Document-based and Term-based Linear Methods for Pseudo-Relevance Feedback

Daniel Valcarce, Javier Parapar, Álvaro Barreiro
Information Retrieval Lab
Department of Computer Science
University of A Coruña, Spain
{daniel.valcarce, javierparapar, barreiro}@udc.es

ABSTRACT

Query expansion is a successful approach for improving Information Retrieval effectiveness. This work focuses on pseudo-relevance feedback (PRF) which provides an automatic method for expanding queries without explicit user feedback. These techniques perform an initial retrieval with the original query and select expansion terms from the top retrieved documents. We propose two linear methods for pseudo-relevance feedback, one document-based and another term-based, that models the PRF task as a matrix decomposition problem. These factorizations involve the computation of an inter-document or inter-term similarity matrix which is used for expanding the original query. These decompositions can be computed by solving a least squares regression problem with regularization and a non-negativity constraint. We evaluate our proposals on five collections against state-of-the-art baselines. We found that the term-based formulation provides high figures of MAP, nDCG and robustness index whereas the document-based formulation provides very cheap computation at the cost of a slight decrease in effectiveness.

CCS Concepts

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

Keywords

Information retrieval; linear methods; pseudo-relevance feedback; query expansion; linear least squares

1. INTRODUCTION

Two natural ways of approaching the enhancing of retrieval effectiveness are by improving the retrieval model or by modifying the query prompted by the user. In this paper, we focus on the latter: how to alter the original query to obtain a better rank. Query expansion techniques aim to add new terms to the query. This expanded query is expected to provide better retrieval results than the initial one. Relevance feedback is one of the most reliable types of query expansion methods, but it requires users to indicate which documents from those retrieved with the original query are relevant [29]. An alternative method for expanding the queries which does not need interaction from the user is pseudo-relevance feedback (PRF). This approach is based on the assumption that the top documents retrieved are relevant. From these pseudo-relevant documents (which form the so-called pseudo-relevant set), PRF techniques extract terms (with their corresponding weights) to expand the original query. This assumption is not too strong if the retrieval model provides decent results. In fact, research has shown that PRF is one the most effective techniques to improve the retrieval quality [28, 27, 8, 26, 4, 13, 6, 14, 15, 21, 16, 23, 41].

The language modeling framework is a fertile area of research for PRF techniques [15, 32, 16]. However, in this article, we propose a novel framework for the PRF task which is not based on language models, but in linear methods, which we call LiMe. In particular, we propose two modelings of the PRF task as matrix decomposition problems called DLiMe (Document-based Linear Methods) and TLiMe (Term-based Linear Methods). LiMe framework and the TLiMe model were first presented in our previous article [39]. In this work, we extend the LiMe framework by proposing DLiMe.

RFMF was the first formulation of PRF as a matrix decomposition problem [41] and computes a latent factor representation of documents/queries and terms using non-negative matrix factorization. In contrast, in this manuscript, we propose a different decomposition that stems from the computation of inter-document or 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 inter-document or inter-term similarities. Our matrix formulations enable to compute these similarities that yield within the query and the pseudo-relevant set. We use the information of these relationships between documents or terms to expand the original query.

Since producing a good rank of expansion terms is critical for a successful PRF technique, the modeling of inter-term similarities seems to be a desirable property. Additionally, computing good weights for those expansion terms is a critical factor in the performance of a PRF technique. We also think that modeling the relationship between pseudo-relevant documents can be a faster way to produce expansion terms because the number of documents is much smaller than the number of terms in the pseudo-relevant set. In fact, our experiments show that the computation of inter-term similarities produces high-quality rankings of expansion terms and weights. In contrast, our proposal based on inter-document similarities is computationally very cheap at the expense of slightly worse expansion terms.

As [41] 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 and previous paper, we explore well-known and straightforward weighting functions which allow us to outperform state-of-the-art techniques.

LiMe modeling of the PRF task paves the way for developing multiple PRF algorithms since the proposed formulations of the matrix decompositions can be calculated in various ways. In this paper, we use a method based on regularized linear least squares regression. On the one hand, we employ a ℓ_2 regularization scheme to avoid overfitting. On the other hand, we use ℓ_1 regularization to enforce sparsity into the learned inter-document or inter-term similarities. This method provides an automatic feature selection which gives us a more compact solution with the corresponding efficiency gains. The combination of ℓ_1 and ℓ_2 regularization for linear least squares problems is also known as an elastic net regression in Statistics [44]. Additionally, we add non-negativity constraints to force the computed similarities to be positive to increase the interpretability of the models.

We thoroughly evaluate DLiMe and TLiMe on five TREC collections. The obtained results show that TLiMe outperforms state-of-the-art baselines regarding several common effectiveness metrics. Moreover, TLiMe achieved high values of robustness compared to the baselines. These findings highlight the applicability of TLiMe as a pseudo-relevance feedback technique. In contrast, DLiMe provides a computationally cheaper alternative with a slight decrease in effectiveness. It is important to note that LiMe framework can exploit different features allowing the exploration of further features schemes.

