Query Expansion in Resource-Scarce Languages: A Multilingual Framework Utilizing Document Structure

ARJUN ATREYA V, ASHISH KANKARIA, PUSHPAK BHATTACHARYYA,
and GANESH RAMAKRISHNAN, IIT Bombay, Mumbai, India

Retrievals in response to queries to search engines in resource-scarce languages often produce no results, which annoys the user. In such cases, at least partially relevant documents must be retrieved. We propose a novel multilingual framework, *MultiStructPRF*, which expands the query with related terms by (i) using a resource-rich assisting language and (ii) giving varied importance to the expansion terms depending on their position of occurrence in the document. Our system uses the help of an assisting language to expand the query in order to improve system recall. We propose a systematic **expansion model** for weighting the expansion terms coming from different parts of the document. To combine the expansion terms from query language and assisting language, we propose a heuristics-based **fusion model**. Our experimental results show an improvement over other PRF techniques in both precision and recall for multiple resource-scarce languages like *Marathi, Bengali, Odia, Finnish,* and the like. We study the effect of different assisting languages on precision and recall for multiple query languages. Our experiments reveal an interesting fact: Precision is positively correlated with the typological closeness of query language and assisting language, whereas recall is positively correlated with the resource richness of the assisting language.

CCS Concepts: • **Information systems → Information retrieval;**

Additional Key Words and Phrases: Query expansion, resource scarce languages, multilingual retrieval

ACM Reference Format:
Arjun Atreya V, Ashish Kankaria, Pushpak Bhattacharyya, and Ganesh Ramakrishnan. 2016. Query expansion in resource scarce languages: A multilingual framework utilizing document structure. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 16, 2, Article 14 (November 2016), 17 pages.
DOI: http://dx.doi.org/10.1145/2997643

1. INTRODUCTION

A user's query to a search engine is often unstructured and incomplete. These factors prevent a search engine from satisfying users' information need correctly and completely. This situation necessitates query expansion.

Various approaches to expand a query by adding similar terms using co-occurrence similarity, query logs, user feedback, pseudo-relevance feedback, and the like have been proposed. *PRF* [Buckeley et al. 1994; Mitra et al. 1998] has been shown to be one of the most effective query expansion techniques [Manning et al. 2008].

PRF, however, is agnostic of the position of the expansion terms in a document. The *title* of a document is usually around 5–10 words long, whereas the *body* of the document contains many more words. Arguably, the terms that occur in the *title* more precisely represent the document as opposed to the terms that occur in the *body* of the document. In this article, we propose a Structure-aware PRF (*UnifiedStructPRF*)

Authors' addresses: A. Atreya V, A. Kankaria, P. Bhattacharyya, and G. Ramakrishnan, Department of Computer Science and Engineering, IIT Bombay, Mumbai, India 400076; emails: {arjun, akankaria13, pb, ganesh}@cse.iitb.ac.in.

DOI: http://dx.doi.org/10.1145/2997643

framework that seamlessly prioritizes the expansion terms located in different parts of documents.

Null query (query retrieving zero results) is an irritation that is often experienced in a search engine built for a resource-scarce language. A language having a relatively small number of documents on the web is called *resource scarce*. If a search engine retrieves no documents for a query, it is bound to annoy the user. Instead, it is better to retrieve partially relevant documents. An assisting language comes in handy for processing null queries. Use of an assisting language to obtain expansion terms by taking help from a resource-rich language like English leads to retrieval of partially relevant documents [Chinnakotla et al. 2010a, 2010b].

In this article, we propose a novel multilingual framework called *MultiStructPRF* that takes the help of an assisting language and gives differing importance to expansion terms coming from different parts of the document. The framework uses a heuristics-based approach to combine the expansion terms from the query language and the assisting language. Weights assigned to expansion terms coming from the assisting language are a function of (i) the relative monolingual performance of the assisting language with respect to the query language, (ii) the resource richness of the assisting language, and (iii) translation confidence (measure of translation quality) between the query language and the assisting language.

We show that our *MultiStructPRF* framework significantly outperforms other variants of PRF-based retrieval techniques both in terms of precision and recall. The percentage improvement in precision over vanilla *PRF* varies between 2% and 180%, whereas the improvement in recall varies between 3% and 18% across seven languages: *Marathi*, *Hindi*, *Bengali*, *Odia*, *Gujarati*, *Spanish,* and *Finnish*.

The performance of *MultiStructPRF* framework depends on the choice of the assisting language. We study the impact of different assisting languages on precision and recall. Our experiments reveal that precision is positively correlated with the typological closeness of the query language and the assisting language. On the other hand, the recall is positively correlated with the resource richness of the assisting language. Hence, the nature of the application determines the choice of an assisting language to be used. For example, a patent search engine in Hindi would demand high recall even at the expense of low precision. In such a case, a resource-rich assisting language like English should be used.

The organization of this article is as follows: In Section 2, we discuss related work. Section 3 explains the architecture of the *MultiStructPRF* framework. Section 4 presents the experimental setup and compares the performance of *MultiStructPRF* against other PRF-based systems. In Section 4.5, we study the influence of different assisting languages. We conclude in Section 6.

2. RELATED WORK

Numerous query expansion techniques have been explored in Information Retrieval (IR). Using an external resource to expand a query is one of the simplest and most intuitive approaches. Unified Medical Language System (UMLS) [Bodenreider 2004] is one such system for querying biomedical research literature using a domain-specific thesaurus. Qiu and Frei [1993] propose the use of a similarity thesaurus for query expansion. The thesaurus is automatically built using domain knowledge. Such systems work well when we have a rich, domain-specific resource.

Voorhees [2005] use WordNet for query expansion by adding synonymous terms and reports negative results. WordNet is also used by Smeaton et al. [1995] to add either generic or specific expansion terms based on the specificity of the query. Cui et al. [2002] develop a system that extracts the expansions terms based on a user's behavior, which is stored in form of query logs. Yin et al. [2009] consider the query log as a bipartite graph that connects query nodes to URL nodes by click edges. These click edges help in finding relevant expansion terms. Random walk models [Collins-Thompson and Callan 2005; Lafferty and Zhai 2001a] are used to learn associations by combining evidence from various sources like WordNet.

Various frameworks like vector space models, language models, and probabilistic IR make use of relevance feedback for query expansion [Buckeley et al. 1994; Jones et al. 2000; Lavrenko and Croft 2001; Zhai and Lafferty 2001]. Croft and Harper [1979] pioneered the technique of pseudo-relevant feedback by using probabilistic models for query expansion. However, they also highlight one fundamental problem in PRF: topic drift. Topic drift is "Tendency of a search to drift away from the original subject of discussion (and thus, from the query), or the results of that tendency" [Macdonald and Ounis 2007]. Several approaches have been proposed to improve *PRF* by (i) refining the relevant document set [Mitra et al. 1998; Sakai et al. 2005], (ii) refining the expansion terms from PRF [Cao et al. 2008], (iii) using selective query expansion [Carpineto and Romano 2012; Cronen-Townsend et al. 2004], and (iv) varying the importance of documents [Tao and Zhai 2006]. Zhai and Lafferty [2001] give the original framework for computing PRF with an expectation maximization technique to extract expansion terms from the initially retrieved top k documents.

