Query Expansion for Language Modeling using Sentence Similarities

Debasis Ganguly, Johannes Leveling, and Gareth J.F. Jones

CNGL, School of Computing, Dublin City University, Ireland
{dganguly, jleveling, gjones}@computing.dcu.ie

Abstract. We propose a novel method of query expansion for Language Modeling (LM) in Information Retrieval (IR) based on the similarity of the query with sentences in the top ranked documents from an initial retrieval run. In justification of our approach, we argue that the terms in the expanded query obtained by the proposed method roughly follow a Dirichlet distribution which, being the conjugate prior of the multinomial distribution used in the LM retrieval model, helps the feedback step. IR experiments on the TREC ad-hoc retrieval test collections using the sentence based query expansion (SBQE) show a significant increase in Mean Average Precision (MAP) compared to baselines obtained using standard term-based query expansion using LM selection score and the Relevance Model (RLM). The proposed approach to query expansion for LM increases the likelihood of generation of the pseudo-relevant documents by adding sentences with maximum term overlap with the query sentences for each top ranked pseudo-relevant document thus making the query look more like these documents. A per topic analysis shows that the new method hurts less queries compared to the baseline feedback methods, and improves average precision (AP) over a broad range of queries ranging from easy to difficult in terms of the initial retrieval AP. We also show that the new method is able to add a higher number of *good* feedback terms (the golden standard of *good* terms being the set of terms added by True Relevance Feedback). Additional experiments on the challenging search topics of the TREC-2004 Robust track show that the new method is able to improve MAP by 5.7% without the use of external resources and query hardness prediction typically used for these topics.

1 Introduction

A major problem in information retrieval (IR) is the mismatch between query terms and terms in relevant documents in the collection. Query expansion (QE) is a popular technique used to help bridge this vocabulary gap by adding terms to the original query. The expanded query is presumed to better describe the information need by including additional or attractive terms likely to occur in relevant documents.

Evidence suggests that in some cases a document as a whole might not be relevant to the query, but a subtopic of it may be highly relevant [1], summarization improves accuracy and speed of user relevance judgments [2], and even relevant documents may act as “poison pills” and harm topic precision after feedback [3],

This problem is compounded in Blind Relevance Feedback (BRF) where all top ranked documents from an initial retrieval run are assumed to be relevant, but often one or more of them will not be. In this case, terms not related to the topic of the query, but which meet the selection criteria are used for QE (e.g. semantically unrelated, but high frequency terms from long pseudo-relevant documents). Using text passages for feedback (sentences instead of full documents) has the potential to be more successful since long documents can contain a wide range of discourse and using whole documents can result in noisy terms being added to the original query, causing a topic shift.

The multi-topic nature of many long documents means that the relevant portion may be quite small, while generally feedback terms should be extracted from relevant portions only. This observation leads to the idea of using smaller textual units (passages[1]) for QE. This proposal dates back to early experiments on the TIPSTER collections [4, 5]. This approach raises questions of how to create the passages, how to select the relevant passages, how passage size influences performance, and how to extract feedback terms from the passages. In the method proposed in this paper, passages correspond to single sentences, which are ranked by their similarity with the original query. Using sentences instead of fixed length (non)overlapping word windows has the implicit advantages that firstly it does not involve choosing the window length parameter, and secondly a sentence represents a more natural semantic unit of text than a passage.

The sentence based QE method for LM proposed in this paper is based on extracting sentences from pseudo-relevant documents, ranking them by similarity with the query sentences, and adding the most similar ones as a whole to expand the original query. This approach to QE, which we call SBQE (Sentence Based Query Expansion), introduces more context to the query than term based expansion. Moreover adding a number of sentences from the document text in its original form has a natural interpretation within the context of the underlying principle of LM in the sense that the modified query more closely resembles the pseudo-relevant documents increasing the likelihood of generating it from these documents.

The remainder of the paper is organized as follows: Section 2 reviews existing and related work on BRF in general and feedback approaches for LM in particular. Section 3 introduces the sentence based query expansion. Section 4 describes our experimental setup and presents the results on TREC adhoc topics, while Section 5 contains a detailed analysis on the robustness of the proposed method and finally Section 6 concludes with directions for future work.

2 Related Work

Standard blind relevance feedback (BRF) techniques for IR assume that the top R documents as a whole are relevant and extract T terms to expand the original query. The various different IR models have corresponding different approaches to QE (see, for example [6–8]). Typically, BRF can increase performance on average for a topic set, but does not perform well on all topics. Some research even questions the usefulness of BRF in general [9]. Many approaches have been proposed to increase the overall IR performance of BRF, for example by adapting the number of feedback terms and documents per topic [10], by selecting only good feedback terms [11, 12], or by increasing

[1] We employ the term passage in its most general sense, denoting phrases, sentences, paragraphs, and other small text units.