In summary, the contributions of this paper are DLiMe and TLiMe, two novel matrix decomposition formulations of the PRF task involving inter-document and inter-term similarities and an algorithm based on constrained elastic net regression for solving the proposed matrix decompositions and computing the expansion terms. The empirical evaluation of the effectiveness of the proposed methods against state-of-the-art baselines shows that DLiMe and TLiMe are competitive PRF techniques.

2. BACKGROUND

In this section, we first describe pseudo-relevance feedback (PRF). Then, we focus on state-of-the-art PRF techniques based on the language modeling framework [24] because they perform notably well in practice [13, 15, 16, 41]. Afterward, we introduce previous work on PRF using matrix factorization [41]. Finally, we introduce linear methods for regression problems since our proposal rests on these models.

2.1 Pseudo-Relevance Feedback (PRF)

Query expansion methods aim to add new terms to the original query. 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 beneficial 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 [8, 26, 4, 42, 13, 6, 14, 15, 21, 23, 16, 41]. 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 has 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 from 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 modeling 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]:

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

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 modeling framework [15].

2.2.1 Relevance-Based Language Models

Relevance-based language models or, for short, 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 initially 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, 35, 36, 37].

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:

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

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 [42, 13]. RM1 is typically called RM3 when it is interpolated with the original query (see Eq. 2) [1].

2.2.2 MEDMM

The Maximum-Entropy Divergence Minimization Model (also known as MEDMM) [16] is a PRF technique based on the Divergence Minimization Model (DMM) [42] which stems from the language modeling 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 minimizes the distance to the language models of the documents of the pseudo-relevant set and, at the same time, maximizes 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:

$$\theta_F = \arg\min_{\theta} \sum_{D \in F} \alpha_D \, H(\theta, \theta_D) - \lambda \, H(\theta_F, \theta_C) - \beta \, H(\theta) \quad (4)$$

where $H(\cdot, \cdot)$ denotes the cross entropy and $H(\cdot)$ denotes the entropy.

MEDMM also gives a weight α_D for each document based on the posterior of the document language model:

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

The analytic solution to MEDMM, obtained with Lagrange multipliers, is given by [16]:

$$p(t|\theta_F) \propto \exp\left(\frac{1}{\beta} \sum_{D \in F} \alpha_D \log p(t|\theta_D) - \frac{\lambda}{\beta} \log p(t|\theta_C)\right) \quad (6)$$

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. The feedback model computed by MEDMM is also interpolated with the original query as in Eq. 2.

2.3 PRF based on Matrix Factorization

Other authors have focused on developing PRF models based on different ideas. In particular, RFMF was the first technique that applied matrix factorization to the PRF task [41]. 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 modeling framework. RFMF reconstructs, through non-negative matrix factorization (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 factorization techniques are commonplace [11]. RFMF finds the 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 minimize the error $\vec{\epsilon}$:

$$\vec{y} = X\vec{w} + \vec{\epsilon} = w_1\vec{x_1} + \cdots + w_n\vec{x_n} + \vec{\epsilon} \quad (7)$$

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:

$$\vec{w}^* = \arg\min_{\vec{w}} \|\vec{\epsilon}\|_2^2 = \arg\min_{\vec{w}} \|\vec{y} - X\vec{w}\|_2^2 \quad (8)$$

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 regularization (this model is also known as ridge regression in Statistics [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:

$$\vec{w}^* = \arg\min_{\vec{w}} \|\vec{y} - X\vec{w}\|_2^2 + \beta_2 \|\vec{w}\|_2^2 \tag{9}$$

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 in Statistics [34]. This approach performs automatic feature selection as the value of the non-negative parameter β_1 grows:

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

Since both, ridge and lasso regressions, have beneficial properties, Zou and Hastie [44] developed a technique combining both ℓ_1 and ℓ_2 regularization: the elastic net, which is a generalization 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:

$$\vec{w}^* = \arg\min_{\vec{w}} \|\vec{y} - X\vec{w}\|_2^2 + \beta_1 \|\vec{w}\|_1 + \beta_2 \|\vec{w}\|_2^2 \tag{11}$$

3. LIME: LINEAR METHODS FOR PRF

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 $n = |V_{F'}|$.

3.1 LiMe Framework

We can define LiMe using matrix or vector formulation. To understand better the idea behind LiMe, we initially present our technique under a matrix formulation. Afterward, we introduce the vector representation which is much more convenient for 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 to 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 factorize this matrix X into the product of itself and another matrix. In the case of TLiMe, we build an inter-term matrix $W = (w_{ij}) \in \mathbb{R}_+^{n \times n}$ whereas in the case of DLiMe, we build an inter-document matrix $Z = (z_{ij}) \in \mathbb{R}_+^{m \times m}$.

