Query Expansion as a Matrix Factorization Problem

Extended Abstract

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

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

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

ABSTRACT

Pseudo-relevance feedback (PRF) provides an automatic method for query expansion in Information Retrieval. These techniques find relevant expansion terms using the top retrieved documents with the original query. In this paper, we present an approach based on linear methods called LiMe that formulates the PRF task as a matrix factorization problem. LiMe learns an inter-term similarity matrix from the pseudo-relevant set and the query that uses for computing expansion terms. The experiments on five datasets show that LiMe outperforms state-of-the-art baselines in most cases.

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. Query Expansion as a Matrix Factorization Problem: Extended Abstract. In *CERI 18: 5th Spanish Conference in Information Retrieval, June 26–27, 2018, Zaragoza, Spain.* ACM, New York, NY, USA, Article 3, 4 pages. https://doi.org/10.1145/3230599.3230603

1 INTRODUCTION

Retrieval can be improved if we exploit user's feedback for the presented results. However, we are not usually provided with relevance feedback [19]. For this reason, pseudo-relevance feedback (PRF) has emerged as an alternative that does not require user feedback [4]. PRF approach assumes that the top retrieved documents for the original query are relevant. These documents form the pseudo-relevant set. PRF techniques extracts and weights terms from this set to expand the original query. The expanded query is used to perform a second retrieval whose output is displayed to the user. This method has shown to be a very successful technique for improving retrieval effectiveness [3, 5, 6, 8–11, 13, 14, 16–18, 24].

In this paper, we summarize our previous contributions in modelling the PRF task as a matrix factorization problem [23]. Our proposal LiMe, based on linear methods, models inter-term similarities using the original query and the pseudo-relevance set. This technique is agnostic of the retrieval model. Additionally, it can employ any document-term feature scheme (we propose TF and TF-IDF). In particular, we compute the factorization using a bound-constrained least-squares solver with an elastic net penalty. The experiments on five TREC datasets shows that LiMe is a competitive PF technique obtaining significant improvements over the state of the art in most scenarios.

2 DESCRIPTION OF LIME

LiMe exploits information from the original query Q and the pseudo-relevant set F to generate an extended query Q'. The set F is formed by the top-k documents obtained using the original query Q. Since LiMe considers the query to be another pseudo-relevant document, we define the extended pseudo-relevant feedback set F' as the pseudo-relevant set plus the original query (i.e., $F' = \{Q\} \cup F$) and we denote its cardinality by $m = |F'| = k + 1$. We define the vocabulary of the extended pseudo-relevant set F' by $V_{F'}$ and its cardinality by $n = |V_{F'}|$. This set is constituted by all the terms that appear in Q or F.

We define the matrix $X = (x_{ij}) \in \mathbb{R}^{m \times n}$ which represents the extended pseudo-relevant set. The first row corresponds to the original query Q and the rest of rows represents the k documents from F. Likewise, each column of X represents a term from $V_{F'}$. To enforce sparsity, we set to zeros all entries that correspond to terms that do not appear in the current document. In the other cases, x_{ij} represents a feature between the document (or query) and the term t_j. Therefore, x_{ij} is given by:

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

where $s(t, D)$ is a weighting function that scores the term t with the document D and $f(t, D)$ is the frequency of term t in document D. We use two popular Information Retrieval weighting functions: TF and TF-IDF; in particular, the following logarithmic smoothed versions [21]:

$$s_{tf}(t, D) = 1 + \log_2 f(t, D) \tag{2}$$

CERI 18, June 26–27, 2018, Zaragoza, Spain

ACM ISBN 978-1-4503-6543-7/18/06…$15.00
https://doi.org/10.1145/3230599.3230603

$$s_{tf\text{-}idf}(t, D) = \left(1 + \log_2 f(t, D)\right) \times \log_2 \frac{|C|}{df(t)} \tag{3}$$

where $|C|$ denotes the number of documents in the collection and $df(t)$ the document frequency of term t. However, other functions are also possible. We leave this exploration for future work.

LiMe factorizes this matrix X into the product of X and another matrix $W = (w_{uv}) \in \mathbb{R}_+^{n \times n}$. The matrix W captures the inter-term similarities between every pair of words in $V_{F'}$. Therefore, entry w_{uv} corresponds to the similarity between terms t_u and t_v. To prevent W from becoming the identity matrix, we constrain the diagonal of W to be zero. Also, we add a positivity constraint to matrix W to enforce interpretability of the similarity coefficients. In this way, LiMe is defined by the following optimization problem:

$$\begin{aligned} X &\approx X\,W \\ s.t.\ \operatorname{diag}(W) &= 0,\ W \geq 0 \end{aligned} \tag{4}$$

To find a solution to this problem, we minimize the square error of the factorization. In addition, we add ℓ_2 regularization to prevent overfitting and ℓ_1 regularization to enforce sparsity. The combination of both regularizers is known as elastic net penalty [26]. Thus, the optimization objective is:

$$\begin{aligned} W^* \;=\; \underset{W}{\arg\min} \quad & \frac{1}{2}\left\|X - X\,W\right\|_F^2 + \beta_1 \left\|W\right\|_{1,1} + \frac{\beta_2}{2}\left\|W\right\|_F^2 \\ \text{s.t.} \quad & \operatorname{diag}(W) = 0,\ W \geq 0 \end{aligned} \tag{5}$$

where $\|\cdot\|_F^2$ denotes the squared Frobenius norm and $\|\cdot\|_{1,1}$ denotes the $\ell_{1,1}$ matrix norm.

