Retrievability based Document Selection for Relevance Feedback with Automatically Generated Query Variants

Anirban Chakraborty
ADAPT Centre, School of Computer Science & Statistics
Trinity College Dublin, Ireland
anirban.chakraborty@adaptcentre.ie

Debasis Ganguly
IBM Research Europe
Dublin, Ireland
debasis.ganguly1@ie.ibm.com

Owen Conlan
ADAPT Centre, School of Computer Science & Statistics
Trinity College Dublin, Ireland
Owen.Conlan@scss.tcd.ie

ABSTRACT

To mitigate the problem of over-dependence of a pseudo-relevance feedback algorithm on the top-M document set, we make use of a set of equivalence classes of queries rather than one single query. These query equivalents are automatically constructed either from a) a knowledge base of prior distributions of terms with respect to the given query terms, or b) iteratively generated from a relevance model of term distributions in the absence of such priors. These query variants are then used to estimate the retrievability of each document with the hypothesis that documents that are more likely to be retrieved at top-ranks for a larger number of these query variants are more likely to be effective for relevance feedback. Results of our experiments show that our proposed method is able to achieve substantially better precision at top-ranks (e.g. higher nDCG@5 and P@5 values) for ad-hoc IR and points-of-interest (POI) recommendation tasks.

CCS CONCEPTS

• **Information systems** → **Personalization**; *Information retrieval diversity*; *Recommender systems*; **Probabilistic retrieval models**.

KEYWORDS

Pseudo-relevance feedback, Query variants, Retrievability

ACM Reference Format:
Anirban Chakraborty, Debasis Ganguly, and Owen Conlan. 2020. Retrievability based Document Selection for Relevance Feedback with Automatically Generated Query Variants . In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3412032

1 INTRODUCTION

Standard pseudo-relevance feedback (PRF) methods, such as the relevance model and its variants, have in general been shown to improve overall retrieval effectiveness, such as mean average precision. However, these relevance feedback methods can sometimes, at the cost of increasing recall, lead to decreasing the precision at the very top ranks (e.g. for ranks up to 5). This mainly happens because the only source of information which is made available to a PRF method is the top-retrieved set of documents retrieved in response to a query. One of the limitations is that the effectiveness of the PRF algorithms depends, to a large extent, on the choice of this set (the top-retrieved M documents), which makes these algorithms less robust and more sensitive to the variations in the chosen set of pseudo-relevant set of documents [9, 28].

Researchers have explored different approaches to increase the overall retrieval performance, e.g., by learning the appropriate number of feedback terms for query expansion [24], or by selectively using effective feedback terms either by supervised [10] or learning an optimal policy for feedback term selection using reinforcement learning [23]. It was reported in [9] that despite an average performance increase over a set of topics, relevance feedback does not perform well on a large number of topics. One major problem with relevance feedback is that a large number of top ranked (pseudo-relevant) documents may not truly be related to the core information need of the query thus leading to a detrimental effect on the retrieval effectiveness for a large number of topics after query expansion. The study [28] argues that some relevant documents may also in fact act as *poison pills* and hurt post-feedback effectiveness specially in terms of precision.

Our work in this paper aligns with the approaches that seek to estimate a robust set of feedback documents by, generally speaking, employing a *document selector function* to decide which documents from the top-ranked ones to include in the feedback set. Instances of such work include [18], which uses overlapping clusters of documents to find a number of *dominant clusters* of documents, and [15], which uses a classification approach to decide which documents to include in the feedback set. A key novelty of our work with respect to the existing thread of work for feedback document selection is that *our approach does not rely on one single query for estimating this selector function.* Specifically, our PRF algorithm makes use of an automatically constructed equivalence class of queries instead of a single query, and then uses the query variants to execute multiple retrieval steps. We then leverage the notion of retrievability [3] of a document to estimate the likelihood of its usefulness for relevance feedback. We rely on the assumption that if a document is retrieved at high ranks for a higher number of query variants, it is more likely to be relevant to the information need of the original query and hence more likely to be useful for PRF.

As a way to automatically generate query variants, we leverage information from semantic associations between term pairs, which act as weak supervision signals affecting the subsequent feedback step (hence we call our proposed feedback method *weakly*

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412032

supervised relevance model, or *WSRM* for short). We argue that our feedback approach is particularly expected to work well in situations where these term pairs are available as manually annotated resources (e.g. knowledge bases). In the absence of a knowledge-base (as in ad-hoc IR), we use a local co-occurrence matrix of term pair relations. Specifically, we construct a graph representing words as nodes, the edge weights between nodes reflecting the co-occurrence likelihoods [27]. We conduct a random walk on this graph to generate the query variants. To demonstrate the efficacy of our feedback approach in both these situations (i.e. without and with available knowledge-bases), we apply our feedback algorithm on two different tasks in this paper, namely ad-hoc IR and points-of-interest (POI) recommendation, respectively.

The rest of the paper is organized as follows. Section 2 surveys literature on pseudo-feedback document selection and studies involving query variants. In Section 3, we describe how query variants are automatically constructed and how are they eventually used to select the set of feedback documents. Section 4 describes how we adapt our feedback approach for POI recommendation where term pair relationships are available in the form of a knowledge base. Section 5 describes the details of our experiment setup, followed by a presentation of results in Section 6. Finally, Section 7 concludes the paper with directions for future work.