3.1.1 TLiMe Formulation

The matrix W represents the inter-term similarity between pairs of words in $V_{F'}$. In particular, each entry w_{ij} symbolizes the similarity between terms t_i and t_j. To increase the interpretability of the model, 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 TLiMe as an algorithm that computes the following decomposition:

$$\begin{gathered} X \approx X\,W \\ s.t.\, \operatorname{diag}(W) = 0,\ W \geq 0 \end{gathered} \tag{12}$$

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

$$\begin{aligned} W^* = \arg\min_{W} \quad & \frac{1}{2} \|X - X\,W\|_F^2 + \beta_1 \|W\|_{1,1} + \frac{\beta_2}{2} \|W\|_F^2 \\ \text{s.t.} \quad & \operatorname{diag}(W) = 0,\ W \geq 0 \end{aligned} \tag{13}$$

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 optimization problem in the following vector form:

$$\begin{aligned} \vec{w}^*_{\cdot j} = \arg\min_{\vec{w}_{\cdot j}} \quad & \frac{1}{2} \|\vec{x}_{\cdot j} - X\vec{w}_{\cdot j}\|_2^2 + \beta_1 \|\vec{w}_{\cdot j}\|_1 + \frac{\beta_2}{2} \|\vec{w}_{\cdot j}\|_2^2 \\ \text{s.t.} \quad & w_{jj} = 0,\ \vec{w}_{\cdot j} \geq 0 \end{aligned} \tag{14}$$

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. For each term j in $V_{F'}$, we train an elastic net [44] with an equality constraint to zero in one coefficient and non-negativity constraints on the rest of the coefficients.

We merge the solutions of the regression problems depicted in Eq. 14 to build the inter-term similarity matrix W^*. We use the computed matrix decomposition to reconstruct the first row of X (which we will denote by $\hat{x}_{1\cdot}$) as follows:

$$\hat{x}_{1\cdot} = \vec{x}_{1\cdot} W^* \tag{15}$$

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 regularization. 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.

3.1.2 DLiMe Formulation

The document-based linear method for PRF (DLiMe) is based on the computation of the matrix $Z = (z_{ij}) \in \mathbb{R}_{+}^{m\times m}$. This matrix represents the inter-document similarity between pairs of elements from the extended pseudo-relevant set F' (i.e., the query and the pseudo-relevant documents). The matrix formulation of DLiMe is analogous to TLiMe:

$$\begin{aligned} X &\approx Z\,X \\ s.t.\, \mathrm{diag}(Z) &= 0,\; Z \geq 0 \end{aligned} \tag{16}$$

We also constrain Z to be non-negative to foster interpretability and enforce the diagonal to be zero to avoid the trivial solution. Since we are only interested in reconstructing the first row of X, we only need to compute the first row of Z. Therefore, DLiMe factorization can be reduced to a single constrained linear least squares optimization problem as follows:

$$\begin{aligned} \vec{z}^{*}_{1\cdot} \;=\; & \underset{\vec{z}_{1\cdot}}{\arg\min} \;\; \frac{1}{2}\left\|\vec{z}_{1\cdot} - \vec{z}_{1\cdot}X\right\|_2^2 + \beta_1\left\|\vec{z}_{1\cdot}\right\|_1 + \frac{\beta_2}{2}\left\|\vec{z}_{1\cdot}\right\|_2^2 \\ & \text{s.t.} \qquad z_{11} = 0,\; \vec{z}_{1i} \geq 0 \end{aligned} \tag{17}$$

Note that compared to TLiMe, where n least squares problem have to be solved, DLiMe is much more efficient because it only involves solving one least squares problem. To reconstruct the first row of X we simply need to perform the following vector-matrix multiplication:

$$\hat{x}_{1\cdot} = \vec{z}^{*}_{1\cdot}X \tag{18}$$

3.2 LiMe Feedback Model

LiMe feedback model is created from $\hat{x}_{1\cdot}$, which can be reconstructed using either DLiMe or TLiMe. We can normalize this vector to obtain a probability estimate. In this way, the probability of the j-th term given the feedback model is given by:

$$p(t_j|\theta_F) = \begin{cases} \dfrac{\hat{x}_{1j}}{\sum_{t_v \in V_{F'}} \hat{x}_{1v}} & \text{if } t_j \in V_{F'}, \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

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, 41]. 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 optimization algorithms. Additionally, LiMe employs elastic net regularization. In contrast, RFMF is based on non-negative factorization which can deal with non-negative and sparse data while LiMe deals with this data by enforcing non-negativity constraints in the optimization problem. Additionally, LiMe discovers inter-document (DLiMe) or inter-term similarities (TLiMe) that yield within the pseudo-relevant set and the query while RFMF learns document and term latent factor representations.

Next, we discuss how we fill matrix $X = (x_{ij})$ with features relating query/documents i with terms j.

3.3 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 decided 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:

$$x_{ij} = \begin{cases} s(w_j, Q) & \text{if } i = 1 \text{ and } f(w_j, Q) > 0, \\ s(w_j, D_{i-1}) & \text{if } i > 1 \text{ and } f(w_j, D_{i-1}) > 0, \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

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

$$s_{tf\text{-}idf}(w, D) = (1 + \log_2 f(w, D)) \times \log_2 \frac{|\mathcal{C}|}{df(w)} \tag{21}$$

where $|\mathcal{C}|$ is the number of documents in the collection and $df(w)$ represents the document frequency of term w (i.e., the number of documents in the collection where the term w occurs).

In any case, other alternatives may be possible. In fact, in previous work, we also reported the performance for the logarithmic TF heuristic [39]. 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 for performing well with simple and well-known features such as TF-IDF is remarkable. Also, this weighting function is supported by decades of research in Information Retrieval.