We can split the above optimization problem in columns obtaining a bound-constrained linear least squares optimization problem:

$$\begin{aligned} \vec{w}_{\cdot j}^{\,*} \;=\; \underset{\vec{w}_{\cdot j}}{\arg\min} \quad & \frac{1}{2}\left\|\vec{x}_{\cdot j} - X\vec{w}_{\cdot j}\right\|_2^2 + \beta_1 \left\|\vec{w}_{\cdot j}\right\|_1 + \frac{\beta_2}{2}\left\|\vec{w}_{\cdot j}\right\|_2^2 \\ \text{s.t.} \quad & w_{jj} = 0,\ \vec{w}_{\cdot j} \geq 0 \end{aligned} \tag{6}$$

where $\|\cdot\|_2^2$ denotes the square ℓ_2 vector norm and $\|\cdot\|_1$ denotes the ℓ_1 vector norm. Note that we represent the j-th column of matrix X by $\vec{x}_{\cdot j}$ and the j-th column of matrix W by $\vec{w}_{\cdot j}$.

We used the BCLS[1] library to compute each column of W^*. Note that the computation of each column is independent and, thus, we can calculate them in parallel. Once we have computed W^*, we can reconstruct the first row of X (denoted by $\hat{x}_{1\cdot}$) which represents the expanded query in the following manner:

$$\hat{x}_{1\cdot} = \vec{x}_{1\cdot} W^* \tag{7}$$

We can calculate a probability estimate of the feedback model θ_F by normalizing this vector $\hat{x}_{1\cdot}$. Thus, the LiMe feedback model is computed as follows:

$$p(t_j|\theta_F) = \begin{cases} \dfrac{\hat{x}_{1j}}{\sum_{t_v \in V_{F'}} \hat{x}_{1v}} & \text{if } t_j \in V_{F'}, \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

In this way, we can rank all those terms that appear in the pseudo-relevant set or the query.

If we use the language modelling framework, we rank documents according to the KL divergence $D(\cdot\|\cdot)$ between the query and the document language models, θ_Q and θ_D. This is rank equivalent to the negative cross-entropy [7]:

$$Score(D, Q) = -D(\theta_Q\|\theta_D) \overset{\text{rank}}{=} \sum_{t \in V} p(t|\theta_Q) \log p(t|\theta_D) \tag{9}$$

where V is the vocabulary of the collection. When performing the second retrieval in pseudo-relevance feedback, we use the extended query model θ'_Q which is the result of the interpolation (controlled by the hyperparameter $\alpha \in [0, 1]$) between the original query model θ_Q and the estimated feedback model θ_F [1, 10]:

$$p(t|\theta'_Q) = (1 - \alpha)\, p(t|\theta_Q) + \alpha\, p(t|\theta_F) \tag{10}$$

Table 1: Collections statistics. We also detail the topics that are used for training and test purposes.

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

3 EXPERIMENTS

In this section, we present the experimental evaluation of LiMe. We perform our experiments on five diverse TREC collections [10, 11, 24]: AP88-89, TREC-678, Robust04, WT10G and GOV2. Since PRF is typically used for expanding short queries, we employ the title query for each topic. We split the topics into training and test: we use the training topics to tune the model hyperparameters that maximize MAP and we use the test topics to assess the performance of the techniques. We present in Table 1 the statistics of each collection and the training and test splits.

We use Terrier framework [12] to conduct these experiments. We applied Porter stemming and stopwords removal because previous work has shown they improve the performance of PRF techniques [10].

We evaluated the methods using MAP and nDCG at a cut-off of 1000 using the `trec_eval`[2] implementation. We also computed the robustness index (RI) [20] to assess how many topics benefit from using PRF. We used the one-tail permutation test at level $p < 0.05$ to perform statistically significance tests in MAP and nDCG [22]. Note that we cannot use a paired statistic with RI because it is a global metric.

We use the language modelling framework as retrieval model [15] with Dirichlet priors smoothing ($\mu = 1000$) [25]. In particular, we use the KL divergence model (see Eq. 9) to incorporate the feedback model [7] into the second retrieval. To assess the performance of LiMe, we use the following baselines:

- **LM** The first retrieval obtained with language models without query expansion.
- **RFMF** This PRF technique is based on non-negative matrix factorization [24].

