Estimation and Use of Uncertainty in Pseudo-relevance Feedback

Kevyn Collins-Thompson and Jamie Callan
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-8213 U.S.A.
{kct | callan}@cs.cmu.edu

ABSTRACT

Existing pseudo-relevance feedback methods typically perform averaging over the top-retrieved documents, but ignore an important statistical dimension: the risk or variance associated with either the individual document models, or their combination. Treating the baseline feedback method as a black box, and the output feedback model as a random variable, we estimate a posterior distribution for the feedback model by resampling a given query's top-retrieved documents, using the posterior mean or mode as the enhanced feedback model. We then perform model combination over several enhanced models, each based on a slightly modified query sampled from the original query. We find that resampling documents helps increase individual feedback model precision by removing noise terms, while sampling from the query improves robustness (worst-case performance) by emphasizing terms related to multiple query aspects. The result is a meta-feedback algorithm that is both more robust and more precise than the original strong baseline method.

Categories and Subject Descriptors:
H.3.3 [**Information Retrieval**]: Retrieval Models

General Terms: Algorithms, Experimentation

Keywords: Query expansion, pseudo-relevance feedback

1. INTRODUCTION

Uncertainty is an inherent feature of information retrieval. Not only do we not know the queries that will be presented to our retrieval algorithm ahead of time, but the user's information need may be vague or incompletely specified by these queries. Even if the query were perfectly specified, language in the collection documents is inherently complex and ambiguous and matching such language effectively is a formidable problem by itself. With this in mind, we wish to treat many important quantities calculated by the retrieval system, whether a relevance score for a document, or a weight for a query expansion term, as random variables whose true value is uncertain but where the uncertainty about the true value may be quantified by replacing the fixed value with a probability distribution over possible values. In this way, retrieval algorithms may attempt to quantify the risk or uncertainty associated with their output rankings, or improve the stability or precision of their internal calculations.

Current algorithms for pseudo-relevance feedback (PRF) tend to follow the same basic method whether we use vector space-based algorithms such as Rocchio's formula [16], or more recent language modeling approaches such as Relevance Models [10]. First, a set of top-retrieved documents is obtained from an initial query and assumed to approximate a set of relevant documents. Next, a single feedback model vector is computed according to some sort of average, centroid, or expectation over the set of possibly-relevant document models. For example, the document vectors may be combined with equal weighting, as in Rocchio, or by query likelihood, as may be done using the Relevance Model[1]. The use of an expectation is reasonable for practical and theoretical reasons, but by itself ignores potentially valuable information about the risk of the feedback model.

Our main hypothesis in this paper is that estimating the uncertainty in feedback is useful and leads to better individual feedback models and more robust combined models. Therefore, we propose a method for estimating uncertainty associated with an individual feedback model in terms of a posterior distribution over language models. To do this, we systematically vary the inputs to the baseline feedback method and fit a Dirichlet distribution to the output. We use the posterior mean or mode as the improved feedback model estimate. This process is shown in Figure 1. As we show later, the mean and mode may vary significantly from the single feedback model proposed by the baseline method. We also perform model combination using several improved feedback language models obtained by a small number of new queries sampled from the original query. A model's weight combines two complementary factors: the model's probability of generating the query, and the variance of the model, with high-variance models getting lower weight.

[1]For example, an expected parameter vector conditioned on the query observation is formed from top-retrieved documents, which are treated as training strings (see [10], p. 62).

SIGIR'07 July 23–27, 2007, Amsterdam, The Netherlands.

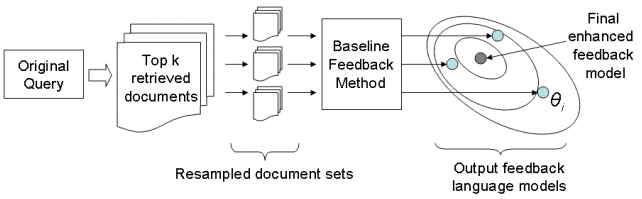


Figure 1: Estimating the uncertainty of the feedback model for a single query.

2. SAMPLING-BASED FEEDBACK

In Sections 2.1–2.5 we describe a general method for estimating a probability distribution over the set of possible language models. In Sections 2.6 and 2.7 we summarize how different query samples are used to generate multiple feedback models, which are then combined.

2.1 Modeling Feedback Uncertainty

Given a query Q and a collection $\mathcal{C}$, we assume a probabilistic retrieval system that assigns a real-valued document score $f(D, Q)$ to each document D in $\mathcal{C}$, such that the score is proportional to the estimated probability of relevance. We make no other assumptions about $f(D, Q)$. The nature of $f(D, Q)$ may be complex: for example, if the retrieval system supports structured query languages [12], then $f(D, Q)$ may represent the output of an arbitrarily complex inference network defined by the structured query operators. In theory, the scoring function can vary from query to query, although in this study for simplicity we keep the scoring function the same for all queries. Our specific query method is given in Section 3.