2 RELATED WORK

A pragmatic approach towards pseudo-relevance feedback (PRF) essentially relies on term level manipulations, e.g., while Ogilvie et. al. [24] for their query expansion method learn the appropriate number of feedback terms, Cao et. al. [10] selectively use good feedback terms for query expansion. Traditional PRF methods, such as Okapi [25], the relevance model (RM) [17] and its variants [12, 26], primarily rely on the set of top-retrieved M documents for the purpose of selecting potential candidate expansion terms. These approaches inevitably fail to perform well for all queries when the initial top retrieved document set is noisy, which eventually degrade the retrieval performance for many topics after query expansion [9]. For term-level manipulations, researchers have also leveraged on semantic matching with embedded vectors to learn retrieval-specific semantic relationships from top documents retrieved with a large number of queries from a query log [29], or to combine the effects of global term semantics within the framework of RM [26].

A comparatively less explored approach towards PRF is the use of document level manipulations with an aim to create a more robust set of feedback documents [6, 18]. Existing research along this thread includes those of [15] where a supervised classification approach was applied for selecting good feedback documents using a number of features, and [18] where a k-NN based resampling method was applied for selecting the *dominant set* of documents for relevance feedback.

Our proposed document selection method is based on document retrievability [3] on *query variants*. Studying query variants recently became popular among researchers. Use of manually created query variants [4] has been shown to yield more consistent retrieval [5, 8] and query performance prediction effectiveness [31]. In a recent work, Lu et al. [20] explored different fusion techniques to combine multiple relevance models estimated on different query variants. They experimented with both manually created query variants (UQV dataset [4]) and query variants automatically created leveraging external resources. Liu et al. [19] conducted a comparative analysis of manual and automatic query variants and reported that they yield comparable retrieval effectiveness. The study [20] showed that manual query variants result in better query performance prediction (QPP) than automatically constructed variants. Benham et al. [7] explored a way of automatically generating query variants with the help of external parallel corpora to mimic the achievable retrieval performance using manually generated query variants. Generating query variants based on some external resources may not always be feasible due to the dependency on the external data. Instead of relying on the availability of human generated query variants, in our work we propose a method to generate this set of *reference queries* automatically for each query, without the help of any external resources.

3 WEAKLY SUPERVISED RELEVANCE MODEL

3.1 Relevance Model

Relevance model (RM) [17] is a PRF method which estimates a term's importance for relevance feedback by making use of the co-occurrence statistics between a set of given query terms and those occurring in the top-retrieved documents. Formally speaking, given a query $Q = \{q_1, \ldots, q_n\}$, RM estimates a term weight distribution $P(w|R) \approx P(w|Q)$. It is assumed that $P(w|R)$ also generate the set of terms in the top-M documents $\mathcal{M} = \{D_1, \ldots, D_M\}$, i.e.,

$$P(w|R) \approx P(w|Q) = \sum_{D \in \mathcal{M}} P(w|D) \prod_{q \in Q} P(q|D). \tag{1}$$

From Equation 1, it is evident that a high $P(w|Q)$ value (RM term weight) results when a term w occurs frequently in a top-retrieved document (large $P(w|D)$ value) in conjunction with the frequent occurrence of a query term $q \in Q$ within D. This original version of the relevance model is commonly known as 'RM1' in the literature. 'RM1' does not take the original query terms into account while estimating the density function, which usually results in a query drift [21]. It has been shown that a mixture model of the estimated density of other term weights in conjunction with the original query terms yields better results [21]. This mixture model, commonly known by the name 'RM3' [16], is represented as

$$P'(w|R) = \lambda P(w|R) + (1 - \lambda) P(w|Q). \tag{2}$$

Each mention of 'relevance model' or 'RM' in this paper is to be interpreted as its more effective mixture model variant, i.e. 'RM3'.

3.2 Equivalence Classes of Query Variants

Generally speaking, a PRF model in IR, e.g. a relevance model [17], estimates for each non-query term - a relevance score, which is essentially its local co-occurrence likelihood with the query terms (i.e., within the top-M retrieved). The estimation is based only on a single query usually with a small number of terms.

What a standard feedback model lacks, is the process of accumulating evidences over an *extended* set of a larger number of queries, which may lead to a more robust estimation of the relevance weights. In fact, prior work has shown that a combination of feedback models involving a number of query variants improves the retrieval

effectiveness corresponding to the underlying information need of the original query [19]. Notably, both pre-combination (combining PRF models) and post-combination (combining the ranked lists from PRF models) work well in practice [20]. A desirable property of this *extended set of query variants*, comprising a multiple number of queries, is that each member of this set should express a similar information need as that expressed in the original query. We call this set the *equivalence class* of query variants.

A way to construct a good representation of the equivalent set of a query is through a controlled study setup, where participants are asked to formulate queries corresponding to a given information need description ('back-story') [4]. It has been found that the query variants obtained this way, i.e., manually under a controlled setup, for standard TREC query sets (specifically, the TREC Robust, and the TREC 2013 and 2014 Web Tracks) are of relatively good quality in that they can be used to yield a more consistent retrieval [5], improved query performance prediction (QPP) [31] and more effective feedback results [19]. Different from existing approaches of using manually constructed query variants, a key component of our proposed methodology involves automatically generate this set of equivalence class of *query variants* for each query.