diversity of terms in pseudo-relevant documents by skipping feedback documents [13]. TREC experiments with BRF use conservative settings for the number of feedback terms and documents (see [7, 14]) using less than 10 documents and 10-30 feedback terms to obtain the best IR effectiveness. In contrast, Buckley et al. [15] performed massive query expansion using the Vector Space Model of the SMART[2] retrieval system for ad-hoc retrieval experiments at TREC 3. The experiments involved Rocchio feedback, a linear combination of re-weighted query terms [6]. In these experiments, 300 to 530 terms and phrases were added for each topic. An improvement in retrieval effectiveness between 7% and 25% in various experiments was observed. Among the existing QE methods in LM the most commonly used ones are: a) Selecting query expansion terms by the LM score [16], b) Estimating an underlying relevance model from query terms and the top ranked documents [17]. Ponte [16] defines a LM score for the words occurring in the R top ranked documents and proposes adding the top T high scoring terms to the original query. He uses the following score:

$$s(w) = \sum_{d \in R} \log \frac{P(w|M_d)}{P(w)}$$

This score prefers the terms which are frequent in the relevant documents and infrequent in the whole collection. Xu and Croft [18] proposed Local Context Analysis (LCA) which involves decomposing the feedback documents into fixed length word windows so as to overcome the problem of choosing terms from the unrelated portions of a long document. and then ranking the terms by a scoring function which depends on the co-occurrence of a word with the query term, the co-occurrence being computed within the fixed word length windows. It also uses the Inverse Document Frequency (idf) score of a word to boost the co-occurrence score of rarely occurring terms as against frequent terms. In our method, we add document sentences which have maximal similarity with the query sentence(s) thus achieving the same effect of filtering out potentially irrelevant parts of a longer document similar to LCA. We do not compute the co-occurrences explicitly nor do we use the idf scores. Lavrenko and Croft [17] provide a solid theoretical footing on the co-occurrence based feedback as done in LCA by proposing the estimation of an underlying relevance model which is supposed to generate the relevant documents as well as the query terms. Considering the event that the query terms are samples from the unknown relevance model, co-occurrence of a word with the query terms can be used to estimate this probability. An attempt to use shorter context for BRF instead of full documents can be found in [19] where document summaries are extracted based on sentence significance scores, which are a linear combination of scores derived from significant words found by clustering, the overlap of title terms and document, sentence position, and a length normalization factor. Järvelin [20] investigated under which conditions IR based on sentence extraction is successful. He investigates user interactions for true relevance feedback. Additional BRF experiments are based on TREC 7-8 data and use the Lemur system. The best result is obtained using 5 documents and 30 terms. Our proposed method can be related to the above mentioned existing works in the following ways:

[2] `ftp://ftp.cs.cornell.edu/pub/smart/`

Table 1: Differences between QE approaches.

Feature	Term based QE	SBQE
QE components	Term-based	Sentence-based
Candidate scoring	Term score/RSV	Sentence similarity
Number of terms	Few terms (5-20)	Many terms ($>$ 100)
Extraction	Terms from feedback documents or segments	Sentences from the whole document
Working Methodology	On the whole set of feedback documents	On one document at a time
Differentiation between feedback documents	Not done	More sentences are selected from a top ranked document as compared to a lower ranked one
idf factor of terms	Used	Not used

i) It utilises the co-occurrence information of LCA and Relevance Model (RLM) in a different way. A word may co-occur with a query term in a document, but they may be placed far apart. The proximity between the two cannot be handled by these two methods. Recent work by Lv and Zhai [21] attempted to address this issue by generalizing the RLM, in a method called PRLM, where non-proximal co-occurrence is down-weighted by using propagated counts of terms using a Gaussian kernel. The difference between our work and LCA and (P)RLM is that co-occurrence of terms is not computed explicitly, since we rely on the intrinsic relationship of a document word with a query term as defined by the proximity of natural sentence boundaries.

ii) A major difference between all the existing methods and our method is that our method processes each feedback document in turn instead of considering the merits all the pseudo-relevant documents (or segments) collectively. This allows us to extract more content from the top-ranked documents and less from the lower ranked ones.

iii) Our method utilizes shorter context as explored in [19] and [20], but differs from these approaches in the sense that these methods follow the traditional term selection approach over the set of extracted shorter segments whereas we do not employ any term selection method from the shorter segments (sentences). Also we do not need to tune parameters such as the window size for passages as in [5].

Existing work on sentence retrieval considering sentences as the retrieval units instead of documents [22, 23]. The difference between this and ours is that our goal is not to retrieve sentences, but on sentence selection as an intermediate step to help BRF.

Table 1 summarizes the major differences between term-based QE and SBQE.

3 Sentence Based Query Expansion (SBQE)

3.1 Motivation

In LM based IR, a document d is ranked by the estimated probability $P(Q|M_d)$ of generating a query Q from the document model M_d underlying in the document D. M_D is modelled to choose $Q = \{t_1, t_2 \dots t_n\}$ as a sequence of independent words as proposed by Hiemstra [8]. The estimation probability is given by Equation 1.

$$P(Q|M_d) = \prod_{i=1}^{n} \lambda_i P(t_i|M_d) + (1 - \lambda_i) P(t_i) \quad (1)$$

The term weighting equation can be derived from Equation 1 by dividing it with $(1 - \lambda_i)P(t_i)$ and taking $\log$ on both sides to convert the product to summation.

$$\log P(Q|M_d) = \sum_{i=1}^{n} \log(1 + \frac{\lambda_i}{1 - \lambda_i} \frac{P(t_i|M_d)}{P(t_i)}) \quad (2)$$

Thus if the query vector $\boldsymbol{q}$ is weighted as $q_k = tf(t_k)$ and the document vector $\boldsymbol{d}$ is weighted as $d_k = log(1 + \frac{P(t_k|M_d)}{P(t_k)} \frac{\lambda_k}{1-\lambda_k})$, the dot product $\boldsymbol{d} \cdot \boldsymbol{q}$ gives the likelihood of generating $\boldsymbol{q}$ from $\boldsymbol{d}$ and can be used as the similarity score to rank the documents. Adding sentences from relevant documents to the query serves the purpose of making the query look more like the relevant documents and hence increases the likelihood of generating the relevant documents by increasing the maximum likelihood estimate $P(t_i|M_d)$.