We treat the feedback algorithm as a black box and assume that the inputs to the feedback algorithm are the original query and the corresponding top-retrieved documents, with a score being given to each document. We assume that the output of the feedback algorithm is a vector of term weights to be used to add or reweight the terms in the representation of the original query, with the vector normalized to form a probability distribution. We view the the inputs to the feedback black box as random variables, and analyze the feedback model as a random variable that changes in response to changes in the inputs. Like the document scoring function $f(D, Q)$, the feedback algorithm may implement a complex, non-linear scoring formula, and so as its inputs vary, the resulting feedback models may have a complex distribution over the space of feedback models (the *sample space*). Because of this potential complexity, we do not attempt to derive a posterior distribution in closed form, but instead use simulation. We call this distribution over possible feedback models the *feedback model distribution*. Our goal in this section is to estimate a useful approximation to the feedback model distribution.

For a specific framework for experiments, we use the language modeling (LM) approach for information retrieval [15]. The score of a document D with respect to a query Q and collection $\mathcal{C}$ is given by $p(Q|D)$ with respect to language models $\hat{\theta}_Q$ and $\hat{\theta}_D$ estimated for the query and document respectively. We denote the set of k top-retrieved documents from collection $\mathcal{C}$ in response to Q by $\mathcal{D}_Q(k, \mathcal{C})$. For simplicity, we assume that queries and documents are generated by multinomial distributions whose parameters are represented by unigram language models.

To incorporate feedback in the LM approach, we assume a model-based scheme in which our goal is take the query and resulting ranked documents $\mathcal{D}_Q(k, \mathcal{C})$ as input, and output an expansion language model $\hat{\theta}_E$, which is then interpolated with the original query model $\hat{\theta}_Q$:

$$\hat{\theta}_{New} = (1 - \alpha) \cdot \hat{\theta}_Q + \alpha \cdot \hat{\theta}_E \quad (1)$$

This includes the possibility of $\alpha = 1$ where the original query mode is completely replaced by the feedback model.

Our sample space is the set of all possible language models $\mathcal{L}_F$ that may be output as feedback models. Our approach is to take samples from this space and then fit a distribution to the samples using maximum likelihood. For simplicity, we start by assuming the latent feedback distribution has the form of a Dirichlet distribution. Although the Dirichlet is a unimodal distribution, and in general quite limited in its expressiveness in the sample space, it is a natural match for the multinomial language model, can be estimated quickly, and can capture the most salient features of confident and uncertain feedback models, such as the overall spread of the distibution.

2.2 Resampling document models

We would like an approximation to the posterior distribution of the feedback model $\mathcal{L}_F$. To accomplish this, we apply a widely-used simulation technique called *bootstrap sampling* ([7], p. 474) on the input parameters, namely, the set of top-retrieved documents.

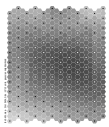
Bootstrap sampling allows us to simulate the approximate effect of perturbing the parameters within the black box feedback algorithm by perturbing the inputs to that algorithm in a systematic way, while making no assumptions about the nature of the feedback algorithm.

Specifically, we sample k documents *with replacement* from $\mathcal{D}_Q(k, \mathcal{C})$, and calculate an expansion language model θ_b using the black box feedback method. We repeat this process B times to obtain a set of B feedback language models, to which we then fit a Dirichlet distribution. Typically B is in the range of 20 to 50 samples, with performance being relatively stable in this range. Note that instead of treating each top document as equally likely, we sample according to the estimated probabilities of relevance of each document in $\mathcal{D}_Q(k, \mathcal{C})$. Thus, a document is more likely to be chosen the higher it is in the ranking.

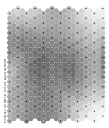
2.3 Justification for a sampling approach

The rationale for our sampling approach has two parts. First, we want to improve the quality of individual feedback models by smoothing out variation when the baseline feedback model is unstable. In this respect, our approach resembles *bagging* [4], an ensemble approach which generates multiple versions of a predictor by making bootstrap copies of the training set, and then averages the (numerical) predictors. In our application, top-retrieved documents can be seen as a kind of noisy training set for relevance.

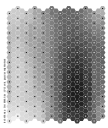
Second, sampling is an effective way to estimate basic properties of the feedback posterior distribution, which can then be used for improved model combination. For example, a model may be weighted by its prediction confidence, estimated as a function of the variability of the posterior around the model.



(a) Topic 401 Foreign minorities, Germany



(b) Topic 402 Behavioral genetics



(c) Topic 459 When can a lender foreclose on property

Figure 2: Visualization of expansion language model variance using self-organizing maps, showing the distribution of language models that results from resampling the inputs to the baseline expansion method. The language model that would have been chosen by the baseline expansion is at the center of each map. The similarity function is Jensen-Shannon divergence.

2.4 Visualizing feedback distributions

Before describing how we fit and use the Dirichlet distribution over feedback models, it is instructive to view some examples of actual feedback model distributions that result from bootstrap sampling the top-retrieved documents from different TREC topics.