3.3 Automatic Construction of Query Variants

Local Term Co-occurrences. We first compute the local co-occurrence matrix between the terms present in the vocabulary of (say) the top-M retrieved documents [30]. Specifically,

$$P(u, v; Q, \mathcal{M}) = \frac{1}{|\mathcal{M}|} \sum_{D \in \mathcal{M}} P(u|D)P(v|D), \;\; u, v \in V_M, \qquad (3)$$

where the usual notation $P(w|D)$ (similar to RM of Equation 1) denotes the probability of sampling a term w from a document D (independent of another term), and the set $V_M = \cup_{D \in \mathcal{M}}\{D\}$ denotes the set of unique terms (vocabulary) of the set of top-retrieved documents $\mathcal{M}$. Similar to RM of Equation 1, we employ a standard collection smoothing (Jelinek-Mercer) based maximum likelihood estimate for computing the probabilities, i.e.,

$$P(u|D) = \lambda \frac{f(u, D)}{|D|} + (1 - \lambda)\frac{f(u)}{f(.)}, \qquad (4)$$

where $f(u, D)$ denotes the frequency of term u in D, $|D|$ denotes the length of D, $f(u)$ denotes the collection frequency of u and $f(.)$ denotes the total aggregate of collection frequencies over all terms (collection size). For generating the variants, we set $\lambda = 0.6$ as per the recommendations in previous studies [17]. As a note for practical implementation, the co-occurrence matrix of Equation 3 can be efficiently implemented by squaring the sparse term-document matrix of the top retrieved M documents, $X \in \mathbb{R}^{M \times |V_M|}$, i.e., yield the desired $|V_M| \times |V_M|$ matrix with the operation $C = X^T X$.

Weighted Graph of Local Co-occurrences. The co-occurrence matrix constructed from each term pair co-occurrence likelihood of Equation 3 represents the adjacency matrix, C, of a graph of $|V_M|$ nodes (each node corresponding to a word). The weight between a pair of nodes in this graph indicates the co-occurrence likelihood between the words (Equation 3). A subset of these nodes constitutes the original query terms, i.e. members of the set Q. The rest, i.e. $|V_M - Q|$, is comprised of candidate terms that could be selected

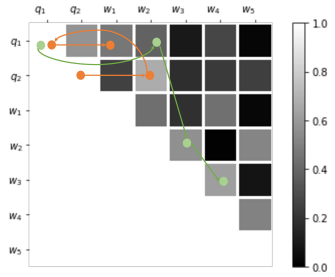


Figure 1: A schematic visualization of query variant construction with the help of random walks. Two sample walks of length 4 each are shown in two different colors. The orange colored walk starts from the query term q_2. The walk then visits node (word) w_2 (a word which has a relatively high co-occurrence likelihood with q_2 as seen from a light shade of gray). The walk then continues to q_1 and terminates at w_1 thus generating a variant, $\hat{Q}_1 = \{q_2, q_1, w_2, w_1\}$ of the original query $Q = \{q_1, q_2\}$.

for forming the query variants. Formally using the definition of $P(u, v; Q, \mathcal{M})$ from Equation 3,

$$G = (Q \cup (V_M - Q), \{(u, v, \omega_{u,v})\}) : \omega_{u,v} = P(u, v; Q, \mathcal{M}) > 0. \quad (5)$$

Random Walk for Query Variant Generation. To select a candidate query variant, we initiate a random walk from one of the query nodes chosen with a uniform probability (this ensures that we include at least one query term in the automatically constructed variant). We continue the walk for a small number of steps (specifically, 3-7 in our experiments). Each walk comprises a set of nodes, the corresponding words of which forms a query variant (strictly speaking, a walk is a sequence of nodes; however, in an IR setup, a query is treated as a set rather than as a sequence of terms). We employ a greedy approach to construct the query variants. In particular, the probability of visiting the next node in the walk is Markovian, i.e., it depends only on the current node visited. The probability of selecting the next node (i.e. that of including the next term in a query variant) is given by the maximum likelihood estimate of the neighboring edge weights. This makes it more likely to select a term that has a high co-occurrence likelihood with the most recent term selected. Formally,

$$P(t_i = v|t_{i-1} = u, \ldots, t_1) = \frac{\omega(u, v)}{\sum_{w \in \mathcal{N}(u)} \omega(u, w)}, \;\; P(t_1 = q) = \frac{1}{|Q|} \qquad (6)$$

where $\mathcal{N}(u)$ denotes the neighborhood (adjacent set of nodes) of the current node u, and t_i denotes the i^{th} term added to the walk.

A schematic illustration of the random walk process of query variants generation is shown in Figure 1. For the purpose of illustration, the figure shows a sample weighted graph visualized as the part above the diagonal of a local co-occurrence matrix (the part to the bottom-left of the diagonal is left blank to avoid confusion). While one of the walks leads to a query variant that also includes both the original query terms (the orange colored walk $\hat{Q}_1 = \{q_2, q_1, w_2, w_1\}$), the green colored walk ($\hat{Q}_2$) is comprised of only one term from the original query.

Characteristics of the Query Variants. Since during each step of the the walk (Equation 6), it is likely to select a word that has a high co-occurrence likelihood with the current word (and by transitivity, also with each word that has already been visited), the set of words eventually included in a walk is likely to represent a query variant that is expected to be semantically related to the original query Q. As the walk proceeds by adding a node at each step to the sequence of nodes already visited, it can happen that a node is visited multiple times. In our query processing stage (Section 3.4), the sequence representation of a walk is transformed into a set representation of terms.