3.2 Methodology

Let R be the number of top ranked documents assumed to be relevant for a query. Each pseudo-relevant document d can be represented as a set comprising of the constituent sentences. Thus $d = \{d^1, \dots d^{\eta(d)}\}$ where $\eta(d)$ denotes the number of sentences in d and d^is are its constituent sentences. Each such sentence d^i is represented as a vector $d^i = (d^i_1, \dots d^i_{\zeta(d)})$, where $\zeta(d)$ is the number of unique terms in d. The components of the vector are defined as $d^i_j = \texttt{tf}(t_j, d^i) \; \forall j \in [1, \zeta(d)]$, where $\texttt{tf}(t_j, d^i)$ denotes the term frequency of term t_j in sentence d^i. Also the query text is similarly mapped to the vector representation. Similarity between a sentence vector and the query vector is computed by measuring the cosine of the angle between the vectors. We choose the cosine similarity because it favours shorter texts [24]. The working steps of the proposed method are as follows:

1. Initialize i to 0.
2. For each sentence vector $\boldsymbol{q^j}$ in the query do Steps 3-5.
3. For each sentence vector $\boldsymbol{d^k} \in d$ (where d is the i^{th} pseudo-relevant document) compute the cosine similarity as $\frac{\boldsymbol{d^k} \cdot \boldsymbol{q^j}}{|\boldsymbol{d^k}||\boldsymbol{q^j}|}$ and store the document sentence similarity pair in a sorted set S ordered by decreasing similarities.

4. Add the first $m_i = \min(\lfloor \frac{1-m}{r-1}(i-1) + m \rfloor, |S|)$ sentences from the set S to the query .
5. Clear the set S. If done with all pseudo-relevant documents then exit; else increment i by 1 and goto Step 2.

The value of m_i is obtained by a linear interpolation as shown in Step 4, the slope of the interpolating line being uniquely determined from the fact that we use m number of sentences for the top ranked document and 1 sentence for the bottom ranked one. This ensures that as we go down through the ranked list we progressively become more selective in adding the sentences.

3.3 A formal justification

For simplicity let the initial query be $\boldsymbol{q} = (q_1, \dots q_n)$ where each unique term q_i occurs only once. From a pseudo-relevant document, we add the sentences with maximum similarity back to the original query. Since a similar sentence must have one or more query terms in it, in other words we can say that whenever one or more query terms are found in a document sentence, we add the same query terms with some more additional terms back to the original query. This methodology can be modeled as a variant of Polya's urn scheme where it is known that the distribution of balls after a sufficiently large number of draws approaches a Dirichlet distribution [25].

The initial query can be thought of as an urn of n balls, the i^{th} ball having a colour c_i. Each pseudo-relevant document can be thought of as a bag of transparent boxes (sentences) so that it is possible to know whether a box (sentence) contains a ball (term) of colour similar to a ball from the query urn. Let us also consider another initially empty urn (expanded query) where we pour in the contents from the selected transparent boxes. A turn in the game comprises of opening a bag, looking at the boxes with matching balls and emptying its contents onto the output urn. If we find more boxes with colour c_i, we are going to end up with more balls of colour c_i in the output urn. Let us assume that after a large number of draws, we have α_i balls of colour c_i where $i = 1, 2, \dots N$ and let the total number of balls be $\alpha_0 = \sum_i^N \alpha_i$. The expectation of finding a ball of colour c_i is

$$E[(X_i = c_i)] = \frac{\alpha_i}{\alpha_0}$$

Thus, after a sufficient number of steps in the game, we could say that the distribution of colours of balls in the output urn follows a Dirichlet distribution $Dir(\alpha_1, \dots \alpha_N)$.

Coming back to the feedback algorithm, the colours are analogous to unique terms and the output urn is the generated expanded query $X \sim Dir(\alpha)$, $X, \alpha \in \mathbb{R}^{\mathbb{N}}$, i.e. the expanded query comprises of N unique terms, the event $X_i = t_i$ denoting that the i^{th} term is t_i. It is well known that the Dirichlet distribution $Dir(X, \alpha)$ is the conjugate prior to the multinomial distribution $Mult(X, \alpha)$. Since $Mult(X, \alpha)$ is the likelihood model of generating a query term used in the LM retrieval, the expanded query can be seen as the posterior probability of the event $X_i = t_i$ after seeing α_i occurrences of t_i in the pseudo-relevant documents. Another way of expressing this is that placing a prior distribution of $Dir(\alpha)$ on the parameters $P(X_i = t_i)$s of the multinomial distribution through the expanded query is equivalent to adding $\alpha_i - 1$ pseudo observations of term

t_i in addition to the actual number of occurrences of t_i in a document (true observation) in the feedback step.

Speaking in simple terms, it is not only the presence of a query term that the feedback method utilizes, but it tries to reproduce the distribution of the original query terms and the new terms co-occurring with the original ones through evidences in the top R document texts as accurately as possible. Thus, in the feedback step this distribution of the pseudo-occurrences of new terms can benefit the conjugate prior used in the LM retrieval model.

In Section 5.3 we experimentally verify the two hypotheses that firstly it is not the mere presence of expansion terms but rather the distribution which helps in the feedback step, and secondly the greater the number of documents we examine, the better our estimation becomes. However, there is a practical limit to the number of documents that should be examined simply because for every new document examined, the chance that we are going to add a set of terms which are not already added, increases. We do not want N (the number of unique terms in the expanded query) to become too large so that we do not end up with an excessive number of hits on the inverted list.

