LiMe: Linear Methods for Pseudo-Relevance Feedback

Daniel Valcarce
University of A Coruña
Department of Computer Science
A Coruña, Spain
daniel.valcarce@udc.es

Javier Parapar
University of A Coruña
Department of Computer Science
A Coruña, Spain
javierparapar@udc.es

Álvaro Barreiro
University of A Coruña
Department of Computer Science
A Coruña, Spain
barreiro@udc.es

ABSTRACT

Retrieval effectiveness has been traditionally pursued by improving the ranking models and by enriching the pieces of evidence about the information need beyond the original query. A successful method for producing improved rankings consists in expanding the original query. Pseudo-relevance feedback (PRF) has proved to be an effective method for this task in the absence of explicit user's judgements about the initial ranking. This family of techniques obtains expansion terms using the top retrieved documents yielded by the original query. PRF techniques usually exploit the relationship between terms and documents or terms and queries. In this paper, we explore the use of linear methods for pseudo-relevance feedback. We present a novel formulation of the PRF task as a matrix decomposition problem which we called LiMe. This factorisation involves the computation of an inter-term similarity matrix which is used for expanding the original query. We use linear least squares regression with regularisation to solve the proposed decomposition with non-negativity constraints. We compare LiMe on five datasets against strong state-of-the-art baselines for PRF showing that our novel proposal achieves improvements in terms of MAP, nDCG and robustness index.

CCS CONCEPTS

• **Information systems → Information retrieval**; **Information retrieval query processing**; **Query reformulation**; **Retrieval models and ranking**;

KEYWORDS

Linear methods, pseudo-relevance feedback, query expansion, linear least squares

ACM Reference Format:
Daniel Valcarce, Javier Parapar, and Álvaro Barreiro. 2018. LiMe: Linear Methods for Pseudo-Relevance Feedback. In *SAC 2018: SAC 2018: Symposium on Applied Computing , April 9–13, 2018, Pau, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3167132.3167207

SAC 2018, April 9–13, 2018, Pau, France

ACM ISBN 978-1-4503-5191-1/18/04…$15.00
https://doi.org/10.1145/3167132.3167207

1 INTRODUCTION

In the beginning, search engines only considered the user's query to produce the document ranking. Soon, it was shown that the retrieval effectiveness can be notably improved considering the user's feedback for the presented results. Although relevance feedback is the most reliable type (users are asked to indicate which documents from the top are relevant) [29], it is often impractical. For this reason, research has focused on improving retrieval quality without relevance information or any further interaction from the user. This research line gave rise to the development of pseudo-relevance feedback (PRF). This approach assumes that the top documents returned by the retrieval engine are relevant and uses them to produce new query terms. In spite of this strong assumption, PRF has shown to be one of the most successful techniques for improving retrieval effectiveness [4, 6, 8, 13–16, 21, 23, 26–28, 37]. The terms obtained from the PRF method can be added to the original query or used in isolation to perform the second retrieval.

A lot of research has focused on improving and extending PRF techniques based on the language modelling framework [15, 16, 32]. In this paper, we propose an alternative formulation of the PRF task. Our proposal is not based on language models, but in linear methods. We introduce a novel formulation of the PRF task as a particular matrix decomposition problem called LiMe (Linear Methods). Most PRF techniques exploit the relationship between terms and documents or terms and queries. In contrast, our proposed factorisation method computes term similarities using the original query and the pseudo-relevance set. RFMF was the first formulation of PRF as a matrix decomposition problem [37] and computes a latent factor representation of documents/queries and items using non-negative matrix factorisation. Instead, in our work, we propose a different decomposition that stems from the computation of inter-term similarities. Previous work on translation models has exploited this concept of inter-term similarities [2, 12]; however, to the best of our knowledge, no state-of-the-art PRF approach directly leverages this information. Our matrix formulation enables to compute inter-term similarities that yield within the query and the pseudo-relevant set. We use the information of these relationships among terms to expand the original query. Since producing a good rank of expansion terms is critical for a successful PRF technique, modelling the relationship correctly among terms seems to be a desirable property. Additionally, computing good weights for those expansion terms is a key factor in the performance of a PRF technique. Our experiments show that the computation of inter-term similarities using information from the query and the pseudo-relevant set produces good rankings of expansion terms and also good weights for those terms.

As [37] showed, an advantage of addressing PRF as a matrix decomposition problem is that it admits different types of features for representing the query and the pseudo-relevant set. Since these features are independent of the retrieval model, LiMe is a general framework for PRF that can be plugged on top of any retrieval engine. Although we can plug in retrieval-dependent features or a theoretical probabilistic weighting function into LiMe if desired, we leave those ideas for future work. In this paper, we explore simple and well-known weighting functions such as TF and TF-IDF which allow us to outperform state-of-the-art techniques.

LiMe modelling of the PRF task paves the way for developing multiple PRF algorithms since the proposed matrix decomposition formulation can be calculated in various ways. In this paper, we use a method based on regularised linear least squares regression. On the one hand, we employ a ℓ_2 regularisation scheme to avoid overfitting. On the other hand, we use ℓ_1 regularisation to enforce sparsity into the learned inter-term similarities. This method provides automatic feature selection which gives us a more compact solution with the corresponding efficiency gains. The combination of ℓ_1 and ℓ_2 regularisation for linear least squares problems is also known as elastic net regression [40]. Additionally, we add non-negativity constraints to force the computed similarities to be positive increasing the interpretability of the decomposition.

We thoroughly evaluate the proposed method on five TREC collections. The obtained results show that LiMe outperforms state-of-the-art baselines in terms of several common effectiveness metrics. Moreover, LiMe achieved high values of robustness compared to the baselines. These findings highlight the applicability of LiMe as a pseudo-relevance feedback technique. Furthermore, LiMe formulation of the PRF task can exploit different features allowing the exploration of further features schemes.

In summary, the contributions of this paper are: (1) LiMe, a new matrix decomposition formulation of the PRF task involving inter-term similarities, (2) an algorithm based on constrained elastic net regression for solving the proposed matrix decomposition and computing expansion terms and (3) an empirical evaluation of the effectiveness of the proposed method against state-of-the-art baselines showing that LiMe is a competitive PRF method.

2 BACKGROUND

In this section, we describe the pseudo-relevance feedback procedure. We focus on state-of-the-art pseudo-relevance feedback techniques based on the language modelling framework [24] because they perform notably well in practice [13, 15, 16, 37]. Afterwards, we introduce linear methods for regression problems since our proposal rests on these models.

2.1 Pseudo-Relevance Feedback (PRF)

Query expansion aims to add new terms to the original query prompted by the user. These techniques can improve the performance of retrieval models when answering the users' information needs. Using true relevance feedback from the user is highly effective, but also difficult to obtain. Hence, automatic query expansion techniques, which do not require feedback from the user, can be extremely useful in practice [5]. Given the utility of these methods, it is not surprising that initial work on automatic query expansion dates from the sixties [18]. Manifold strategies for approaching this problem have been developed [5]; however, the foundations of PRF were established in the late seventies [8]. Pseudo-relevance feedback (also known as blind relevance feedback) is a highly effective strategy to improve the retrieval accuracy without user intervention [4, 6, 8, 13–16, 21, 23, 26, 37, 38]. Instead of using explicit feedback information from the user, the top retrieved documents by the user's original query are assumed to be relevant. These documents constitute the pseudo-relevant set. PRF techniques produce an expanded version of the original query using the information from the pseudo-relevant set. PRF methods use the expanded query for a second retrieval, and the results of the second ranking are presented to the user.

