Information Retrieval Journal (2021) 24:269–297
https://doi.org/10.1007/s10791-021-09393-5

Pseudo relevance feedback optimization

Avi Arampatzis[1] · Georgios Peikos[1] · Symeon Symeonidis[1]

Received: 4 November 2020 / Accepted: 16 April 2021 / Published online: 25 May 2021

Abstract

We propose a method for automatic optimization of pseudo relevance feedback (PRF) in information retrieval. Based on the conjecture that the initial query's contribution to the final query may not be necessary once a good model is built from pseudo relevant documents, we set out to optimize per query only the number of top-retrieved documents to be used for feedback. The optimization is based on several query performance predictors for the initial query, by building a linear regression model discovering the optimal machine learning pipeline via genetic programming. Even by using only 50–100 training queries, the method yields statistically-significant improvements in MAP of 18–35% over the initial query, 7–11% over the feedback model with the best fixed number of pseudo-relevant documents, and up to 10% (5.5% on median) over the standard method of optimizing both the balance coefficient and the number of feedback documents by grid-search in the training set. Compared to state-of-the-art PRF methods from the recent literature, our method outperforms by up to 21% with an average of 10%. Further analysis shows that we are still far from the method's effectiveness ceiling (in contrast to the standard method), leaving amble room for further improvements.

Keywords Pseudo relevance feedback · Blind relevance feedback · Optimization · Query performance predictors · Query difficulty · Regression

1 Introduction

One of the major topics in the field of Information Retrieval is automated ways to optimize retrieval effectiveness. Nowadays, new commercial search applications are in high public demand, therefore, search engines should be equipped with techniques that can process

✉ Avi Arampatzis
avi@ee.duth.gr

Georgios Peikos
georpeik1@ee.duth.gr

Symeon Symeonidis
ssymeoni@ee.duth.gr

[1] Database & Information Retrieval research unit, Department of Electrical & Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece

user queries extensively and yield good search results, such as Pseudo (or Blind) Relevance Feedback (PRF), among others.

PRF is an age-old method for improving retrieval effectiveness (Salton 1971; Croft and Harper 1979; Xu and Croft 1996). It is (commonly) a two-step process that enables us to utilize information about the initial query with respect to the collection, by using information from documents retrieved by the initial query in order to formulate and issue a better query. The recent literature about PRF attempts to produce new linguistic models to select terms from the retrieved documents or other external sources, e.g. (Jaleel et al. 2004; Tao and Zhai 2006; Lv and Zhai 2009b, 2010, 2014), or uses mathematical models to re-weight the chosen terms and reformulate the final query, e.g. (Singh et al. 2017; Valcarce et al. 2018).

PRF, in its classic form, involves three parameters: the number of top-ranked documents by the initial query that will be considered relevant so as to build a positive feedback model, the relative weight of the feedback model against the initial query, and the number of terms to keep in the improved query. In this paper, we deal with the automated optimization of such parameters. Based on the conjecture that the initial query's contribution to the final query may not be necessary once a good model is built from pseudo relevant documents, we set out to optimize per query only the number of top-retrieved documents to be used for feedback.

The idea for the conjecture originates in (Arampatzis et al. 2000; Arampatzis 2001), where in an adaptive filtering context the authors introduced and employed *initial query elimination/degradation*. Quoting Arampatzis (2001): *"The initial query is considered as carrying a worth of a certain number of relevant documents. As a result, the contribution of an initial query in training a classifier decreases with the number of relevant training documents."* Thus, as more and more training documents were becoming available during adaptive filtering, the contribution of the initial filtering query was gradually diminishing in adapting the classifier. The technique was applied successfully in TREC-9 and TREC-10 Filtering Tracks, assuming a worth of 10 and 2 relevant documents, respectively. We investigate this method at the far end of the spectrum by eliminating the initial query's contribution altogether.

The contributions of the present study are the following. We explore the initial query elimination conjecture by arguing theoretically and investigating experimentally whether it holds some truth also in a PRF context. In this respect, we develop a PRF optimization method which disregards the initial query but builds a better positive feedback model by optimizing, per query, the number K of top-retrieved documents to be used for feedback. The optimization is based on several query performance predictors (QPPs) for the initial query, used as inputs to a linear regression model for predicting the optimal K. The machine learning pipeline of the linear regression model itself is also optimized using genetic programming via a tool which it intelligently explores thousands of possible pipelines to find the best one for the data at hand. The approach requires training data, and while it may be computationally-heavy in training, it is quite fast in query-time.

Despite this interesting perspective, to the best of our knowledge, only a small number of previous studies have tried to solve this or a similar optimization problem in the context of PRF, e.g. (Sakai et al. 2005; Lv and Zhai 2009a; Parapar et al. 2014), and none of them used QPPs except Amati et al. (2004) who employed QPPs only to decide whether to apply PRF to a query or not. Over the years, query performance prediction has become an important research area, consisting of two primary methodologies, i.e. pre- and post-retrieval. The former studies the expected query performance before the retrieval takes place, i.e. using only the query and collection statistics. The latter takes also into consideration data produced by the retrieval, such as the result list (Markovits et al. 2012; Shtok et al. 2012). Since in PRF the initial query is always run, the latter methodology seems more suitable and expected to be more beneficial.

The rest of the paper is organized as follows. Section 2 gives a brief overview of related works. Section 3 introduces and elaborates on the proposed method. Section 4 presents the experimental evaluation. Section 5 provides further discussion and insight, before conclusions are drawn and several directions for further research are pointed out in Sect. 6.

2 Related work

Over the years, there has been a considerable interest in Query Expansion (QE). QE approaches are classified into two groups: global, which have as a primary goal the extraction of a set of terms from various data sources (external or internal, e.g. thesauri) to meaningfully augment user's original query, and local, which expand and re-weight user's original query with terms derived from the analysis of the result set. There is a large volume of studies describing the role of QE, focusing primarily on techniques to improve retrieval effectiveness, e.g. Mitra et al. (1998); Kekäläinen and Järvelin (1998); Crouch et al. (2002); Cronen-Townsend et al. (2002); Ruthven and Lalmas (2003); Abdelmgeid Amin (2008); Azad and Deepak (2019).

There are two ways to expand and/or re-weight the user's original query with local methods techniques, relevance feedback and pseudo relevance feedback (PRF). A well-known method for relevance feedback is Rocchio's (1971) which is based on the vector space model, and another primary study is that of Croft and Harper (1979) which is a probabilistic approach.

Karisani et al. (2016) proposed a method to extract the most informative terms in a set of documents for PRF. A set of documents is retrieved using the user's initial query and then a weight is assigned to each document describing the document's closeness to the user's information need. These weights are used to recalculate the final query's term weights. The experimental results in standard English and Persian test collections increased MAP up to 7%.

Another method, that achieved significant improvements in retrieval effectiveness, was proposed by Singh et al. (2017). They explored the possibility of using fuzzy logic-based QE approach to improve overall efficiency. The weights of each word were mixed using fuzzy rules to infer the weights of the additional query terms. At the end, after the fuzzy logic approach, they filter out semantically irrelevant terms to further improve their results.

Lv and Zhai (2010) proposed a novel positional relevance model to reward terms close to the initial query terms in the feedback documents and avoid including irrelevant terms in the feedback model. Their proposed method is an extension of relevance models so they set the parameters, such as feedback coefficient, number K of feedback documents, and number of expansion terms to fixed values. However, in order to check the robustness of the proposed method regarding the K value, the authors also tried K values varied from 0 to 100. Their methods proved robust and effective compared to the standard relevance feedback models.

Parapar and Barreiro (2011) presented two different approaches for the Relevance Model (RM) (Lavrenko and Croft 2001), promoting terms under the Language Modelling framework to improve divergence in the PRF context. The first approach (KLD3) is built upon Kullback–Leibler Divergence based on query expansion in the language modelling framework. The second approach (RM3DT) is based on the Relevance Model with the promotion of Divergent Terms. The authors evaluated the performance of the proposed models on TREC collections, and the RM3DT method outperformed the baseline Language Modelling (LM) retrieval model by 11–31%, and the RM3 feedback model by 0.5–23%.

Valcarce et al. (2018) examined the use of linear methods for PRF. They proposed the Lime model, a novel formulation of the PRF task as a matrix decomposition problem. To expand the original query, they used a factorization that includes the computation of an inter-term similarity matrix. Also, for the proposed decomposition, they applied linear least squares regression with regularisation. The proposed LiMe-TF and LiMe-TF-IDF outperform the LM (12–34%) and RM3 (0.6–5.5%) baselines, on five TREC datasets. In both of the last-mentioned studies, the number K of feedback documents is tuned, per PRF model, based on training data, to the same fixed number (one of 5,10,25,50,75,100) for all queries.

An important limitation found in the aforementioned studies which use local QE techniques to improve retrieval effectiveness lies in the fact that the number K of pseudo-relevant documents is set to a specific *fixed value for all queries* (irrespective of whether it is optimized on some training set or not), with the most common being 5, 10, 20, 30, 50 (Raiber and Kurland 2014). Only a few researchers attempted to optimize, per query, the balance α between initial query and feedback information (Lv and Zhai 2009a), or realized the importance of a good PRF document set (Sakai et al. 2005) or K with respect to the query (Parapar et al. 2014). These constitute the more related works, which we will see next.

Lv and Zhai (2009a) proposed three heuristics to adaptively predict the optimal balance between initial query and feedback information in PRF. To predict the balance coefficient α, several features were examined and combined by using a regression approach which led to robust and effective results compared with the regular fixed-coefficient feedback. In our study, we focus on the K parameter instead, eliminating α.

An attempt to adjust the number of pseudo-relevant documents per query was proposed by Sakai et al. (2005), called Selective Sampling. The method assumes that some of the initial top-ranked documents are not useful, so it skips those documents while it creates the set of pseudo-relevant documents S. Three parameters are introduced, $P_{\min}, P_{\max}, P_{\text{scope}}$, which are the minimum/maximum number of pseudo-relevant documents required and the total number of pseudo-relevant documents examined per query. The algorithm uses these three parameters as cutoffs, so that $P_{\min} \leq |S| \leq P_{\max} \leq P_{\text{scope}}$, which were set via training with the NTCIR-3 Japanese test collection to 3, 10, and 50, respectively. They used 40 expansion terms which were down-weighted by a factor of 0.25 compared to the initial query terms. An evaluation on the NTCIR-4 Japanese/English test collection found that Selective Sampling outperforms traditional PRF methods almost as often as traditional PRF methods outperform Selective Sampling, which is rather a tie. Our work is quite different, since we do not skip documents or optimize K with a fixed value for all queries but optimize it per query, and it will be proven clearly effective (as will see in this paper).

A method (SDRM3) that tried to optimize the number of pseudo relevant documents per query was proposed by Parapar et al. (2014). The authors investigated the score distribution of the initial retrieval and tried to break it down to its relevant and non-relevant components; they formulated the problem as a threshold optimization task (similarly to what was proposed before, e.g., in Arampatzis et al. 2009) and evaluated the model's performance on TREC collections. Significant improvements were found compared to the baseline LM retrieval model (8–17%). Although there were also improvements over the baseline (RM3) feedback model (2.3%), these were not statistically significant.