Each point in our sample space is a language model, which typically has several thousand dimensions. To help analyze the behavior of our method we used a Self-Organizing Map (via the SOM-PAK package [9]), to 'flatten' and visualize the high-dimensional density function[2].

The density maps for three TREC topics are shown in Figure 2 above. The *dark* areas represent regions of high similarity between language models. The *light* areas represent regions of low similarity – the 'valleys' between clusters. Each diagram is centered on the language model that would have been chosen by the baseline expansion. A single peak (mode) is evident in some examples, but more complex structure appears in others. Also, while the distribution is usually close to the baseline feedback model, for some topics they are a significant distance apart (as measured by Jensen-Shannon divergence), as in Subfigure 2c. In such cases, the mode or mean of the feedback distribution often performs significantly better than the baseline (and in a smaller proportion of cases, significantly worse).

2.5 Fitting a posterior feedback distribution

After obtaining feedback model samples by resampling the feedback model inputs, we estimate the feedback distribution. We assume that the multinomial feedback models $\{\hat{\theta}_1, \ldots, \hat{\theta}_B\}$ were generated by a latent Dirichlet distribution with parameters $\{\alpha_1, \ldots, \alpha_N\}$. To estimate the $\{\alpha_1, \ldots, \alpha_N\}$, we fit the Dirichlet parameters to the B language model samples according to maximum likelihood using a generalized Newton procedure, details of which are given in Minka [13]. We assume a simple Dirichlet prior over the $\{\alpha_1, \ldots, \alpha_N\}$, setting each to $\alpha_i = \mu \cdot p(w_i \mid \mathcal{C})$, where μ is a parameter and $p(\cdot \mid \mathcal{C})$ is the collection language model estimated from a set of documents from collection $\mathcal{C}$. The parameter fitting converges very quickly – typically just 2 or 3 iterations are enough – so that it is practical to apply at query-time when computational overhead must be small. In practice, we can restrict the calculation to the vocabulary of the top-retrieved documents, instead of the entire collection. Note that for this step we are re-using the existing retrieved documents and not performing additional queries.

Given the parameters of an N-dimensional Dirichlet distribution $Dir(\alpha)$ the mean μ and mode x vectors are easy to calculate and are given respectively by

$$\mu_i = \frac{\alpha_i}{\sum \alpha_i} \quad (2) \quad \text{and} \quad \mathrm{x}_i = \frac{\alpha_i - 1}{\sum \alpha_i - N}. \quad (3)$$

We can then choose the language model at the mean or the mode of the posterior as the final enhanced feedback model. (We found the mode to give slightly better performance.)

For information retrieval, the number of samples we will have available is likely to be quite small for performance reasons – usually less than ten. Moreover, while random sampling is useful in certain cases, it is perfectly acceptable to allow deterministic sampling distributions, but these must be designed carefully in order to approximate an accurate output variance. We leave this for future study.

2.6 Query variants

We use the following methods for generating variants of the original query. Each variant corresponds to a different assumption about which aspects of the original query may be important. This is a form of deterministic sampling. We selected three simple methods that cover complimentary assumptions about the query.

No-expansion Use only the original query. The assumption is that the given terms are a complete description of the information need.

Leave-one-out A single term is left out of the original query. The assumption is that one of the query terms is a noise term.

Single-term A single term is chosen from the original query. This assumes that only one aspect of the query, namely, that represented by the term, is most important.

After generating a variant of the original query, we combine it with the original query using a weight α_{SUB} so that we do not stray too 'far'. In this study, we set $\alpha_{SUB} = 0.5$. For example, using the Indri [12] query language, a leave-one-out variant of the initial query that omits the term 'ireland' for TREC topic 404 is:

```
#weight(0.5 #combine(ireland peace talks)
0.5 #combine(peace talks))
```

2.7 Combining enhanced feedback models from multiple query variants

When using multiple query variants, the resulting enhanced feedback models are combined using Bayesian model combination. To do this, we treat each word as an item to be classified as belonging to a relevant or non-relevant class, and derive a class probability for each word by combining the scores from each query variant. Each score is given by that term's probability in the Dirichlet distribution. The term scores are weighted by the inverse of the variance of the term in the enhanced feedback model's Dirichlet distribution. The prior probability of a word's membership in the relevant class is given by the probability of the original query in the entire enhanced expansion model.

[2]Because our points are language models in the multinomial simplex, we extended SOM-PAK to support Jensen-Shannon divergence, a widely-used similarity measure between probability distributions.

3. EVALUATION

In this section we present results confirming the usefulness of estimating a feedback model distribution from weighted resampling of top-ranked documents, and of combining the feedback models obtained from different small changes in the original query.