Wikipedia has been used as source of expansion terms [Al-Shboul and Myaeng 2011; Ganesh and Verma 2009; Voorhees 2005; Xu et al. 2009a, 2009b]. Atreya et al. [2013] suggest an approach utilizing the structure of the document for assigning weights to expansion terms. However, the assignment is ad hoc and fixed by trial and error, which may not be scalable. Gao et al. [2008] use English to improve the performance of Chinese queries.

Use of an assisting language has proved useful in extracting rich semantic information for a community-based question retrieval system [Zhou et al. 2012, 2013, 2016]. Trieschnigg et al. [2010] uses a cross-lingual IR framework for biomedical information retrieval. Text representation and conceptual representation of queries and documents are treated as two languages, and a translation model was built to service the queries in a biomedical domain. Chinnakotla et al. [2010a] and Chinnakotla et al. [2010b] show that an assisting language can help in improving retrieval performance. But the way in which expansion terms from the query language and the assisting language are combined is not systematic (i.e., they use ad hoc weights). Using an assisting language makes sense only if we are able to systematically combine the expansion terms from both languages.

3. OUR SYSTEM

We propose a multilingual framework for query expansion in resource-scarce languages that uses the help of a resource-rich assisting language. The framework also utilizes document structure by giving different importance to the expansion terms from different parts of the document. Our framework customizes the expectation maximization technique proposed by Zhai and Lafferty [2001]. We use Wikipedia corpus to extract expansion terms. Wikipedia documents are structured in four parts: *title*, *body*, *categories,* and *infobox*. Our framework gives a principled approach to assigning weight to expansion terms coming from the *title*, *body*, *categories,* and *infobox* parts of the document in both the query language and the assisting language. The framework uses a heuristic model for combining the expansion terms coming from the query language and the assisting language.

Figure 1 illustrates the architecture of *MultiStructPRF*. The work flow is as follows:

(1) Translate the query Q from a query language L_Q to an assisting language L_A, where L_A is more resource rich than L_Q. The translated query is Q_T: **Translation Model**

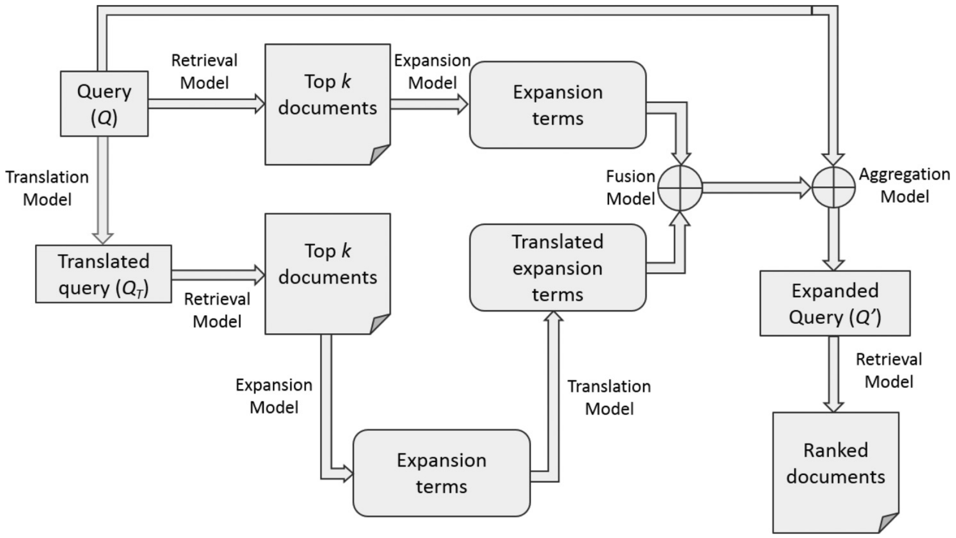


Fig. 1. Architecture of MultiStructPRF.

(2) Retrieve the ranked list of relevant documents for queries Q and Q_T from the Wikipedia collection independently for the languages L_Q and L_A, respectively: **Retrieval Model**

(3) Extract expansion terms from top k documents in L_Q and L_A using an expectation maximization algorithm across multiple parts of the document: **Expansion Model**

(4) Translate the expansion terms from L_A to L_Q: **Translation Model**

(5) Obtain the final set of expansion terms E by heuristically combining the expansion terms from L_Q and the expansion terms translated from L_A to L_Q: **Fusion Model**

(6) The updated query Q' is generated by merging the user query Q with the expansion terms E: **Aggregation Model**

(7) Retrieve the ranked list of relevant documents in language L_Q for the updated query Q': **Retrieval Model**

3.1. Retrieval Model

The retrieval model accomplishes the process of retrieving the ranked list of relevant documents for a query. Each document is ranked using a KL divergence score [Lafferty and Zhai 2001b]. For every document D in the collection, a language model θ_D is generated and stored. A language model θ_Q is generated for each query Q. We use a KL-divergence (D_{KL}) metric to measure the similarity between θ_Q and θ_D, as shown in Equation (1). The more relevant is D, the less is D_{KL}. We rank the documents in an increasing order of their divergence score.

$$D_{KL}\left(\theta_Q|\theta_D\right) = -\sum_{w} p(w|\theta_Q) * log\ p(w|\theta_D). \tag{1}$$

The retrieval model is used in two different stages of *MultiStructPRF*: (i) retrieve top k documents from Wikipedia to generate expansion terms, and (ii) retrieve a ranked list of documents for the expanded query.

3.2. Expansion Model

The expansion model accomplishes the process of extracting the expansion terms from the set of relevant documents. Top k documents retrieved are assumed to be relevant.

We extract the expansion terms in query language and assisting language separately using this model. This model **utilizes the structure of the document** for query expansion. The *UnifiedStructPRF* system uses this expansion model with *PRF* without the use of *Translation* and *Fusion* models.

Terms from all the documents in the corpus constitute a *collection set*. *A relevant set* consists of the terms from top *k* relevant documents retrieved for a query. Terms in the relevant set act as potential expansion terms. We customize the EM algorithm suggested by Zhai and Lafferty [2001] for generating expansion terms. The EM algorithm is a process of iteratively assigning a probability to expansion terms. In each iteration, the probability of the terms that uniquely represent the relevant set is increased while the probability of other terms is decreased, thus yielding the representative terms of relevant set. This model assumes that the terms, which uniquely represent the relevant set, are related to the query. This is a reasonable assumption since the relevant set represents the query, and the expansion terms uniquely represent the relevant set.