Thus, the study of Parapar et al. (2014) is the most related, as it tries to solve the same problem using the score distribution of the initial retrieval; however, there are still major differences. Firstly, in their approach, they use the training set to optimize the number of expansion terms and the balance coefficient (both with fixed values irrespective of the query), and the smoothing parameter; we eliminate the initial query and do not tune any other parameter. Secondly, while the mixture model of the relevant and non-relevant score distributions is tightly-coupled to the retrieval model employed (Arampatzis and Robertson 2011), our approach based on query performance predictors is—in principle—retrieval model invariant and certainly much faster in query-time (recovering the parameters of a mixture model iteratively is much more expensive than calculating our predictors). Finally, we achieve larger improvements over the initial retrieval and over the baseline feedback model, as will see later in our experiments.

Thus, the current study attempts to solve the optimization problem at hand by using a novel approach. To the best of our knowledge, this is the first time anyone explores query performance predictors (QPPs) to determine the optimal number of pseudo-relevant documents per query. Amati et al. (2004) who used QPPs only to determine whether or not to apply PRF in a query, used a fixed $K = 10$ when their method told them so; such a *selective* PRF is also included our method, since we detect queries for which it would not be beneficial and switch it off. Moreover, our method can be used independently of the retrieval and PRF models, as QPPs can be calculated using the initial retrieval scores and a regression model can be built with a relatively few training queries.

3 Optimizing pseudo relevance feedback

Let Q_0 be the initial user query, expressed for some information need. Traditionally, pseudo relevance feedback (PRF) involves three parameters: the number K of top-ranked documents retrieved by Q_0 to be considered as pseudo-relevant, the Q_0's weight α against the positive feedback query/model $Q_{r,K}$ built from the K pseudo-relevant documents, and the number T of top-weighted feedback terms to be retained in the modified query Q_m. Assuming vector representations for Q_0, $Q_{r,K}$, Q_m, the modified query is calculated as:

$$Q_m = \alpha Q_0 + (1-\alpha) Q_{r,K}\,, \quad 0 \leq \alpha \leq 1\,. \tag{1}$$

Taking as an example Rocchio's formula, it uses three weights: α, β, γ. Since there is no negative feedback in PRF, γ is set to zero or eliminated. Additionally, $\beta = 1 - \alpha$, since what matters practically is the relative weight of the contributions of Q_0 and $Q_{r,K}$ to Q_m, i.e. there is a single free weight after all: α. Rocchio builds $Q_{r,K}$ as the average pseudo-relevant document vector or centroid.

Of the three parameters involved (α, K, and T), the latter has been deemed as the least important after decades of experimentation. The number of terms used for query expansion with PRF is less significant than the quality of terms selected, as stated many times before in the literature (e.g. Sihvonen and Vakkari (2004)), so commonly T is set to 20. Since optimization of T does not seem to worth the effort, we also set $T = 20$ and focus on the former two parameters.

Most previous research in PRF, pre-set K and α to fixed values *independent* of the Q_0 at hand, such as $\alpha = 0.5$ and $K = 10$. These actual fixed values are usually determined experimentally by selecting the α and K which maximize, on average, some effectiveness measure on a set of training queries on some benchmark corpus. We will refer to this optimization method as *standard* throughout the paper. Note that there is no single α/K combination that optimizes all evaluation measures, but the optimal values depend on the measure of interest.

The value of α denotes the degree of distrust we have in the feedback model $Q_{r,K}$: the larger the α, the less the confidence we have in $Q_{r,K}$ with respect to its quality. For a given Q_0 and its initial ranking, the quality of $Q_{r,K}$ depends solely on the choice of K, for which two factors come at play:

1. The number R of documents relevant to the information need. Assuming Q_0 yields a perfect ranking (i.e. all R relevant documents are ranked above all non-relevant ones), K should not be set greater than R, otherwise $Q_{r,K}$ (and consequently Q_m) may *drift* away from Q_0 and achieve a worse ranking. Setting K less than R may also have an adverse effect due to a possible insufficient *coverage* of the topic in $Q_{r,K}$. Accounting for imperfections in training $Q_{r,K}$ statistical anomalies and other effects, we can say that the best K is *around* R, when Q_0 produces a perfect ranking.
2. The Q_0's effectiveness or quality of its ranking. For a less-than-perfect Q_0 ranking, K should be set *lower* than R, since the density of relevant documents generally increases when going up the ranking and decreases when going down. In other words, from the two alternative sets of top-K documents, $K = R - \delta$ or $K = R + \delta$ (for a positive integer δ), the former is expected to have a larger fraction of relevant documents than the latter. Therefore, this strategy produces a 'cleaner' pseudo-relevant set with respect to the fraction of relevant documents it contains.[1] In any case, when Q_0 is imperfect, we pay for drift and coverage problems.

Based on the above, the optimal K can take a value up to around R. The more effective the Q_0, the nearer the optimal K is to R. The less effective the Q_0, the further the optimal K moves away from R to smaller values. Thus, positive correlations between the optimal K and both R and Q_0 effectiveness are expected.

Since R is unknown and Q_0 is imperfect in practice, it is difficult to achieve the delicate balance between drift and coverage in $Q_{r,K}$. To alleviate these effects from spilling into Q_m and keep focus to the user's information need, α is usually set to a value > 0.5, retaining a significant (safe) contribution of Q_0 to Q_m, more than $Q_{r,K}$.[2] In combination with using the same fixed α and K values for all incoming queries, PRF's potential may not be squeezed out in its entirety.

Based on the above, we argue that once one has a method for optimizing *K per query*, the α parameter becomes much less important and could even be eliminated/set-to-zero discarding Q_0's contribution. A perfect $Q_{r,K}$ could potentially encapsulate all Q_0's information, deeming perhaps Q_0's contribution to Q_m unnecessary.[3] While Q_0's effectiveness cannot be controlled during PRF (it depends on the query issued, retrieval model, collection pre-processing/indexing, etc.), since PRF is always (at least) a two-stage process, Q_0's effectiveness and R could be

[1] Nevertheless, this may not be an effective strategy when the set of pseudo-relevant gets too small. When the amount of training data is not sufficient, perhaps a larger but 'dirtier' set is preferable to a smaller but 'cleaner' one.

[2] For example, PRF in the Terrier search engine defaults at $\alpha = 0.6$ and $K = 3$.

[3] Even though a perfect $Q_{r,K}$ could potentially encapsulate all Q_0's information, it may also contain additional information that is not necessary to the information need. This may also be the case for Q_0, depending on how well the user has expressed the information need, so a (large) contribution of Q_0 may still not have a desirable effect.

estimated, guiding the selection of a better than a pre-set fixed K. Ideally, in the extreme case, such an optimization method should even predict a $K = 0$, meaning that no PRF would be beneficial and only the Q_0 should be used.

Thus, the method we propose employs query performance predictors (also known as *query difficulty*) to determine/predict Q_0's effectiveness, and uses their values to predict an optimal K per query that maximizes a given effectiveness measure. In this study, we will not consider any R-predictors, although they constitute an obvious and perhaps effective extension.

3.1 Post-retrieval query performance predictors

In the Query Performance Prediction (QPP) literature, there are several quantities correlated to retrieval effectiveness, usually to MAP (Hauff 2010), but also to other measures since many measures are correlated— in their turn—to MAP, e.g. Precision@R (Manning et al. 2008). There exist pre- and post-retrieval QPPs. Since in a PRF setting, the initial query will always run, it makes sense to focus on post-retrieval QPPs.

There are three main categories of post-retrieval QPP methods. The first one is clarity-based methods that directly measure the ambiguity of the results list with respect to the corpus (Cronen-Townsend et al. 2002). The second is robustness-based methods, which evaluate how robust the results are to perturbations in the query, the result list, and the retrieval method (Zhou and Croft 2007; Yom-Tov et al. 2005). Lastly, the score distribution-based methods analyze the score distribution of the results list.

According to Zhou and Croft (2007), the methods of the first two categories are time-consuming. Since PRF alone more than doubles the runtime, it is not desirable to burden it further. For instance, to calculate robustness there is the need to generate a random collection by sampling from document models of the documents in the original collection, and then perform retrieval on both collections. The similarity between the two rankings is the robustness score. To calculate the clarity score one needs to estimate the query's and the collection's language model. Although the collection's language model can be pre-computed during indexing, the query language model is estimated by sampling documents after the initial retrieval. For these reasons, we resort to QPPs which are based on the score distribution of the initial results list, which are easy and fast to calculate.

Consequently, we employ three post retrieval QPPs, namely, WIG, NQC, and SMV; all three have been widely used in recent studies (Zhou and Croft 2007; Shtok et al. 2012; Tao and Wu 2014).

3.1.1 Weighted information gain (WIG)

The Weighted Information Gain (WIG) predictor was introduced by Zhou and Croft (2007) as an approach to predict query performance in web search environments. It measures the divergence between the mean retrieval score of some top documents in the result list and that of a random document in the whole corpus. Equation 2 is a simplified version of the WIG predictor formula which, according to Zhou (2008), is efficient and uses only the scores of the results:

$$\mathrm{WIG}(q, \mathcal{M}) = \frac{1}{n} \sum_{d \in \mathcal{D}_q^{[n]}} \frac{1}{\sqrt{|q|}} (\mathrm{Score}(d) - \mathrm{Score}(\mathcal{D}))\,, \tag{2}$$

where n is a free parameter equal to the number of top-ranked documents used for calculating the predictor, $\mathcal{D}_q^{[n]}$ is the set of the top-n documents, and $|q|$ is the query length. Score(d) is the score assigned to document d by the retrieval model $\mathcal{M}$. Finally, Score($\mathcal{D}$) is the average score of all retrieved results.

This predictor has been used in previous studies (Tao and Wu 2014; Shtok et al. 2012). According to Markovits et al. (2012), the normalization of the WIG by the query length $|q|$ harms the prediction quality on TREC benchmark collections, so we removed this normalization in our experiments. Lastly, we set $n = 5$, as in Zhou (2008).

3.1.2 Normalized query commitment (NQC)

The Normalized Query Commitment (NQC) predictor, proposed by Shtok et al. (2012), estimates the amount of query drift in the list of top-retrieved documents using the standard deviation of their retrieval scores:

$$\mathrm{NQC}(q, \mathcal{M}) = \frac{\sqrt{\frac{1}{n}\sum_{d \in \mathcal{D}_q^{[n]}} (\mathrm{Score}(d) - \hat{\mu})^2}}{\mathrm{Score}(\mathcal{D})}, \tag{3}$$

where $\hat{\mu}$ is the average score of the top-n results in $\mathcal{D}_q^{[n]}$. We set $n = 100$, as recommended by Shtok et al. (2012).