3.1 General method

We evaluated performance on a total of 350 queries derived from four sets of TREC topics: 51-200 (TREC-1&2), 351-400 (TREC-7), 401-450 (TREC-8), and 451-550 (wt10g, TREC-9&10). We chose these for their varied content and document properties. For example, wt10g documents are Web pages with a wide variety of subjects and styles while TREC-1&2 documents are more homogeneous news articles. Indexing and retrieval was performed using the Indri system in the Lemur toolkit [12] [1]. Our queries were derived from the words in the title field of the TREC topics. Phrases were not used. To generate the baseline queries passed to Indri, we wrapped the query terms with Indri's `#combine` operator. For example, the initial query for topic 404 is:

```
#combine(ireland peace talks)
```

We performed Krovetz stemming for all experiments. Because we found that the baseline (Indri) expansion method performed better using a stopword list with the feedback model, all experiments used a stoplist of 419 common English words. However, an interesting side-effect of our resampling approach is that it tends to remove many stopwords from the feedback model, making a stoplist less critical. This is discussed further in Section 3.6.

3.2 Baseline feedback method

For our baseline expansion method, we use an algorithm included in Indri 1.0 as the default expansion method. This method first selects terms using a log-odds calculation described by Ponte [14], but assigns final term weights using Lavrenko's relevance model[10].

We chose the Indri method because it gives a consistently strong baseline, is based on a language modeling approach, and is simple to experiment with. In a TREC evaluation using the GOV2 corpus [6], the method was one of the top-performing runs, achieving a 19.8% gain in MAP compared to using unexpanded queries. In this study, it achieves an average gain in MAP of 17.25% over the four collections.

Indri's expansion method first calculates a log-odds ratio $o(v)$ for each potential expansion term v given by

$$o(v) = \sum_D \log \frac{p(v|D)}{p(v|C)} \tag{4}$$

over all documents D containing v, in collection C. Then, the expansion term candidates are sorted by descending $o(v)$, and the top m are chosen. Finally, the term weights $r(v)$ used in the expanded query are calculated based on the relevance model

$$r(v) = \sum_D p(q|D)p(v|D)\frac{p(v)}{p(D)} \tag{5}$$

The quantity $p(q|D)$ is the probability score assigned to the document in the initial retrieval. We use Dirichlet smoothing of $p(v|D)$ with $\mu = 1000$.

This relevance model is then combined with the original query using linear interpolation, weighted by a parameter α. By default we used the top 50 documents for feedback and the top 20 expansion terms, with the feedback interpolation parameter $\alpha = 0.5$ unless otherwise stated. For example, the baseline expanded query for topic 404 is:

```
#weight(0.5 #combine(ireland peace talks) 0.5
#weight(0.10 ireland 0.08 peace 0.08 northern ...)
```

3.3 Expansion performance

We measure our feedback algorithm's effectiveness by two main criteria: precision, and robustness. Robustness, and the tradeoff between precision and robustness, is analyzed in Section 3.4. In this section, we examine average precision and precision in the top 10 documents (P10). We also include recall at 1,000 documents.

For each query, we obtained a set of B feedback models using the Indri baseline. Each feedback model was obtained from a random sample of the top k documents taken with replacement. For these experiments, $B = 30$ and $k = 50$. Each feedback model contained 20 terms. On the query side, we used leave-one-out (LOO) sampling to create the query variants. Single-term query sampling had consistently worse performance across all collections and so our results here focus on LOO sampling. We used the methods described in Section 2 to estimate an enhanced feedback model from the Dirichlet posterior distribution for each query variant, and to combine the feedback models from all the query variants. We call our method 'resampling expansion' and denote it as RS-FB here. We denote the Indri baseline feedback method as Base-FB. Results from applying both the baseline expansion method (Base-FB) and resampling expansion (RS-FB) are shown in Table 1.

We observe several trends in this table. First, the average precision of RS-FB was comparable to Base-FB, achieving an average gain of 17.6% compared to using no expansion across the four collections. The Indri baseline expansion gain was 17.25%. Also, the RS-FB method achieved consistent improvements in P10 over Base-FB for every topic set, with an average improvement of 6.89% over Base-FB for all 350 topics. The lowest P10 gain over Base-FB was +3.82% for TREC-7 and the highest was +11.95% for wt10g. Finally, both Base-FB and RS-FB also consistently improved recall over using no expansion, with Base-FB achieving better recall than RS-FB for all topic sets.

3.4 Retrieval robustness

We use the term *robustness* to mean the worst-case average precision performance of a feedback algorithm. Ideally, a robust feedback method would never perform worse than using the original query, while often performing better using the expansion.

To evaluate robustness in this study, we use a very simple measure called the *robustness index* (RI)[3]. For a set of queries Q, the RI measure is defined as:

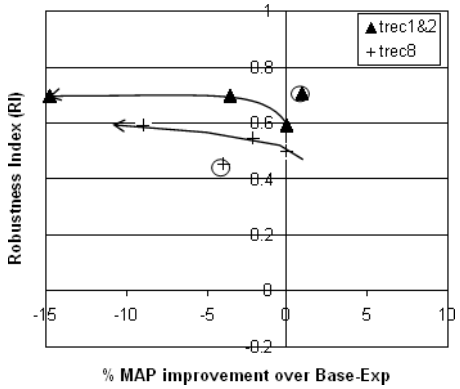
$$RI(Q) = \frac{n_+ - n_-}{|Q|} \tag{6}$$

where n_+ is the number of queries helped by the feedback method and n_- is the number of queries hurt. Here, by 'helped' we mean obtaining a higher average precision as a result of feedback. The value of RI ranges from a minimum