A plethora of strategies for weighting the candidate expansion terms using the pseudo-relevant set information have been developed. The Rocchio framework [28] was one of the very early successful methods presented in the context of the vector space model. Rocchio algorithm modifies the query vector in a direction which is closer to the centroid of the relevant documents vectors and further to the centroid of non-relevant documents vectors. In [4], the authors used this framework with different term weighting functions including those based on pseudo-relevant feedback instead of relevance feedback such as the Binary Independence Model [27], the Robertson Selection Value [26], the Chi-square method [4] or the Kullback-Leibler distance method [4].

2.2 PRF based on Language Models

Among all the PRF techniques in the literature, those developed within the Statistical Language Model framework [24] are arguably the most prominent ones because of their sound theoretical foundation and their empirical effectiveness [15]. Within the language modelling framework, documents are ranked according to the KL divergence $D(\cdot\|\cdot)$ between the query and the document language models, θ_Q and θ_D, which is rank equivalent to the negative cross-entropy [12]:

$$Score(D, Q) = -D(\theta_Q \| \theta_D) \stackrel{\text{rank}}{=} \sum_{t \in V} p(t|\theta_Q) \log p(t|\theta_D) \quad (1)$$

where V is the vocabulary of the collection. To obtain better results, instead of using the original query model θ_Q, we use θ'_Q which is the result of the interpolation between θ_Q and the estimated feedback model θ_F [1, 15]:

$$p(t|\theta'_Q) = (1 - \alpha)\, p(t|\theta_Q) + \alpha\, p(t|\theta_F) \quad (2)$$

where $\alpha \in [0, 1]$ controls the relative importance of the feedback model with respect to the query model. Therefore, the task of a PRF technique under this framework is to provide an estimate of θ_F given the pseudo-relevant set F. Next, we remind two state-of-the-art PRF techniques based on the language modelling framework [15].

2.2.1 Relevance-Based Language Models. Relevance-based language models or, simply, relevance models (RM) are a state-of-the-art PRF technique that explicitly introduces the concept of relevance in language models [13]. Although RM were originally conceived for standard PRF [13], they have been used in different ways such as the generation of query variants [6], cluster-based retrieval [14] or collaborative filtering recommendation [22].

Lavrenko and Croft [13] proposed two models for estimating the relevance: RM1 (which uses i.i.d. sampling) and RM2 (based on conditional sampling). We remind solely RM1 since it has shown to be more effective than RM2 [15]. RM1 estimates can be computed as follows when assuming uniform document prior probabilities:

$$p(t|\theta_F) \propto \sum_{D \in F} p(t|\theta_D) \prod_{q \in Q} p(q|\theta_D) \tag{3}$$

where $p(t|\theta_D)$ is the smoothed maximum likelihood estimate (MLE) of the term t under the language model of the document D with Dirichlet priors as the preferred smoothing technique [13, 38]. RM1 is typically called RM3 when it is interpolated with the original query (see Eq. 2) [1].

2.2.2 Maximum-Entropy Divergence Minimisation Model. The maximum-entropy divergence minimisation model (also known as MEDMM) [16] is a PRF technique based on the divergence minimisation model (DMM) [38] which stems from the language modelling framework. It is similar to the Rocchio algorithm from the vector space model if we use the pseudo-relevant set to compute the relevant documents vectors and the collection model for the non-relevant documents vectors [28]. MEDMM aims to find a feedback model θ_F which minimises the distance to the language models of the documents of the pseudo-relevant set and, at the same time, maximises the distance to the collection model θ_C (the assumed non-relevant model). This model has a parameter λ to control the IDF effect and parameter β to control the entropy of the feedback language model and is also interpolated with the original query. The analytic solution to MEDMM is given by [16]:

$$p(t|\theta_F) \propto \exp\left(\frac{1}{\beta} \sum_{D \in F} \alpha_D \log p(t|\theta_D) - \frac{\lambda}{\beta} \log p(t|\theta_C)\right) \tag{4}$$

where $p(t|\theta_D)$ is the smoothed MLE of the term t under the language model θ_D using additive smoothing with parameter γ. On the other hand, $p(t|\theta_C)$ represents the MLE of the term t in the collection. Finally, MEDMM gives a weight for each document based on the posterior of the document language model:

$$\alpha_D = p(\theta_D|Q) = \frac{p(Q|\theta_D)}{\sum_{D' \in F} p(Q|\theta'_D)} = \frac{\prod_{t \in Q} p(t|\theta_D)}{\sum_{D' \in F} \prod_{t' \in Q} p(t'|\theta'_D)} \tag{5}$$

2.3 PRF based on Matrix Factorisation

Other authors have focused on developing PRF models based on different ideas. In particular, RFMF was the first PRF technique based on matrix factorisation [37]. This approach builds a document-term matrix X from the query and the pseudo-relevant set. They built this matrix using TF-IDF or weights derived from the language modelling framework. RFMF reconstructs, through non-negative matrix factorisation (NMF), the document-term matrix and use the new weights as a scoring function to rank candidates terms for expansion. This approach is inspired by the Recommender Systems literature where matrix factorisation techniques are commonplace [11]. RFMF finds latent document and term factors with a particular parameter for the number of dimensions d of the latent factors.

Formally, NMF is a matrix factorization algorithm which decomposes the matrix $X \in \mathbb{R}_+^{m \times n}$ in two matrices $U \in \mathbb{R}_+^{m \times d}$ and $V \in \mathbb{R}_+^{d \times n}$ such that $X \approx UV$. U represents the latent factors of the query and the pseudo-relevant documents whereas V represents the latent factors of the terms.

2.4 Linear Methods

Linear methods are a simple but successful collection of techniques that have been used for regression and classification tasks. Given n features and m data points, $\vec{y} = (y_1, \ldots, y_m)^T$ is the column vector which contains the response and $\vec{x_1}, \ldots, \vec{x_n}$ are the m-dimensional vectors that contains each of the n features of the m observations. A linear method try to predict the response $\vec{y}$ using a linear combination of $\vec{x_1}, \ldots, \vec{x_n}$. The vectors of features can be arranged in the form of a matrix X of m rows and n columns. Linear regression aims to find the optimal values of the coefficients $\vec{w} = (w_1, \ldots, w_n)^T$ that minimise the error $\vec{\epsilon}$:

$$\vec{y} = X\vec{w} + \vec{\epsilon} = w_1\vec{x_1} + \cdots + w_n\vec{x_n} + \vec{\epsilon} \tag{6}$$

In particular, ordinary linear least squares models try to find the best approximate solution of this system of linear equations where the sum of squared differences between the data and the prediction made by the model serves as the measure of the goodness of the approximation:

$$\vec{w}^* = \arg\min_{\vec{w}} \|\vec{\epsilon}\|_2^2 = \arg\min_{\vec{w}} \|\vec{y} - X\vec{w}\|_2^2 \tag{7}$$

Linear least squares loss is strictly convex; thus, it has a unique minimum. Moreover, the simplicity of the model favours its explainability and interpretability. However, this model suffers from overfitting. For tackling this problem, it is common to add ℓ_2 or Tikhonov regularisation (this model is also known as ridge regression [9]). Imposing a penalty based on the squared ℓ_2-norm of the coefficients $\vec{w}$ produces a shrinking effect which is controlled by the non-negative parameter β_2:

$$\vec{w}^* = \arg\min_{\vec{w}} \|\vec{y} - X\vec{w}\|_2^2 + \beta_2 \|\vec{w}\|_2^2 \tag{8}$$