3.1.3 Score magnitude and variance (SMV)

According to Tao and Wu (2014), WIG and NQC tend to work in some situations and fail in others; as a result, they developed another post-retrieval predictor, namely, the Score Magnitude and Variance (SMV):

$$\mathrm{SMV(q}, \mathcal{M}) = \frac{\frac{1}{n}\sum_{d \in \mathcal{D}_q^{[n]}} \left(\mathrm{Score}(d) \left| \ln \frac{\mathrm{Score}(d)}{\hat{\mu}} \right| \right)}{\mathrm{Score}(\mathcal{D})}. \tag{4}$$

Once more, we set $n = 100$, as recommended by Shtok et al. (2012).

3.2 Predicting *K* optimal

First, we investigated how the optimal K for MAP ($K_{\mathrm{opt\_MAP}}$) looks like on real data. In initial experiments with a benchmark dataset (which is described in detail in Sect. 4.1), we run a Q_0, built on its results all positive-only PRF queries $Q_m = Q_{r,K}$ for $K = 1 \ldots 200$ with Rocchio, and evaluated them on the test corpus in order to find $K_{\mathrm{opt\_MAP}}$. We did that for 150 different Q_0s. The min/med/avg/max $K_{\mathrm{opt\_MAP}}$ found was 1/13/46.6/200. Two topics hit the 200 mark, suggesting that we should have searched also higher Ks; nevertheless, the distribution is quite skewed to the downside, so we are confident that these few topics which should have had $K_{\mathrm{opt\_MAP}} > 200$ will not affect our overall experimental results.

Figure 1 shows how MAP changes with K for four queries, and Table 1 gives some more quantitative information. As it can be seen in the figure, MAP as a function of K is neither smooth nor monotonic. These four queries were selected as representatives of four broad and rough categories of behaviour we saw in the data: multiple modals, rising-and-falling, falling, rising.

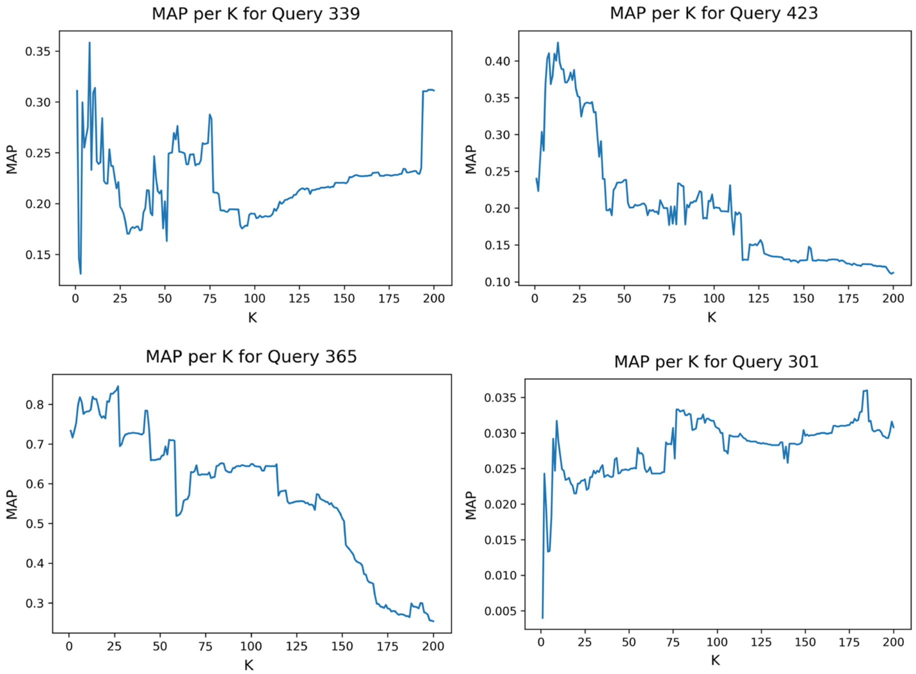

Fig. 1 MAP as a function of using the top-K documents as pseudo-relevant for PRF (with a zero contribution of the initial query)

Table 1 Example topics/queries

topic	R	MAP@Q_0	$K_{\text{opt\_MAP}}$	MAP@$Q_{r,K_{\text{opt\_MAP}}}$	WIG	NQC	SMV
339	10	.3389	8	.3585	16.274	1.696	1.371
423	21	.5685	13	.4251	14.694	0.622	0.509
365	35	.6955	27	.8457	11.900	0.716	0.570
301	474	.0280	185	.0360	6.960	0.147	0.123

In the table, we can see than 3 out of 4 topics have an improved MAP with $Q_{r,K_{\text{opt\_MAP}}}$ over Q_0; for these, optimizing K produces a $Q_{r,K}$ better than Q_0, so such an optimization method is beneficial. However, queries similar to 423 cannot be improved with any positive-only PRF model disregarding the initial query. Of the 150 topics in our dataset, only 19 (12.7%) fall in this category. In such cases, tuning of the parameter α may be needed, but instead, we decided to incorporate these cases into our prediction model in order to switch off PRF when they are detected (we will see how below). In any case, using the same fixed K for all queries seems rather naive, and it only works on average as long as a proper fixed K is selected.

We calculated the three QPPs (Sect. 3.1) on the result lists of the 150 Q_0s and measured their correlation to the MAP of Q_0, as shown in Table 2. There are statistically significant positive correlations everywhere, with a strength typical for QPPs. So, they are doing their intended job, but do they also predict $K_{\text{opt\_MAP}}$?

Table 2 Correlation of MAP@Q_0 to QPPs (significance levels .05° and .001$^{\bullet}$)

MAP@Q_0 corr.	WIG	NQC	SMV
Pearson	0.237°	0.198°	0.191°
Spearman	$0.361^{\bullet}$	$0.406^{\bullet}$	$0.387^{\bullet}$
Kendall	$0.265^{\bullet}$	$0.291^{\bullet}$	$0.279^{\bullet}$

Table 3 Correlation of $K_{\text{opt\_MAP}}$ to QPPs (all statistically insignificant)

$K_{\text{opt\_MAP}}$ corr.	WIG	NQC	SMV
Pearson	-0.137^{-}	-0.040^{-}	-0.024^{-}
Spearman	-0.069^{-}	-0.051^{-}	-0.041^{-}
Kendall	-0.058^{-}	-0.051^{-}	-0.039^{-}

Table 3 shows the correlation of $K_{\text{opt\_MAP}}$ to QPPs. Unfortunately, individual QPPs seem to have almost no predictive power for $K_{\text{opt\_MAP}}$.[4] But it may be the case that their predictive power becomes significant when their values are scaled and/or non-linearly transformed and all three are combined into a single regression model.

To achieve this, we used TPOT[5] in order to transform and scale the independent variables, and project them into a high dimensional feature space via a kernel-based method, so as to become more compatible with linear regression. TPOT is a tool that optimizes machine learning pipelines using genetic programming. It intelligently explores thousands of possible pipelines to find the best one for the data at hand. Among the transformation methods explored by TPOT are those which handle multicollinearity. Multicollinearity occurs when independent variables in a regression model are correlated (Rosipal et al. 2001); in our case, NQC and SMV exhibit a strong, statistically significant Pearson correlation.

The approach requires training data, which in our case are the values of QPPs for a set of Q_0s together with their corresponding $K_{\text{opt\_MAP}}$. If a Q_0's MAP is greater than the MAP of $Q_{r,K_{\text{opt\_MAP}}}$, then we set $K_{\text{opt\_MAP}} = 0$ in order to enable prediction of cases where PRF would not be beneficial. TPOT iterates through different regression models paired with feature selectors and transformers, and each iteration produces a pipeline. When some pipeline is used for regression, it is bound to have some forecast error (loss) as measured by a specified loss function. As a loss function, we selected the mean absolute error (MAE) between $K_{\text{opt\_MAP}}$ and $K_{\text{pred\_MAP}}$.[6]

Within the training set, the evaluation of each pipeline's loss is performed in a 5-fold cross-validation fashion to avoid over-fitting. Finally, the pipelines are ranked in an increasing order of their loss, and the best one is selected and re-trained on the entire set of

[4] Although it is risky to draw conclusions on non-significant correlations, the widespread *negative* co-efficients in Table 3 are worrisome. They seem to suggest a flaw in our argument at the beginning of Section 3 (i.e. the better the Q_0's effectiveness, the larger the $K_{\text{opt\_MAP}}$), but there is no flaw. We will investigate and discuss this in Section 5.1.

[5] http://epistasislab.github.io/tpot

[6] From a theoretical point of view, the MAE is not the right forecast error to be measured/minimized here for the problem at hand. Nevertheless, in practice, MAE has worked better than alternatives. We will elaborate on this in Section 5.2.

Table 4 Statistics of Post Retrieval Query Performance Predictors

	WIG	NQC	SMV
Average	9.3370	0.4946	0.3945
Median	9.2550	0.3710	0.2920
STD	3.5874	0.5013	0.4104

training samples. The selected pipeline is used to predict $K_{\text{pred\_MAP}}$ for an unknown query. If the pipeline predicts a $K_{\text{pred\_MAP}} <= 0$, then PRF does not take place and the Q_0 is used.

4 Evaluation

First, in Sect. 4.1, we describe the experimental setup, i.e. the main benchmark corpus, train–test splits, evaluation measures, and search engine used along with parameter settings. Then, we run two main experiments in Sect. 4.2. Finally, in Sect. 4.3, we provide results on additional corpora and/or splits as well as comparisons to state-of-the-art PRF methods.

4.1 Experimental setup

We evaluated on a TREC[7] corpus, namely, TREC Volumes 4 and 5 minus the Congressional Record (Voorhees and Harman 1999), which consists of newswire articles. We used TREC topics 301–450, in several different train/test splits as it will be described below, for training and testing our regression model. The min/median/average/max numbers of relevant documents these topics have in the corpus are 3/67/93.4/474. Relevance feedback is more effective and recommended for short queries, so we used only the titles of the TREC topics.

As evaluation measures, we employed MAP, Precision@30, Precision@R, and Recall@1000, where R is the number of relevant documents of a query. While in our experiments we target to optimize only the first measure (MAP), the latter three serve as auxiliary measures in order to get more insight. We report the macro-averages of these measures across the queries of multiple test-sets. In the published literature of PRF methods, it is usual to report also the Robustness Index (RI), introduced by Sakai et al. (2005), which gives information about the reliability of the improvements. For a set Q of test queries, it is defined as $\text{RI}(Q) = (n_+ - n_-)/|Q| \in [-1, 1]$, where n_+ and n_- are the numbers of queries that are respectively helped or hurt by the feedback method according to some evaluation measure.

For indexing the collection, we used Terrier[8] v4.2 with Porter Stemmer, without removing stop words. For retrieval, we used Terrier's default language model, which is the inverse document frequency model for randomness (InL2) (Amati and van Rijsbergen 2002).