3.4 Implementation Details

Equation 14 shows that 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. In contrast, DLiMe only requires to solve one least squares problem (Eq. 17). To solve these regression problems, we used the highly efficient BCLS[1] (Bound-Constrained Least

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

Squares) library, which implements a two-metric projected-descent method for solving bound-constrained least squares problems.

An additional optimization for TLiMe 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. 15). Therefore, we only need those rows that correspond to a term in the original query. If we only store those similarities, we save much space since the number of terms in a query prompted by a user is tiny compared to the number of rows.

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 explain the choice of baselines and the parameter setting. Finally, we present and analyze the results comparing the behavior of LiMe concerning the baselines.

4.1 Evaluation Methodology

We conducted the experiments on diverse TREC collections commonly used in PRF literature [15, 16, 41]: 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 found the model hyperparameters that maximize 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 (normalized 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 randomizations and $p < 0.05$ to measure if the improvements regarding MAP and nDCG were statistically significant [33]. We cannot apply a paired statistic to RI because it is a global metric.

We used title queries from TREC topics. We preprocessed the collections with the standard Terrier stopwords removal and Porter stemmer since previous work recommended the use of stemming and stopwords removal [15].

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

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

4.2 Baselines and Parameter Setting

We employed the state-of-the-art language modeling framework for performing the first and second stage retrievals [24]. In particular, we used the 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 [43] with parameter $\mu = 1000$. To compare the effectiveness of our proposals, we employed the following state-of-the-art baselines:

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

RFMF We included this PRF technique because it is based on the non-negative factorization of a document-term matrix obtained from the query and the pseudo-relevant set [41]. We set the number of dimensions of the factorization, d, to the size of the relevant set plus one as the authors recommended [41]. We used the TF-IDF weighting function.

MEDMM We also employed the maximum-entropy divergence minimization model which is recognized 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 modeling 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 [13, 1]. We set the Dirichlet priors smoothing parameter μ' to 1000 as it is typically done [15, 16, 41].

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 selected those parameters that maximize the values of MAP in the training set.

4.3 Results and Discussion

The results of the experiments regarding MAP, nDCG, and RI are summarized in Table 2. Overall, all the PRF techniques outperform the language modeling baseline without query expansion. However, TLiMe is the only method that offered significant improvements over LM in MAP and nDCG on all collections. DLiMe showed competitive effectiveness concerning MEDMM and RM3.

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

On all datasets, TLiMe achieved the highest results regarding MAP and nDCG. No baseline outperformed TLiMe on any dataset. TLiMe significantly surpassed RFMF on four out of five datasets regarding MAP and nDCG. Regarding RM3, TLiMe significantly outperformed RM3 on three collections (concerning MAP or nDCG). The strongest baseline, MEDMM, was only significantly surpassed by TLiMe on the AP88-89 collection. However, on all datasets, TLiMe showed higher values in nDCG and MAP than MEDMM. Although no baseline significantly improved TLiMe, MEDMM significantly surpassed RM3 and DLiMe regarding nDCG and MAP on the TREC-678 collection. Also, DLiMe, RM3, and MEDMM significantly improved RFMF in terms of MAP and nDCG on several datasets.

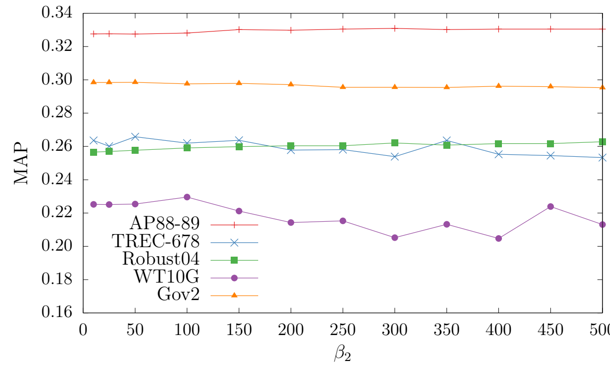
It is interesting to remark that the PRF techniques achieved the smallest improvements in the WT10G collection. This small improvement is probably due to the nature of the web which is a noisy media. Also, the values of RI on this dataset are the lowest.

Regarding the differences between DLiMe and TLiMe, the latter approach showed better figures of MAP and nDCG on all datasets. Nevertheless, the differences are significant only on the TREC-678 collections. In contrast, DLiMe provided higher RI than TLiMe on GOV2 and the same figure on AP88-89 collections

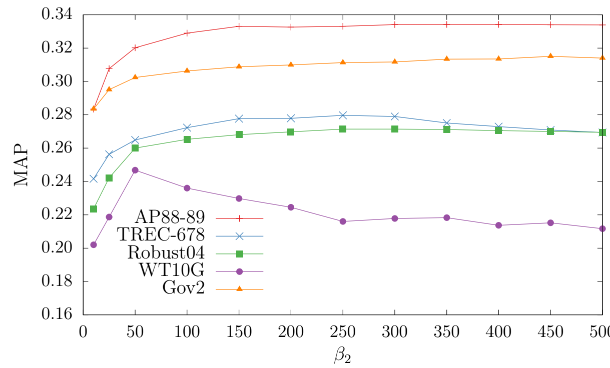
4.3.1 Query Analysis

To provide insights into the good results achieved by DLiMe and TLiMe, we manually studied the expanded queries produced by the tested PRF methods. Table 3 shows 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". Those terms seem 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 DLiMe provided good expansion terms. Still, this approach included the term "hey" which we think is uninformative. In this case, TLiMe yielded the best expansion terms. All of them are specific and related to the



a) DLiMe



b) TLiMe

Figure 1: Sensitivity of DLiMe and TLiMe techniques to β_2 on each collection. The rest of the parameters were fixed to their optimal values.