We customize the EM algorithm to utilize the structure of Wikipedia documents to generate expansion terms (i.e., *title*, *body*, *categories,* and *infobox*). The terms in the relevant set are divided into multiple sets based on their position of occurrence in the document. Each part of the document contributes to the expansion terms with different importance. For instance, a *title* precisely represents the document, whereas the *body* of the document is more generic in nature, elaborating the title, as it were. Hence, expansion terms from the *title* are more important than terms from the *body* of the document.

Atreya et al. [2013] utilize the document structure by assigning different weights to the expansion terms from different parts of the document. For every part of the document, a separate EM algorithm is used to generate expansion terms. The expansion terms from various parts of the documents are merged using the weights chosen in an ad hoc manner. In contrast, we propose a systematic approach called *UnifiedStructPRF*, which dynamically assigns the weights to each part of the document. For a document with *four* parts, there are *four* relevant sets and *four* collection sets corresponding to terms from *four* parts of the documents. Thus, the total number of sources in our model becomes:

$$no.\ of\ sources = 2 * (no.\ of\ parts)$$

For ease of explanation, we consider a document with *two* parts (*title* and *body*). The proposed model is scalable to any number of parts. We describe the mathematical formulation of the customized EM algorithm in the following section.

Mathematical Model. In this section, we describe the expectation maximization algorithm to seamlessly extract the expansion terms from different parts of the document.

Notations

—N = observation sequence; all terms from relevant set
—M = sources; relevant set(title) R_T, relevant set(body) R_B, collection set (title) C_T, and collection set (body) C_B
—L = outcome; all terms from collection
—P_{jk} is the probability that the k^{th} expansion term uniquely represents the j^{th} source.
—π_j is the probability of selecting thej^{th} source.
—X_{ik} is the indicator variable representing whether the i^{th} term from the observation is same as the k^{th} expansion term or not.
—Z_{ij} is the hidden variable representing whether the source of the i^{th} term is j or not.

Maximum likelihood expression
We need to boost the probability of the expansion terms that uniquely represent the relevant set. The EM algorithm iteratively performs this task and stores the probability

of a term k, uniquely representing the relevant set's *title* and *body*, in P_{R_Tk} and $P_{R_Bk,}$ respectively.

$L(\theta)$ is the likelihood of all the terms from the relevant set being expansion terms. The likelihood estimate $L(\theta)$ of the unknown parameter θ is described in terms of marginal likelihood of the observed data (X_{ik} and Z_{ij}). We formulate $L(\theta)$ as shown in Equation (2):

$$L(\theta) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left(\pi_j * \prod_{k=1}^{L} P_{jk}^{X_{ik}}\right)^{Z_{ij.}} \tag{2}$$

For ease of calculation, we take the log of the expression to get the log likelihood, as in Equation (3):

$$LL(\theta) = \sum_{i=1}^{N}\sum_{j=1}^{M} E(z_{ij})\left(log(\pi_j) + \sum_{k=1}^{L} X_{ik}log(P_{jk})\right). \tag{3}$$

We maximize the log likelihood to obtain the equations for the P_{R_Tk} and P_{R_Bk} subject to the constraints $\sum_{j=1}^{M}\pi_j = 1$ and $\sum_{k=1}^{L} P_{jk} = 1$ using Lagrange multipliers α and β. Equation (4) represents the maximum log likelihood expression:

$$MLE = LL(\theta) - \alpha * \left(\sum_{j=1}^{M}\pi_j - 1\right) - \beta * \left(\sum_{k=1}^{L} P_{jk} - 1\right). \tag{4}$$

Maximization step: M step

We obtain the expressions for π_j and P_{jk} by partially differentiating MLE with respect to π_j and $P_{jk,}$ respectively. Equations (5) and (6) constitute the maximization step:

$$\pi_j = \frac{\sum_{i=1}^{N} E(Z_{ij})}{\sum_{i=1}^{N}\sum_{j=1}^{M} E(Z_{ij})} \tag{5}$$

$$P_{jk} = \frac{\sum_{i=1}^{N} x_{ik} * E(z_{jk})}{\sum_{i=1}^{N}\sum_{k=1}^{L} x_{ik} * E(z_{jk})}. \tag{6}$$

π_j is the probability of choosing a source. Zhai and Lafferty [2001] keep π_j constant and compute only P_{jk} in the M step. We allow π_j to vary in both *UnifiedStructPRF* and *MultiStructPRF*, as per Equation (5), which is mathematically correct.

Expectation Step: E Step

Equation 7 allows us to estimate the probability of the source given the terms in the relevant set:

$$E(z_{ij}) = \frac{\pi_j * \prod_{k=1}^{L} P_{jk}^{x_{ik}}}{\sum_{j=1}^{M}(\pi_j * \prod_{k=1}^{L} P_{jk}^{x_{ik}})}. \tag{7}$$

In order to find terms uniquely representing source R_T, we consider source R_T as the relevant set and remaining sources R_B, C_T, and C_B as the collection set. The important point to note is that R_B is considered noise for R_T, which ensures that we find the expansion terms that uniquely represent only the relevant set(title). Similarly, we consider R_T as noise for finding the expansion terms in the relevant set(body). The terms representing R_T and R_B individually constitute the final set of expansion terms. In *MultiStructPRF*, we scale this mathematical model to accommodate the four parts of the document: *title*, *body*, *categories,* and *infobox*.

3.3. Translation Model

Translation model accomplishes the process of translating the query from the query language to the assisting language and translating the expansion terms from the assisting language to the query language. We use the IndoWordNet[1] for translating the terms across Indian languages and between the Indian language and English. This dictionary may have multiple translations for a word w. This is handled by uniformly distributing the probability of the word w among all its translations. Every translated word tw_i is assigned a probability value, as in Equation (8). $\#trans(w)$ is a function that returns the number of translations of the word w:

$$P(tw_i|\theta_{Q_T}) = \frac{P(w|\theta_Q)}{\#trans(w)},\ 1 \leq i \leq \#trans(w). \tag{8}$$

Named entities, foreign words, and out of vocabulary words (OOVs) cannot be translated using IndoWordNet. We have built an in-house transliteration system across Indian languages and also between the Indian language and English. The combination of translation and transliteration helps in transforming the words from the query language to the assisting language and vice versa.

Apart from Indowordnet, we can use other statistical machine translation techniques or existing of-the-shelf translation services like *Google translate.*[2] Our experience in using Google translate between Indian languages suggested that the output is of poor quality since the translations bridges through English. This led to the usage of Indowordnet for word-based translations across languages in our work.

3.4. Fusion Model

The fusion model accomplishes the process of merging the expansion terms from the query language and the translated expansion terms from the assisting language. This model **captures the multilinguality** of our framework. Using an assisting language demands a systematic approach to combine the expansion terms. We propose a heuristic to combine the expansion terms from query and assisting language.