In initial test runs, we investigated the values produced by the post-retrieval QPPs (Equations 2, 3 and 4) for all initial queries Q_0. Table 4 shows some statistics for the topics

[7] https://trec.nist.gov

[8] http://terrier.org

301–450. One query appeared to be an outlier. Query 368 had WIG, NQC, and SMV values of 15.0, 12.8, and 10.5, respectively. Its NQC and SMV are both well above two standard deviations from the mean, while its WIG is also very large. While these do not seem like unreasonable values given that 368's Q_0 achieves a high MAP (0.4291) but not the highest in the dataset (according to Table 1), even one outlier can have adverse effects in regression; thus, we typically excluded topic 368 from all experiments when it occurred in a training set.

In order to investigate the sensitivity of our regression model to the *selection* of queries used for training, we generated three different training/test splits of the queries 301–450. Split 1 (SPLT1) consists of the 50 queries with numbers $301 + 3k, k = 0 \dots 49$, for training, and the rest for testing. Similarly, the selection formulae for the training sets of SPLT2 and SPLT3 are $302 + 3k$ and $303 + 3k$, respectively; in all cases, the remaining 100 queries constitute the test set. Furthermore, in order to investigate the sensitivity of our regression model to the *number* of queries used for training, we generated three additional splits, this time with 100 training queries each. For SPLT4–6, we just reverse the training/test sets of SPLT1–3, respectively. Thus, we can train on double the amount of queries, however, we will test on only the 50 remaining instead of 100. As mentioned earlier in this section, we excluded query 368 wherever it occurred.

As explained in Sect. 3.2, we use the TPOT tool, developed by Olson et al. (2016), for determining the optimal pipeline for building our regression models on the training data. Table 5 lists the optimal settings produced by TPOT on the training set of each of the splits. Among the transformation methods obtained by the grid search are Power transformation, L1 Normalization, Robust Scaling of the input data, and kernel-based methods such as PCA, FastICA, Nystroem, and RBFsampler.

For PRF, we adopted Rocchio's formula, with an initial query Q_0 weight α, a positive feedback weight $\beta = 1 - \alpha$, and $\gamma = 0$ (i.e. no negative feedback used in PRF). The number of query expansion terms T is set to 20 for all experiments. Next, we will name all our runs in an "α/K" fashion, referring to the two parameters of Equation 1.

4.2 Experimental results

First, we investigate how our proposed model performs against the initial query and fixed-K positive-only PRF disregarding the initial query. Then, we compare against the standard PRF optimization retaining the initial query.

4.2.1 Initial query elimination

Table 6 shows the results for optimizing K for MAP per query with our proposed method (i.e., the 0/pM run, meaning that Q_0 is eliminated and $K = K_{\text{pred\_MAP}}$), for all six train/test splits. The second column (1/0) shows the effectiveness of the initial retrieval (Q_0), while the next five columns show the effectiveness of positive-only PRF (i.e. no initial query) for five fixed values of K. Finally, the last column (0/oM) shows the effectiveness when the optimal K for MAP ($K_{\text{opt\_MAP}}$) per query is used (i.e. the Ks we set out to predict); these MAP numbers represent the ceiling of possible effectiveness or upper bound, when the initial query is eliminated.

We remind that whenever $K_{\text{pred\_MAP}} <= 0$, 0/pM drops back to 1/0 for that topic, i.e. only the initial query Q_0 is used with no PRF. This happens 4, 8, 3, 5, 4, 8 times in

Table 5 Pipelines produced by TPOT on the training sets of the splits

	Pipelines	Parameters
SPLT1	PowerTransformer	(method=’box-cox’, standardize=True)
	RobustScaler	(quantile_range=(40,45),with_centering=False,with_scaling=True)
	LinearSVR	(C=100,dual=True,epsilon=0.01,intercept_scaling=85, loss=’squared_epsilon_insensitive’,random_state=28,tol=0.0001)
SPLT2	FastICA	(algorithm=’parallel’,fun=’logcosh’,n_components=23, random_state=28,tol=0.0001,whiten=True)
	RobustScaler	(quantile_range=(80, 85),with_centering=True,with_scaling=True)
	LinearSVR	(C=100,dual=True,epsilon=0.001,intercept_scaling=10, loss=’epsilon_insensitive’,random_state=28,tol=0.0001)
SPLT3	Normalizer	(norm=’l1’)
	RobustScaler	(quantile_range=(70,75),with_centering=False,with_scaling=True)
	LinearSVR	(C=10.0,dual=True,epsilon=1.0,intercept_scaling=35, loss=’epsilon_insensitive’,random_state=28,tol=0.01)
SPLT4	FastICA	(algorithm=’parallel’,fun=’logcosh’,n_components=27, random_state=28,tol=0.0001,whiten=True)
	RobustScaler	(quantile_range=(50,55),with_centering=True,with_scaling=True)
	LinearSVR	(C=15.0,dual=True,epsilon=1.0,intercept_scaling=10, loss=’epsilon_insensitive’,random_state=28,tol=0.0001)
SPLT5	RBFSampler	(gamma=8.4, n_components=35, random_state=28)
	Nystroem	(gamma=2.2,kernel=’linear’,n_components=25,random_state=28)
	LinearSVR	(C=25.0,dual=True,epsilon=0.0001,intercept_scaling=20, loss=’epsilon_insensitive’,random_state=28,tol=0.001)
SPLT6	PCA	(iterated_power=5,n_components=3,random_state=28, svd_solver=’randomized’,tol=0.01,whiten=True)
	RobustScaler	(quantile_range=(50,55),with_centering=False,with_scaling=True)
	LinearSVR	(C=20.0,dual=True,epsilon=0.001,intercept_scaling=10, loss=’epsilon_insensitive’,random_state=28,tol=0.01)

Table 6 Effectiveness of positive-only PRF (initial query eliminated)

SPLT1–6	blind feedback parameters α/K:							
	1/0	0/5	0/10	0/20	0/30	0/50	0/pM (vs. 0/10, vs. 1/0)	0/oM
MAP	.2206	.2310	**.2473**	.2398	.2303	.2052	.2648°(+7.1%, +20.0%)	.3499
Prec@R	.2708	.2574	**.2710**	.2642	.2630	.2426	.2873°(+6.0%, +6.1%)	.3657
Prec@30	.3236	.3138	**.3414**	.3343	.3269	.3057	.3407^{-}(−0.2%, +5.3%)	.4535
Rec@1000	.6266	.6738	.6973	.6969	**.7019**	.6917	.7031^{-}(+0.8%, +12.2%)	.7618
MAP	.2139	.2232	**.2366**	.2306	.2196	.2003	.2531◍(+7.0%, +18.3%)	.3447
Prec@R	**.2665**	.2543	.2638	.2613	.2526	.2328	.2780°(+5.4%, +4.3%)	.3630
Prec@30	.3133	.3007	.3127	**.3147**	.3023	.2863	.3193^{-}(+2.1%, +1.9%)	.4270
Rec@1000	.5876	.6406	**.6667**	.6631	.6610	.6444	.6619^{-}(−0.7%, +12.6%)	.7274
MAP	.2187	.2288	**.2451**	.2376	.2283	.2034	.2636◍(+7.5%, +20.5%)	.3468
Prec@R	.2693	.2556	**.2697**	.2627	.2616	.2415	.2818^{-}(+4.5%, +4.6%)	.3636
Prec@30	.3223	.3123	**.3400**	.3327	.3257	.3047	.3507^{-}(+3.1%, +8.8%)	.4507
Rec@1000	.6219	.6686	.6922	.6918	**.6968**	.6867	.6969^{-}(+0.7%, +12.1%)	.7562
MAP	.2061	.2347	**.2374**	.2285	.2159	.2076	.2623◍(+10.5%, +27.3%)	.3294
Prec@R	.2655	**.2689**	.2623	.2643	.2449	.2336	.2884°(+10.0%, +8.6%)	.3487
Prec@30	.3127	**.3180**	.2967	.3007	.2847	.2793	.3200°(+7.9%, +2.3%)	.4187
Rec@1000	.5438	.5950	**.6173**	.6078	.6088	.5864	.6095^{-}(−1.3%, +12.1%)	.6914
MAP	.2195	.2506	**.2589**	.2470	.2374	.2179	.2844◍(+9.8%, +29.6%)	.3396
Prec@R	.2742	.2754	**.2770**	.2703	.2657	.2533	.3035$^{\bullet}$(+9.6%, +10.7%)	.3539
Prec@30	.3333	.3449	**.3544**	.3401	.3340	.3184	.3952°(+11.5%, +18.6%)	.4721
Rec@1000	.6216	.6611	.6780	.6751	**.6904**	.6809	.6892^{-}(+1.7%, +10.9%)	.7601
MAP	.2097	.2392	**.2416**	.2327	.2198	.2114	.2671◍(+10.6%, +27.4%)	.3354
Prec@R	.2686	**.2727**	.2649	.2673	.2474	.2357	.2916°(+10.1%, +8.6%)	.3527
Prec@30	.3150	**.3211**	.2986	.3034	.2864	.2810	.3238°(+8.4%, +2.8%)	.4238
Rec@1000	.5516	.6040	**.6261**	.6166	.6174	.5946	.6184^{-}(−1.2%, +12.1%)	.7012

SPLT1–6, respectively, i.e. 4–16% of the topics with a median of 7%.[9] Similarly, whenever the MAP of Q_0 is greater than the MAP of $Q_{r,K}$ for all K, 0/oM drops back to the effectiveness of Q_0 (1/0) for that topic. This happens 19 times in the 149 topics (12.7%), or 12, 13, 13, 7, 6, 6 times in SPLT1–6, respectively; i.e. 12–14% of the topics.

The best result per measure, across the Q_0-only (1/0) or fixed-K runs (0/5–0/50), is in boldface. Across these runs, MAP is maximized for $K = 10$ (0/10) in all experiments/splits.[10] The MAP improvements of 0/10 over 1/0 are between 10.6% and 17.9%.[11] Thus, a positive-only PRF, disregarding the initial query, can result to large improvements in effectiveness, even by using a fixed K for all queries as long as a proper K is selected; here, we confirm once more that the widely-used value of $K = 10$ gives the best MAP results.

A valuable K optimization method must outperform Q_0 (1/0), as well as all fixed K runs (0/5–0/50). We see that our proposed method (0/pM) always outperforms the best fixed

[9] These are approximate (but very close) numbers, not accounting for the excluded topic 368 whenever this is missing from a test set of a split.

[10] We tried a higher resolution for K (see lines at $\alpha = 0$ in Table 8 in Sect. 4.2.2), and $K = 10$ was still the best overall in MAP. While Table 8 seems like it shows MAP in the training sets of splits, we remind that SPLT1's training set is SPLT4's test set, and so on.

[11] These percentage improvements are resulting from Table 6, but not shown in the table due to the limited space.