Lavrenko and Croft state that "Many popular techniques in Information Retrieval, such as relevance feedback and automatic query expansion have a very intuitive interpretation in the traditional probabilistic models, but are conceptually difficult to integrate into the language modeling framework ..."[17]. As we have shown, SBQE is a simple, yet conducive to an intuitive interpretation in LM framework where we can argue the evidence collected from the pseudo-relevant documents generates an expanded query *looking* like the pseudo-relevant documents (formally speaking following a Dirichlet prior of pseudo-observations of the pseudo-relevant documents) thus benefiting the feedback step.

4 Experimental Results

4.1 Description and settings

To evaluate the effectiveness of our proposed method, we used the TREC adhoc document collection (disks 1-5) and title fields of the adhoc topic sets 101-200 (TREC 2-4) and 300-450 (TREC 6-8). We do not use the TREC-5 queries as these queries comprise terms which are poor or negative indicators of relevance (see [26] for more details). Retrieval for all the TREC topic sets were done indexing the corresponding official document sets i.e. TREC 2 and 3 retrievals use disks 1 and 2, TREC 4 uses disks 2 and 3, and TREC 6-8 use disks 4 and 5.

Instead of trying to achieve optimal results by parameter tuning for each topic set, we aim to investigate the robustness and portability of SBQE for unseen topics. We used the TREC 2 query relevance judgments to train our system. The best settings obtained were then used on TREC 3 and 4 topic sets. Similarly TREC 6 query relevance judgments were used for training, and testing was done on TREC 7-8 topics. The reason behind this break-up of the training and test sets is that TREC topic sets 2,3 and 4 resemble each other in terms of the average number of relevant documents. These tasks benefit from using a higher value of R (the number of pseudo-relevant documents) in

Table 2: Summarization of our experimental setup based on the average number of relevant documents for the topic sets

Adhoc-set 1				Adhoc-set 2			
Data set	Topic #	Usage	Avg. # relevant	Data set	Topic #	Usage	Avg. # relevant
TREC-2	101-150	Training	232.9	TREC-6	301-350	Training	92.2
TREC-3	151-200	Testing	196.1	TREC-7	351-400	Testing	93.4
TREC-4	201-250	Testing	130.0	TREC-8	401-450	Testing	94.5

BRF experiments. Whereas the average number of relevant documents is much less for the TREC 6-8 topic sets and a smaller value of R proves to be effective for these topic sets. Table 2 summarizes the average number of relevant documents for the individual topic sets.

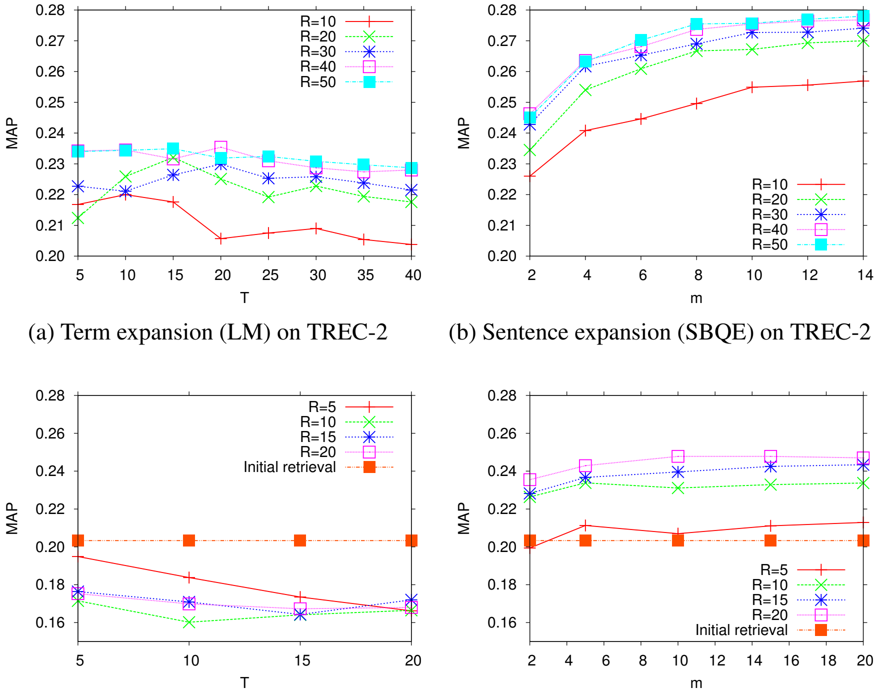
We used the LM module implemented in SMART by one of the authors for indexing and retrieval. Extracted portions of documents were indexed according to Equation 1 and using single terms and a pre-defined set of phrases (using the standard SMART phrase list) according to Equation 1. The retrieval used $\lambda_i = 0.3$ for all query terms. Sentence boundaries were detected using the Morphadorner package[3]. Stopwords were removed using the standard SMART stopword list. Words were stemmed using the Porter stemmer [27].

To compare our approach with the existing feedback approaches in LM, we selected two baselines, the first being the LM term based query expansion, hereafter referred to as LM, as in Equation 1 which was implemented in SMART. The second baseline used the RLM implementation of Indri [28] with default settings. For RLM on TREC topics set 2-4, we used the 50 top documents to estimate the relevance model as reported by Lavrenko and Croft [17]. For the TREC 6-8 topics, our experiments with Indri revealed that the best MAP obtained is by using 5 pseudo-relevant documents, and hence for these topics we used 5 documents to estimate the relevance model. As far as the number of expansion terms is concerned, best results are obtained with no expansion terms (which is also the default settings in Indri) for the RLM implementation. While it may seem that it is unfair to choose the number of expansion terms to be zero for RLM, it is important to note that RLM relies particularly on estimating a relevance model and reordering the initially retrieved documents by KL-divergence from the estimated relevance model. Additional expansion terms do not play a key role in the working principle of the model.