The equivalence class of a query generated by this stochastic random walk is likely to constitute a fair mixture of both specializations and generalizations of the original query. It may happen that some query variants contain the original query as a part of them, e.g. the orange colored walk of Figure 1. The additional terms in these queries is likely to specialize the information need of the original query [11]. Some queries, on the other hand, contain only a subset of the original query terms and hence is likely to lead to generalizing the information need.

Random Walk Length. While each query variant should seek to address the same information need as that of the original query, it should also contain additional semantically similar terms that could potentially enrich the information need (without drifting it away from the information need of the original query). This requires a careful trade-off between *exploitation* (utilizing what has been constructed till the current stage) and *exploration* (seeking to explore more terms to construct more variants). While too conservative an exploration (a short and compact random walk) may result in a small number of variants to be constructed (thus leading to a small post-feedback effect), a too ambitious exploration (a long and spread walk) may result in a large number of variants, the information need of most of which may in fact be substantially different from that of the original query.

With a manual inspection and some of the initial trends in our experiments, we found that a walk length of 7 works well in practice. Moreover, we set the number of generated query variants (each with a separate instance of a random walk) to 50 after observing a set of initial trends in the feedback results. Since we eventually use each query variant to retrieve ranked lists of documents to aggregate retrieval rank likelihoods of documents, too large a number of variants would contribute to increased run-times, as a result of which the number of variants was set to a modest value of 50.

3.4 Query Variants to Feedback Documents

Combining Evidences from Query Variants. After describing the method of automatically constructing query variants, we now describe how to make use of these variants for improving the effectiveness of relevance feedback. The fact that the information need of a manually formulated query variant is quite similar to that of the original query contributes to the effectiveness of feedback and the QPP models [20, 31] that use these variants. However, in the absence of the manually annotated variants (which in fact is representative of a more realistic situation), it is likely that the automatically constructed ones may potentially contain a number of terms that could cause a drift in the information need. This necessitates developing a more robust approach of combining the information retrieved with these query variants. To do so, rather than relying on using a single query variant at a time and then eventually combining their feedback models [20], we instead, for each query Q make use of the *entire* set of its automatically generated variants $\hat{\boldsymbol{Q}}$, to aggregate a collective belief about the usefulness of a document for relevance feedback.

Retrievability based Document Selection. We now describe how, starting with an equivalence class of automatically generated queries, we obtain a candidate set of documents that could be used for relevance feedback. Specifically, we make use of the concept of *retrievability* [3], which is a quantitative score associated with the likelihood of a document D to be retrieved within the top-M ranks in response to a set of queries sampled from a collection. In the context of our problem, the notion of the collection corresponds to the local set of the top-M retrieved documents. Formally,

$$s(D, \hat{\boldsymbol{Q}}) = \sum_{\hat{Q} \in \hat{\boldsymbol{Q}}} r(D, \hat{Q}), \tag{7}$$

where $r(D, \hat{Q})$ is the rank at which document D is retrieved for a query variant $\hat{Q}$. For implementation purpose, we retrieved the top-1000 documents for a query, and $r(D, \hat{Q})$ is set to 1001 if D is not retrieved within top-1000.

Intuitively, Equation 7 aggregates for each document D, the ranks at which each query variant $\hat{Q}$ retrieves D. A low value of these aggregated ranks (lower the better) for a particular document, say D, indicates that D is retrieved *towards top-ranks for a large number of query variants*. These aggregated rank values are then used to preferentially select documents for relevance feedback with the hypothesis that the documents with small (better) values of aggregated ranks are the ones that are consistently retrieved at top ranks for a large number of query variants. This in turn accumulates evidence for the belief that these documents are strongly related to the information need of the original query and hence should be useful for relevance feedback. While on one hand, consistency in the top-retrieved documents for the good quality query variants may help to select the relevant documents, this way of aggregation is also expected to *discount the noisy contributions* from the (possibly) drifted variants on the other.

Differences with the existing notion of *retrievability*. The notion of retrievability that we use in Equation 7 is different in two ways from its original definition [3]. First, in [3] it relied on a parameter r_{max} that specified the upper bound of the rank, and second, it accumulated Boolean values (1/0) indicating whether the rank of a document was within this specified bound. In our approach, firstly, we do not restrict the rank computation with a bound (because for PRF it is difficult to foresee the rank cut-off). Secondly, we aggregate the rank values themselves instead of the Boolean indicator variables to get a better estimate of the likelihood, which also makes the overly restrictive rank cut-off unnecessary.

Feedback with Selected Documents. Next, we sort the feedback (top-M) documents in ascending order of the aggregated retrievability (rank aggregated) scores computed by Equation 7. The

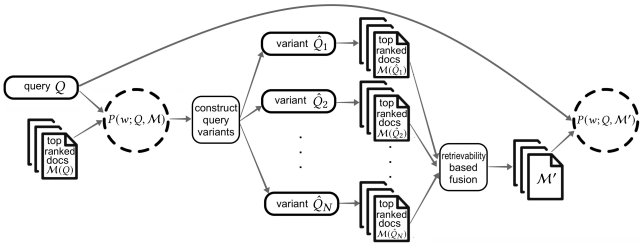


Figure 2: A schematic workflow diagram of our proposed weakly supervised relevance model (WSRM).

top-M' documents from this set are then used for relevance feedback, where M' is a parameter. PRF with this set of documents thus combines evidences across a range of different queries and is thus expected to yield better retrieval effectiveness. In particular, we employ RM (Equation 1) on the top-M' documents from this set. The parameter M' is independent of M, the number of documents used to compute the local co-occurrence graph (Equation 3) and the random walk on it (Equation 6).