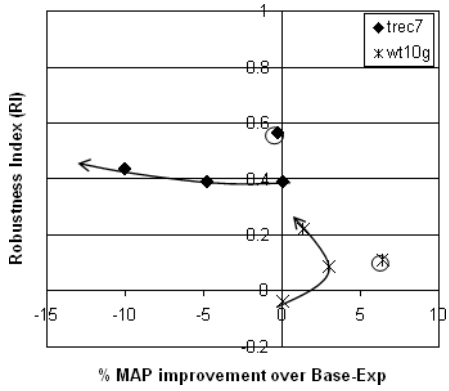
[3]This is sometimes also called the *reliability of improvement index* and was used in Sakai *et al.* [17].

Collection		NoExp	Base-FB	RS-FB
TREC 1&2	AvgP	0.1818	0.2419 (+33.04%)	0.2406 (+32.24%)
	P10	0.4443	0.4913 (+10.57%)	0.5363 (+17.83%)
	Recall	15084/37393	19172/37393	15396/37393
TREC 7	AvgP	0.1890	0.2175 (+15.07%)	0.2169 (+14.75%)
	P10	0.4200	0.4320 (+2.85%)	0.4480 (+6.67%)
	Recall	2179/4674	2608/4674	2487/4674
TREC 8	AvgP	0.2031	0.2361 (+16.25%)	0.2268 (+11.70%)
	P10	0.3960	0.4160 (+5.05%)	0.4340 (+9.59%)
	Recall	2144/4728	2642/4728	2485/4728
wt10g	AvgP	0.1741	0.1829 (+5.06%)	0.1946 (+11.78%)
	P10	0.2760	0.2630 (-4.71%)	0.2960 (+7.24%)
	Recall	3361/5980	3725/5980	3664/5980

Table 1: Comparison of baseline (Base-FB) feedback and feedback using re-sampling (RS-FB). Improvement shown for Base-FB and RS-FB is relative to using no expansion.



(a) TREC 1&2 (upper curve); TREC 8 (lower curve)



(b) TREC 7 (upper curve); wt10g (lower curve)

Figure 3: The trade-off between robustness and average precision for different corpora. The x-axis gives the change in MAP over using *baseline expansion* with $\alpha = 0.5$. The y-axis gives the Robustness Index (RI). Each curve through *uncircled* points shows the RI/MAP tradeoff using the simple small-α strategy (see text) as α decreases from 0.5 to zero in the direction of the arrow. *Circled* points represent the tradeoffs obtained by resampling feedback for $\alpha = 0.5$.

Collection	N	Base-FB		RS-FB	
		n_-	RI	n_-	RI
TREC 1&2	103	26	+0.495	15	+0.709
TREC 7	46	14	+0.391	10	+0.565
TREC 8	44	12	+0.455	12	+0.455
wt10g	91	48	–0.055	39	+0.143
Combined	284	100	+0.296	76	+0.465

Table 2: Comparison of robustness index (RI) for baseline feedback (Base-FB) vs. resampling feedback (RS-FB). Also shown are the actual number of queries hurt by feedback (n_-) for each method and collection. Queries for which initial average precision was negligible (≤ 0.01) were ignored, giving the remaining query count in column N.

of -1.0, when all queries are hurt by the feedback method, to $+1.0$ when all queries are helped. The RI measure does not take into account the magnitude or distribution of the amount of change across the set Q. However, it is easy to understand as a general indication of robustness.

One obvious way to improve the worst-case performance of feedback is simply to use a smaller fixed α interpolation parameter, such as $\alpha = 0.3$, placing less weight on the (possibly risky) feedback model and more on the original query. We call this the 'small-α' strategy. Since we are also reducing the potential gains when the feedback model is 'right', however, we would expect some trade-off between average precision and robustness. We therefore compared the precision/robustness trade-off between our resampling feedback algorithm, and the simple small-α method. The results are summarized in Figure 3. In the figure, the curve for each topic set interpolates between trade-off points, beginning at x=0, where $\alpha = 0.5$, and continuing in the direction of the arrow as α decreases and the original query is given more and more weight. As expected, robustness continuously increases as we move along the curve, but mean average precision generally drops as the gains from feedback are eliminated. For comparison, the performance of resampling feedback at $\alpha = 0.5$ is shown for each collection as the *circled* point. *Higher* and *to the right* is better. This figure shows that resampling feedback gives a somewhat better trade-off than the small-α approach for 3 of the 4 collections.