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 factorization approaches. LiMe was capable of selecting very specific and relevant terms such as "smithsonian" or "chumash". 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 based on inter-term or inter-doc similarities can select relevant terms while RFMF factorization 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 DLiMe and TLiMe 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 regularization into LiMe models is still beneficial since it provides sparsity to the learned matrix W with the corresponding space sav-

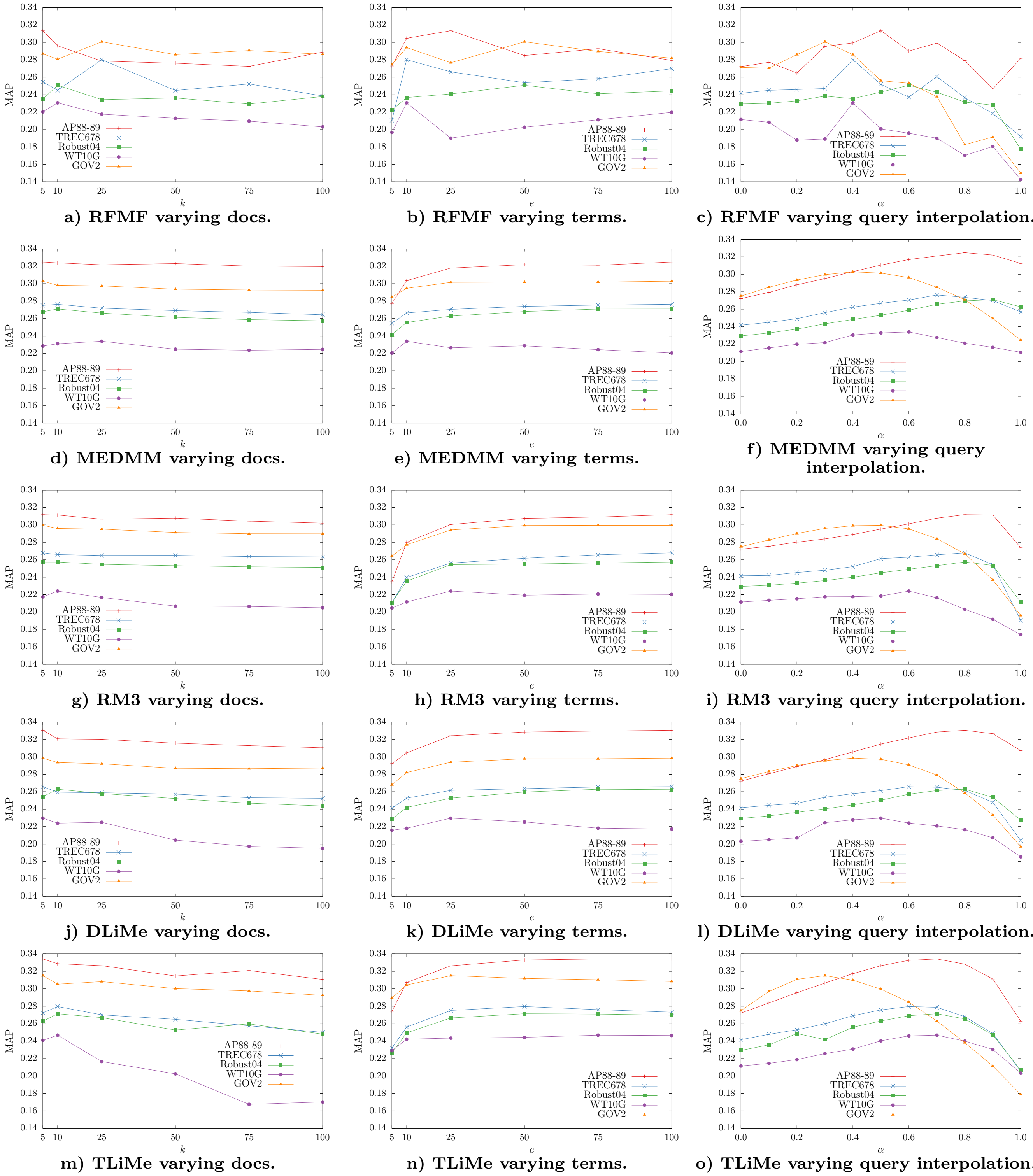


Figure 2: Sensitivity of RFMF, MEDMM, RM3, DLiMe and TLiMe to k (the number of feedback documents), e (the number of expansion terms) and α (the interpolation parameter of the original query with the expansion terms) on each collection. The rest of the parameters were fixed to their optimal values.

Table 2: Values of MAP, P@5, nDCG and RI for LM, RFMF, MEDMM, RM3, DLiMe and TLiMe techniques on each collection. Statistically significant improvements according to permutation test (p<0.05) w.r.t. to LM, RFMF, MEDMM, RM3, DLiMe and TLiMe are superscripted with a, b, c, d, e and f, respectively.