We call our model the 'Weakly Supervised RM' (**WSRM**) because the retrieval position likelihoods captured with the aggregated retrievability scores (Equation 7) act as weak supervision signals for estimating document relevance. A schematic overview of the relevance feedback workflow for WSRM is depicted in Figure 2.

3.5 Manually Annotated Query Variants

The main advantage of our proposed feedback method is that it *does not need to rely on manually formulated query variants*. Since recent studies have shown that manually annotated query variants are useful to improve the effectiveness of relevance feedback and query performance prediction (QPP) [7, 20], we in our proposed feedback method (WSRM), also incorporate information from manually formulated query variants. For this, instead of estimating the co-occurrence weights on the top-M retrieved documents (Equation 3), we adapt the idea of [20] where separate relevance models are estimated with each manually constructed individual query variant. Consequently, instead of a single set of top-retrieved set of documents, $\mathcal{M}$, we obtain a total of N such different sets of documents one for each query, where N denotes the number of manual query variants. We then compute the local co-occurrence weights by aggregating the evidences from the top-M documents of each query, i.e.,

$$P(u,v;Q_1,\ldots,Q_N,\mathcal{M}_1\ldots,\mathcal{M}_N) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{M}_i|}\sum_{D\in\mathcal{M}_i}P(u|D)P(v|D). \tag{8}$$

Similar to the single query input, these local co-occurrence values of Equation 8 are used to define a graph with weighted edges, the only difference being that the random walk can now start from an arbitrary query term in any of the manual variants. The rest of the methodology is the same, i.e., we use the retrievability based rank aggregation mechanism (Equation 7) to construct the final set of feedback documents, $\mathcal{M}'$.

4 WEAK SUPERVISION WITH PRIORS

In this section, we describe how the weakly supervised relevance model proposed in Section 3 can be applied in the case of POI (point-of-interest) recommendation, where additional prior beliefs about the contextual appropriateness of a term can act as weak signals to improve RM estimation. POI recommendation, being a precision-oriented task [1], provides an interesting use-case to study the robustness effects of relevance feedback.

4.1 Contextual Recommendation

In an IR-based contextual POI recommendation framework, a system needs to return a ranked list of POIs based on a user's preference history and also his current contextual constraints. Examples of contextual constraints include the current location of the user, the purpose of the trip such as 'holiday' etc. To draw an analogy from the problem of contextual POI recommendation to that of IR, it can be considered that the user preference history and the current contextual constraints in a recommendation system are analogous to the notion of a query in IR, whereas the candidate POIs are analogous to documents [12, 13].

Specifically, a query in contextual recommendation problem is personalized in nature, comprising a) a description of the POI that the user has visited in the past, b) the reviews posted by the user on location-based social networks and the tags associated with the reviews, and c) the ratings associated with the past POI visits. In addition, each query is also associated with a current location of the user, which imposes a *hard constraint* that the recommended POIs must be from the current location of the user. Furthermore, a query also contains a list of *soft constraints*, corresponding to a list of categorical values representing *trip qualifiers*, e.g., 'trip-type = {business, holiday,. . .}', 'trip-duration = {day-trip, night-out,. . .}', 'accompanied-by = {alone, family,. . .}' etc.

Following the work of [12, 13] we represent a query as a structured document of the form (t_u, q_u) comprised of the review-text or tags from the user profile and the trip-qualifier contexts, respectively. A two-step factored RM-based approach that uses both the query and the top-retrieved documents was proposed in [12, 13] to obtain a combined RM of the form

$$\begin{aligned} P(w|\theta_{q_u}) &= \sum_{d\in D_u} rP(w|d)\psi(w,q_u)\prod_{t\in t_u}P(t|d) \\ P(w|\theta_{q_u,l_u}) &= \sum_{d\in D_M(\theta_{q_u}):L(d)=l_u} P(w|d)\psi(w,q_u)\prod_{t\in\theta_{qu}}P(t|d), \end{aligned} \tag{9}$$

where D_u denotes the set of documents (POIs that the user had visited in the past), $L(d) = l_u$ lists the candidate set of POIs in the current location (the hard constraint), $D_M(\theta_{q_u})$ denotes the top-M retrieved POIs with θ_{q_u} as the expanded query, $P(w|d)$ denotes the normalized term frequency of a word w in document d, and $\psi(w,q_u)$ denotes a prior belief on a *contextual appropriateness score* of a term w with respect to a context term q_u (which is explained later in Equation 10).

To see how our proposed relevance feedback framework may be useful to estimate $P(w|\theta_{q_u})$ (Equation 9), note that the first step of constructing the local co-occurrences graph (Section 3.3) can be substituted with that of leveraging information from the

co-occurrence graph of the prior beliefs of manually annotated contextual appropriateness scores between term pairs ($\psi(w, q_u)$ of Equation 9). Next, we describe how to generate the query variants with random walk applied on the graph of binary relations of the term appropriateness scores.