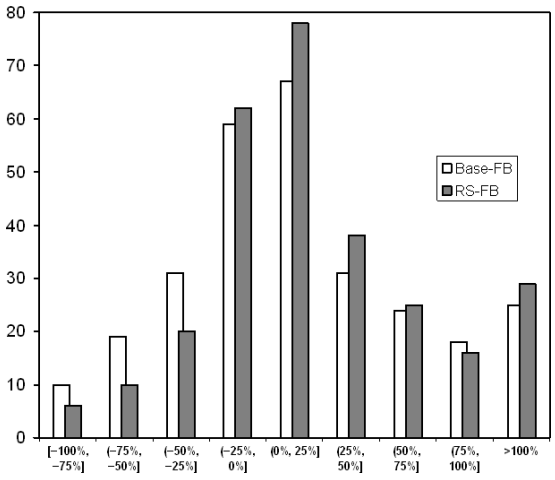


Figure 4: Histogram showing improved robustness of resampling feedback (RS-FB) over baseline feedback (Base-FB) for all datasets combined. Queries are binned by % change in AP compared to the unexpanded query.

Collection		DS + QV	DS + No QV
TREC 1&2	AvgP	0.2406	0.2547 (+5.86%)
	P10	0.5263	0.5362 (+1.88%)
	RI	0.7087	0.6515 (-0.0572)
TREC 7	AvgP	0.2169	0.2200 (+1.43%)
	P10	0.4480	0.4300 (-4.02%)
	RI	0.5652	0.2609 (-0.3043)
TREC 8	AvgP	0.2268	0.2257 (-0.49%)
	P10	0.4340	0.4200 (-3.23%)
	RI	0.4545	0.4091 (-0.0454)
wt10g	AvgP	0.1946	0.1865 (-4.16%)
	P10	0.2960	0.2680 (-9.46%)
	RI	0.1429	0.0220 (-0.1209)

Table 3: Comparison of resampling feedback using document sampling (DS) with (QV) and without (No QV) combining feedback models from multiple query variants.

Table 2 gives the Robustness Index scores for Base-FB and RS-FB. The RS-FB feedback method obtained higher robustness than Base-FB on three of the four topic sets, with only slightly worse performance on TREC-8.

A more detailed view showing the distribution over relative changes in AP is given by the histogram in Figure 4. Compared to Base-FB, the RS-FB method achieves a noticable reduction in the number of queries significantly hurt by expansion (i.e. where AP is hurt by 25% or more), while preserving positive gains in AP.

3.5 Effect of query and document sampling methods

Given our algorithm's improved robustness seen in Section 3.4, an important question is what component of our system is responsible. Is it the use of document re-sampling, the use of multiple query variants, or some other factor? The results in Table 3 suggest that the model combination based on query variants may be largely account for the improved robustness. When query variants are turned off and the original query is used by itself with document sampling, there is little net change in average precision, a small decrease in P10 for 3 out of the 4 topic sets, but a significant drop in robustness for all topic sets. In two cases, the RI measure drops by more than 50%.

We also examined the effect of the document sampling method on retrieval effectiveness, using two different strategies. The 'uniform weighting' strategy ignored the relevance scores from the initial retrieval and gave each document in the top k the same probability of selection. In contrast, the 'relevance-score weighting' strategy chose documents with probability proportional to their relevance scores. In this way, documents that were more highly ranked were more likely to be selected. Results are shown in Table 4.

The relevance-score weighting strategy performs better overall, with significantly higher RI and P10 scores on 3 of the 4 topic sets. The difference in average precision between the methods, however, is less marked. This suggests that uniform weighting acts to increase variance in retrieval results: when initial average precision is high, there are many relevant documents in the top k and uniform sampling may give a more representative relevance model than focusing on the highly-ranked items. On the other hand, when initial precision is low, there are few relevant documents in the bottom ranks and uniform sampling mixes in more of the non-relevant documents.

For space reasons we only summarize our findings on sample size here. The number of samples has some effect on precision when less than 10, but performance stabilizes at around 15 to 20 samples. We used 30 samples for our experiments. Much beyond this level, the additional benefits of more samples decrease as the initial score distribution is more closely fit and the processing time increases.

3.6 The effect of resampling on expansion term quality

Ideally, a retrieval model should not require a stopword list when estimating a model of relevance: a robust statistical model should down-weight stopwords automatically depending on context. Stopwords can harm feedback if selected as feedback terms, because they are typically poor discriminators and waste valuable term slots. In practice, however, because most term selection methods resemble a $tf \cdot idf$ type of weighting, terms with low idf but very high tf can sometimes be selected as expansion term candidates.

This happens, for example, even with the Relevance Model approach that is part of our baseline feedback. To ensure as strong a baseline as possible, we use a stoplist for all experiments reported here. If we turn off the stopword list, however, we obtain results such as those shown in Table 5 where four of the top ten baseline feedback terms for TREC topic 60 (said, but, their, not) are stopwords using the Base-FB method. (The top 100 expansion terms were selected to generate this example.)

Indri's method attempts to address the stopword problem by applying an initial step based on Ponte [14] to select less-common terms that have high log-odds of being in the top-ranked documents compared to the whole collection. Nevertheless, this does not overcome the stopword problem completely, especially as the number of feedback terms grows.