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

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

Table 2: Values of MAP, P@5, nDCG and RI for LM, RFMF, MEDMM, RM3, LiMe-TF and LiMe-TF-IDF techniques on each dataset. We superscripted with a, b, c, d, e and f all statistically significant improvements (permutation test $p < 0.05$) with respect to LM, RFMF, MEDMM, RM3, LiMe-TF and LiMe-TF-IDF, 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}	$\mathbf{0.3149}^{abcde}$
	nDCG	0.5637	0.5749^{a}	0.5955^{ab}	0.6005^{ab}	0.6003^{ab}	$\mathbf{0.6085}^{ab}$
	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}	$\mathbf{0.2357}^{abd}$
	nDCG	0.4518	0.4746	0.5115^{abd}	0.4987^{ab}	0.5051^{ab}	$\mathbf{0.5198}^{abde}$
	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}	$\mathbf{0.3517}^{abe}$
	nDCG	0.5830	0.5884	0.6227^{ab}	0.6251^{ab}	0.6223^{ab}	$\mathbf{0.6294}^{ab}$
	RI	–	0.07	0.32	**0.37**	0.23	**0.37**
WT10G	MAP	0.2194	0.2389^{a}	0.2472^{a}	0.2470^{a}	$\mathbf{0.2484}^{a}$	0.2476^{a}
	nDCG	0.5212	0.5262	0.5324	0.5352	$\mathbf{0.5416}^{a}$	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}	$\mathbf{0.3830}^{ab}$
	nDCG	0.6325	0.6453	0.6653^{ab}	0.6618^{ab}	0.6656^{ab}	$\mathbf{0.6698}^{abd}$
	RI	–	0.42	0.66	0.60	**0.68**	0.62

MEDMM This PRF technique, the maximum-entropy divergence minimization model, is regarded as one of the most competitive PRF techniques [11].

RM3 The relevance-based language model with i.i.d. sampling [1, 8].

Note that all the PRF techniques are interpolated with the original query as shown in Eq. 10.

We present the results in terms of MAP, nDCG and RI in Table 2. The language modelling baseline is outperformed by all PRF techniques; however, only LiMe-TF and LiMe-TF-IDF provide significant improvements over LM in MAP and nDCG on all datasets.

The values of robustness index are positive for all PRF techniques which means that PRF provides, in general, a beneficial impact on retrieval results. We can see that LiMe techniques achieve the highest values of RI on every collection except for MEDMM on WT10G.

LiMe-TF-IDF achieves the highest value of MAP and nDCG on all collections except on WT10G where LiMe-TF obtains the highest results. Additionally, LiMe was not outperformed by any baseline. We observe that LiMe-TF-IDF significantly outperforms RFMF on four out of five datasets in terms of MAP and on three out of five collections in terms of nDCG. With respect to RM3, LiMe-TF-IDF significantly surpasses RM3 on two collections. Finally, MEDMM was only significantly outperformed by LiMe-TF-IDF on AP88-89. Nevertheless, LiMe-TF and LiMe-TF-IDF achieve higher values in nDCG and MAP than MEDMM on every collection. Although no baseline significantly improves LiMe, MEDMM significantly surpasses RM3 in terms of nDCG and MAP on the TREC-678 collection. Also, RM3 and MEDMM significantly outperform RFMF in terms of MAP and nDCG on several datasets.

Between LiMe-TF and LiMe-TF-IDF, we can see that the TF-IDF weighting scheme provides better figures of MAP and nDCG on all collections except on WT10G. However, these differences are significant only on AP88-89, TREC-678 and Robust04. Note also that LiMe-TF is slightly more robust that LiMe-TF-IDF on three out of five datasets. As we commented, WT10G is a quite noisy web crawl.

4 CONCLUSIONS AND FUTURE WORK

We presented a query expansion technique based on linear methods called LiMe which models the pseudo-relevance feedback task as a matrix decomposition problem. This is an extended abstract that summarizes previous work [23]. LiMe computes inter-term similarities using information from the original query and the pseudo-relevant. This can be applied on top of any retrieval model. Extensive experimentation showed that LiMe achieves state-of-the-art performance on five TREC collections.

As future work, we plan to experiment with different feature schemes (in addition to TF and TF-IDF) as well as explore the meaning behind the item-item similarities computed by LiMe. In addition, we envision to study other inter-term similarity measures such as those used in translation models [2, 7].

ACKNOWLEDGMENTS

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

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] 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

[4] 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

[5] 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

[6] 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

[7] 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

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

[9] 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

[10] 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

[11] 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

[12] 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.

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

[14] 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

[15] 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

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

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

[18] 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.

[19] 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

[20] 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

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

[22] 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

[23] Daniel Valcarce, Javier Parapar, and Álvaro Barreiro. 2018. LiMe: Linear Methods for Pseudo-Relevance Feedback. In *Proceedings of the 33st Annual ACM Symposium on Applied Computing (SAC '18)*. ACM, New York, NY, USA, 678–687. https://doi.org/10.1145/3167132.3167207

[24] 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

[25] 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

[26] 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.