4.2 Query Variants with a Knowledge-base

Knowledge-base to Weighted Graph. A knowledge resource of term-category associations was compiled in [2], which comprises lists of pairs constituting a term and a non-location trip-qualifier with manually judged relevance scores of the form (t, q, a), where t is a term (e.g. `food`), q is a single category (e.g. `holiday`) and $a = 1(a \in [0, 1])$ is the appropriateness score. An example of a non-relevant pair with a lower score is (`nightlife, business, 0.1`). We formally denote this knowledge resource as

$$\kappa : (w, q) \mapsto [0, 1], w \in V, q \in Q_i, i \in \{1, \ldots, c\}), \quad (10)$$

where Q denotes the set of *joint* non-location type contexts, Q_i denotes a context category, and V denotes the vocabulary set of the review text and tags. For a given non-location contextual constraint vector q_u in the user query, we use embedded word vector representations to aggregate the similarities of each word in the review text/tag of a user profile with the seed words assessed as relevant for a context q_u. Formally, $\forall w \in P_U$ we define a function,

$$\psi(w, q_u) = \max(\mathbf{w} \cdot \mathbf{s}),\ s \in \cup\{t : \kappa(t, q_U) = 1\}. \quad (11)$$

Equation 11 indicates that for each word w (embedded vector of which is represented as $\mathbf{w}$) contained in the text from the profile of a user, we compute its maximum similarity over a subset of seed words relevant only for the given context, i.e., the words for which $\kappa(s, q_U) = 1$. In our experiments, we make use of the `word2vec` (skipgram algorithm) [22] for the purpose of embedding the vector representation of a word.

The reason for using the maximum as the aggregate function in Equation 11 is that a word is usually semantically similar to a small number of seed set of words relevant for a given context. To illustrate this with an example, let the 3-dimensional query context comprising trip-type, duration and company be set to the value of '(vacation, day-trip, friends)'. The relevant seed set in this example constitutes words such as 'base-ball stadium', 'beer-garden', 'salon', 'sporting-goods-shop' etc. However, a word such as 'pub' is similar to only one member of this seed set, namely 'beer-garden', which means that other aggregation functions, such as averaging, can lead to a low aggregated value, which in this case is not desirable.

WSRM with Edge Weights from Knowledge-base. The values indicating term pair relations, $\psi(w, q_u)$, computed by Equation 11 are then used to define a weighted graph (similar to the one of Equation 5). After defining the graph this way, we then apply the random walk based method (Equation 6) to initiate a number of different walks from the query terms. In this case, therefore, the walks are comprised of tags and trip qualifier terms.

As a novel contribution of this paper different to that of [12, 13], we then modify the RM estimation of Equation 9 with the weak-supervised approach based on query variants. Specifically, instead of applying RM over the top-M retrieved documents $D_M(\theta_{q_u})$, we use the documents with the lowest rank aggregation scores obtained from the query variants (Equation 7). This weak supervised RM is able to take into account the prior beliefs in the contextual appropriateness of terms from a knowledge resource.

Table 1: Dataset Overview

Collection	Topic Set	#Topics	Fields	Qry Ids	Avg.$\lvert Q \rvert$	Avg.#Rel
Disks 4 and 5 minus CR	TREC 6	50	title	301-350	2.48	92.22
	TREC 7	50	title	351-400	2.42	93.48
	TREC 8	50	title	401-450	2.38	94.56
	TREC Rb	99	title	601-700	2.88	37.20
TREC-CS	2016	61	tags	700-922	10.36	35.26

5 EXPERIMENTAL SETUP

We evaluate our PRF approach on two different tasks - a) standard ad-hoc IR, where our proposed feedback algorithm works with the automatic query variants generated with the local co-occurrence information (WSRM), and b) POI recommendation, where we leverage information from term-level contextual appropriateness scores to formulate the query variants (WSRM-KB).

5.1 Dataset

For the ad-hoc task, we performed our experiments on TREC 6-8 and Robust topic sets comprising 150 and 99 topics respectively. The target documents collection is TREC ad-hoc IR collection from disks 4 and 5 without the congressional records. A summary of the dataset is shown in Table 1. For the POI recommendation task, we use the TREC-CS 2016 dataset (phase-1 setup) [14]. The task requires a system to return a ranked list of 50 POIs from a given query collection (user profiles), that best fit the user preference history and the user's current contextual constraints. A user's contextual constraint is a 3-dimensional vector of categorical values (corresponding to non-location type trip qualifiers) as outlined in Table 2. The overall collection comprises over 1.2M of POIs in total, and the number of context queries used in our experiments is 61 (part of the TREC-CS 2016 dataset).

UQV Dataset for manually obtained query variants. In a more realistic use-case, the only information available to an IR model is a single query (as entered by a user). Our PRF algorithm WSRM constructs the variants automatically by employing random walks on the local co-occurrence matrix of top retrieved documents. Recent literature has investigated the effectiveness of feedback models on manually formulated query variants, e.g. using the UQV dataset. In this dataset, given a manually constructed back-story (a narrative illustrating the information seeking situation) corresponding to a TREC query, participants were asked to formulate queries. These queries were then post-processed (e.g. duplicates removed, spelling errors corrected etc.) and released as a resource for the purpose of conducting experiments with query variants.

Although the pre-existence of query variants represents a somewhat unrealistic experiment setup, nonetheless for the sake of comparing our proposed feedback approaches with the other feedback methods reported in the literature, e.g. [7, 20], we also conduct PRF experiments on manually formulated variants.

Table 2: TREC-CS trip-qualifier categories with their values.