Using resampling feedback, however, appears to mitigate

Collection		QV + Uniform weighting	QV + Relevance-score weighting
TREC 1&2	AvgP	0.2545	0.2406 (-5.46%)
	P10	0.5369	0.5263 (-1.97%)
	RI	0.6212	0.7087 (+14.09%)
TREC 7	AvgP	0.2174	0.2169 (-0.23%)
	P10	0.4320	0.4480 (+3.70%)
	RI	0.4783	0.5652 (+18.17%)
TREC 8	AvgP	0.2267	0.2268 (+0.04%)
	P10	0.4120	0.4340 (+5.34%)
	RI	0.4545	0.4545 (+0.00%)
wt10g	AvgP	0.1808	0.1946 (+7.63%)
	P10	0.2680	0.2960 (+10.45%)
	RI	0.0220	0.1099 (+399.5%)

Table 4: Comparison of uniform and relevance-weighted document sampling. The percentage change compared to uniform sampling is shown in parentheses. QV indicates that query variants were used in both runs.

Baseline FB	$p(w_i\|\mathcal{R})$	Resampling FB	$p(w_i\|\mathcal{R})$
said	0.055	court	0.026
court	0.055	pay	0.018
pay	0.034	federal	0.012
but	0.026	education	0.011
employees	0.024	teachers	0.010
their	0.024	employees	0.010
not	0.023	case	0.010
federal	0.021	their	0.009
workers	0.020	appeals	0.008
education	0.020	union	0.007

Table 5: Feedback term quality when a stoplist is not used. Feedback terms for TREC topic 60: *merit pay vs seniority*.

the effect of stopwords automatically. In the example of Table 5, resampling feedback leaves only one stopword (their) in the top ten. We observed similar feedback term behavior across many other topics. The reason for this effect appears to be the interaction of the term selection score with the top-m term cutoff. While the presence and even proportion of particular stopwords is fairly stable across different document samples, their relative position in the top-m list is not, as sets of documents with varying numbers of better, lower-frequency term candidates are examined for each sample. As a result, while some number of stopwords may appear in each sampled document set, any given stopword tends to fall below the cutoff for multiple samples, leading to its classification as a high-variance, low-weight feature.

4. RELATED WORK

Our approach is related to previous work from several areas of information retrieval and machine learning. Our use of query variation was inspired by the work of YomTov et al. [20], Carpineto et al. [5], and Amati et al. [2], among others. These studies use the idea of creating multiple subqueries and then examining the nature of the overlap in the documents and/or expansion terms that result from each subquery. Model combination is performed using heuristics. In particular, the studies of Amati et al. and Carpineto et al. investigated combining terms from individual distributional methods using a term-reranking combination heuristic. In a set of TREC topics they found wide average variation in the rank-distance of terms from different expansion methods. Their combination method gave modest positive improvements in average precision.

The idea of examining the overlap between lists of suggested terms has also been used in early query expansion approaches. Xu and Croft's method of Local Context Analysis (LCA) [19] includes a factor in the empirically-derived weighting formula that causes expansion terms to be preferred that have connections to multiple query terms.

On the document side, recent work by Zhou & Croft [21] explored the idea of adding noise to documents, re-scoring them, and using the stability of the resulting rankings as an estimate of query difficulty. This is related to our use of document sampling to estimate the risk of the feedback model built from the different sets of top-retrieved documents. Sakai et al. [17] proposed an approach to improving the robustness of pseudo-relevance feedback using a method they call *selective sampling*. The essence of their method is that they allow skipping of some top-ranked documents, based on a clustering criterion, in order to select a more varied and novel set of documents later in the ranking for use by a traditional pseudo-feedback method. Their study did not find significant improvements in either robustness (RI) or MAP on their corpora.

Greiff, Morgan and Ponte [8] explored the role of variance in term weighting. In a series of simulations that simplified the problem to 2-feature documents, they found that average precision degrades as term frequency variance – high noise – increases. Downweighting terms with high variance resulted in improved average precision. This seems in accord with our own findings for individual feedback models.

Estimates of output variance have recently been used for improved text classification. Lee *et al.* [11] used query-specific variance estimates of classifier outputs to perform improved model combination. Instead of using sampling, they were able to derive closed-form expressions for classifier variance by assuming base classifiers using simple types of inference networks.

Ando and Zhang proposed a method that they call structural feedback [3] and showed how to apply it to query expansion for the TREC Genomics Track. They used r query

variations to obtain R different sets S_r of top-ranked documents that have been intersected with the top-ranked documents obtained from the original query q_{orig}. For each S_i, the normalized centroid vector $\hat{w}_i$ of the documents is calculated. Principal component analysis (PCA) is then applied to the $\hat{w}_i$ to obtain the matrix Φ of H left singular vectors ϕ_h that are used to obtain the new, expanded query

$$q_{exp} = q_{orig} + \Phi^T \Phi q_{orig}. \quad (7)$$

In the case $H = 1$, we have a single left singular vector ϕ:

$$q_{exp} = q_{orig} + (\phi^T q_{orig})\phi$$

so that the dot product $\phi^T q_{orig}$ is a type of dynamic weight on the expanded query that is based on the similarity of the original query to the expanded query. The use of variance as a feedback model quality measure occurs indirectly through the application of PCA. It would be interesting to study the connections between this approach and our own model-fitting method.