An alternative strategy to ridge regression is imposing a penalty based on the ℓ_1-norm of the coefficient vector. This approach is commonly known as lasso regression [34]. This approach performs automatic feature selection as the value of the non-negative parameter β_1 grows:

$$\vec{w}^* = \arg\min_{\vec{w}} \|\vec{y} - X\vec{w}\|_2^2 + \beta_1 \|\vec{w}\|_1 \tag{9}$$

Since both, ridge and lasso regressions, have beneficial properties, Zou and Hastie [40] developed a technique combining both ℓ_1 and ℓ_2 regularisation. The elastic net is a generalisation of ridge and lasso regression. This approach can perform shrinkage and feature selection at the same time controlled by the non-negative parameters β_1 and β_2:

$$\vec{w}^* = \arg\min_{\vec{w}} \|\vec{y} - X\vec{w}\|_2^2 + \beta_1 \|\vec{w}\|_1 + \beta_2 \|\vec{w}\|_2^2 \tag{10}$$

3 LIME: LINEAR METHODS FOR PRF

Our proposal, LiMe, is designed for ranking the candidate terms for producing an expanded query Q'. As it is usual in PRF, LiMe uses only information about the original query Q and the pseudo-relevant set F. The set F is composed of the top-k documents retrieved using the original query Q. We should note that LiMe treats the query as another document. Thus, for convenience, we define the extended feedback set F' as the pseudo-relevant set plus the original query ($F' = \{Q\} \cup F$) and we denote its cardinality by $m = |F'| = k + 1$. We consider as candidate terms the subset of words from the collection vocabulary V that appear in F'. We refer to this set by $V_{F'}$ and we denote its cardinality by $n = |V_{F'}|$.

In contrast to LiMe which considers the query and the pseudo-relevant documents jointly to compute the expansion terms, RM3 and MEDMM exploit the query for weighting the documents of the pseudo-relevant set.

It is interesting to remark that LiMe is a general method. This means that it is independent of the algorithm used for computing the pseudo-relevant set F (first retrieval) and also independent of the algorithm employed for producing the final ranking with the expanded query (second retrieval).

3.1 LiMe Formulation

We can define LiMe using a matrix or a vector formulation. To understand better the idea behind LiMe, we initially present our technique under a matrix formulation. Afterwards, we introduce the vector representation which is much more convenient from the point of view of its implementation.

Considering the query as another pseudo-relevant document, we define the matrix $X = (x_{ij}) \in \mathbb{R}^{m \times n}$. The first row represents the original query Q while the rest rows correspond the k documents from F. Each column of X corresponds to a term from $V_{F'}$. Each element x_{ij} represents a feature between the document (or query) corresponding to the i-th position and the term t_j represented with the j-th column of X. Therefore, each row of X is a sparse feature vector representing the query or a pseudo-relevant document.

The objective of LiMe is to factorise this matrix X into the product of itself and another matrix $W = (w_{uv}) \in \mathbb{R}_{+}^{n \times n}$. This new matrix represents the inter-term similarity among the words in $V_{F'}$. In particular, each entry w_{uv} symbolises the similarity between terms t_u and t_v. To favour interpretability, we constrain the similarities to be non-negative. Moreover, to avoid the trivial solution (W equal to the identity matrix) we enforce that the main diagonal of W are all zeros. Formally, we define LiMe as an algorithm that computes the following decomposition:

$$\begin{aligned} X &\approx X\,W \\ s.t.\ \operatorname{diag}(W) &= 0,\ W \geq 0 \end{aligned} \tag{11}$$

We formulate this matrix decomposition task as a constrained linear least squares optimisation problem. We want to minimise the residual sum of squares of the factorisation. Additionally, to avoid overfitting and to enforce a sparse solution we apply the elastic net penalty which combines ℓ_1 and ℓ_2 regularisation. In this way, the objective function of LiMe is the following one:

$$\begin{aligned} W^* = \ & \underset{W}{\arg\min} \quad \frac{1}{2}\|X - X\,W\|_F^2 + \beta_1 \|W\|_{1,1} + \frac{\beta_2}{2}\|W\|_F^2 \\ & \text{s.t.} \qquad \operatorname{diag}(W) = 0,\ W \geq 0 \end{aligned} \tag{12}$$

Note that the matrix $\ell_{1,1}$-norm (denoted by $\|\cdot\|_{1,1}$) is equivalent to the sum of the ℓ_1-norm of the columns. On the other hand, the squared Frobenius norm (denoted by $\|\cdot\|_F^2$) is calculated as the sum of the squares of each matrix element which is equivalent to the sum of the squared ℓ_2-norm of the columns. Using these equivalences between the matrix and vector norms, we can split this matrix formulation by columns rewriting the optimisation problem in the following vector form:

$$\begin{aligned} \vec{w}_{\cdot j}^* = \ & \underset{\vec{w}_{\cdot j}}{\arg\min} \quad \frac{1}{2}\left\|\vec{x}_{\cdot j} - X\vec{w}_{\cdot j}\right\|_2^2 + \beta_1 \left\|\vec{w}_{\cdot j}\right\|_1 + \frac{\beta_2}{2}\left\|\vec{w}_{\cdot j}\right\|_2^2 \\ & \text{s.t.} \qquad w_{jj} = 0,\ \vec{w}_{\cdot j} \geq 0 \end{aligned} \tag{13}$$

where the non-negativity constraint is applied to the elements of $\vec{w}_{\cdot j}$ vector which is the j-th column of the W matrix. Similarly, $\vec{x}_{\cdot j}$ represents the j-th column of the X matrix. In this way, for each term j in $V_{F'}$, we train an elastic net [40] with an equality constraint to zero in one coefficient and non-negativity constraints on the rest of the coefficients.

Once the regression problems depicted in Eq. 13 are solved for each column (i.e., each term in $V_{F'}$), we merge their solutions to build the inter-term similarity matrix W^*. Now, we employ the computed matrix decomposition to reconstruct the first row of X (which we will denote by $\hat{x}_{1\cdot}$) as follows:

$$\hat{x}_{1\cdot} = \vec{x}_{1\cdot} W^* \tag{14}$$

Note that, by construction, X is a sparse matrix (hence also the row vector $\vec{x}_{1\cdot}$) and W^* will be a sparse matrix due to the ℓ_1 regularisation. Thus, the product between the row vector $\vec{x}_{1\cdot}$ and the matrix W^* is highly efficient. We use the pseudo-relevant documents for learning the inter-term similarities, but we reconstruct the first row of X because we want to expand only the query.

We can normalise the reconstructed vector $\hat{x}_{1\cdot}$ to obtain a probability estimate. The probability of the j-th term given the LiMe feedback model is given by:

$$p(t_j|\theta_F) = \begin{cases} \dfrac{\hat{x}_{1j}}{\sum_{t_v \in V_{F'}} \hat{x}_{1v}} & \text{if } t_j \in V_{F'}, \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

We only rank those terms that appear in the pseudo-relevant set or the query. Although some PRF techniques can rank all the terms in the collection, in practice, it is common to only rank those appearing in the pseudo-relevant set or the query [13, 37]. In fact, scoring terms that do not appear in F' would contradict the foundations of PRF since this approach is based on local information (i.e., the pseudo-relevant set and the query).