4.2 Feedback Effect

One of the parameters to vary for both LM and SBQE is the number of documents to be used for the BRF which we refer to as R. The other parameter to vary for SBQE is m which is the number of sentences to add. We vary both R and T (the number of terms to add for LM) in the range of [5, 50] in steps of 5.

[3] http://morphadorner.northwestern.edu/morphadorner/



(a) Term expansion (LM) on TREC-2

(b) Sentence expansion (SBQE) on TREC-2

(c) Term expansion (LM) on TREC-6

(d) Sentence expansion (SBQE) on TREC-6

Fig. 1: Iso-R plots demonstrating the effect of varying the number of terms for LM and the parameter m for SBQE on TREC-2 and TREC-6 topics. R is the number of pseudo-relevant documents.

Figures 1a and 1c suggest that for LM, the MAP degrades with an increase in the number of expansion terms, but with SBQE there is no noticeable degradation in MAP with an increase in m, as is evident from Figures 1b and 1d. Also the LM graphs are more parameter sensitive as can be seen from the larger average distances between iso-T points and greater number of intersections of the iso-R lines as compared to the SBQE graphs.

In Table 3 we report the MAPs obtained via all the three approaches for all the 300 topics, the training being done on TREC 2 and 6 topics. This table also reports the percentage changes in MAPs computed with reference to the initial retrieval MAPs for the corresponding approach. The percentage changes under the RLM column is measured relative to the Indri initial retrieval whereas the ones under LM and SBQE columns have been calculated with respect to the SMART initial retrieval.

It can be observed that SBQE outperforms both LM and RLM on these test topics. The statistically significant improvements (measured by Wilcoxon test) in MAP with SBQE over LM and RLM are shown with a $^+$ and * respectively. It can also be observed that although the TREC 4 topic set uses a different collection, the same parameter set-

Table 3: Three BRF approaches (LM, RLM and SBQE) on TREC 2-4 and 6-8 "title-only" topics (trained respectively on TREC 2 and TREC 6 topics).

Topic	LM Initial retrieval		MAP			Avg. # of terms	
set	SMART	Indri	LM	RLM	SBQE	LM	SBQE
101-150	0.169	0.195	0.236 (+39.4%)	0.206 (+5.5%)	**0.278**$^{+*}$ (+64.5%)	13.78	1007.26
151-200	0.215	0.234	0.288 (+34.2%)	0.242 (+3.6%)	**0.327**$^{+*}$ (+52.5%)	14.50	1141.24
201-250	0.204	0.181	0.228 (+12.2%)	0.185 (+1.9%)	**0.255**$^{+*}$ (+25.3%)	17.96	1513.66
301-350	0.207	0.217	0.195 (-6.10%)	0.226 (+4.2%)	**0.248**$^{+}$ (+19.4%)	7.48	404.84
351-400	0.161	0.185	0.163 (+0.90%)	0.187 (+0.8%)	**0.196**$^{+}$ (+21.4%)	7.42	445.90
401-450	0.241	0.241	0.213 (-11.4%)	0.245 (+1.7%)	**0.289**$^{+}$ (+12.8%)	7.38	465.88

tings works fairly well. This is suggestive of the relative insensitiveness of the method to precise parameter settings.

For TREC-6 (Figures 1c and 1d), we see that using LM, the best MAP we obtain is 0.1949 (which is worse than the initial retrieval) using 5 documents and 5 terms as seen in Figure 1c. Although term expansion performs very poorly on these topics, all the retrieval results being worse compared to the initial retrieval, SBQE does perform well on these topics with a significant increase in MAP compared to the initial retrieval.

The SBQE plots of Figures 1b and 1d bring out an experimental verification of the hypothesis proposed in Section 3.3, that greater the number of documents we use for predicting the Dirichlet distribution of terms in the expanded query, better the predictions for the conjugate prior become and better is the retrieval effectiveness in the feedback step. It can be observed that the MAP values of the feedback steps for increasing values of R form a strict monotonically increasing sequence.

5 Posthoc Analysis

In this section we begin with an opening subsection on query drift analysis of SBQE as compared to the other feedback methods. The following subsection aims to investigate the effectiveness of SBQE on the *hard* topics of the TREC 2004 Robust Track. This is followed by an examination of the term frequencies of the expanded query where we aim to find experimental verification of the fact that the distribution of terms in the bag-of-words model of the expanded query do play a pivotal role in the feedback step. The section ends with a subsection where we see how close this new feedback method gets to QE using true relevance feedback for TREC 6-8 ad-hoc tasks.

5.1 Query drift analysis

It has been found that traditional QE techniques degrade performance for many topics [9]. This can arise particularly if most of the top ranked pseudo-relevant documents are actually not relevant to the query. In these cases, QE can add terms not associated with the focus of the query, and cause the expanded query vector to draft further away from

Table 4: Feedback effects on the 5 topic categories for LM, RLM and SBQE. 300 topics TREC 2-4 and 6-8("title-only") were used for the analysis.

LM Initial retrieval	# Queries		# Queries improved			# Queries hurt			% change in AP		
precision interval	SMART	Indri	LM	RLM	SBQE	LM	RLM	SBQE	LM	RLM	SBQE
$[0-0.1)$	117	120	57	51	**63**	60	69	54	+56.3	-1.6	**+75.0**
$[0.1-0.2)$	77	59	50	34	**57**	27	25	20	+34.1	+2.0	**+64.0**
$[0.2-0.3)$	37	42	23	28	**31**	14	14	6	+22.5	+7.4	**+37.1**
$[0.3-0.4)$	23	25	14	13	**19**	9	12	4	+7.7	+2.0	**+27.5**
$[0.4-0.5)$	18	21	10	**15**	**15**	8	6	3	+4.4	+5.8	**+23.5**
$[0.5-1]$	28	33	15	**24**	18	13	9	10	-5.0	**+1.3**	+1.3
Total	300	300	169	165	**203**	131	135	97			

the centroid of the relevant documents and as a result the feedback retrieval can lead to worse AP for these topics.