Collection	Metric	LM	RFMF	MEDMM	RM3	DLiMe	TLiMe
AP88-89	MAP	0.2349	0.2774^{a}	0.3010^{ab}	0.3002^{ab}	0.3112^{ab}	$\mathbf{0.3149}^{abcd}$
	nDCG	0.5637	0.5749^{a}	0.5955^{ab}	0.6005^{ab}	0.6058^{ab}	$\mathbf{0.6085}^{ab}$
	RI	–	0.42	0.42	0.50	**0.52**	**0.52**
TREC-678	MAP	0.1931	0.2072	0.2327^{abde}	0.2235^{ab}	0.2206^{ab}	$\mathbf{0.2357}^{abde}$
	nDCG	0.4518	0.4746	0.5115^{abde}	0.4987^{ab}	0.4936^{ab}	$\mathbf{0.5198}^{abde}$
	RI	–	0.23	0.26	0.40	0.44	0.46
Robust04	MAP	0.2914	0.3130^{a}	0.3447^{ab}	0.3488^{ab}	0.3435^{ab}	$\mathbf{0.3517}^{ab}$
	nDCG	0.5830	0.5884	0.6227^{ab}	0.6251^{ab}	0.6247^{ab}	$\mathbf{0.6294}^{ab}$
	RI	–	0.07	0.32	**0.37**	0.32	**0.37**
WT10G	MAP	0.2194	0.2389^{a}	0.2472^{a}	0.2470^{a}	0.2368^{a}	$\mathbf{0.2476}^{a}$
	nDCG	0.5212	0.5262	0.5324	0.5352	0.5290	$\mathbf{0.5398}^{a}$
	RI	–	0.30	**0.36**	0.20	0.26	0.30
GOV2	MAP	0.3310	0.3580^{a}	0.3790^{ab}	0.3755^{ab}	0.3731^{ab}	$\mathbf{0.3830}^{ab}$
	nDCG	0.6325	0.6453	0.6653^{ab}	0.6618^{ab}	0.6588^{ab}	$\mathbf{0.6698}^{abd}$
	RI	–	0.42	0.66	0.60	**0.72**	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	b) MEDMM term	weight	c) RM3 term	weight	d) DLiMe term	weight	e) TLiMe term	weight
indian	0.1725	indian	0.1511	indian	0.1285	indian	0.1392	indian	0.1392
museum	0.1685	museum	0.0802	american	0.0895	museum	0.1365	museum	0.1364
american	0.1505	american	0.0780	museum	0.0874	american	0.1257	american	0.1256
professor	0.0193	cultur	0.0210	year	0.0219	smithsonian	0.0394	tribe	0.0393
tribal	0.0160	year	0.0177	will	0.0209	artifact	0.0307	artifact	0.0306
ancient	0.0155	live	0.0153	west	0.0182	hey	0.0272	cultur	0.0272
dolphin	0.0153	nation	0.0148	1	0.0167	tribal	0.0271	tribal	0.0271
rafaela	0.0140	artifact	0.0146	tribal	0.0158	cultur	0.0250	nation	0.0249
activist	0.0137	part	0.0139	time	0.0149	chumash	0.0219	chumash	0.0219
racist	0.0137	tribal	0.0127	new	0.0147	tribe	0.0213	smithsonian	0.0212

ings. Regarding β_2, we plotted the values of MAP achieved by DLiMe and TLiMe with different amount of ℓ_2 regularization in Fig. 1. Except for the WT10G collection, the parameter β_2 is relatively stable among the values 150 and 400 for both DLiMe and TLiMe.

We also studied how DLiMe and TLiMe behave 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 summarizes the results of the sensitivity analysis regarding 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 never higher than 25. LiMe methods and RM3 are quite stable, and they behave optimally with 5-10 documents. In contrast, RFMF and MEDMM may require up to 25 documents in the pseudo-relevant set depending on the dataset.

The optimal number of expansion terms is 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, DLiMe and TLiMe are situated between the two extremes.

Regarding the interpolation parameter α, except for the GOV2 collection, we observed that the optimal values for DLiMe and TLiMe lie 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 adequately tune it. Overall, the performance of RFMF is very unstable when we vary α (to a lesser extent, 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 DLiMe, TLiMe, and MEDMM generate better feedback models to use in isolation.

5. RELATED WORK

Pseudo-relevance feedback (PRF) is a fertile area of research in Information Retrieval [28, 27, 8, 26, 4, 13, 6, 14, 15, 32, 21, 23, 16, 41]. Among the PRF techniques, those based on the language modeling framework have showed great effectiveness [15]. Therefore, we used them as baselines and described them in Section 2. Additionally, we included RFMF as a baseline because it was the first work that modeled the PRF task as a matrix factorization problem [41].

PRF methods have been adapted to collaborative filtering recommendation with great success [22]. In particular, relevance-based language models [22, 36, 37, 38] and the Rocchio framework [35]. Conversely, RFMF is a case of a recommendation technique applied to PRF [41].