The heuristic calculates α, which is the weight assigned to expansion terms from the assisting language. The final set of expansion terms E is a weighted addition of the expansion terms from the query language (E_Q) and the translated expansion terms from the assisting language (E_A), as in Equation (9):

$$E = \alpha * E_A + (1 - \alpha) * E_Q. \tag{9}$$

α depends on three parameters: (i) monolingual performance, (ii) resource richness, and (iii) translation confidence.

Relative monolingual performance: We use F-score as a metric of monolingual performance. We calculate the F-score of the system built for the query language and the assisting language independently. Using these scores, we compute the Relative Monolingual Performance (RMP) of the assisting language as shown in Equation (10):

$$RMP_A = \frac{fscore(L_A)}{fscore(L_Q) + fscore(L_A).} \tag{10}$$

It is intuitive that if the monolingual performance of the system is greater, then the system produces better expansion terms. We use an assisting language to enrich the quality of the expansion terms; hence, the assisting language chosen should have a high relative monolingual score compared to the query language. The higher the score, the better will be the quality of expansion terms from the assisting language.

[1] IndoWordNet is a linked lexical knowledge base of WordNets of Indian languages.

[2] https://translate.google.co.in/.

Relative resource richness: Resource richness is an indication of the quantity and quality of the document collection. More documents in a collection imply more topic coverage. So, a resource-rich assisting language produces better expansion terms. The resource richness of a language depends on two factors: (i) Total number of documents, and (ii) diversity of the documents.

It is challenging to compute the diversity of a collection due to its subjective nature, but the total number of documents alone is also a good indicator of resource richness. This assumes that the diversity of a collection depends on the number of documents in the collection. Thus, the number of documents is a *loose* indicator of diversity. We calculate the relative resource richness of an assisting language with respect to the query language using number of documents. English has around *10,000K* Wikipedia documents, whereas Hindi has only *600K* documents. The relative resource richness of English is 94.34%. This hugely biases the importance toward English. Thus, we need a smoothing factor to reduce the bias. We introduce *log* as a smoothing factor and calculate the relative resource richness (*RR*) of the assisting language as shown in Equation (11):

$$RR_A = \frac{log(\#docs(L_A))}{log(\#docs(L_Q)) + log(\#docs(L_A))}, \tag{11}$$

where $\#docs(L)$ is the number of documents in Language L.

Translation confidence: It is possible that the monolingual performance of the assisting language is excellent, but the translation quality between the query language and the assisting language is poor. The query may get translated incorrectly due to an erroneous dictionary. Although the expansion model for the translated query would produce excellent expansion terms, the terms may not be related to the original query. Also, while translating the expansion terms to the query language, there is a high possibility that the expansion terms are incorrectly translated. This is bound to degrade the quality of expansion terms and, in turn, the quality of retrieval. So it is important to incorporate the effect of translation while calculating α.

We use Translation Confidence (TC) to reduce the importance of the assisting language in order to account for poor translation. Thus, TC accounts for the loss of information during translation. In an ideal scenario of perfect translation, α depends only on the relative monolingual performance and the relative resource richness.

We use WordNet to calculate the TC for a pair of languages X and Y. The TC of each term t in X is

$$TC_{X\to Y}(t) = \frac{1}{\#trans(t),}$$

$TC_{X\to Y}(t)$ is the scaling factor used in the Equation (8). The TC from X to Y is the average of the TC across all the terms. In MultiStructPRF, first we translate the query into an assisting language, followed by translation of expansion terms from the assisting language to the query language. So, the translation confidence between the language pair X and Y is a product of TC from X to Y and TC from Y and X, as shown in Equation (12):

$$TC(X, Y) = \frac{\sum_t TC_{X\to Y}(t)}{N_1} * \frac{\sum_t TC_{Y\to X}(t)}{N_2}, \tag{12}$$

where N_1 and N_2 are the total terms in dictionaries of X and Y, respectively. We use the relative monolingual performance and the relative resource richness with equal weights, while the translation confidence is used as a scaling factor to calculate α, as

Table I. Feedback and Target Corpus Statistics

Language	Feedback	Target (Year)	#queries
English	3835K	–	–
Hindi	600K	113K (2010)	50 (76–125)
Marathi	66K	69K (2010)	50 (76–125)
Bengali	384K	416K (2012)	50 (176–225)
Gujarati	54K	313K (2011)	50 (126–175)
Odia	41K	17K (2012)	50 (176–225)
Spanish	1772K	460K (2009)	160 (41–200)
Finnish	432K	55K (2009)	120 (131–250)

shown in Equation (13):

$$\alpha = \left(\frac{1}{2} * RMP_A + \frac{1}{2} * RR_A\right) * TC(L_Q, L_A). \qquad (13)$$

3.5. Aggregation Model

The aggregation model accomplishes the process of combining expansion terms E generated by the fusion model and the initial query Q. Let λ be the weight assigned to the query terms. Query is an actual input from the user, whereas the expansion terms are automatically generated by the expansion model. So, Q is more important than E (i.e., $\lambda > 0.5$). The expanded query *Q'* is generated using Equation (14). We ran experiments varying the value of λ between *0.5* and *0.9* and empirically found that the value of λ at *0.6* gives the best results. Chinnakotla et al. [2010b] also suggests that the value of λ be *0.6*:

$$Q' = \lambda * Q + (1 - \lambda) * E. \qquad (14)$$

4. EXPERIMENTAL RESULTS

In this section, we present the quantitative and qualitative experiments that demonstrate the effectiveness of *MultiStructPRF*. The parameters of the quantitative evaluation are precision and recall. We conduct experiments for seven languages: *Hindi*, *Marathi*, *Bengali*, *Odia*, *Gujarati*, *Spanish,* and *Finnish*. In *MultiStructPRF*, we use English as an assisting language to extract the expansion terms.

4.1. Experimental Setup

We use the Wikipedia corpus for extracting the expansion terms and use the corpus from evaluation forums to retrieve the final ranked list of documents. The corpus for extracting the expansion terms is called the *feedback corpus* and the one used to retrieve final results is called the *target corpus*. By the definition of PRF, a target corpus itself must be used as a feedback corpus. However, we use Wikipedia corpus for both due to the unavailibility of structure in the target corpus provided by the evaluation forums.

The corpus, queries, and relevance judgment pool used to evaluate the performance for Indian languages are from the Forum for Information Retrieval (FIRE),[3] whereas Spanish and Finnish are from the ELRA-E0036[4] dataset used in the Cross-Lingual Evaluation Forum (CLEF)[5] [Braschler and Peters 2004]. Table I details the corpus and number of queries used for evaluation. We use the top 10 documents from the initial retrieval as the relevant set and pick the top 30 expansion terms. Both FIRE and CLEF datasets that are being used are collections of news documents. As detailed in

[3] http://www.isical.ac.in/~fire/.

[4] http://catalog.elra.info/product_info.php?products_id=1127.

[5] http://www.clef-initiative.eu/.