Although both LiMe and RFMF decomposes a similar matrix, they use different objective functions and optimisation algorithms. Additionally, LiMe employs elastic net regularisation. In contrast, RFMF is based on non-negative factorisation which can deal with non-negative and sparse data while LiMe deals with this data by enforcing non-negativity constraints in the optimisation problem. Additionally, LiMe discovers inter-term similarities that yield within the pseudo-relevant set and the query while RFMF learns term latent factor representations.

Next, we discuss how we fill matrix $X = (x_{ij})$ with features relating query/documents i with terms j. Last, we provide implementation details about how to solve this constraint linear squares problem since the original method for solving the elastic net [40] does not consider our constraints.

3.2 Feature Schemes

One advantage of LiMe is its flexibility: we can use any feature scheme to build matrix X. To foster sparsity in matrix X, we decide to fill with zeros all those entries that correspond to terms that do not appear in the current document. This approach will provide a quite sparse matrix which can be more efficiently decomposed than a dense one.

Let $s(w, D)$ be the function that assigns a score to the term w given the document D and let $f(w, D)$ be the frequency of occurrence of term w in document D, the matrix X is filled in the following manner:

$$x_{ij} = \begin{cases} s(w_j, Q) & \text{if } i = 1 \text{ and } f(w_j, Q) > 0, \\ s(w_j, D_{i-1}) & \text{if } i > 1 \text{ and } f(w_j, D_{i-1}) > 0, \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

We explored several strategies based on well-known weighting functions used in Information Retrieval. We studied several term frequency measures: raw frequency counts, binarised counts and logarithmic versions. Additionally, we tried different TF-IDF formulations. We achieved the best results using the following TF and TF-IDF weighting functions proposed by Salton [31]:

$$s_{tf}(w, D) = 1 + \log_2 f(w, D) \tag{17}$$

$$s_{tf\text{-}idf}(w, D) = \big(1 + \log_2 f(w, D)\big) \times \log_2 \frac{|C|}{df(w)} \tag{18}$$

where $|C|$ refers to the number of documents in the collection and $df(w)$ the document frequency of term w (i.e., the number of documents in the collection where the term w occurs).

Anyway, other alternatives may be possible. For example, it may be worth exploring features related to the first retrieval such as the contribution of an individual term to the document score within a particular retrieval model; however, in that case, LiMe would not be independent of the retrieval technique. Also, we could derive probabilistic weighting functions (as RFMF does) at the expense of introducing a few new parameters to tune into the model. We leave for future work the investigation of additional features schemes. Nevertheless, the ability of LiMe of performing well with simple and well-known features such as TF and TF-IDF is remarkable. Also, these heuristics are supported by decades of research in Information Retrieval.

3.3 Implementation Details

As illustrated in Eq. 13, the computation of matrix W^* can be divided in multiple linear regression problems, one for each vector $\vec{w}^*_{\cdot j}$ which represents a term in $V_{F'}$. Thus, each column of matrix W^* can be computed separately and, if needed, in parallel without any dependencies among them.

To solve each regression problem, we used the highly efficient BCLS[1] (Bound-Constrained Least Squares) library, which implements a two-metric projected-descent method for solving bound-constrained least squares problems.

An additional optimisation is to drop part of the matrix W^*. This matrix is used for computing expansion terms when multiplied by vector $\vec{x}_{1\cdot}$ (see Eq. 14). Therefore, we only need those rows that correspond to a term in the original query. If we only store those similarities, we save a lot of space since the number of terms in a query prompted by a user is tiny compared to the number of rows.

[1] See http://www.cs.ubc.ca/~mpf/bcls

Table 1: Collections statistics.

Collection	#docs	Avg doc length	Topics	
			Training	Test
AP88-89	165k	284.7	51-100	101-150
TREC-678	528k	297.1	301-350	351-400
Robust04	528k	28.3	301-450	601-700
WT10G	1,692k	399.3	451-500	501-550
GOV2	25,205k	647.9	701-750	751-800

4 EXPERIMENTS

In this section, we assess the performance of LiMe against state-of-the-art techniques. The experiments were performed using Terrier [17] on five TREC collections. We describe the evaluation methodology and then we explain the choice of baselines and the parameter setting. Finally, we present and analyse the results comparing the behaviour of LiMe with respect to the baselines.

4.1 Evaluation Methodology

We conducted our experiments on five TREC collections commonly used in PRF literature [15, 16, 37]: AP88-89, TREC-678, Robust04, WT10G and GOV2. The first one is a subset of the Associated Press collection from years 1988 and 1989. The second collection is based on TREC disks 4 and 5. The third dataset was used in the TREC Robust Track 2004 and consists of poorly performing topics. The fourth one, the WT10G collection, is a general web crawl used in the TREC Web track 2000-2001. Finally, we also ran our experiments on a large dataset, the GOV2 collection, which is a web crawl of `.gov` websites from 2004 (used in the TREC Terabyte track 2004-2006 and the Million query track 2007-2008). We applied training and test evaluation on all collections. We tuned the model hyperparameters that maximise MAP (mean average precision) using the training topics and we used the test topics to evaluate the performance of the methods. Table 1 describes each collection and the training and test splits.

We produced a rank of 1000 documents per query. We evaluated MAP and nDCG (normalised discounted cumulative gain) using `trec_eval`[2] at a cut-off of 1000. Additionally, we measured the RI (robustness index or reliability of improvement [30]) against the non-expanded query. This metric, which ranges in the interval $[-1, 1]$, is computed as the number of topics improved by using PRF minus the number of topics hurt by the PRF technique divided by the number of topics. We employed one-tail permutation test with 10,000 randomisations and $p < 0.05$ to measure if the improvements in terms of MAP and nDCG were statistically significant [33]. We cannot apply any paired statistic to RI because it is a global metric.

We used queries based only on the title field of the TREC topics because short queries are the most common scenario for the application of PRF techniques. Previous work showed that stemming and stopwords removal is beneficial for the PRF task [15]. For this reason, we preprocessed the collections with the standard Terrier stopwords removal and Porter stemmer.

[2] See http://trec.nist.gov/trec_eval

4.2 Baselines and Parameter Setting

We employed the state-of-the-art language modelling framework for performing the first and second stage retrievals [24]. In particular, we used KL divergence model (see Eq. 1) which allow us to introduce a feedback model easily [12]. For smoothing the document language models, we used Dirichlet priors smoothing [39] with parameter μ set to 1000. As baselines, we selected a representative set of state-of-the-art techniques.

LM First, we should always compare a PRF technique against the performance of a retrieval model without feedback information. We used the same retrieval settings (Dirichlet priors $\mu = 1000$).

RFMF We included this technique because it is based on matrix factorisation. It builds a document-term matrix from the query and the pseudo-relevant set and employs non-negative matrix factorisation [37]. We set the number of dimensions of the factorisation, d, to the size of the relevant set plus one as the authors recommended [37]. We employed the TF-IDF weight function.

MEDMM We also employed the maximum-entropy divergence minimisation model which is recognised as one of the most competitive PRF techniques [16]. We followed the recommendations of the authors and we set the IDF parameter λ to 0.1, the entropy parameter β to 1.2 and the additive smoothing parameter γ to 0.1 [16].

RM3 Relevance-based language models are an effective PRF technique based on the language modelling framework. We use Dirichlet priors for smoothing the maximum likelihood estimate of the relevance models. We used RM3, the most effective estimate, which uses i.i.d. sampling method and interpolates the original query with the feedback model [1, 13]. We set the Dirichlet priors smoothing parameter μ' to 1000 as it is typically done [15, 16, 37].