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 regularization. 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 optimized 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 factorization, they compute different decompositions. The differences in performance are explained by the use of different objective functions and optimization algorithms. LiMe minimizes the elastic net loss and RFMF minimizes the KL-divergence of the NMF 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 maximize 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 framework where the PRF task is modeled as a matrix decomposition problem which involves the computation of inter-term similarities. In previous work, we proposed TLiMe, a technique based on inter-term similarities. In this extended version, we also present DLiMe which is based on an inter-document matrix. TLiMe and DLiMe factorizations are solved as linear least squares problems with ℓ_1 and ℓ_2 regularization 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 TLiMe outperforms state-of-the-art baselines on five TREC datasets whereas DLiMe shows competitive effectiveness with a reduced computational cost.

This work paves the way for further investigation on linear methods for pseudo-relevance feedback. The obtained results reveal 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 DLiMe and TLiME using only TF-IDF indicate that there may be room for improvements. Therefore, 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 [40]. 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 learned inter-term similarities from training data [2] or employed mutual information [10].

7. ACKNOWLEDGMENTS

This work was supported by project TIN2015-64282-R (MINECO/ERDF), project GPC ED431B 2016/035 (Xunta de Galicia) and accreditation 2016-2019 ED431G/01 (Xunta de Galicia/ERDF). The first author also acknowledges the support of grant FPU014/01724 (MECD).

8. REFERENCES

[1] N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. UMass at TREC 2004: Novelty and HARD. In *TREC 2004*, pages 1–13, 2004.

[2] A. Berger and J. Lafferty. Information Retrieval as Statistical Translation. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 222–229, New York, NY, USA, 1999. ACM.

[3] D. Carmel and E. Yom-Tov. Estimating the Query Difficulty for Information Retrieval. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 2(1):1–89, 2010.

[4] C. Carpineto, R. de Mori, G. Romano, and B. Bigi. An Information-Theoretic Approach to Automatic Query Expansion. *ACM Transactions on Information Systems*, 19(1):1–27, 2001.

[5] C. Carpineto and G. Romano. A Survey of Automatic Query Expansion in Information Retrieval. *ACM Computing Surveys*, 44(1):1:1–1:50, 2012.

[6] K. Collins-Thompson and J. Callan. Estimation and use of uncertainty in pseudo-relevance feedback. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 303, New York, NY, USA, 2007. ACM.

[7] P. Cremonesi, Y. Koren, and R. Turrin. Performance of Recommender Algorithms on Top-N Recommendation Tasks. In *Proceedings of the 4th ACM Conference on Recommender Systems*, RecSys '10, pages 39–46, New York, NY, USA, 2010. ACM.

[8] W. B. Croft and D. J. Harper. Using Probabilistic Models of Document Retrieval Without Relevance Information. *Journal of Documentation*, 35(4):285–295, 1979.

[9] A. E. Hoerl and R. W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67, 1970.

[10] M. Karimzadehgan and C. Zhai. Estimation of Statistical Translation Models Based on Mutual Information for Ad Hoc Information Retrieval. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, page 323, New York, NY, USA, 2010. ACM.

[11] Y. Koren and R. Bell. Advances in Collaborative Filtering. In F. Ricci, L. Rokach, and B. Shapira, editors, *Recommender Systems Handbook*, pages 77–118. Springer US, 2nd edition, 2015.

[12] J. Lafferty and C. Zhai. 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, pages 111–119, New York, NY, USA, 2001. ACM.

[13] V. Lavrenko and W. B. Croft. Relevance-Based Language Models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 120–127, New York, NY, USA, 2001. ACM.

[14] K. S. Lee, W. B. Croft, and J. Allan. A Cluster-based Resampling Method for Pseudo-relevance Feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 235–242, New York, NY, USA, 2008. ACM.

[15] Y. Lv and C. Zhai. A Comparative Study of Methods for Estimating Query Language Models with Pseudo Feedback. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 1895–1898, New York, NY, USA, 2009. ACM.

[16] Y. Lv and C. Zhai. Revisiting the Divergence Minimization Feedback Model. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 1863–1866, New York, NY, USA, 2014. ACM.

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

[18] M. E. Maron and J. L. Kuhns. On Relevance, Probabilistic Indexing and Information Retrieval. *Journal of the ACM*, 7(3):216–244, 1960.

[19] D. Metzler and W. B. Croft. Linear Feature-Based Models for Information Retrieval. *Information Retrieval*, 10(3):257–274, 2007.

[20] X. Ning and G. Karypis. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, pages 497–506, Washington, DC, USA, 2011. IEEE Computer Society.

[21] J. Parapar and Á. Barreiro. Promoting Divergent Terms in the Estimation of Relevance Models. In *Proceedings of the 3rd Cnternational Conference on Advances in Information Retrieval Theory*, ICTIR '11, pages 77–88. Springer-Verlag, sep 2011.

[22] J. Parapar, A. Bellogín, P. Castells, and Á. Barreiro. Relevance-based language modelling for recommender systems. *Information Processing and Management*, 49(4):966–980, 2013.

[23] J. Parapar, M. A. Presedo-Quindimil, and Á. Barreiro. Score Distributions for Pseudo Relevance Feedback. *Information Sciences*, 273:171–181, 2014.

[24] J. M. Ponte and W. B. Croft. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM.