Table II. MAP Values for Different Query Expansion Techniques

Language	**PRF**	**StructPRF**	**UnifiedStructPRF**	**MultiPRF**	**MultiStuctPRF**
Hindi	0.2364	0.2529	0.2717	0.2938	**0.2946**
Marathi	0.1827	0.2611	0.3023	0.3173	**0.3186**
Odia	0.1100	0.1400	0.1300	0.1527	**0.1615**
Gujarati	0.0670	0.1024	0.1183	0.1531	**0.1606**
Bengali	0.0640	0.1267	0.1218	0.1528	**0.1539**
Spanish	0.1352	0.1778	0.2702	0.3562	**0.3791**
Finnish	0.2477	0.2517	0.2524	0.2530	**0.2693**

the second column of Table I, the date information indicates the version of the dataset used for experimentation in each language.

4.2. Evaluation

We evaluate the search engine on the queries provided by evaluation forums. The per query retrieved documents are then stored and compared with the relevance judgment pools. We then use the trec_eval[6] script to calculate the precision and recall of our system. In this experiment, we evaluate our search engine for five Indian languages and two European languages. For each language, we evaluate four different query expansion techniques, as listed here:

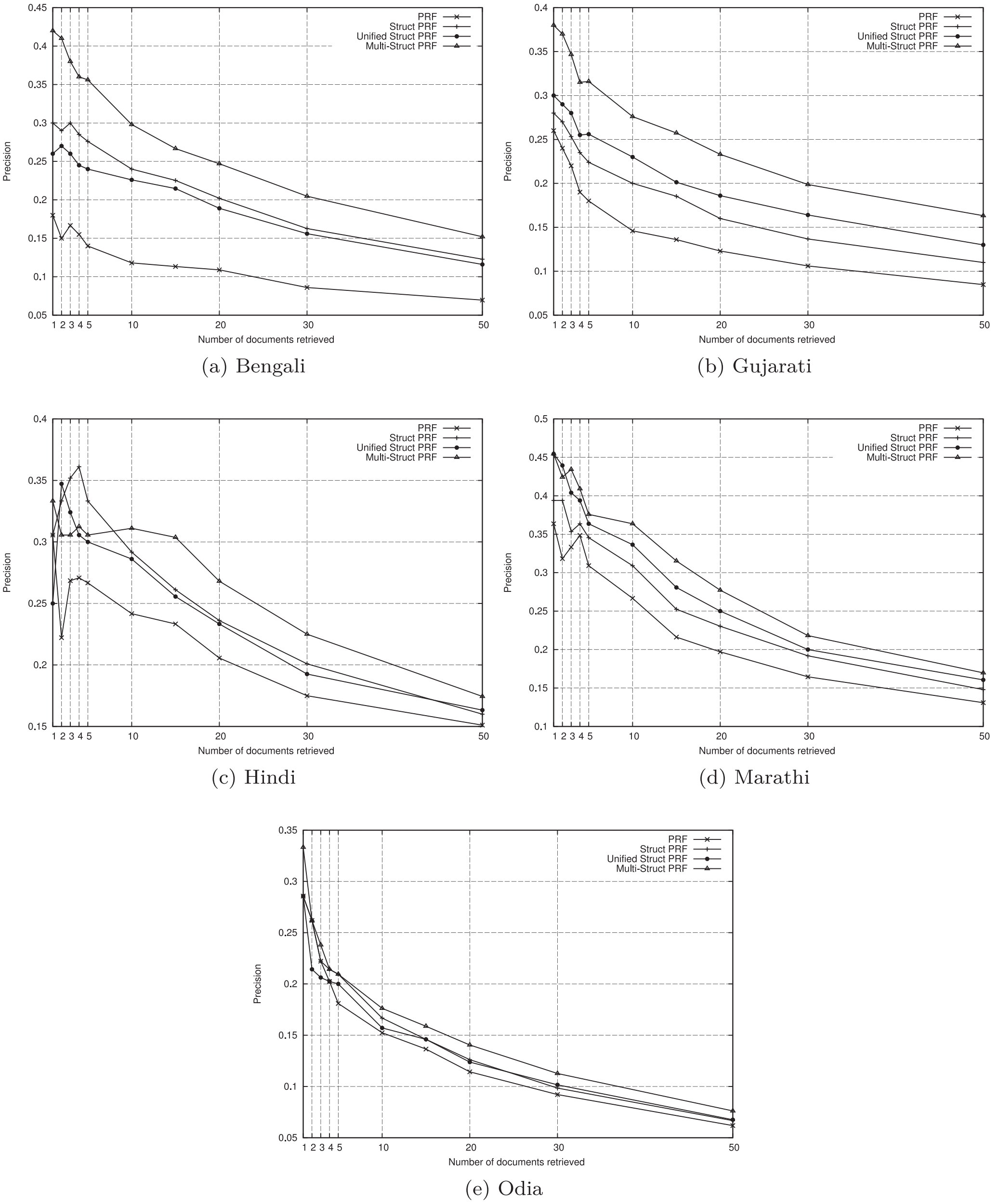
(1) ***PRF*****:** This is the basic version of PRF as proposed by Zhai and Lafferty [2001].
(2) ***StructPRF*****:** This is the structure-cognizant PRF system proposed by Atreya et al. [2013], which uses ad hoc weights.
(3) ***UnifiedStructPRF*****:** This is our proposed approach that uses a unified framework to extract expansion terms from various parts of documents seamlessly, as described in Section 3.2.
(4) ***MultiPRF*****:** This is the approach proposed by Chinnakotla et al. [2010b] which uses expansion terms from source and assisting languages without taking cognizance of document structure.
(5) ***MultiStructPRF*****:** This is our proposed multilingual framework that uses the help of an assisting language (*English*) and utilizes the document structure while extracting the expansion terms, as explained in Section 3.

In Table II, we list the Mean Average Precision (MAP) values for all variants of *PRF* across multiple languages. The *StructPRF* system improves precision over vanilla *PRF*. For most languages, our *UnifiedStructPRF* outperforms both *PRF* and *StructPRF*. Our *UnifiedStructPRF* system is more principled compared to *StructPRF* and thus is scalable. As expected, *MultiStructPRF* significantly outperforms all other variants of *PRF* for every language including *MultiPRF* proposed by Chinnakotla et al. [2010b]. The improvement in MAP of *MultiStructPRF* over *MultiPRF* is not considerably high, but the manual qualitative analysis shows that the expansion terms generated by *MultiStructPRF* have less topic drift compared to *MultiPRF*.

For *Bengali*, we see an improvement of around 150% over *PRF* and around 22% over *StructPRF* in MAP values. For *Spanish*, the improvement in MAP values is 173% and 108% with respect to *PRF* and *StructPRF,* respectively. Similarly, *MultiStructPRF* has an improvement of 87% over *PRF* and 22% over *StructPRF* for *Marathi*. Similar trends are observed in other languages.