For all the PRF models, we swept the number of top k documents retrieved in the first stage among $\{5, 10, 25, 50, 75, 100\}$ and the number of expansion terms e among $\{5, 10, 25, 50, 75, 100\}$. We swept the interpolation parameter α from 0 to 1 in steps of 0.1. Regarding LiMe, we trained the β_1 and β_2 parameters. We tuned the values of β_1 among $\{0.01, 0.1, 1.0\}$ and parameter β_2 among $\{10, 25, 50, 100, 150, 200, 250, 300, 350, 400, 450\}$. We tested both the TF and the TF-IDF feature schemes. We selected those parameters that maximise the values of MAP in the training set.

4.3 Results and Discussion

The results of the experiments regarding MAP, nDCG and RI are summarised in Table 2. Overall, all the PRF techniques outperform the language modelling baseline without query expansion. LiMe-TF and LiMe-TF-IDF are the only methods that offered significant improvements over LM in MAP and nDCG on all collections.

To further analyse if PRF techniques are beneficial, we measured the robustness index. This value is positive for all the methods on every collection. This means that, in average, more queries were improved rather than worsened due to the PRF techniques. Either LiMe-TF or LiMe-TF-IDF achieved the higher figures in RI on every dataset except for MEDMM on the WT10G collection. Additionally, RM3 achieve the same robustness index as LiMe-TF-IDF on Robust04.

On all datasets, at least one of the LiMe techniques achieved the highest results in terms of MAP and nDCG. No baseline outperformed LiMe on any dataset. LiMe-TF-IDF significantly surpassed RFMF on four out of five datasets in terms of MAP and on three out of five collections in terms of nDCG. Regarding RM3, LiMe-TF-IDF significantly outperformed RM3 on two collections. The strongest baseline, MEDMM, was only significantly surpassed by LiMe-TF-IDF on the AP88-89 collection. However, on all datasets, both LiMe-TF and LiMe-TF-IDF showed higher values in nDCG and MAP than MEDMM. Although no baseline significantly improved LiMe, MEDMM significantly surpassed RM3 in terms of nDCG and MAP on the TREC-678 collection. Also, RM3 and MEDMM significantly improved RFMF in terms of MAP and nDCG on several datasets.

It is interesting to remark that the PRF techniques achieved the smallest improvements on the WT10G collection. This is probably due to the nature of the web which is a noisy media. Also, the values of RI on this dataset are the lowest.

Regarding the differences between LiMe-TF and LiMe-TF-IDF, the latter approach showed better figures of MAP and nDCG on all datasets except on the WT10G collection. Nevertheless, the differences are significant only on the non-web datasets (AP88-89, TREC-678 and Robust04). In contrast, LiMe-TF provided a slightly higher RI on three out of five datasets compared to LiMe-TF-IDF. As we commented, WT10G is a quite noisy web crawl. This result may indicate that LiMe-TF is only adequate on noisy datasets. Also, the optimal number of expansion terms e is smaller for LiMe-TF-IDF than for LiMe-TF in four out of five collections which is also a good feature in terms of efficiency.

4.3.1 Query Analysis. To provide insights into the good results achieved by LiMe-TF and LiMe-TF-IDF, we manually studied the expanded queries produced by the tested PRF methods. We show in Table 3 the top 10 expansion terms for the TREC topic 664 (“American Indian Museum”) on the Robust04 collection.

RM3 provided bad expansion terms by adding very common uninformative terms such as “will”, “1” or “new”. This seems to be a problem of low IDF effect. In contrast, MEDMM yielded much better expansion terms. However, some of them are of dubious utility such as “live” or “part”. RFMF provided specific terms but some of them are completely unrelated to the topic (e.g., “dolphin” or “rafaela”). Hence, the inferior performance of RFMF is likely to be due to the introduction of noisy terms. Regarding our methods, we can see than LiMe-TF provided good expansion terms. Still, this approach included the term “part” (as MEDMM did) which we think is uninformative. In this case, LiMe-TF-IDF yielded the best expansion terms. All of them are specific and related to the topic.

In the light of the results, we can claim that RM3 and MEDMM tend to foster those terms that appear in the majority of the pseudo-relevant set in contrast to matrix factorisation approaches. LiMe-TF-IDF was capable of selecting very specific and relevant terms such as “smithsonian”. RFMF was also able to include relevant terms such as “professor” but it also added non-related terms. Therefore, the main advantage of the matrix formulation is its ability to select discriminative words without being biased to popular and non-informative terms in the pseudo-relevant set. However, our approach base on inter-term similarities is able to select relevant terms while RFMF factorisation approach based on document and term latent factors is incapable of filtering non-related terms.

4.3.2 Sensitivity Analysis of Parameters. Regarding the parameters of LiMe, we observed that the differences in effectiveness between LiMe-TF and LiMe-TF-IDF when we changed the value of β_1 were minor. We can set β_1 to 0.01 reducing the number of parameters to tune and obtaining good results. Nevertheless, the inclusion of ℓ_1 regularisation into the LiMe model is still beneficial since it provides sparsity to the learned matrix W with the corresponding space savings. Regarding β_2, we plotted the values of MAP achieved by LiMe-TF and LiMe-TF-IDF with different amount of ℓ_2 regularisation in Fig. 1. Except for the WT10G collection, the parameter β_2 is fairly stable among the values 150 and 400 for both LiMe-TF and LiMe-TF-IDF.

We also studied how LiMe-TF-IDF behaves varying the size of the pseudo-relevant set k, the number of expansion terms e and the interpolation parameter α against the baselines RFMF, MEDMM and RM3. Figure 2 summarises the results of the sensitivity analysis in terms of MAP. The general trend is that a high number of pseudo-relevant documents hurts the performance of the PRF techniques. The optimal number of feedback documents was not greater than 25. LiMe methods and RM3 are quite stable and they behave optimally with 5-10 documents. In constrast, RFMF and MEDMM require 25 documents in the pseudo-relevant set depending on the dataset.

The optimal values of parameter e are quite variable. MEDMM and RM3 require more expansion terms than any other approach except on the WT10G dataset which is the noisiest one. LiMe methods are robust to noisy collections and work well with a high number of terms on WT10G. In contrast, RFMF is the technique that requires the smallest number of expansion terms in general. Finally, LiMe techniques are situated between the two extremes.

Regarding the interpolation parameter α, except for the GOV2 collection, we observed that the optimal value for LiMe-TF-IDF lies within a narrower interval than the optimal values for RFMF, MEDMM and RM3. Nevertheless, we can see that α has a notable impact on any PRF technique and we should properly tuned it. Overall, the performance of RFMF is very unstable when we vary α (to a lesser extend, this is also true when varying the other parameters). We also found that when we do not interpolate the feedback model with the original query by setting $\alpha = 1$ (i.e., when we use the feedback model as the expanded query), RM3 showed the lowest performance. In general, we observed that LiMe-TF-IDF and MEDMM generate better feedback models to use in isolation.