Finally, in language modeling approaches to feedback, Tao and Zhai [18] describe a method for more robust feedback that allows each document to have a different feedback α. The feedback weights are derived automatically using regularized EM. A roughly equal balance of query and expansion model is implied by their EM stopping condition. They propose tailoring the stopping parameter η based on a function of some quality measure of feedback documents.

5. CONCLUSIONS

We have presented a new approach to pseudo-relevance feedback based on document and query sampling. The use of sampling is a very flexible and powerful device and is motivated by our general desire to extend current models of retrieval by estimating the risk or variance associated with the parameters or output of retrieval processes. Such variance estimates, for example, may be naturally used in a Bayesian framework for improved model estimation and combination. Applications such as selective expansion may then be implemented in a principled way.

While our study uses the language modeling approach as a framework for experiments, we make few assumptions about the actual workings of the feedback algorithm. We believe it is likely that any reasonably effective baseline feedback algorithm would benefit from our approach. Our results on standard TREC collections show that our framework improves the robustness of a strong baseline feedback method across a variety of collections, without sacrificing average precision. It also gives small but consistent gains in top-10 precision. In future work, we envision an investigation into how varying the set of sampling methods used and the number of samples controls the trade-off between robustness, accuracy, and efficiency.

Acknowledgements

We thank Paul Bennett for valuable discussions related to this work, which was supported by NSF grants #IIS-0534345 and #CNS-0454018, and U.S. Dept. of Education grant #R305G03123. Any opinions, findings, and conclusions or recommendations expressed in this material are the authors. and do not necessarily reflect those of the sponsors.

6. REFERENCES

[1] The Lemur toolkit for language modeling and retrieval. http://www.lemurproject.org.

[2] G. Amati, C. Carpineto, and G. Romano. Query difficulty, robustness, and selective application of query expansion. In *Proc. of the 25th European Conf. on Information Retrieval (ECIR 2004)*, pages 127–137.

[3] R. K. Ando and T. Zhang. A high-performance semi-supervised learning method for text chunking. In *Proc. of the 43rd Annual Meeting of the ACL*, pages 1–9, June 2005.

[4] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[5] C. Carpineto, G. Romano, and V. Giannini. Improving retrieval feedback with multiple term-ranking function combination. *ACM Trans. Info. Systems*, 20(3):259 – 290.

[6] K. Collins-Thompson, P. Ogilvie, and J. Callan. Initial results with structured queries and language models on half a terabyte of text. In *Proc. of 2005 Text REtrieval Conference*. NIST Special Publication.

[7] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley and Sons, 2nd edition, 2001.

[8] W. R. Greiff, W. T. Morgan, and J. M. Ponte. The role of variance in term weighting for probabilistic information retrieval. In *Proc. of the 11th Intl. Conf. on Info. and Knowledge Mgmt. (CIKM 2002)*, pages 252–259.

[9] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. SOMPAK: The self-organizing map program package. Technical Report A31, Helsinki University of Technology, 1996. http://www.cis.hut.fi/research/papers/som_tr96.ps.Z.

[10] V. Lavrenko. *A Generative Theory of Relevance*. PhD thesis, University of Massachusetts, Amherst, 2004.

[11] C.-H. Lee, R. Greiner, and S. Wang. Using query-specific variance estimates to combine Bayesian classifiers. In *Proc. of the 23rd Intl. Conf. on Machine Learning (ICML 2006)*, pages 529–536.

[12] D. Metzler and W. B. Croft. Combining the language model and inference network approaches to retrieval. *Info. Processing and Mgmt.*, 40(5):735–750, 2004.

[13] T. Minka. Estimating a Dirichlet distribution. Technical report, 2000. http://research.microsoft.com/ minka/papers/dirichlet.

[14] J. Ponte. *Advances in Information Retrieval*, chapter Language models for relevance feedback, pages 73–96. 2000. W.B. Croft, ed.

[15] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proc. of the 1998 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281.

[16] J. Rocchio. *The SMART Retrieval System*, chapter Relevance Feedback in Information Retrieval, pages 313–323. Prentice-Hall, 1971. G. Salton, ed.

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

[18] T. Tao and C. Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *Proc. of the 2006 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 162–169.

[19] J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Trans. Inf. Syst.*, 18(1):79–112, 2000.

[20] E. YomTov, S. Fine, D. Carmel, and A. Darlow. Learning to estimate query difficulty. In *Proc. of the 2005 ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 512–519.

[21] Y. Zhou and W. B. Croft. Ranking robustness: a novel framework to predict query performance. In *Proc. of the 15th ACM Intl. Conf. on Information and Knowledge Mgmt. (CIKM 2006)*, pages 567–574.