For a detailed analysis, we plot *P@k* for multiple values of *k* ranging from 1 to 50 for five languages: *Marathi*, *Hindi*, *Bengali*, *Odia,* and *Gujarati*. The graphs in Figure 2

[6] http://trec.nist.gov/trec_eval/index.html.



(a) Bengali

(b) Gujarati

(c) Hindi

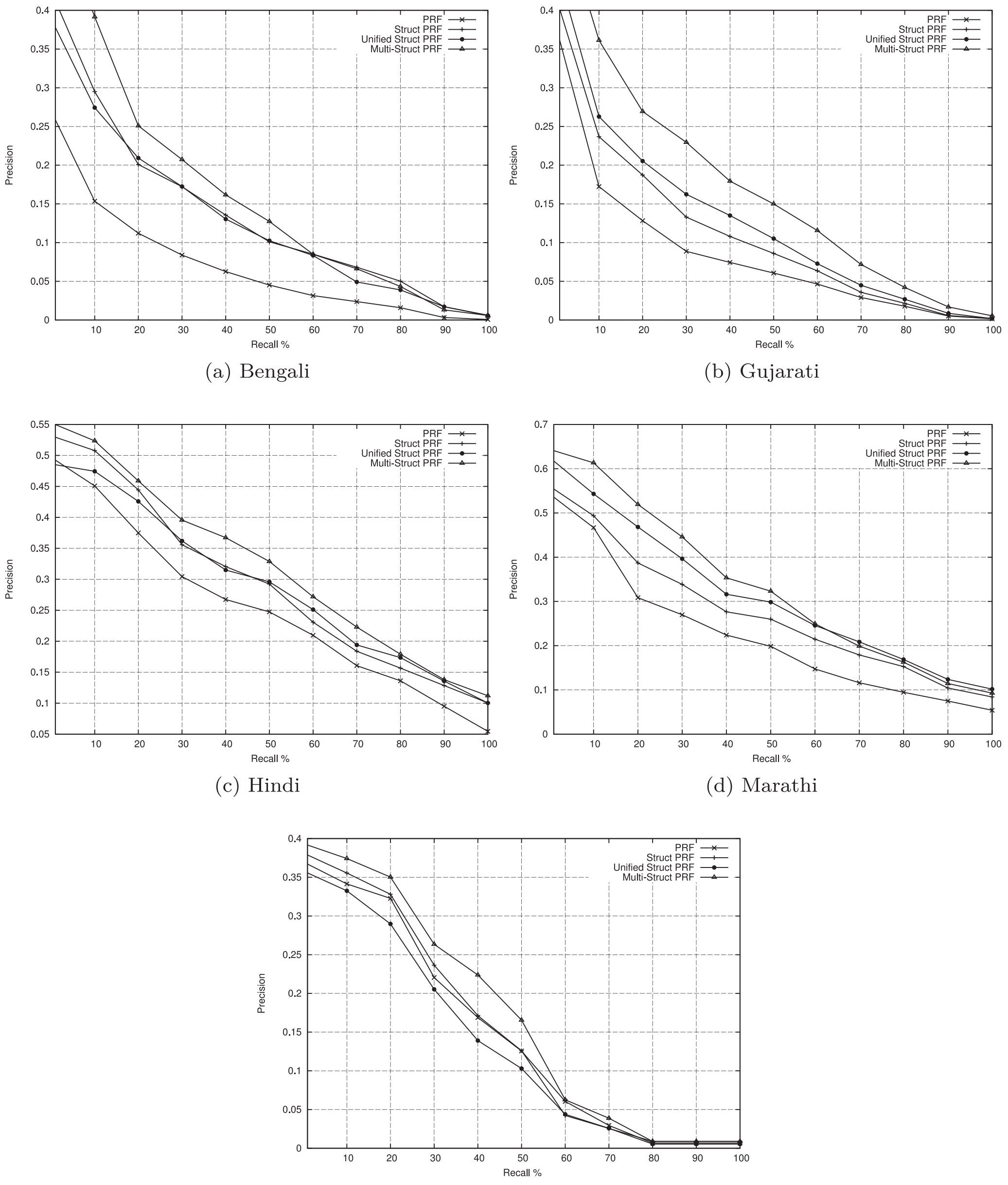
(d) Marathi

(e) Odia

Fig. 2. *P@k* values.

show the effectiveness of *MultiStructPRF*. There is a consistent improvement over *PRF* and *StructPRF* ranging from 5% to 152%. It is also evident that *MultiStructPRF* performs better than others for each value of *k*. This validates our hypothesis that using the help of a resource-rich assisting language and utilizing the document structure improve the performance of the search engine, which leads to better user satisfaction.

We plot precision-recall curves in Figure 3. We observe that the precision of *MultiStructPRF* is better than *PRF* and *StructPRF* for all the languages for almost all recall



(a) Bengali

(b) Gujarati

(c) Hindi

(d) Marathi

(e) Odia

Fig. 3. Precision-Recall curves.

values. This indicates that most of the relevant documents are pushed higher in the ranked list of documents.

More often than not, it is usually the top 10 documents that the user is interested in. In order to examine the effectiveness of our approach for such cases, we evaluate our system for *P@1* and *P@10*. We present the results in Table III. For *Bengali*, the relative improvement at *P@1* is 133% and at *P@10* is a 152% over *PRF*, while the relative improvement over Struct PRF at *P@1* and *P@10* is 40% and 24%, respectively.

Table III. Percentage Improvement in *MultiStructPRF* at *P@1* and *P@10*

	over *PRF(%)*		over *StructPRF(%)*	
Language	P@1	P@10	P@1	P@10
Marathi	25.00	36.33	15.38	17.63
Bengali	133.33	152.54	40.00	24.17
Odia	16.66	15.62	16.66	5.70
Gujarati	46.15	89.04	35.71	38.00
Hindi	9.06	28.71	9.06	6.65

Table IV. Results of T-test

Language	**t-value**	**Confidence**
Marathi	1.1	80%
Bengali	3.39	99%
Odia	0.423	Not significant
Gujarati	2.37	95%
Hindi	0.942	60%
Spanish	4.15	99.9%
Finnish	0.619	Not significant

Similarly for *Gujarati,* the relative improvement over *PRF* at *P@1* is 46% and at *P@10* is 89%, while the relative improvement over *StructPRF* at *P@1* is 35% and at *P@10* is 38%. We observe consistent improvement across all languages over both *PRF* and *StructPRF*.

4.3. Test of Significance

We carried out t-test significance in order to evaluate the significance of *MultiStructPRF* results compared to *PRF* results. Table IV represents the results of the significance test performed across various languages. The results obtained for Spanish are most significant, and we can say with 99.9% confidence that our results are significant. Also for Bengali and Gujarati, we can say that our results are 99% and 95% significant, respectively. The results for Marathi and Hindi are 80% and 60% significant. The relative improvement in *MAP* for Finnish and Odia was not much and hence the t-test results in “not significant.” However, Table III show that the improvement at lower positions (*P@1* and *P@10*) is significantly high.