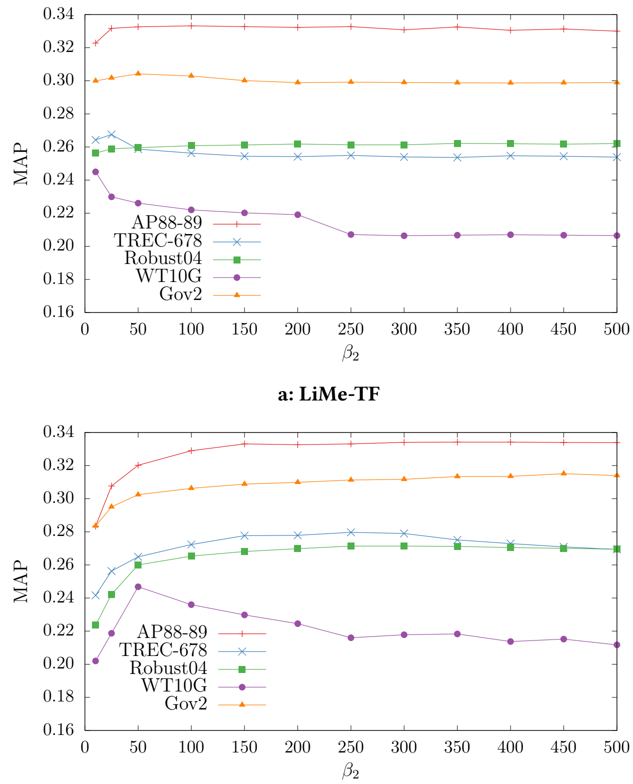
5 RELATED WORK

Pseudo-relevance feedback has become a fertile area of research [4, 6, 8, 13–16, 21, 23, 26–28, 32, 37]. Those PRF techniques based on language models are among the most effective [15] and for this reason were used as baselines. Also, since RFMF was the first work on applying matrix factorisation to PRF, we also included it as baseline and described in the background section.

PRF methods have been adapted to collaborative filtering recommendation with great success [22]. RM [22] or the Rocchio framework [35] are two examples of PRF approaches used as recommender systems. Conversely, RFMF is a case of a recommendation technique applied to PRF [37].

Following this analogy between PRF and collaborative filtering, we can find a state-of-the-art recommendation technique, SLIM [20], which is also based on linear methods. SLIM decomposes the full user-item feedback producing an item-item similarity matrix using ℓ_1 and ℓ_2 regularisation. With this decomposition, they reconstruct the full user-item feedback matrix to generate recommendations. In contrast, we only need to predict the first row of X since we only have to expand the query. As SLIM does, LiMe fills with zeros all the missing values of the input matrix. In the beginning, in Recommender Systems, those unknown values were not set to zero. Instead, the objective function was optimised only for the known elements. However, later research found that this procedure produces worse rankings than dealing with the whole matrix considering all missing values as zeros [7].

Although RFMF and LiMe are PRF techniques based on matrix factorisation, they compute different decompositions. The differences in performance are explained by the use of different objective functions and optimisation algorithms. LiMe minimises the elastic net loss and RFMF minimises the KL-divergence of the NMF



a: LiMe-TF

b: LiMe-TF-IDF

Figure 1: Sensitivity of LiMe methods to β_2 on the AP88-89, TREC-678, Robust04, WT10G and GOV2 collection. The rest of the parameters were fixed to their optimal values.

Table 2: Values of MAP, P@5, nDCG and RI for LM, RFMF, MEDMM, RM3, LiMe-TF and LiMe-TF-IDF techniques on the AP88-89, TREC-678, Robust04, WT10G and GOV2 collection. Statistically significant improvements according to permutation test ($p < 0.05$) w.r.t. to LM, RFMF, MEDMM, RM3, LiMe-TF and LiMe-TF-IDF are superscripted with a, b, c, d, e and f, respectively.

Collection	Metric	LM	RFMF	MEDMM	RM3	LiMe-TF	LiMe-TF-IDF
AP88-89	MAP	0.2349	0.2774^{a}	0.3010^{a}	0.3002^{a}	0.3062^{a}	**0.3149**^{abcde}
	nDCG	0.5637	0.5749^{a}	0.5955^{ab}	0.6005^{ab}	0.6003^{ab}	**0.6085**^{ab}
	RI	–	0.42	0.42	0.50	0.38	**0.52**
TREC-678	MAP	0.1931	0.2072	0.2327^{abd}	0.2235^{a}	0.2267^{a}	**0.2357**^{abd}
	nDCG	0.4518	0.4746	0.5115^{abd}	0.4987^{ab}	0.5051^{ab}	**0.5198**^{abde}
	RI	–	0.23	0.26	0.40	**0.48**	0.46
Robust04	MAP	0.2914	0.3130^{a}	0.3447^{ab}	0.3488^{ab}	0.3388^{ab}	**0.3517**^{abe}
	nDCG	0.5830	0.5884	0.6227^{ab}	0.6251^{ab}	0.6223^{ab}	**0.6294**^{ab}
	RI	–	0.07	0.32	**0.37**	0.23	**0.37**
WT10G	MAP	0.2194	0.2389^{a}	0.2472^{a}	0.2470^{a}	**0.2484**^{a}	0.2476^{a}
	nDCG	0.5212	0.5262	0.5324	0.5352	**0.5416**^{a}	0.5398^{a}
	RI	–	0.30	**0.36**	0.20	0.32	0.30
GOV2	MAP	0.3310	0.3580^{a}	0.3790^{ab}	0.3755^{ab}	0.3776^{ab}	**0.3830**^{ab}
	nDCG	0.6325	0.6453	0.6653^{ab}	0.6618^{ab}	0.6656^{ab}	**0.6698**^{abd}
	RI	–	0.42	0.66	0.60	**0.68**	0.62

Table 3: Top 10 expansion terms for the TREC topic 664 ("American Indian Museum") when using the different PRF methods on the Robust04 collection.

a: RFMF

term	weight
indian	0.1725
museum	0.1685
american	0.1505
professor	0.0193
tribal	0.0160
ancient	0.0155
dolphin	0.0153
rafaela	0.0140
activist	0.0137
racist	0.0137

b: MEDMM

term	weight
indian	0.1511
museum	0.0802
american	0.0780
cultur	0.0210
year	0.0177
live	0.0153
nation	0.0148
artifact	0.0146
part	0.0139
tribal	0.0127

c: RM3

term	weight
indian	0.1285
american	0.0895
museum	0.0874
year	0.0219
will	0.0209
west	0.0182
1	0.0167
tribal	0.0158
time	0.0149
new	0.0147

d: LiMe-TF

term	weight
indian	0.0895
american	0.0873
museum	0.0855
year	0.0189
cultur	0.0184
nation	0.0181
part	0.0177
time	0.0168
tribal	0.0157
artifact	0.0146

e: LiMe-TF-IDF

term	weight
indian	0.1392
museum	0.1364
american	0.1256
tribe	0.0393
artifact	0.0306
cultur	0.0272
tribal	0.0271
nation	0.0249
chumash	0.0219
smithsonian	0.0212