Categories	Values
trip-type	{business, holiday, other}
trip-duration	{day-trip, longer, night-out, weekend-trip}
accompanied-by	{alone, family, friends, other}

5.2 Baselines and Parameter Settings

Single-Query Baselines. Some of the standard baseline approaches are only able to make use of a single query for retrieving a ranked list of documents. These baselines include **BM25** and the standard relevance model, **RM** ('RM3' version) [16, 17].

Top-Document Set Permutation Baselines. Instead of blindly assuming that the top-M retrieved documents are useful for relevance feedback, our method essentially relies on permuting this set of top documents (based on the rank aggregation scores of Equation 7) and select a new top set (M') of documents for feedback. To demonstrate the effectiveness of this method, we undertake a number of baselines that employ some form of a document reordering mechanism to choose a set of documents, different from the top-retrieved ones.

A simple such permutation function is to sort the top-M retrieved set of documents by document length, and then select the top M' ones for feedback ($M' < M$). Since the input to the selection function is the set of documents that are retrieved within the top M ranks, they have high similarity scores with the query. A further filtering based on their lengths may serve as a useful heuristic to choose the ones that could potentially improve feedback. A different choice of the permutation order yields two different baselines.

(1) 'Shortest Document First' (**SDF**), which assumes that the shortest documents will be more useful for feedback because they are more likely to be focused on the query topic.
(2) 'Longest Document First' (**LDF**), which assumes that the longest documents will be more useful for feedback because they are likely to contain a higher number of terms that eventually could be useful to enrich the initial query.

Clustering-based Resampling Baseline. The clustering based resampling method, proposed in [6, 18], employs a document neighborhood induced permutation on the top retrieved M documents. Specifically, the method involves finding neighborhoods of documents (called 'overlapping clusters' by the authors of [18]). The method assumes that *dominant* documents for a query are the ones with several nearest neighbors with high similarities, i.e., the neighborhoods with the highest aggregated retrieval scores (essentially assuming that such a neighborhood effectively represents the core topic of the information need). Since this cluster based resampling method estimates a new set of documents that is used for feedback, we employ this approach as another baseline, which we call '**kNN**'.

In fact, in addition to selecting the documents for feedback, since the cluster-based resampling method also involves making use of a cluster-based smoothed query likelihood model, for a fair comparison with our approach, we incorporate the neighborhood-based smoothed mechanism for computing the maximum likelihood estimates (MLEs) of the local co-occurrences (Equation 3). Specifically, instead of using Equation 4 for computing the MLEs at the level of documents, we employ

$$P(w|C) = \lambda \frac{f(w, C)}{|C|} + (1 - \lambda) \frac{f(u)}{f(.)}, \tag{12}$$

which differs from Equation 4 in that it samples terms from the bag-of-words representation of a neighborhood (overlapping cluster), C, of documents. Applying Equation 12 for computing the local co-occurrence graph, subsequently followed by a random walk based query variant generation and rank aggregation for selecting the feedback document set, constitutes *a variant of our proposed method* for relevance feedback, which we call 'Cluster-based Weakly Supervised RM' (**CWSRM**).

Fusion Baselines for Single and Multi-Queries. A recent work [20] shows that both the approaches of - a) combining separate relevance models estimated with each input query (variant) as a single feedback model **AriRM**, and b) separately executing feedback models on the individual query variants and then finally merging the results **MultiRM**, improve retrieval effectiveness. To investigate if our proposed rank aggregation method of document selection for relevance feedback is effective, as baselines we employ the fusion based approaches AriRM and MultiRM for both single query setup and multi-query setup (i.e. with and without the UQV query variants for the TREC topic sets). In the single query setup ($N = 1$), we applied AriRM and MultiRM on query variants that were generated automatically by our proposed approach. For the multi-query case, we made use of only the supplied query variants from the UQV dataset alone (inclusive of the original TREC query) to fuse the feedback model (AriRM), or the result-lists (Multi-RM).

The parameters of each method, namely - a) (k, b) for BM25, b) number of clusters, $|C|$ for kNN, c) the number of feedback documents and terms, (M, T) respectively, for RM, kNN, AriRM, and MultiRM, and d) the number of feedback documents (in the second-stage after document selection) and terms (M', T) respectively for WSRM and CWSRM - were tuned individually by grid search on the TREC-8 dataset with respect to the metric P@5. The decision to use TREC-8 as the development dataset was arbitrary. The optimal parameter settings (as obtained on the development dataset) were then applied for each method on the rest of the topic sets, namely TREC 6, 7 and Robust.

5.3 POI Recommendation Settings

Similar to the ad-hoc IR setup, for contextual suggestion we also employ BM25 and RM as the standard baselines. Since a factored version of relevance model (FRM) has been shown to be effective for the contextual suggestion task [12, 13], we employ this method as one of our baselines. Concretely speaking, FRM [12, 13] first enriches the user history and tags to better match the POI descriptors, and then follows this up with a standard RM feedback on POI descriptors using this enriched user history.

To investigate if rank aggregation on automatically generated query variants can improve FRM, we investigate two variants of the weak supervised RM (WSRM) for the contextual suggestion experiments. First, we investigate **WSRM**, which uses Equation 11 to constitute the query variants by leveraging information from the knowledge base of manually assessed appropriateness scores.