4.4. Qualitative Analysis

In this section, we study the quality of expansion terms generated by *MultiStructPRF* using case studies.[7] Table V lists the top three expansion terms extracted in each of the *PRF* variants for multiple languages. It is evident from the table that the *MultiStructPRF* finds expansion terms that are closer to the query. For example, the top three expansion terms from *MultiStructPRF* for the query सचिन तेंडूलकर (sachin tendulkar) in *Marathi* include क्रिकेटपटू (cricket-patu, cricketer) and क्रिकेट (cricket). These terms are very closely related to the query but did not appear in the top three expansion terms of *PRF* and *StructPRF*. Similarly, we see that the expansion terms from *MultiStructPRF* are better than the expansion terms from *PRF* and *StructPRF* in *Gujarati* and *Hindi,* as shown in Table V. The same trend is observed for other languages as well.

[7] All non-English words used in case studies are associated with the corresponding transliteration and translation for ease of understanding.

Table V. Top 3 Expansion Terms from Various *PRF* Techniques

Language	**Query**	**PRF**	**StructPRF**	**MultiStructPRF**
Marathi	सचिन तेंडूलकर (sachin tendulkar)	मुंबई (mumbai) इंडियन्स (indians) पुणे (pune)	रमेश (ramesh) अंबाती (ambati) सामनावीर (saamnaveer, man of the match)	क्रिकेट (cricket) क्रिकेटपटू (cricket-patu, cricketer) मुंबई (mumbai)
Gujarati	ટાટા ની નેનો ગાડી (tata ni nano gaadi, tata's nano car)	પરીવહન (parivahan, vehicle) ઇન્ડિકા (indica) ઈન્ડિગો (indigo)	સેદાન (sedan) હેચબેક્સ (hatchback) ઓટોમોબાઇલ્સ (automobile)	કાર (car) ઉત્પાદક (utpadak, manufacturer) હેચબેક્સ (hatchback)
Hindi	स्वाईन फ्लू (swine flu)	रोग (rog, disease) विश्वमारी (vishwamari, pandemic) स्वास्थ्य (swastya, health)	इन्फ्लूएंजा (influenza) विश्वमारी (vishwamari, pandamic) तामीफ्लू (tamiflu)	वायरस(virus) इन्फ्लूएंजा(influenza) h1n1

Table VI. Comparison of MAP Values for Various Base and Assisting Languages

L_Q \ L_A	Odia	Marathi	Bengali	Gujarati	Hindi	English
Odia	–	0.1627	**0.1631**	0.1575	0.1584	0.1615
Marathi	0.3112	–	0.3171	0.3142	**0.3333**	0.3186
Bengali	**0.1684**	0.1559	–	0.1648	0.1631	0.1539
Gujarati	0.1538	**0.1627**	0.1544	–	0.1600	0.1606
Hindi	0.2694	**0.2974**	0.2882	0.2875	–	0.2946

4.5. Effect of Choice of Assisting Language

So far, we have seen that using the help of an assisting language and utilizing document structure significantly improves the performance of a search engine. But the question that remains unanswered is which assisting language to choose for a given query language.

In this section, we argue that the choice of an assisting language depends on the kind of application the search engine is going to serve. Our experiments show that for a search engine demanding high precision, like web search, we must choose an assisting language that is typologically closer to the query language, preferably belonging to same family as the query language. On the other hand, if a search engine demands high recall (e.g., patent search), then we should choose a resource-rich assisting language like English.

Table VI lists the precision values for various base and assisting language combinations across five Indian languages and English. These observations validate our hypothesis that typological closeness between the query language and the assisting language impacts the precision of the system. *Bengali* and *Odia* are culturally similar and share vocabulary. On the other hand, *Marathi*, *Hindi,* and *Gujarati* are lexically, syntactically, and typologically very similar languages. They all belong to same Indo-Aryan family of languages. A large amount of vocabulary is shared among *Hindi*, *Marathi,* and *Gujarati*. From the results we observe that *Bengali* acts as the best assisting language for *Odia* and vice versa. On the other hand, *Marathi* is the best choice for *Gujarati* and *Hindi*. Similarly, *Hindi* acts as the best assisting language for *Marathi*.

Table VII lists the recall values for various pairs of query and assisting languages. English Wikipedia has larger number of documents as compared to any other language, and it is more diverse and bound to cover more topics. As per the discussion in Section 3.4, the resource richness of a language is a factor of (i) number of documents and (ii) diversity of documents. Thus, English is more resource rich than any other language. Among Indian languages, *Hindi* is the most resource-rich language.

Table VII. Comparison of Recall Values for Various Base and Assisting Languages

L_A L_Q	Odia	Marathi	Bengali	Gujarati	Hindi	English
Odia	–	0.5238	**0.5291**	**0.5291**	**0.5291**	**0.5291**
Marathi	0.7744	–	0.7761	0.7827	0.7927	**0.7960**
Bengali	0.5358	0.5486	–	0.5304	0.5517	**0.5962**
Gujarati	0.7715	0.7799	0.7727	–	**0.7848**	0.7755
Hindi	0.9195	0.9098	0.9037	0.9051	–	**0.9224**

From the results, we observe the highest recall when English is used as an assisting language for all the query languages. It is important to note that the recall is positively correlated with the resource richness of an assisting language, whereas the precision is positively correlated with typological closeness between the query and the assisting languages. The above set of observations validates the claims that choice of an assisting language depends on the type of application the search engine is going to serve.

5. DISCUSSIONS

In this section, we discuss some of the factors which influence the performance of *MultiStructPRF*.

5.1. Limitations of Relevant Judgment Pool

The Relevance Judgment (RJ) pool is used to evaluate the retrieval performance. Evaluation forums use pooling technique to build an RJ pool, as explained by Sanderson and Braschler [2009]. This technique combines results from multiple search engines for a query to create a pool. An assessor then judges the relevance of every document in the pool. If systems chosen for pooling use *keyword search*, then documents that are relevant to the query but do not contain any query terms are not included in the pool. *MultiStructPRF* tries to retrieve relevant documents from the collection irrespective of the query term being present in the document or not. This technique of RJ pool creation may degrade the performance of *MultiStructPRF*.

5.2. Query and Document Processing

Terms in both the query and the document collection are morphologically analyzed before expansion. We use WordNet analyzer[8] for *English*, snowball stemmers[9] for *Spanish* and *Finnish,* and Indian language morphological analyzers[10] for Indian languages. Even though we have not estimated the impact of these stemmers on *MultiStructPRF* in this work, we believe that a linguistically rich morphological analyzer is more helpful than having a statistical stemmer. Since *MultiStructPRF* works based on the evidence of a term in the relevant set and the collection set, multiple stems for multiple inflections of the same term would mislead the term probability.