The topics for which the initial retrieval AP is fairly good can be termed easy topics and the ones for which it is poor as difficult or hard ones. An ideal QE technique is expected to perform well over a broad range of the spectrum defined by initial retrieval AP values, ideally not degrading retrieval effectiveness for any topics. To see the effect of SBQE, we categorize all the topics (TREC 2-4 and 6-8) into classes defined by a range over the initial retrieval AP values hereafter referred to as bins. Five equal length intervals are chosen as $\{[i, i+0.1)\}$ where $i \in \{0, 0.1 \ldots 0.4\}$. Since there are not many topics with initial retrieval AP over 0.5, the last interval is chosen as $[0.5, 1]$ so as to maintain a balance in the number of queries in each bin for meaningful comparisons. Thus the first bin contains the topics for which the initial retrieval AP is between 0 and 0.1, the second bin consists of topics for which the it is between 0.1 and 0.2 and so on. For each bin, the AP is computed by considering only the queries of that current bin. A similar analysis was presented in [29] the only difference being that they used discrete integer values of P@20 to define the bins.

In Table 4 we report statistics computed for each query class for the three expansion techniques. It can be observed that the SBQE achieves a positive change in AP for each query class. RLM exhibits maximal improvement (in terms of change in AP) for the group defined by the range $[0.2, 0.3)$ of initial precision values whereas both LM and SBQE work best for the topics whose initial retrieval AP is less than 0.1. LM suffers from a degradation in AP for queries with initial AP higher than 0.5, whereas SBQE and RLM improve the AP for this group, the improvement being slightly more for RLM. But improvements of AP for the other groups in SBQE are considerably higher than RLM. It is worth noting that the number of queries hurt by feedback for every query class other than the last one (where RLM is the winner with 9 hurts compared to 10 for SBQE) is the minimum for SBQE, thus making it an attractive query expansion method. The total number of queries being hurt is far less as compared to LM and RLM.

Table 5: Feedback effect on the hard topics categorized into 4 failure classes and 100 new topics for the TREC 2004 robust track. For all queries only the title field was used. All the methods use the best parameter settings obtained by training on TREC-6 topics.

Topic Category	LM Initial retrieval		MAP		
	SMART	Indri	LM	RLM	SBQE
2: General technical failures such as stemming	0.225	0.116	0.127 (-9.7%)	0.118 (+0.2%)	**0.253** (+2.9%)
3: Systems all emphasize one aspect, miss another required term	0.081	0.083	0.162 (+8.1%)	0.088 (+0.5%)	**0.179** (+9.8%)
4: Systems all emphasize one aspect, miss another aspect	0.089	0.071	0.147 (+5.7%)	0.074 (+0.3%)	**0.152** (+6.2%)
5: Some systems emphasize one aspect, some another, need both	0.103	0.118	0.119 (+1.5%)	0.118 (+0.0%)	**0.140** (+3.7%)
Overall	0.108	0.092	0.139 (+3.1%)	0.094 (+0.2%)	**0.165** (+5.7%)
601-700 (New topics for TREC 2004 Robust track)	0.261	0.262	0.277 (+1.6%)	0.274 (+1.2%)	**0.354** (+9.3%)

5.2 Feedback effect on TREC-2004 Robust track topics

The TREC Robust track explored retrieval for a challenging set of topics from the TREC ad hoc tasks [30]. The 45 topics from TREC topics 301-450 were categorized as *hard* based on Buckley's failure analysis [31]. Buckley [32] categorized the topics into failure classes with increasing difficulty and required natural level language understanding. He suggests that retrieval could be improved for the 17 topics in categories 2-5 if systems could predict in the category to which topic a topic belongs.

We applied SBQE on the *hard* topics (categories 2-5) without the use of external resources and/or selective query expansion methods. We also ran all the three methods used in experiments on 100 new topics (601-700) designed for the TREC robust track as instances of challenging topics. Our results for individual groups of topics are shown in Table 5. From these results, we can see clearly that SBQE outperforms LM and RLM for the *hard* topics. SBQE achieves a MAP of *0.354* which ranks third among the official set of results [30] behind *pircRB04t3* [33] and *fub04Tge* [34] both of which employ web documents as an external resource for BRF. The important observation to be made here is that SBQE, without any separate training on *hard* topics, is able to achieve good precision without the use of any external resources and without employing selective query expansion (which itself consumes additional computation time).

We take a sample query from each category and report some terms added by SBQE, but not by LM term expansion. For topic 445 - "women clergy" belonging to category 3, true feedback adds terms like *stipend*, *church*, *priest*, *ordain*, *bishop*, *England* etc. The description of the topic reads "What other countries besides the United States are considering or have approved women as clergy persons". While LM expansion adds the terms *church*, *priest* and *ordain*, SBQE adds the additional terms (*bishop*, 7), (*England*, 10), (*stipend*, 7), (*ordain*, 11) where the numbers beside the terms indicate their occurrence frequencies in the expanded query. Common sense suggests that according to the description of this topic, *England* is indeed a good term to add. A look at topic 435 - “curbing population growth” belonging to category 4, reveals that term based LM feedback adds terms like *billion*, *statistics*, *number*, while it misses terms representing the other aspect of relevance (the aspect of contraceptive awareness in rural areas to prevent population growth - emphasized by terms like *rural*, *contraceptive* etc.), which are added by SBQE.