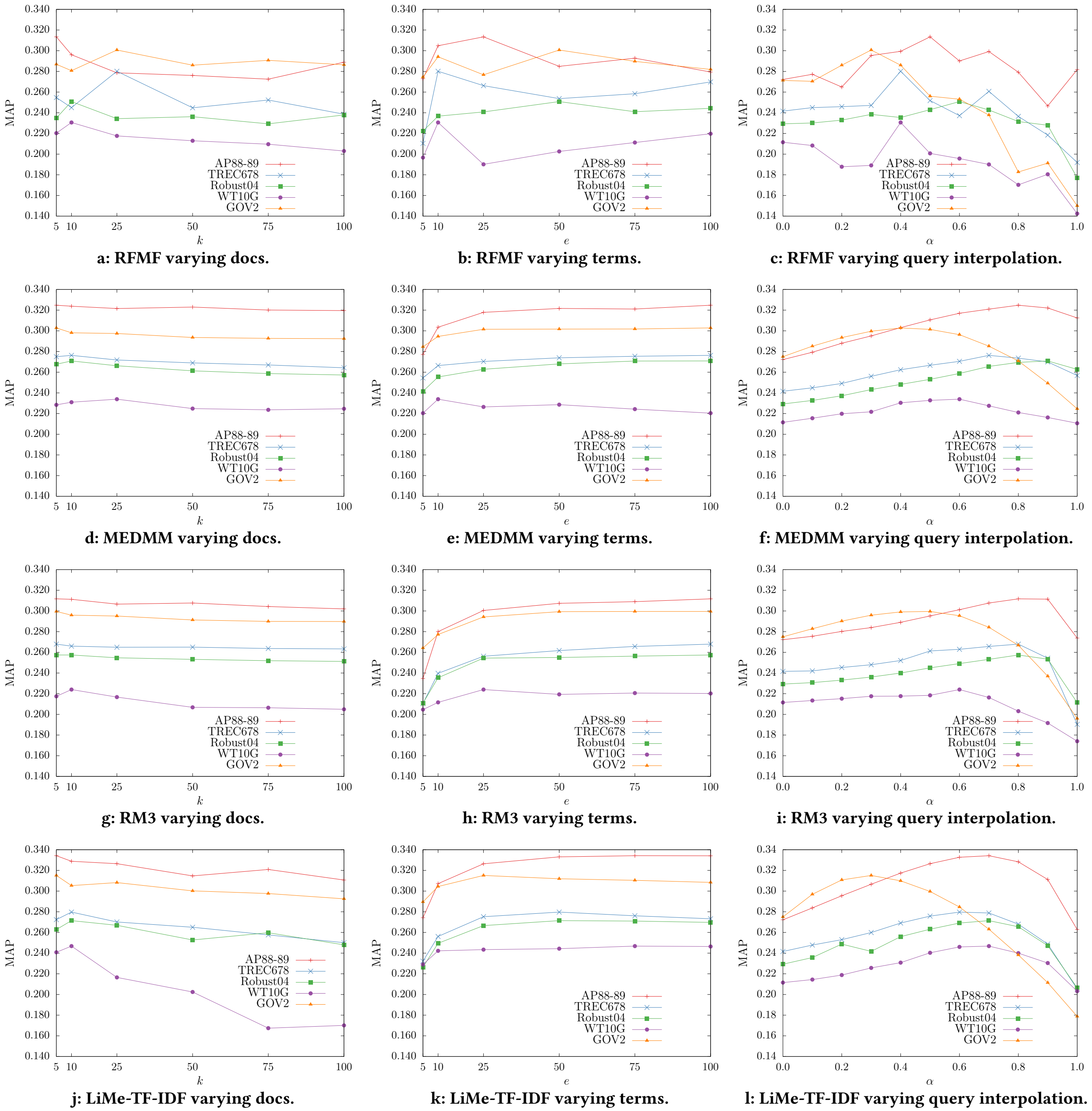
decomposition. This diversity in performance is also found in collaborative filtering where approaches such as SLIM outperforms several alternative matrix factorization techniques [20].

Linear methods have also been used in Information Retrieval. For example, [19] proposed a learning to rank approach based on linear models that directly maximise MAP. Moreover, linear methods have been applied to other tasks such a query difficulty prediction [3]. In the context of PRF, [25] used logistic regression (a linear classification method) to discriminate between relevant and non-relevant terms. However, to the best of our knowledge, multiple elastic net models have never been applied before to the PRF task.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we presented LiMe, a pseudo-relevance feedback model based on linear methods. LiMe models the PRF task as a particular matrix decomposition problem which involves the computation of inter-term similarities. We propose a solution of this decomposition based on linear least squares problems with ℓ_1 and ℓ_2 regularisation and non-negativity constraints. For that purpose, we use not only the information from the pseudo-relevant set but also the original query before expansion. The experimental evaluation showed that our proposal outperforms state-of-the-art baselines on five TREC datasets.

This work paves the way for further investigation on linear methods for pseudo-relevance feedback. The obtained results reveal



a: RFMF varying docs. b: RFMF varying terms. c: RFMF varying query interpolation.

d: MEDMM varying docs. e: MEDMM varying terms. f: MEDMM varying query interpolation.

g: RM3 varying docs. h: RM3 varying terms. i: RM3 varying query interpolation.

j: LiMe-TF-IDF varying docs. k: LiMe-TF-IDF varying terms. l: LiMe-TF-IDF varying query interpolation.

Figure 2: Sensitivity of RFMF, MEDMM, RM3 and LiMe-TF-IDF to k (the number of feedback documents), e (the number of expansion terms) and α (the interpolation parameter of the original query with the expansion terms) on the AP88-89, TREC-678, Robust04, WT10G and GOV2 collections. The rest of the parameters were fixed to their optimal values.

the potential of LiMe as a general PRF method usable on top of any retrieval model. LiMe is a flexible framework that allows the introduction of different document-term features. The good results achieved by LiMe using popular Information Retrieval features such as TF and TF-IDF indicate that there may be room for improvements. Thus, exploring alternative feature schemes seems to be a promising research direction.

We also envision to include a richer representation of text features into the model. For example, the use of features extracted from Wikipedia has proved to be beneficial in the PRF task [36].

Additionally, we plan to study how other similarity measures may be useful for PRF. In particular, we plan to study translation models because they usually rely on inter-term similarities [2, 12]. Previous work on translation models learnt inter-term similarities from training data [2] or employed mutual information [10].

ACKNOWLEDGMENTS

This work has received financial support from the i) "Ministerio de Economía y Competitividad" of the Government of Spain and the ERDF (project TIN2015-64282-R), ii) Xunta de Galicia – "Consellería de Cultura, Educación e Ordenación Universitaria" (project GPC ED431B 2016/035), and iii) Xunta de Galicia – "Consellería de Cultura, Educación e Ordenación Universitaria" and the ERDF ("Centro Singular de Investigación de Galicia" accreditation 2016-2019 ED431G/01). The first author also acknowledges the support of the "Ministerio de Educación, Cultura y Deporte" of the Government of Spain (grant FPU014/01724).

REFERENCES

[1] Nasreen Abdul-Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. In *TREC 2004*. 1–13.

[2] Adam Berger and John Lafferty. 1999. Information Retrieval as Statistical Translation. In *SIGIR '99*. ACM, 222–229. https://doi.org/10.1145/312624.312681

[3] David Carmel and Elad Yom-Tov. 2010. Estimating the Query Difficulty for Information Retrieval. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 2, 1 (2010), 1–89. https://doi.org/10.2200/S00235ED1V01Y201004ICR015

[4] Claudio Carpineto, Renato de Mori, Giovanni Romano, and Brigitte Bigi. 2001. An Information-Theoretic Approach to Automatic Query Expansion. *ACM Trans. Inf. Syst.* 19, 1 (2001), 1–27. https://doi.org/10.1145/366836.366860

[5] Claudio Carpineto and Giovanni Romano. 2012. A Survey of Automatic Query Expansion in Information Retrieval. *ACM Comput. Surv.* 44, 1 (2012), 1:1–1:50. https://doi.org/10.1145/2071389.2071390

[6] Kevyn Collins-Thompson and Jamie Callan. 2007. Estimation and use of uncertainty in pseudo-relevance feedback. In *SIGIR '07*. ACM, 303. https://doi.org/10.1145/1277741.1277795