Table 7 Robustness Index (RI) for MAP

	0/pM vs. 1/0	0/pM vs. 0/10	0/10 vs. 1/0
SPLT1	.23	.31	.05
SPLT2	.17	.15	.05
SPLT3	.15	.05	.05
SPLT4	.16	.32	.04
SPLT5	.42	.38	.02
SPLT6	.30	.38	.02

K run (0/10) by 7.0% to 10.6% in MAP, depending on the split (but most-affected by the amount of training data). The $K_{\text{pred\_MAP}}$ results (0/pM) are significance-tested with a bootstrap test, one-tailed, at significance levels 0.05 ($^{\circ}$), 0.01 ($^{◉}$), and 0.001 ($^{\bullet}$), against the fixed $K = 10$ run (0/10); ($^{-}$) means non-significant. The MAP improvements achieved by $K_{\text{pred\_MAP}}$ (0/pM) over Q_0 (1/0) are between 18.3% and 29.6%.

The auxiliary precision-oriented measures, Prec@R and Prec@30, also yield mostly statistically-significant improvements in tandem with MAP. In 0/pM vs. 0/10, Prec@R improves by 4.5% to 10.1% and Prec@30 by -0.2% to 11.5%, depending again on the split. These measures tend to get maximized at $K = 5$ in some splits, but we still get large improvements over these 0/5 runs. Rec@1000 does not show any significant differences and it tends to get maximized for larger fixed Ks (e.g. 0/30). These results were expected due to the fact that precision-oriented measures are correlated to MAP (MAP is also mostly sensitive to the top of the ranking), while Rec@1000 is a high-recall measure. Nevertheless, note that we still get large increases in Rec@1000 over Q_0, from 10.9 to 12.6%. In any case, all these are extra improvements, since our target has been MAP optimization where we achieve the largest and most significant improvements as we showed earlier.

Regarding the sensitivity of our method (0/pM) to the selection of training queries, we can see that SPLT1–3 show similar percentage improvements in MAP (7.0% to 7.5% over 0/10). The same goes for SPLT4–6 (9.8% to 10.6% over 0/10). Thus, our method is robust. We remind that SPLT1–3 use three different training sets of 50 queries each, while SPLT4–6, three different training sets of 100 queries each.[12]

Regarding the sensitivity of our method to the size of the training set of queries, we can see that SPLT4–6 perform better than SPLT1–3. Thus, more training queries lead to a better performance. Nevertheless, one could argue that even with just 100 training queries, we may be already seeing some diminishing returns: the MAP achieved by the proposed method is nearer to the ceiling of possible achievable effectiveness (0/oM) than to the effectiveness of Q_0 (1/0) in SPLT5. All in all, if not 100, a few hundreds of training queries may be sufficient.

Table 7 reports the RI values for MAP. The proposed PRF method (0/pM) improves a significant amount of queries over the Q_0-only run 1/0 (1st column) and the fixed $K = 10$ run 0/10 (2nd column). The third column shows the RI of the fixed $K = 10$ run over the Q_0-only run. Once again we confirm that using a fixed K value is not an effective approach as it hurts almost as many queries as it improves, while our proposed method is much more reliable.

[12] Again, not accounting for the excluded topic 368, whenever this occurs in a training set.

Table 8 MAP on the train sets of splits for different combinations of α/K

SPLT1–6		K						
		3	5	10	15	20	30	50
α	0	.2395	.2347	.2373	.2305	.2285	.2159	.2076
	0.2	.2297	.2186	.2200	.2139	.2140	.2054	.2007
	0.4	.2529	.2538	.2510	.2430	.2400	.2387	.2367
	0.5	.2529	**.2588**	.2573	.2496	.2466	.2442	.2436
	0.6	.2480	.2558	.2534	.2493	.2462	.2428	.2457
	0.8	.2305	.2357	.2325	.2321	.2296	.2311	.2322
α	0	.2541	.2506	.2589	.2518	.2470	.2374	.2179
	0.2	.2397	.2414	.2427	.2410	.2372	.2213	.2064
	0.4	.2605	.2590	**.2636**	.2588	.2542	.2521	.2358
	0.5	.2590	.2589	.2586	.2559	.2534	.2520	.2406
	0.6	.2532	.2565	.2531	.2491	.2528	.2494	.2426
	0.8	.2359	.2419	.2399	.2376	.2399	.2404	.2356
α	0	.2440	.2392	.2416	.2347	.2327	.2198	.2114
	0.2	.2340	.2229	.2239	.2178	.2180	.2091	.2042
	0.4	.2575	.2586	.2554	.2474	.2443	.2430	.2409
	0.5	.2575	**.2637**	.2618	.2541	.2511	.2486	.2480
	0.6	.2525	.2606	.2579	.2538	.2507	.2471	.2501
	0.8	.2346	.2400	.2366	.2362	.2337	.2352	.2363
α	0	.2357	.2310	.2473	.2460	.2398	.2303	.2052
	0.2	.2210	.2177	.2262	.2295	.2303	.2210	.1966
	0.4	.2479	.2467	.2622	.2608	.2553	.2470	.2278
	0.5	.2472	.2511	**.2666**	.2640	.2598	.2483	.2353
	0.6	.2464	.2518	.2615	.2593	.2575	.2445	.2360
	0.8	.2352	.2410	.2446	.2441	.2416	.2390	.2338
α	0	.2286	.2232	.2366	.2354	.2306	.2196	.2003
	0.2	.2162	.2065	.2150	.2160	.2188	.2131	.1938
	0.4	.2442	.2442	.2559	.2529	.2482	.2404	.2283
	0.5	.2443	.2512	**.2659**	.2608	.2563	.2445	.2369
	0.6	.2439	.2516	.2615	.2593	.2541	.2415	.2377
	0.8	.2325	.2379	.2408	.2413	.2364	.2344	.2321
α	0	.2336	.2288	.2451	.2437	.2376	.2283	.2034
	0.2	.2190	.2156	.2242	.2274	.2282	.2190	.1949
	0.4	.2457	.2445	.2599	.2584	.2530	.2448	.2258
	0.5	.2450	.2488	**.2643**	.2617	.2575	.2462	.2333
	0.6	.2442	.2496	.2592	.2570	.2552	.2423	.2340
	0.8	.2332	.2389	.2425	.2420	.2395	.2369	.2318

To conclude, our proposed PRF method, i.e. disregarding the contribution of the initial query and optimizing the number of pseudo relevant documents (K) per query, is a viable, robust, and effective method. It yields significant improvements both over the initial query and positive-only PRF with a fixed K for all queries.

Table 9 Effectiveness of PRF with standard MAP optimization (std-pM), against the pure proposed method (0/pM) and when retaining a 50% contribution of the initial query (0.5/pM). The ceiling of std-pM is std-oM, when the optimal fixed α/K for a test set is found and used

SPLT1–6	run name or blind feedback parameters α/K:			
	std-pM	0/pM (vs. std-pM)	.5/pM (vs. std-pM, vs. 0/pM)	std-oM
MAP	.2511	$.2648^{\circ}$(+5.5%)	$.2691^{\color{gray}\bullet}$(+7.2%, +1.6%)	.2666
Prec@R	.2844	$.2873^{-}$(+1.0%)	$.2962^{\circ}$(+4.1%, +3.1%)	.2962
Prec@30	.3468	$.3407^{-}$(−1.8%)	$.3566^{-}$(+2.8%, +4.7%)	.3576
Rec@1000	.6947	$.7031^{-}$(+1.2%)	$.7085^{-}$(+2.0%, +0.7%)	.7053
MAP	.2559	$.2531^{-}$(−1.1%)	$.2613^{-}$(+2.1%, +3.2%)	.2659
Prec@R	.2846	$.2780^{-}$(−2.3%)	$.2911^{-}$(+2.3%, +4.7%)	.2991
Prec@30	.3263	$.3193^{-}$(−2.1%)	$.3371^{-}$(+3.3%, +5.6%)	.3367
Rec@1000	.6813	$.6619^{-}$(−2.8%)	$.6750^{-}$(−0.9%, +2.0%)	.6821
MAP	.2488	$.2636^{\circ}$(+5.9%)	$.2688^{\color{gray}\bullet}$(+8.0%, +2.0%)	.2643
Prec@R	.2828	$.2818^{-}$(−0.4%)	$.2982^{\circ}$(+5.4%, +5.8%)	.2946
Prec@30	.3450	$.3507^{-}$(+1.7%)	$.3620^{-}$(+4.9%, +3.2%)	.3567
Rec@1000	.6892	$.6969^{-}$(+1.1%)	$.7040^{\circ}$(+2.1%, +1.0%)	.7001
MAP	.2573	$.2623^{-}$(+1.9%)	$.2660^{\circ}$(+3.4%, +1.4%)	.2588
Prec@R	.2940	$.2884^{-}$(−1.9%)	$.3005^{-}$(+2.2%, +4.2%)	.2943
Prec@30	.3167	$.3200^{-}$(+1.0%)	$.3240^{-}$(+2.3%, +1.2%)	.3340
Rec@1000	.6371	$.6095^{\circ}$(−4.3%)	$.6334^{-}$(−0.6%, +3.9%)	.6275
MAP	.2586	$.2844^{\color{gray}\bullet}$(+10.0%)	$.2789^{\bullet}$(+7.8%, −1.9%)	.2636
Prec@R	.2879	$.3035^{-}$(+5.4%)	$.3071^{\color{gray}\bullet}$(+6.7%, +1.2%)	.2917
Prec@30	.3585	$.3952^{\circ}$(+10.2%)	$.3966^{\bullet}$(+10.6%, +0.3%)	.3585
Rec@1000	.6831	$.6892^{-}$(+0.9%)	$.6950^{-}$(+1.7%, +0.8%)	.6897
MAP	.2618	$.2671^{-}$(+2.0%)	$.2702^{\circ}$(+3.2%, +1.2%)	.2637
Prec@R	.2971	$.2916^{-}$(−1.9%)	$.2995^{-}$(+0.8%, +2.7%)	.2978
Prec@30	.3177	$.3238^{-}$(+1.9%)	$.3245^{-}$(+2.1%, +0.2%)	.3374
Rec@1000	.6464	$.6184^{-}$(−4.3%)	$.6434^{-}$(−0.5%, +4.0%)	.6369

4.2.2 Retaining initial query with fixed or predicted *K*

Table 8 shows the MAP on the training sets of splits for different combinations of α/K; the maximum MAP per split is in boldface. What we call the *standard PRF optimization method for MAP* (std-pM) would use in the test set the best combination of α/K found in the training set, i.e. 0.5/5 for SPLT1, 0.4/10 for SPLT2, and so on. It can be seen that, in general across all splits, MAP is maximized between 0.4–0.5 for α and 5–10 for K, with 0.5/10 being overall the best combination.

Table 9 shows the results achieved by the standard method (std-pM) on the test sets of the splits. The 0/pM column re-iterates the results from Table 6 of our proposed method, but this time they are compared and statistically-tested for significance against the standard method. It can be seen that, even by eliminating the initial query, our proposed method outperforms the standard method in 5 out of 6 splits (significantly in 3 of those 5) in MAP. The only decrease in MAP is in SPLT2, and there are some decreases in the auxiliary evaluation measures (which we do not try to optimize anyway), but none of them is significant. Thus, the improvements in MAP range from -1.1% to +10.0% with a median/mean of +3.7%/+4.0%.