5.3 Term frequency analysis of expanded query

To justify the hypothesis of the estimated Dirichlet prior for the expanded query as the key working reason behind SBQE, we perform a series of experiments on the generated expanded query for the TREC 8 topic set. Firstly we set the term frequencies for each unique term to 1, thus reducing the expanded query to a uniform distribution where every term is equally likely to occur. Next, we seek an answer to the question of whether all terms that we added to the query are indeed useful for retrieval or could we filter out some of the rarely occurring terms from the expanded query. We therefore remove terms falling below a cut-off threshold of frequency 10, 2 and 1. Table 6a reports the observations and clearly shows that the frequencies indeed play a vital role because retrieval effectiveness decreases either when we set the term frequencies to one ignoring the evidence we collected from each feedback document or when we leave out some of the terms. Since we add a large number of terms to the original query, the expanded query at a first glance might intuitively suggest a huge query drift. But the observation which needs to be made here is that a vast majority of the terms are of low frequency. *Important* are those terms which have maximal evidence of occurrence in the feedback documents in proximity to the original query terms, the notion of proximity being defined by natural sentence boundaries. However, frequency alone is not the only criterion for the *goodness* of a term. Some low frequency terms are beneficial for the feedback step too as suggested by the fact that simply cutting off the terms based on frequency has a negative effect on precision.

5.4 Comparison with True Relevance Feedback

To see if SBQE is indeed able to add the *important* query terms to the original query we run a series of true relevance feedback (TRF) experiments which involve selecting

Terms	MAP
All terms	0.2887
$tf(t_i) \leftarrow 1$ (Frequencies set to 1)	0.1805
Terms with frequency > 1	0.280
Terms with frequency > 2	0.273
Terms with frequency > 10	0.248

(a) Term frequency variations on the expanded TREC-8 topics

Method	System	Avg. # terms	Time (s)
LM	SMART	7.38	7
RLM	Indri	2.38	209
SBQE	SMART	465.88	91

(b) Run-time measures on TREC-8 topics

Table 6: Intersection of BRF terms with the gold-standard TRF terms

Topic set	TRF		LM		SBQE	
	MAP	$\|T_{TRF}\|$	MAP	$\|T_{TRF} \cap T_{LM}\|$	MAP	$\|T_{TRF} \cap T_{SBQE}\|$
TREC-6	0.409	1353	0.195	316 (23.3%)	0.248	901 (66.6%)
TREC-7	0.422	1244	0.163	311 (25.0%)	0.196	933 (75.0%)
TREC-8	0.376	1234	0.213	317 (25.7%)	0.289	977 (79.1%)

terms by the LM term selection values as done in our standard BRF experiments, the only difference being that we use only the true relevant out of the top R documents of the initial ranked list for feedback.

While we do not expect that SBQE could outperform TRF, this experiment was designed with a purpose of testing how close the performance of SBQE can get to the ideal scenario. Our main aim was to define a gold-standard for the feedback terms by restricting the LM term selection value to the set of true relevant documents with the assumption that the terms hence selected for feedback provide the best possible evidence of *good* feedback terms. An overlap between the terms obtained by SBQE and the *good* terms found this way can be a measure of the effectiveness of SBQE.

We do the TRF experiments for both TREC 6-8 topic sets. The choice of true relevant documents was left on the top 20 documents from the initial retrieval ranked list. In Table 6 we report the intersection of the set of terms obtained by LM and SBQE with TRF terms. We also re-report the MAPs from Table 3 for convenience of reading. We observe from Table 6 that SBQE is able to add more *important* terms due to the higher degree of overlap with TRF terms.

5.5 Run-time comparisons

One may think that using more than 400 terms for retrieval can lead to poor retrieval performance. But a careful analysis reveals that the time complexity of retrieval for a query of n terms, under a sorted inverted-list implementation scheme as in SMART, is $O(\sum_{i=1}^{n} |L_i|)$, L_i being the size of the inverted list for the i^{th} query term. On the simplified assumption that $L_i = L \; \forall i \in [1, n]$, the retrieval complexity reduces to $O(nL)$. In the worst case, if $n = O(L)$, then the run-time complexity becomes $O(L^2)$. But for SBQE, n, which is in hundreds, is still much less than the average document frequency of query terms. For example in TREC topic 301 - " International Organized Crime", the sum over the document frequencies for the terms is 215839. The SBQE expanded query comprises of 225 terms which is much less as compared to the total size of the inverted lists for the query terms. Our runtime experiments reported in Table 6b reveal that SBQE is faster than the RLM implementation of Indri.

6 Conclusions and Future Work

The main contribution of the paper is the proposal of a novel method of QE by adding sentences in contrast to the traditional approach of adding terms. The proposed method aims to make the query look more like the top ranked documents hence increasing the probability of generating the query from the top ranked documents. We also show that the method behaves like a variant of Polya's urn, and the resulting distribution of terms in the expanded query tends to the conjugate prior of the multinomial distribution used for LM retrieval. While we do not formally derive the output distribution for the variant of Polya's urn, we can explore more on this in our future research.

Although conceptually simple and easy to implement, our method significantly outperforms existing LM feedback methods on 300 TREC topics. A detailed per topic analysis reveals that SBQE increases the AP values for all types of queries when they are categorized from hardest to easiest based on initial retrieval APs. Applying SBQE on the challenging topics from the TREC robust track shows that it significantly outperforms LM and RLM without the use of external resources or selective QE.