[25] K. Raman, R. Udupa, P. Bhattacharya, and A. Bhole. On Improving Pseudo-Relevance Feedback Using Pseudo-Irrelevant Documents. In *ECIR '10*, pages 573–576. Springer-Verlag, 2010.

[26] S. E. Robertson. On Term Selection for Query Expansion. *Journal of Documentation*, 46(4):359–364, 1990.

[27] S. E. Robertson, K. Spärck Jones, and K. S. Jones. Relevance Weighting of Search Terms. *Journal of the American Society for Information Science*, 27(3):129–146, may 1976.

[28] J. J. Rocchio. Relevance Feedback in Information Retrieval. In G. Salton, editor, *The SMART Retrieval System - Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971.

[29] I. Ruthven and M. Lalmas. A Survey on the Use of Relevance Feedback for Information Access Systems. *The Knowledge Engineering Review*, 18(2):95–145, 2003.

[30] T. Sakai, T. Manabe, and M. Koyama. Flexible Pseudo-Relevance Feedback via Selective Sampling. *ACM Transactions on Asian Language Information Processing*, 4(2):111–135, 2005.

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

[32] J. Seo and W. B. Croft. Geometric Representations for Multiple Documents. In *Proceedings of the 33rd*

International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10, page 251, New York, NY, USA, 2010. ACM.

[33] M. D. Smucker, J. Allan, and B. Carterette. A Comparison of Statistical Significance Tests for Information Retrieval Evaluation. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, page 623, New York, NY, USA, 2007. ACM.

[34] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[35] D. Valcarce, J. Parapar, and Á. Barreiro. Efficient Pseudo-Relevance Feedback Methods for Collaborative Filtering Recommendation. In *Proceedings of the 38th European Conference on Information Retrieval*, ECIR '16, pages 602–613, Berlin, Heidelberg, 2016. Springer International Publishing.

[36] D. Valcarce, J. Parapar, and Á. Barreiro. Item-based Relevance Modelling of Recommendations for Getting Rid of Long Tail Products. *Knowledge-Based Systems*, 103:41–51, 2016.

[37] D. Valcarce, J. Parapar, and Á. Barreiro. Language Models for Collaborative Filtering Neighbourhoods. In *Proceedings of the 38th European Conference on Information Retrieval*, ECIR '16, pages 614–625, Berlin, Heidelberg, 2016. Springer.

[38] D. Valcarce, J. Parapar, and Á. Barreiro. Axiomatic Analysis of Language Modelling of Recommender Systems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 25(Suppl. 2):113–127, 2017.

[39] D. Valcarce, J. Parapar, and Á. Barreiro. LiMe: Linear Methods for Pseudo-Relevance Feedback. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, SAC '18, pages 678–687, New York, NY, USA, 2018. ACM.

[40] Y. Xu, G. J. Jones, and B. Wang. 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, page 59, New York, NY, USA, 2009. ACM.

[41] H. Zamani, J. Dadashkarimi, A. Shakery, and W. B. Croft. Pseudo-Relevance Feedback Based on Matrix Factorization. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 1483–1492, New York, NY, USA, 2016. ACM.

[42] C. Zhai and J. Lafferty. Model-based Feedback in the Language Modeling Approach to Information Retrieval. In *Proceedings of the 10th International Conference on Information and Knowledge Management*, CIKM '01, page 403, New York, NY, USA, 2001. ACM.

[43] C. Zhai and J. Lafferty. A Study of Smoothing Methods for Language Models Applied to Information Retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, 2004.

[44] H. Zou and T. Hastie. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

ABOUT THE AUTHORS:

Daniel Valcarce is a Ph.D. student at the University of A Coruña (Spain). He obtained his B.Sc. and M.Sc. degrees in Computer Science from the same university. His research interests cover recommender systems, information retrieval and machine learning. He is particularly interested in adapting information retrieval models to solve recommendation problems. He has been reviewer for multiple conferences such as ECIR, RecSys, CIKM and SIGIR. For more information, go to https://www.dc.fi.udc.es/~dvalcarce.

Javier Parapar is an Assistant Professor at the University of A Coruña (Spain). He was President of the Spanish Society for IR from 2014 to 2018. Javier Parapar holds a B.Sc.+M.Sc. in Computer Science and he got his Ph.D. in Computer Science (cum laude) in 2013, both from the University of A Coruña (Spain). His current research interests include but are not limited to information retrieval evaluation, recommender systems, text mining, and summarization. He regularly serves as reviewer and PC member of conferences such as ACM RecSys, The Web Conference (WWW), ACM SIGIR, ECIR, etc. He is a member of IP&M editorial board and a regular reviewer for journals such as ACM TOIS, IRJ, DKE and TKDE. For more information, go to https://www.dc.fi.udc.es/~parapar.

Álvaro Barreiro is a Professor in Computer Science at the University of A Coruña (Spain) and the group leader of the Information Retrieval Lab. He has supervised five doctoral theses and five research projects of the National R&D program, as well as other regional projects and R&D projects with companies. His research interests include information retrieval models, efficiency in information retrieval systems, text and data analysis and classification, evaluation and recommender systems. He has been acknowledged as ACM Senior Member and he is a member of ACM and ACM-SIGIR, BCS and BCS-IRSG, AEPIA and SERI societies. For more information, go to https://www.dc.fi.udc.es/~barreiro.