for Forum for Information Retrieval Evaluation (FIRE) 2008 Workshop.

Udupa, R., Saravanan, K., Bakalov, A. and Bhole, A. 2009. *"They Are Out There, If You Know Where to Look": Mining Transliterations of OOV Query Terms for Cross-Language Information Retrieval.* In 31th European Conference on IR Research, ECIR 2009.

Udupa, R., Saravanan, K., Kumaran, A. and Jagarlamudi, J. 2009. *MINT: A Method for Effective and Scalable Mining of Named Entity Transliterations from Large Comparable Corpora.*Proceedings of the 12th Conference of the European Chapter of the ACL 2009.

Zhai, C. and Lafferty, J. 2002. *Two Stage Language Models for Information Retrieval*. Proceedings of SIGIR 2002.

Zhai, C. and Lafferty, J. 2004. *A study of smoothing algorithms for language models applied to information retrieval*. ACM Transactions on Information Systems, 22(2):179–214.