For term expansion, it is observed that a variable number of expansion terms chosen dynamically for the individual topics provides best effective results [10]. As future work we would like to explore whether using different m values across topics yields further improvement in the retrieval effectiveness. The method can also be extended to handle fixed length word windows (pseudo-sentences).

Whether involving any of the sentence scoring mechanisms outlined in [22, 23] in our method instead of the simple cosine similarity for selecting the candidate sentences for feedback proves more beneficial will form a part of our future work as well.

Acknowledgments

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (CNGL) project.

References

1. Wilkinson, R.: Effective retrieval of structured documents. In: SIGIR, Springer New York, Inc. (1994) 311–317
2. Tombros, A., Sanderson, M.: Advantages of query biased summaries in information retrieval. In: SIGIR 1998, ACM (1998) 2–10
3. Terra, E.L., Warren, R.: Poison pills: harmful relevant documents in feedback. In: CIKM 2005, ACM (2005) 319–320
4. Callan, J.P.: Passage-level evidence in document retrieval. In: SIGIR 1994, ACM/Springer (1994) 302–310
5. Allan, J.: Relevance feedback with too much data. In: SIGIR 1995, ACM Press (1995) 337–343
6. Rocchio, J.J.: Relevance feedback in information retrieval. In: The SMART retrieval system – Experiments in automatic document processing. Prentice Hall (1971)
7. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M.: Okapi at TREC-3. In: Overview of the Third Text Retrieval Conference (TREC-3), NIST (1995) 109–126
8. Hiemstra, D.: Using Language Models for Information Retrieval. PhD thesis, Center of Telematics and Information Technology, AE Enschede (2000)
9. Billerbeck, B., Zobel, J.: Questioning query expansion: An examination of behaviour and parameters. In: ADC 2004. Volume 27., Australian Computer Society, Inc. (2004) 69–76

10. Ogilvie, P., Vorhees, E., Callan, J.: On the number of terms used in automatic query expansion. Information Retrieval **12**(6) 666–679
11. Cao, G., Nie, J.Y., Gao, J., Robertson, S.: Selecting good expansion terms for pseudo-relevance feedback. In: SIGIR 2008, ACM (2008) 243–250
12. Leveling, J., Jones, G.J.F.: Classifying and filtering blind feedback terms to improve information retrieval effectiveness. In: RIAO 2010, CID (2010)
13. Sakai, T., Manabe, T., Koyama, M.: Flexible pseudo-relevance feedback via selective sampling. ACM Transactions on Asian Language Processing **4**(2) (2005) 111–135
14. Robertson, S., Walker, S., Beaulieu, M., Willett, P.: Okapi at TREC-7: Automatic ad hoc, filtering, vlc and interactive track. In **21** (1999) 253–264
15. Buckley, C., Salton, G., Allan, J., Singhal, A.: Automatic query expansion using SMART: TREC 3. In: Overview of the Third Text REtrieval Conference (TREC-3), NIST (1994) 69–80
16. Ponte, J.M.: A language modeling approach to information retrieval. PhD thesis, University of Massachusetts (1998)
17. Lavrenko, V., Croft, B.W.: Relevance based language models. In: SIGIR 2001, ACM (2001) 120–127
18. Xu, J., Croft, W.B.: Query expansion using local and global document analysis. In: SIGIR 1996, ACM (1996) 4–11
19. Lam-Adesina, A.M., Jones, G.J.F.: Applying summarization techniques for term selection in relevance feedback. In: SIGIR 2001, ACM (2001) 1–9
20. Järvelin, K.: Interactive relevance feedback with graded relevance and sentence extraction: simulated user experiments. In: CIKM 2009, ACM (2009) 2053–2056
21. Lv, Y., Zhai, C.: Positional relevance model for pseudo-relevance feedback. In: SIGIR 2010, ACM (2010) 579–586
22. Murdock, V.: Aspects of Sentence Retrieval. PhD thesis, University of Massachusetts - Amherst (2006)
23. Losada, D.E.: Statistical query expansion for sentence retrieval and its effects on weak and strong queries. Inf. Retr. **13** (2010) 485–506
24. Wilkinson, R., Zobel, J., Sacks-Davis, R.: Similarity measures for short queries. In: In Fourth Text REtrieval Conference (TREC-4). (1995) 277–285
25. Blackwell, D., James, M.: Fergusson distributions via Polya urn schemes. Annals of Statistics (1973) 353–355
26. Xu, J., Croft, W.B.: Improving the effectiveness of informational retrieval with Local Context Analysis. ACM Transactions on information systems **18** (2000) 79–112
27. Porter, M.F.: An algorithm for suffix stripping. Program **14**(3) (1980) 130–137
28. Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: a language-model based search engine for complex queries. In: Online Proceedings of the International Conference on Intelligence Analysis. (2005)
29. Mitra, M., Singhal, A., Buckley, C.: Improving automatic query expansion. In: SIGIR 1998, ACM (1998) 206–214
30. Voorhees, E.M.: Overview of the TREC 2004 robust track. In: TREC. (2004)
31. Harman, D., Buckley, C.: The NRRC Reliable Information Access (ria) workshop. In: SIGIR 2004, New York, NY, USA, ACM (2004) 528–529
32. Buckley, C.: Why current IR engines fail. In: SIGIR 2004, New York, NY, USA, ACM (2004) 584–585
33. Kwok, K.L., Grunfeld, L., Sun, H.L., Deng, P.: TREC 2004 robust track experiments using PIRCS. In: TREC. (2004)
34. Amati, G., Carpineto, C., Romano, G.: Fondazione Ugo Bordoni at TREC 2004. In: TREC. (2004)