This phenomenon will affect all approaches built on the principle of *PRF*, but the impact will not be equal. *MultiStructPRF* utilizes multiple parts of the document for query expansion. Parts of the document having fewer terms, like *title,* will be significantly hampered by this phenomenon. This may degrade the performance of *MultiStructPRF*.

[8] http://projects.csail.mit.edu/jwi.

[9] http://snowball.tartarus.org/.

[10] http://www.cfilt.iitb.ac.in/Tools.html.

6. CONCLUSION

In this article, we introduced a novel multilingual framework for query expansion in resource-scarce languages. The framework uses the help of an assisting language and utilizes the document structure by giving differing importance to expansion terms from different parts of the document. We proposed a systematic model for weighting the expansion terms coming from different parts of the document. We then proposed a heuristic to combine the expansion terms from the query language and the assisting language. Our experimental results showed that *MultiStructPRF* significantly outperforms various PRF techniques. We also establish that the choice of an assisting language depends on the nature of the application. Applications demanding high recall must use a resource-rich assisting language, whereas applications demanding high precision must use an assisting language that is typologically closer to the query language.

REFERENCES

Bashar Al-Shboul and Sung-Hyon Myaeng. 2011. Query phrase expansion using Wikipedia in patent class search. In *AIRS*. 115–126.

Arjun Atreya, Yogesh Kakde, Pushpak Bhattacharyya, and Ganesh Ramakrishnan. 2013. Structure cognizant pseudo relevance feedback. In *Proceedings of IJCNLP*. 982–986.

Olivier Bodenreider. 2004. The unified medical language system (UMLS): Integrating biomedical terminology. *Nucleic Acids Research* 32, suppl 1 (2004), D267–D270.

Martin Braschler and Carol Peters. 2004. Cross-language evaluation forum: Objectives, results, achievements. *Information Retrieval* 7, 1–2 (2004), 7–31.

C. Buckeley, G. Salton, J. Allan, and A. Stinghal. 1994. Automatic query expansion using SMART. In *Proceedings of the 3rd Text Retrieval Conference*. 69–80.

Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 243–250.

Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)* 44, 1 (2012), 1.

Manoj K. Chinnakotla, Karthik Raman, and Pushpak Bhattacharyya. 2010a. Multilingual PRF: English lends a helping hand. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 659–666.

Manoj K. Chinnakotla, Karthik Raman, and Pushpak Bhattacharyya. 2010b. Multilingual pseudo-relevance feedback: Performance study of assisting languages. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 1346–1356.

Kevyn Collins-Thompson and Jamie Callan. 2005. Query expansion using random walk models. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*. ACM, 704–711.

W. Bruce Croft and David J. Harper. 1979. Using probabilistic models of document retrieval without relevance information. *Journal of Documentation* 35, 4 (1979), 285–295.

Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2004. A framework for selective query expansion. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management*. ACM, 236–237.

Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. 2002. Probabilistic query expansion using query logs. In *Proceedings of the 11th International Conference on World Wide Web*. ACM, 325–332.

Surya Ganesh and Vasudeva Verma. 2009. Exploiting structure and content of Wikipedia for query expansion in the context. In *International Conference RANLP*. 103–106.

Wei Gao, John Blitzer, and Ming Zhou. 2008. Using english information in non-english web search. In *Proceedings of the 2nd ACM Workshop on Improving Non-English Web Searching*. ACM, 17–24.

K. Sparck Jones, Steve Walker, and Stephen E. Robertson. 2000. A probabilistic model of information retrieval: Development and comparative experiments: Part 1. *Information Processing & Management* 36, 6 (2000), 779–808.

John Lafferty and Chengxiang Zhai. 2001a. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 111–119.

John Lafferty and Chengxiang Zhai. 2001b. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*. ACM, New York, 111–119. DOI:http://dx.doi.org/10.1145/383952.383970

Victor Lavrenko and W. Bruce Croft. 2001. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 120–127.

Craig Macdonald and Iadh Ounis. 2007. Expertise drift and query expansion in expert search. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM'07)*. ACM, New York, NY, USA, 341–350. DOI:http://dx.doi.org/10.1145/1321440.1321490

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Vol. 1. Cambridge University Press.

Mandar Mitra, Amit Singhal, and Chris Buckley. 1998. Improving automatic query expansion. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 206–214.

Yonggang Qiu and Hans-Peter Frei. 1993. Concept based query expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93)*. ACM, New York, 160–169. DOI:http://dx.doi.org/10.1145/160688.160713

Tetsuya Sakai, Toshihiko Manabe, and Makoto Koyama. 2005. Flexible pseudo-relevance feedback via selective sampling. *ACM Transactions on Asian Language Information Processing (TALIP)* 4, 2 (2005), 111–135.

M. Sanderson and M. Braschler. 2009. *Best Practices for Test Collection Creation and Information Retrieval System Evaluation*. Technical Report. TrebleCLEF Project.

Alan F. Smeaton, Fergus Kelledy, and Ruairi O'Donnell. 1995. TREC-4 experiments at Dublin City University: Thresholding posting lists, query expansion with WordNet and POS tagging of Spanish. *Harman [6]* (1995), 373–389.

Tao Tao and Cheng Xiang Zhai. 2006. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 162–169.

Dolf Trieschnigg, Djoerd Hiemstra, Franciska de Jong, and Wessel Kraaij. 2010. A cross-lingual framework for monolingual biomedical information retrieval. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, 169–178.

Ellen M. Voorhees. 2005. The TREC robust retrieval track. *ACM SIGIR Forum*, Vol. 39. ACM, 11–20.

Yang Xu, Gareth J. F. Jones, and Bin Wang. 2009a. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 59–66.

Yang Xu, Gareth J. F. Jones, and Bin Wang. 2009b. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09)*. ACM, New York, 59–66. DOI:http://dx.doi.org/10.1145/1571941.1571954

Zhijun Yin, Milad Shokouhi, and Nick Craswell. 2009. Query expansion using external evidence. In *Advances in Information Retrieval*. Springer, 362–374.

Chengxiang Zhai and John Lafferty. 2001. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th International Conference on Information and Knowledge Management*. ACM, 403–410.

Guangyou Zhou, Fang Liu, Yang Liu, Shizhu He, Jun Zhao, and others. 2013. Statistical machine translation improves question retrieval in community question answering via matrix factorization. *ACL (1)*. 852–861.

Guangyou Zhou, Kang Liu, Jun Zhao, and others. 2012. Exploiting bilingual translation for question retrieval in community-based question answering. In *COLING*. 3153–3170.

Guangyou Zhou, Zhiwen Xie, Tingting He, Jun Zhao, and Xiaohua Tony Hu. 2016. Learning the multilingual translation representations for question retrieval in community question answering via non-negative matrix factorization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24, 7 (2016), 1305–1314.

Received March 2016; revised July 2016; accepted September 2016