Table 10 Robustness Index (RI) for MAP

	0/pM vs. std-pM	.5/pM vs. std-pM
SPLT1	.22	.26
SPLT2	−.22	.14
SPLT3	.14	.14
SPLT4	.14	.34
SPLT5	.22	.36
SPLT6	.14	.20

While we are satisfied with these improvements, we are also interested to see what happens if we also use some contribution of the initial query in our model. For this purpose, we did an extra run using the best overall $\alpha = 0.5$ (according to Table 8) but with building the positive feedback component with our model/method; these results are shown in the .5/pM column and compared against the standard method (std-pM). This time, there are MAP improvements in all splits, ranging from +2.1% to +8.0% with a median and mean of +5.3%, significant in 5 out of 6 splits. The auxiliary measures also mostly improve, some significantly so.

To conclude, our proposed model outperforms the standard PRF optimization method. Nevertheless, there are still further improvements to be gained—albeit small ones—by using some contribution of the initial query. While our positive-only feedback model is built to be optimal without a contribution of the initial query, it still performs a bit better in 5 out of 6 splits (from +1.2% to +3.2% in MAP) when 50% of such a contribution is used. These small improvements are mostly statistically insignificant, however, it seems that there is some robustness to be gained, as it can been seen in Table 10.

On a final note, the right-most column of Table 9 (std-oM) shows the potential effectiveness ceiling of the standard optimization method. For this, we grid-searched for the α/K combination that maximizes MAP, but this time directly in each test set. It can be seen that the results achieved by the standard method (std-pM) are very near to their ceiling (std-oM), thus there is not much potential left for improvements. In contrast, even our best runs so far (0/pM and 0.5/pM) are far from their potential ceiling (0/oM column in Table 6), leaving amble room for improvement.

4.3 Additional experiments

In this section, first we will confirm the effectiveness of our proposed method on an additional benchmark corpus, and then compare it to PRF methods from the recent literature on several other setups.

4.3.1 Experiments with web data

Table 11 presents our results on the WT10g test collection. We applied the same preprocessing pipeline as in (Parapar and Barreiro 2011; Parapar et al. 2014; Valcarce et al. 2018), i.e. stopword removal and Porter Stemmer, as well as used the same split, i.e.

Table 11 Effectiveness of positive-only PRF — WT10g corpus

WT10g	blind feedback parameters α/K:							
	1/0	0/5	0/10	0/20	0/30	0/50	0/pM (vs. 0/5, vs. 1/0)	0/oM
MAP	.1940	**.2443**	.2365	.2225	.2319	.2123	.2616°(+7.1%, +34.9%)	.3410
Prec@R	.2444	.2521	**.2649**	.2509	.2547	.2417	.2896°(+14.9%, +18.5%)	.3734
Prec@30	.2820	.2961	.3027	.2893	**.3047**	.2963	.3060^{-}(+3.3%, +8.5%)	.3880
Rec@1000	.7233	**.7997**	.7926	.7816	.7876	.7810	.7900^{-}(−1.2%, +9.2%)	.8452

Table 12 Effectiveness of PRF with standard MAP optimization (std-pM), against the pure proposed method (0/pM) and when retaining a 50% contribution of the initial query (0.5/pM). The ceiling of std-pM is std-oM, when the optimal fixed α/K for the test set is found and used

WT10g	run name or blind feedback parameters α/K:			
	std-pM	0/pM (vs. std-pM)	.5/pM (vs. std-pM, vs. 0/pM)	std-oM
MAP	.2469	.2616°(+5.9%)	.2477^{-}(+0.3%, −5.3%)	.2492
Prec@R	.2598	.2896°(+11.5%)	.2725°(+4.9%, −5.9%)	.2594
Prec@30	.2888	.3060°(+5.9%)	.3078^{-}(+6.6%, +0.6%)	.3007
Rec@1000	.7824	.7900^{-}(+0.9%)	.7876^{-}(+0.7%, −0.3%)	.7926

trained our model using TREC topics 451–500 and evaluated using topics 501–550. Again, we optimized for MAP.

The best fixed-K for MAP is 5 (0/5 run); against this run as a baseline, the proposed method (0/pM) yields statistically significant results for MAP and Prec@R. The MAP improvement (+7.1%) is in-line with what we reported earlier in Table 6 when training with 50 queries only. While the Prec@R improvement is much larger in the WT10g corpus, this also has to do with the fact that each measure is maximized at a different fixed-K value. Against the initial query (1/0), the proposed PRF method yields a larger improvement (+34.9%) than the best we saw before in Table 6 (+29.6%). These results confirm the effectiveness of our method, strengthening the evidence. However, the proposed method is quite far from its potential ceiling of effectiveness (0/oM column), leaving amble room for improvements.

We run the standard PRF optimization method for MAP (std-pM), as in Sect. 4.2.2, and found that MAP is maximized at .5/5; these results are shown in the 1st column of Table 12. Our proposed method (0/pM) significantly outperforms std-pM. Moreover, std-pM is practically at its potential ceiling of performance (std-oM column) resulting when using the optimal parameters found in the *test* set (i.e. .6/5). Thus, these results so far confirm our findings in Sect. 4.2.2. But now, using a fixed 50% contribution of the initial query together with the variable Ks predicted by our proposed method (i.e. .5/pM run), is generally worse than the pure proposed method (0/pM). This is in contrast to the small insignificant improvements we found in Sect. 4.2.2, supporting our conjecture of initial query elimination.

Table 13 provides the RIs of some run comparisons from Tables 11–12. All RIs confirm our commentary above, except a single surprise: despite the overall worse performance (-5.3% in MAP) of .5/pM compared to 0/pM, .5/pM yields a larger RI over the std-pM.

Table 13 Robustness Index (RI) for MAP — WT10g corpus

	RI
0/5 vs. 1/0	.12
0/pM vs. 1/0	.38
0/pM vs. 0/5	.32
0/pM vs. std-pM	.22
.5/pM vs. std-pM	.42
0/pM vs. .5/pM	.26

Table 14 MAP-based comparisons to previous works on TS50, TS100, and WT10g testbeds

	LM	RM3	best-perf-PRF	1/0	0/pM	(vs. RM3,	vs. best)	0/oM
TS50	.1915	.2194	.2245 (SDRM3)	.1882	.2718	(+24.0%,	+21.0%)	.3376
	.1931	.2235	.2357 (LiMe-TF-IDF)			(+21.6%,	+15.3%)	
TS100	.2190	.2589	.2700 (RM3DT)	.2150	.2736	(+5.7%,	+1.3%)	.3517
WT10g	.2182	.2468	.2478 (RM3DT)	.1940	.2616	(+6.0%,	+5.6%)	.3410
	.2182	.2402	.2322 (SDRM3)			(+8.9%,	+12.7%)	
	.2194	.2470	.2484 (LiMe-TF)			(+5.9%,	+5.3%)	

This surprise is in line with Valcarce et al. (2018), who also found that while their proposed method was the most effective it did not achieve the best RI value. They attributed this to the noisy nature of the WT10g collection. This suggests again, as in Sect. 4.2.2, that there may be at least some robustness to be gained when using some contribution of the initial query, but this time it comes at a cost of lower average effectiveness. On the other hand, rather weirdly, the RI is positive when comparing 0/pM to .5/pM.

4.3.2 Comparison to the state-of-the-art

Let us now compare to other state-of-the-art PRF methods from the literature, specifically, the RM3 model which is one of the most effective PRF methods based on the language modeling framework (see e.g. Lv and Zhai (2009b, 2010)), and the methods of Parapar and Barreiro (2011); Parapar et al. (2014); Valcarce et al. (2018). We reproduce the experimental setups of these works on the TREC Volumes 4 and 5 (minus the Congressional Record) collection. While this is the same corpus we also used in the previous sections, other studies considered different training/testing splits, namely, TREC topics 301–350 for training and 351–400 or 351–450 for testing; we refer to these setups/splits as TS50 and TS100, respectively. On the WT10g corpus, the aforementioned works used the setup we described earlier in this section.

In Table 14, the first three columns present the MAP reported in the aforementioned works for LM, RM3, and the best-performing PRF method of the ones proposed in each of the studies. On TS50, SDRM3 is the best in Parapar et al. (2014), and LiMe-TF-IDF is the best in Valcarce et al. (2018). On TS100, RM3DT is the best in Parapar and Barreiro (2011). On WT10g, RM3DT is the best in Parapar and Barreiro (2011), SDRM3 is the best in Parapar et al. (2014), and LiMe-TF is the best in Valcarce et al. (2018). As a retrieval model for the initial retrieval, the baseline Language Modelling

Table 15 Robustness Index (RI) for MAP

	0/pM vs. 1/0	best-perf-PRF vs. LM
TS50	.30	.29 (SDRM3) .46 (LiMe-TF-IDF)
TS100	.37	.38 (RM3DT)
WT10g	.46	.36 (RM3DT) .12 (SDRM3) .32 (LiMe-TF)

(LM) with Dirichlet smoothing was used. The RM3 model was well-tuned; using topics 301–350 for training, the optimal values of μ, e, r, λ reported in Parapar et al. (2014) were 500, 100, 10, 0.2, respectively. The λ value suggests that the contribution of the initial query should be high (since $\alpha = 1 - \lambda = 0.8$) for the TS50 and TS100 benchmark datasets. For the WT10g dataset, the values reported for μ, e, r, λ were 500, 10, 5, 0.6, respectively, suggesting a more balanced contribution.

Table 14 also presents our results of the initial query (1/0) as well as our proposed method (0/pM) on those three setups. Note that our initial query run underperforms LM in all three testbeds (we use Terrier's default/untuned language model, i.e. the inverse document frequency model for randomness (InL2) (Amati and van Rijsbergen 2002)). Nevertheless, starting even from such a worse initial retrieval, our proposed PRF method outperforms both RM3 and the best-performing PRF method from previous literature in all testbeds, in some cases by far. The outperformance over literature's best runs ranges from 1.3% to 21.0% with an average of 10.2%. Table 15 shows the RI values of our method and of literature's best-performing methods against the initial query runs (either 1/0 or LM). Our RIs are larger in 4 out of 6 cases, while they are overall high. Last, according to our method's potential ceiling of effectiveness (0/oM), there is amble room for further improvements.

Once more, as explained in Section 3.2, we used the TPOT tool, developed by Olson et al. (2016), to determine the optimal machine learning pipelines for our method. For completeness, Table 16 lists the obtained parameters, which are similar and in-line with the previous experiments reported in Section 4.2. The transformation methods obtained by the grid search are L2 Normalization, StandardScaling, PolynomialFeatures of the input data, and kernel-based methods such as KernelPCA, FastICA, and RBFsampler.

5 Discussion

The empirical evaluation has shown that the proposed method is robust and effective. Here, we will look deeper into some data in order to gain more insight on why it works.