[7] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of Recommender Algorithms on Top-N Recommendation Tasks. In *RecSys '10*. ACM, 39–46. https://doi.org/10.1145/1864708.1864721

[8] W. Bruce Croft and David J. Harper. 1979. Using Probabilistic Models of Document Retrieval Without Relevance Information. *J. Doc.* 35, 4 (1979), 285–295. https://doi.org/10.1108/eb026683

[9] Arthur E Hoerl and Robert W Kennard. 1970. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics* 12, 1 (1970), 55–67. https://doi.org/10.1080/00401706.1970.10488634

[10] Maryam Karimzadehgan and ChengXiang Zhai. 2010. Estimation of Statistical Translation Models Based on Mutual Information for Ad Hoc Information Retrieval. In *SIGIR '10*. ACM, 323. https://doi.org/10.1145/1835449.1835505

[11] Yehuda Koren and Robert Bell. 2015. Advances in Collaborative Filtering. In *Recommender Systems Handbook* (2nd ed.), Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer US, 77–118. https://doi.org/10.1007/978-1-4899-7637-6_3

[12] John Lafferty and Chengxiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01*. ACM, 111–119. https://doi.org/10.1145/383952.383970

[13] Victor Lavrenko and W. Bruce Croft. 2001. Relevance-Based Language Models. In *SIGIR '01*. ACM, 120–127. https://doi.org/10.1145/383952.383972

[14] Kyung Soon Lee, W Bruce Croft, and James Allan. 2008. A Cluster-based Resampling Method for Pseudo-relevance Feedback. In *SIGIR '08*. ACM, 235–242. https://doi.org/10.1145/1390334.1390376

[15] Yuanhua Lv and ChengXiang Zhai. 2009. A Comparative Study of Methods for Estimating Query Language Models with Pseudo Feedback. In *CIKM '09*. ACM, 1895–1898. https://doi.org/10.1145/1645953.1646259

[16] Yuanhua Lv and ChengXiang Zhai. 2014. Revisiting the Divergence Minimization Feedback Model. In *CIKM '14*. ACM, 1863–1866. https://doi.org/10.1145/2661829.2661900

[17] Craig Macdonald, Richard McCreadie, Rodrygo L. T. Santos, and Iadh Ounis. 2012. From Puppy to Maturity: Experiences in Developing Terrier. In *Proceedings of the SIGIR 2012 Workshop in Open Source Information Retrieval*. 60–63.

[18] M. E. Maron and J. L. Kuhns. 1960. On Relevance, Probabilistic Indexing and Information Retrieval. *J. ACM* 7, 3 (1960), 216–244. https://doi.org/10.1145/321033.321035

[19] Donald Metzler and W. Bruce Croft. 2007. Linear Feature-Based Models for Information Retrieval. *Inf. Retr.* 10, 3 (2007), 257–274. https://doi.org/10.1007/s10791-006-9019-z

[20] Xia Ning and George Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *ICDM '11*. IEEE Computer Society, 497–506. https://doi.org/10.1109/ICDM.2011.134

[21] Javier Parapar and Álvaro Barreiro. 2011. Promoting Divergent Terms in the Estimation of Relevance Models. In *ICTIR '11*. Springer-Verlag, 77–88.

[22] Javier Parapar, Alejandro Bellogín, Pablo Castells, and Álvaro Barreiro. 2013. Relevance-based Language Modelling for Recommender Systems. *Inf. Process. Manage.* 49, 4 (2013), 966–980. https://doi.org/10.1016/j.ipm.2013.03.001

[23] Javier Parapar, Manuel A. Presedo-Quindimil, and Álvaro Barreiro. 2014. Score Distributions for Pseudo Relevance Feedback. *Inf. Sci.* 273 (2014), 171–181. https://doi.org/10.1016/j.ins.2014.03.034

[24] Jay M. Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *SIGIR '98*. ACM, 275–281. https://doi.org/10.1145/290941.291008

[25] Karthik Raman, Raghavendra Udupa, Pushpak Bhattacharya, and Abhijit Bhole. 2010. On Improving Pseudo-Relevance Feedback Using Pseudo-Irrelevant Documents. In *ECIR '10*. Springer-Verlag, 573–576. https://doi.org/10.1007/978-3-642-12275-0_50

[26] Stephen E. Robertson. 1990. On Term Selection for Query Expansion. *J. Doc.* 46, 4 (1990), 359–364. https://doi.org/10.1108/eb026866

[27] Stephen E. Robertson and Karen Sparck Jones. 1976. Relevance Weighting of Search T erms. *J. Am. Soc. Inf. Sci.* 27, 3 (1976), 129–146.

[28] Joseph J. Rocchio. 1971. Relevance Feedback in Information Retrieval. In *The SMART Retrieval System - Experiments in Automatic Document Processing*, Gerard Salton (Ed.). Prentice Hall, 313–323.

[29] Ian Ruthven and Mounia Lalmas. 2003. A Survey on the Use of Relevance Feedback for Information Access Systems. *Knowl. Eng. Rev.* 18, 2 (2003), 95–145. https://doi.org/10.1017/S0269888903000638

[30] Tetsuya Sakai, Toshihiko Manabe, and Makoto Koyama. 2005. Flexible Pseudo-Relevance Feedback via Selective Sampling. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 4, 2 (2005), 111–135. https://doi.org/10.1145/1105696.1105699

[31] Gerard Salton. 1971. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc.

[32] Jangwon Seo and W. Bruce Croft. 2010. Geometric Representations for Multiple Documents. In *SIGIR '10*. ACM, 251–258. https://doi.org/10.1145/1835449.1835493

[33] Mark D Smucker, James Allan, and Ben Carterette. 2007. A Comparison of Statistical Significance Tests for Information Retrieval Evaluation. In *CIKM '07*. ACM, 623–632. https://doi.org/10.1145/1321440.1321528

[34] Robert Tibshirani. 1996. Regression Shrinkage and Selection via the Lasso. *J. R. Stat. Soc. Ser. B-Stat. Methodol.* 58, 1 (1996), 267–288.

[35] Daniel Valcarce, Javier Parapar, and Álvaro Barreiro. 2016. Efficient Pseudo-Relevance Feedback Methods for Collaborative Filtering Recommendation. In *ECIR '16*. Springer International Publishing, 602–613. https://doi.org/10.1007/978-3-319-30671-1_44

[36] Yang Xu, Gareth J.F. Jones, and Bin Wang. 2009. Query Dependent Pseudo-Relevance Feedback Based on Wikipedia. In *SIGIR '09*. ACM, 59. https://doi.org/10.1145/1571941.1571954

[37] Hamed Zamani, Javid Dadashkarimi, Azadeh Shakery, and W. Bruce Croft. 2016. Pseudo-Relevance Feedback Based on Matrix Factorization. In *CIKM '16*. ACM, 1483–1492. https://doi.org/10.1145/2983323.2983844

[38] ChengXiang Zhai and John Lafferty. 2001. Model-based Feedback in the Language Modeling Approach to Information Retrieval. In *CIKM '01*. ACM, 403. https://doi.org/10.1145/502585.502654

[39] ChengXiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.* 22, 2 (2004), 179–214. https://doi.org/10.1145/984321.984322

[40] Hui Zou and Trevor Hastie. 2005. Regularization and Variable Selection via the Elastic Net. *J. R. Stat. Soc. Ser. B-Stat. Methodol.* 67, 2 (2005), 301–320.