Table 16 Processing pipelines produced by TPOT for TS50, TS100 & WT10g

	Pipelines	Parameters
TS50	KernelPCA	(alpha=1.0, coef0=1,copy_X=True,degree=3,eigen_solver=’auto’, gamma=’scale’,kernel=’linear’, random_state=28)
	RBFSampler	(gamma=4.3,n_components=22,random_state=28)
	LinearSVR	(C=5.0, dual=True,epsilon=0.01,fit_intercept=True, intercept_scaling=15, loss=’epsilon_insensitive’,max_iter=1000, random_state=28,tol=1e-05,verbose=0)
TS100	StandardScaler	(copy=True,with_mean=True,with_std=False)
	PolynomialFeatures	(degree=2, include_bias=False,interaction_only=True,order=’F’)
	LinearSVR	(C=25.0,dual=True,epsilon=0,fit_intercept=True, intercept_scaling=25,loss=’epsilon_insensitive’,max_iter=1000, random_state=28,tol=0.001,verbose=0)
WT10G	Normalizer	(copy=True,norm=’l2’)
	FastICA	(algorithm=’parallel’,fun=’logcosh’,fun_args=None, max_iter=200,n_components=27,random_state=28,tol=0.0001, w_init=None, whiten=True)
	LinearSVR	(C=25.0,dual=True,epsilon=0.1,fit_intercept=True, intercept_scaling=20,loss=’epsilon_insensitive’ max_iter=1000,random_state=28,tol=0.0001,verbose=0)

Table 17 Correlation of $K_{\text{opt\_MAP}}$ to MAP@Q_0, R, MAP@$Q_{r,K_{\text{opt\_MAP}}}$

$K_{\text{opt\_MAP}}$ corr.	MAP@Q_0	R	MAP@$Q_{r,K_{\text{opt\_MAP}}}$
Pearson	−0.220●	0.196°	−0.225◉
Spearman	−0.160°	0.189°	−0.108^{-}
Kendall	−0.109°	0.127°	−0.073^{-}

Table 18 Correlation of R to QPPs and Q_0's effectiveness

R corr.	WIG	NQC	SMV	MAP@Q_0
Pearson	-0.208°	-0.142^{-}	-0.132^{-}	−0.183°
Spearman	-0.229°	-0.321°	-0.260°	−0.112^{-}
Kendall	-0.159°	-0.222°	-0.180°	−0.080^{-}

5.1 Optimal *K* and query performance

In Table 3, we saw some worrying signs (i.e. although non-significant, all correlations are negative) that may indicate a flaw in half of our argument at the beginning of Sect. 3: *positive correlations between the optimal K and both R and Q_0 effectiveness are expected.* Since Table 3 measures the correlation of $K_{\text{opt\_MAP}}$ to QPPs and not directly to the MAP of Q_0, we provide Table 17 in order to determine whether our argument holds or not. Table 17 confirms the first part of the argument (i.e. the larger the R, the larger the optimal K), but shows an anti-correlation between $K_{\text{opt\_MAP}}$ and MAP@Q_0 (and therefore good QPPs). Since, as we argued in Sect. 3, the effectiveness of Q_0 comes to 'correct' the optimal K, moving it further away from R to lower values the more difficult the Q_0 is, we investigate R further.

Table 18 shows the correlation of R to QPPs and Q_0's effectiveness. These are all negative correlations, mostly statistically significant. This appears to be counter-intuitive, since among the easiest topics there are many which possess a small number of relevant documents, and many difficult topics have many relevant documents. Amati et al. (2004) noticed this before, and attributed it to topic/query generality with respect to the collection. Specific queries have few relevant documents, their query terms have few occurrences in the collection, thus their relevant documents are easier to find/discriminate.

Consequently, what we think is happening with our argument/model is the following. We detect effective queries via QPPs (Table 2). These happen to have small R (Table 18), plausibly due to the argument of Amati et al. (2004) mentioned above, generating a negative correlation between QPPs and R. Since, as we argued, the optimal K should be smaller than R,[13] the negative correlation is also transferred to between QPPs and optimal K, making our argument look flawed while it is not. There is a positive correlation between QPPs and optimal K, which is almost totally overridden (Table 3) by the small R of easy queries and large R of the difficult ones—a much stronger correlation. Indeed, when we measure the correlation between $K_{\text{opt\_MAP}}$ and QPPs but only for queries with R in a tight range, it turns positive most often. Therefore, the latter half of our argument should be better re-formulated as: a positive correlation between the optimal K and Q_0 effectiveness is expected *for topics with a similar R.*

[13] There are 116 out of the 150 queries (77.3%) with $K_{\text{opt\_MAP}} \leq R$ in our dataset. Also note that the Rs are not the real ones but under-estimated—in various degrees—by TREC's pooling process.

Although we decided to focus on Q_0's effectiveness in this study, R has turn out to be a very important variable too. Luckily, QPPs predict R also (Table 18) via the discussed anti-correlation, which seems to have helped our regression model considerably.

5.2 Loss functions for model selection

In Section 3.2 it was stated that we use the mean absolute error (MAE) between the observed optimal K (K_{opt}) and our forecast (K_{pred}) as a loss function for model selection in training the regression model. Further experiments revealed that the choice of loss function is critical in our training method, determining its success or failure.

The absolute error (AE) is defined as $\text{AE}(K_{\text{opt}}, K_{\text{pred}}) = |K_{\text{opt}} - K_{\text{pred}}|$. Theoretically, however, AE is not the right choice in the context of PRF. Its problem is that it penalizes equally a certain AE in forecast irrespective of the magnitude of the observed value, e.g. AE(100,105) = AE(10,15). In PRF, the former error is expected to have a less dramatic impact in PRF effectiveness than the latter. By treating those two cases equally, using MAE for model selection in PRF produces systematically larger-than-desirable forecasts.

An alternative is the mean absolute percentage error (MAPE). The absolute percentage error is defined as $\text{APE}(K_{\text{opt}}, K_{\text{pred}}) = |(K_{\text{opt}} - K_{\text{pred}})/K_{\text{opt}}|$. APE normalizes AE's unsuitable behaviour by measuring percentage differences. However, it puts a heavier penalty on negative errors, i.e. $K_{\text{opt}} < K_{\text{pred}}$, than on positive ones. For example, APE(5,10)=100%, but APE(10,5)=50%. As a consequence, when MAPE is used to compare the accuracy of prediction methods, it is biased in that it will systematically select a method whose forecasts are too low. While this is typically considered as a drawback in the literature, it is a desirable bias in our task since negative errors are expected to decrease the density of relevant documents in the PRF set while positive errors are expected to increase it. Still, note that APE(10,5)=50% but APE(10,15)=50% also (i.e. APE is symmetric on the percentage scale), while, based again on our relevant document density argument, the former error is preferable.

Other error variants, such as the Adjusted/Symmetric MAPE (SMAPE), seem even more unsuitable. SMAPE is the mean of the symmetric APE (sAPE), which is defined as $\text{sAPE}(K_{\text{opt}}, K_{\text{pred}}) = |K_{\text{opt}} - K_{\text{pred}}|/((|K_{\text{opt}}| + |K_{\text{pred}}|)/2)$. The sAPE is asymmetric (despite its name) on the percentage scale, e.g. sAPE(10,5)= 66.6% but sAPE(10,15)=40%, but its asymmetry is the opposite from the desirable: the former error is preferable. Moreover, it eliminates (as it is designed to do so) APE's desirable bias, e.g. sAPE(5,10) and sAPE(10,5) are now equal. A measure proposed by Tofallis (2015) is the log of the accuracy ratio (LAR), which in our case it should be taken as an absolute value (ALAR). It is defined as $\text{ALAR}(K_{\text{opt}}, K_{\text{pred}}) = |\log(K_{\text{pred}}/K_{\text{opt}})| = |\log K_{\text{pred}} - \log K_{\text{opt}}|$. The mean of ALAR (MALAR) can be used as a loss function. Inspecting its properties: ALAR(100,105) < ALAR(10,15) (desirable), ALAR(5,10) = ALAR(10,5) (undesirable), ALAR(10,5) > ALAR(10,15) (undesirable). Therefore, among the MAE, MAPE, SMAPE, and MALAR, qualitatively more suitable for our task (although still not the ideal) is the MAPE.

A drawback of MAPE (as well as SMAPE and MALAR) is that it cannot be used if there are observed values (and/or predicted values for SMAPE and MALAR) equal to zero because there would be a division by zero (or a log of zero); we have a few of such zeros in

Table 19 Ranking of splits by their test-set loss

MAE	MAPE	SMAPE	MALAR	MREE	0/oM−0/pM
6	5	6	4	5	5
2	6	4	6	4	4
1	4	1	5	6	6
4	3	5	1	3	3
3	1	2	3	1	1
5	2	3	2	2	2

our problem. Nevertheless, since we are not interested in (or going to interpret) the actual MAPE value but use MAPE as a loss function, all observed and predicted data could be shifted by adding a small positive value ϵ without any material impact to our use. Thus, we shifted with $\epsilon = 0.5$.

In the discussion above, we have focused on how to best measure the error between optimal and predicted K. In PRF, however, we are ultimately interested in the impact this error has on retrieval effectiveness, not in the error itself. In this respect, the most suitable loss function for our problem is the mean of retrieval effectiveness error (MREE). For MAP, the retrieval effectiveness error is defined as $\mathrm{REE}(K_{\mathrm{opt\_MAP}}, K_{\mathrm{pred\_MAP}}) = \mathrm{MAP@}Q_{r,K_{\mathrm{opt\_MAP}}} - \mathrm{MAP@}Q_{r,K_{\mathrm{pred\_MAP}}}$, with a drop-back to MAP@Q_0 to any of the two MAPs if $K_{\mathrm{opt\_MAP}} = 0$ or $K_{\mathrm{pred\_MAP}} \leq 0$. Minimizing MREE minimizes the average distance from the potential ceiling of effectiveness, i.e. the effectiveness distance between the 0/oM and 0/pM runs in Table 6.

We experimented with all the above loss functions. While the MREE is the theoretically correct one, it produces unstable fits across the splits leading to overall worse effectiveness. From the ones focusing on measuring the K-error, MAPE—which is the most suitable but still not perfect—has exactly the same problem as MREE. The rest, SMAPE, MALAR, and MAE, while they do not have the properties desired by the task, they are more forgiving, with the MAE being the most robust and performing the best.

This counter-intuitive behavior of loss functions can be attributed to the training data: the distribution of $K_{\mathrm{opt\_MAP}}$ in our training sets is skewed to the downside (as resulting from the numbers in the first paragraph of Sect. 3.2). This skew is sufficient to produce the desirable under-forecasts, and when it is combined with the additional under-forecasting of MAPE it becomes too much, leading to an excessive number of forecasts with $K_{\mathrm{pred\_MAP}} <= 0$ which degenerate the method. A similar thing may be happening with MREE. Table 17 shows an anti-correlation between the optimal K and the effectiveness of the positive-only feedback query built at the optimal K, meaning that the most effective feedback happens at small Ks. Since there are many more small $K_{\mathrm{opt\_MAP}}$ than large ones, TPOT focuses overly at low forecasts. However, at large MAPs, diminishing returns kick in: while one can achieve an absolute increase of +0.25 when MAP is e.g. at 0.05, he can only achieve at best +0.20 when MAP is already e.g. at 0.80; inversely, while one can loose -0.20 from 0.80, he can only loose at worst -0.05 from 0.05. The *macro-averaged* MAP we evaluate with is sensitive to these absolute differences.

As an indirect proof that we are right in our analysis above, Table 19 ranks the splits by their effectiveness in the test set as measured by the different loss functions. The right-most column (0/oM−0/pM) ranks the splits with the difference in the mean MAP between the ceiling and our method (minimizing this difference is the target). Obviously, this is exactly the same as the MREE, since the difference of means (0/oM−0/pM) equals the mean of the differences (MREE); therefore, the MREE is optimal. The MAPE is the second best (or the best among the ones focusing on K-errors rather than MAP-errors), with only a single permutation of adjacent splits from the optimal ranking. Note that the good functions rank splits 4–6 above splits 1–3; since splits 4–6 have double the amount of training data than splits 1–3, smaller errors are achieved in the test sets of the former splits.

All in all, one should keep in mind that the theoretical optimal loss function for the task is the MREE, while the MAPE also has most of the desirable properties. Given a larger and more *balanced* training set of queries (i.e. with more uniformly distributed K_{opt} for the effectiveness measure of interest), the MREE or MAPE are the loss functions one should employ. For our few and unbalanced training data, the simplest option of MAE has been the most forgiving, robust, and effective.

6 Conclusions & directions for further research

We have proposed a method for automatic optimization of pseudo relevance feedback (PRF) in information retrieval. Based on the conjecture that the initial query's contribution to the final/feedback query may not be necessary once a good model is built from pseudo relevant documents, we have set out to optimize—per query—*only* the number of top-retrieved documents to be used for feedback. The optimization has been based on several post-retrieval query performance predictors for the initial query by building a linear regression model. The regression model itself has been optimized via genetic programming by intelligently exploring thousands of possible machine learning pipelines to find the best one for the data at hand.

The approach requires training data. Experiments on several train/test splits of standard TREC benchmark corpora have shown that even by using only 50–100 training queries, the method yields statistically-significant improvements in MAP of 18–35% over the initial query, 7–11% over the positive-only feedback model with the best fixed number of pseudo-relevant documents, and up to 10% (5.5% on median) over the standard method of optimizing *both* most-important PRF parameters (i.e. the initial query's contribution/weight *and* the number of feedback documents) by exhaustive/grid search in the training set. Compared to state-of-the-art PRF methods from the recent literature, our method outperforms by up to 21.0% with an average of 10%. Moreover, the method does not seem very sensitive to the selection of training queries, although it may benefit from an increased number of them. While the training phase may be computationally heavy, the prediction phase is quite fast and usable in query-time. This has to do with the choice of query performance predictors which are easy and fast to calculate.

A further analysis of the experimental results has shown that we are still far from the method's effectiveness ceiling (in contrast to the standard method which seems to have reached its saturation point), leaving amble room for further improvements in several directions, notably: tuning of the query performance predictors' parameters (we have merely used standard values recommended in the literature), more query performance predictors (potentially also pre-retrieval ones), more (and more balanced) training data. Additionally, based on our theoretical arguments, the method may benefit from using predictors for the number of relevant documents, but this is quite a problem on its own.

Additional improvements, which extend beyond the explored conjecture, could be to try to optimize *both* most-important PRF parameters, employing the proposed process. The experimental results have shown that there may still be some benefit from using some contribution from the initial query, perhaps in terms of improvements in robustness rather than in average MAP; nevertheless, we believe that this is due to not yet utilizing the proposed method's potential in its entirety. In any case, optimizing two parameters is more difficult than optimizing just one, and may require super-linearly more training data. As a middle ground, one could assume a fixed contribution of the initial query, e.g. 20%, take it into account when training the regression model, and go for the other parameter with our proposed method.

References

Abdelmgeid Amin, A. (2008). Using a query expansion technique to improve document retrieval. *International Journal Information Technologies and Knowledge, 2*(4), 343–348.

Amati, G., & van Rijsbergen, C. J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS), 20*(4), 357–389.

Amati, G., Carpineto, C., & Romano, G. (2004). Query difficulty, robustness, and selective application of query expansion. In *Advances in Information Retrieval, 26th European Conference on IR Research, ECIR 2004, Proceedings, Springer, Lecture Notes in Computer Science, 2997*, pp 127–137

Arampatzis, A. (2001). Unbiased S-D threshold optimization, initial query degradation, decay, and incrementality, for adaptive document filtering. In *Proceedings of The Tenth Text REtrieval Conference, TREC 2001, National Institute of Standards and Technology (NIST), 250*, pp 596–603.

Arampatzis, A., & Robertson, S. (2011). Modeling score distributions in information retrieval. *Information Retrieval, 14*(1), 26–46.

Arampatzis, A., Beney, J., Koster, CHA., & van der Weide, TP. (2000). Incrementality, half-life, and threshold optimization for adaptive document filtering. In *Proceedings of The Ninth Text REtrieval Conference, TREC 2000, National Institute of Standards and Technology (NIST)*, NIST Special Publication, *249*.

Arampatzis, A., Kamps, J., & Robertson, S. (2009). Where to stop reading a ranked list?: threshold optimization using truncated score distributions. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009*, ACM, pp 524–531.

Azad, H. K., & Deepak, A. (2019). Query expansion techniques for information retrieval: A survey. *Information Processing & Management, 56*(5), 1698–1735.

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

Cronen-Townsend, S., Zhou, Y., & Croft, WB. (2002). Predicting query performance. In: SIGIR 2002: *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, pp 299–306.

Crouch, C. J., Crouch, D. B., Chen, Q., & Holtz, S. J. (2002). Improving the retrieval effectiveness of very short queries. *Information Processing & Management, 38*(1), 1–36.

Hauff, C. (2010). Predicting the effectiveness of queries and retrieval systems. *SIGIR Forum, 44*(1), 88.

Jaleel, NA., Allan, J., Croft, WB., Diaz, F., Larkey, LS., Li, X., Smucker, MD., & Wade, C. (2004). Umass at TREC 2004: Novelty and HARD. In *Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, National Institute of Standards and Technology (NIST)*, NIST Special Publication, vol 500-261.

Karisani, P., Rahgozar, M., & Oroumchian, F. (2016). A query term re-weighting approach using document similarity. *Information Processing & Management, 52*(3), 478–489.

Kekäläinen, J., & Järvelin, K. (1998). The impact of query structure and query expansion on retrieval performance. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, pp 130–137.

Lavrenko, V., & Croft, WB. (2001). Relevance-based language models. In *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, pp 120–127.

Lv, Y., & Zhai, C. (2009a). Adaptive relevance feedback in information retrieval. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM 2009, ACM, pp 255–264.

Lv, Y., & Zhai, C. (2009b). 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 2009, ACM, pp 1895–1898.

Lv, Y., & Zhai, C. (2010). Positional relevance model for pseudo-relevance feedback. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2010, ACM, pp 579–586.

Lv, Y., & Zhai, C. (2014). Revisiting the divergence minimization feedback model. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM 2014, ACM, pp 1863–1866.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval.* Cambridge University Press.

Markovits, G., Shtok, A., Kurland, O., & Carmel, D. (2012). Predicting query performance for fusion-based retrieval. In *21st ACM International Conference on Information and Knowledge Management*, CIKM'12, ACM, pp 813–822.

Mitra, M., Singhal, A., & Buckley, C. (1998). Improving automatic query expansion. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, pp 206–214.

Olson, RS., Bartley, N., Urbanowicz, RJ., & Moore, JH. (2016). Evaluation of a tree-based pipeline optimization tool for automating data science. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*, ACM, pp 485–492.

Parapar, J., & Barreiro, A. (2011). Promoting divergent terms in the estimation of relevance models. In *Advances in Information Retrieval Theory - Third International Conference*, ICTIR 2011. Proceedings, Springer, Lecture Notes in Computer Science, vol 6931, pp 77–88.

Parapar, J., Quindimil, M. A. P., & Barreiro, A. (2014). Score distributions for pseudo relevance feedback. *Information Sciences, 273,* 171–181.

Raiber, F., & Kurland, O. (2014). Query-performance prediction: setting the expectations straight. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, ACM, pp 13–22.

Rocchio, JJ. (1971). Relevance feedback in information retrieval. In *The SMART Retrieval System - Experiments in Automatic Document Processing*, Prentice Hall, pp 313–323.

Rosipal, R., Girolami, M. A., Trejo, L. J., & Cichocki, A. (2001). Kernel PCA for feature extraction and de-noising in nonlinear regression. *Neural Computing & Applications, 10*(3), 231–243.

Ruthven, I., & Lalmas, M. (2003). A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review, 18*(2), 95–145.

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

Salton, G. (1971). *The SMART Retrieval System-Experiments in Automatic Document Processing*. USA: Prentice-Hall Inc.

Shtok, A., Kurland, O., Carmel, D., Raiber, F., & Markovits, G. (2012). Predicting query performance by query-drift estimation. *ACM Transactions on Information Systems (TOIS) 30*(2):11:1–11:35.

Sihvonen, A., & Vakkari, P. (2004). Subject knowledge improves interactive query expansion assisted by a thesaurus. *Journal of Documentation, 60*(6), 673–690.

Singh, J., Prasad, M., Daraghmi, Y., Tiwari, P., Yadav, P., Bharill, N., Pratama, M., & Saxena A. (2017). Fuzzy logic hybrid model with semantic filtering approach for pseudo relevance feedback-based query expansion. In *2017 IEEE Symposium Series on Computational Intelligence*, SSCI 2017, IEEE, pp 1–7.

Tao, T., & Zhai, C. (2006). Regularized estimation of mixture models for robust pseudo-relevance feedback. In S*IGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, pp 162–169.

Tao, Y., & Wu, S. (2014). Query performance prediction by considering score magnitude and variance together. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM 2014, ACM, pp 1891–1894.

Tofallis, C. (2015). A better measure of relative prediction accuracy for model selection and model estimation. *Journal of the Operational Research Society, 66*(8), 1352–1362.

Valcarce, D., Parapar, J., & Barreiro, Á. (2018). Lime: linear methods for pseudo-relevance feedback. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, SAC 2018, Pau, ACM, pp 678–687.

Voorhees, EM., & Harman, D. (1999). Overview of the eighth text retrieval conference (TREC-8). In *Proceedings of The Eighth Text Retrieval Conference*, TREC 1999, National Institute of Standards and Technology (NIST), vol Special Publication 246.

Xu, J., & Croft, WB. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, pp 4–11.

Yom-Tov, E., Fine, S., Carmel, D., & Darlow, A. (2005). Metasearch and federation using query difficulty prediction. In *Proceedings of the ACM SIGIR Workshop on Predicting Query Difficulty*.

Zhou, Y. (2008). *Retrieval performance prediction and document quality*. PhD thesis, University of Massachusetts Amherst.

Zhou, Y., & Croft, WB. (2007). Query performance prediction in web search environments. In: *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* ACM, pp 543–550.