Table 3: Ad-hoc IR relevance feedback experiments with *single queries as input*, i.e. without using manually annotated query variants on the TREC ad-hoc IR topic sets. Parameters for each method were tuned separately on TREC 8, and then each method was tested on the remaining topic sets with the optimal parameter configurations. Statistical significance of the proposed methods (WSRM and CWSRM) in comparison to the three most effective baselines - kNN, RM and AriRM, are denoted with '*', '†' and '‡', respectively (t-test with $p = 0.05$).

	Params	Development Set			Test Set								
	tuned on	TREC 8			TREC 6			TREC 7			TREC Rb		
Method	dev set	P@5	P@10	MAP	P@5	P@10	MAP	P@5	P@10	MAP	P@5	P@10	MAP
BM25	k=0.5, b=0.5	0.4960	0.4780	0.2619	0.4680	0.4280	0.2306	0.4760	0.4400	0.1943	0.5051	0.4404	0.2896
RM	M=3, T=160	0.5360	0.5020	0.2803	0.4680	0.4400	0.2504	0.5080	0.4840	0.2284	0.5354	0.4657	0.3292
SDF	M=20, T=160	0.5200	0.4960	0.2685	0.4520	0.4220	0.2343	0.4200	0.4000	0.1928	0.4909	0.4465	0.3004
LDF	M=20, T=160	0.4400	0.4000	0.2429	0.3680	0.3240	0.2019	0.4240	0.3720	0.1834	0.4061	0.3455	0.2332
kNN	$\|C\|$=2, T=100	0.5680	0.5200	0.2847	0.4600	0.4320	0.2455	0.4840	0.4340	0.2186	**0.5576**	**0.4859**	0.3377
AriRM	M=3, T=160	0.5360	0.4760	0.2480	0.4600	0.3880	0.2170	0.4640	0.4380	0.2222	0.5212	0.4465	0.3076
MultiRM	M=3, T=160	0.5280	0.4820	0.2392	0.4440	0.3960	0.2164	0.4680	0.4220	0.2142	0.5111	0.4303	0.2996
WSRM	M'=3, T=160	0.5680†‡	0.5200‡	**0.2887**	**0.5120***†‡	**0.4540***‡	**0.2600**	**0.5320***†‡	**0.4880***‡	**0.2387**	**0.5576**†‡	0.4727	0.3340
CWSRM	M'=3, T=160	**0.6000***†‡	**0.5340**†‡	0.2849	0.4840*‡	0.4360‡	0.2480	0.4840	0.4540	0.2205	0.5455	0.4808	**0.3415**

As the second variant of our proposed approach for contextual recommendation, we investigate **WSFRM**, which is the factored counterpart of WSRM (as FRM is to RM), i.e. instead of applying WSRM only for generating query variants and rank aggregating the retrieved POIs for better feedback document (POI description) selection, we also apply WSRM to enrich the information in the user context as well. The weights in the local co-occurrence graph are estimated with the ψ function representing the manually annotated contextual appropriateness scores (Equation 11). The parameters for each method were tuned independently by conducting a grid search with respect to the metric nDCG@5.

Table 4: Examples of automatically constructed query variants (stemmed words) for two sample queries of TREC 8.

Query: `foreign minor germany`
`foreign germani feder european minor dai govern germani romania mar great minor practic poland minor polici type union econom past kinkel`
Query: `behavioral genetics`
`behavior determin problem thoma state time genet twin genet gene embryo part behavior time children behavior profil parent genet famili environ`

6 RESULTS

6.1 Ad-hoc IR Experiments

Without Manual Query Variants. From Table 3, we observe the following. First, although RM improves MAP substantially for each topic set, it is seen that the improvements in P@5 are marginal even when compared to a relatively simple baseline such as SDF (e.g. compare the TREC-8 P@5 values for RM and SDF). This indicates that a more effective approach may potentially improve precision at top ranks even further. This, conforming to observations of previous studies [18, 28], also confirms that a more robust document selection approach could potentially improve PRF quality.

Second, it is observed that the baseline method kNN [6, 18] is able to substantially improve precision at top ranks (as compared to RM). This reinforces the importance of effectively selecting the set of feedback documents. In fact, our proposed method, WSRM, achieves comparable results with that of kNN. However, an important point to observe is that kNN does not generalize well to the test topic sets (TREC 6 and 7), which indicates that this method is overly sensitive to the choice of its parameters. On the other hand, the facts that WSRM achieves similar effectiveness on the development set and that it also generalizes well on the test data indicates that WSRM is more resilient to parameter variation effects. This also confirms that leveraging information from *rank aggregation statistics* offers a better way to select the candidate set of documents for relevance feedback in comparison to the *neighborhood-based estimation* in kNN for a document's likely usefulness for feedback.

Third, somewhat to our surprise, we observed that the retrieval effectiveness of the pre-fusion and post-fusion based feedback methods (i.e., AriRM and MultiRM respectively) was not satisfactory (compare the AriRM and MultiRM results with those of RM for each topic set). This corroborates the fact that fusion based approaches tend to work well with manually annotated query variants, when each query variant points to the exact same information need.

Finally, we observe that the neighborhood based smoothing of [18] for estimating the local co-occurrences eventually help to further improve the quality of relevance feedback (as can be seen from the CWSRM results of Table 3 in comparison to the WSRM ones). However, the combination method does not generalize well for the test sets of topics. This happens due to the percolating parameter sensitivity effect of kNN method onto CWSRM. As an illustrative example for the quality of the automatically generated query variants, Table 4 shows these variants obtained with WSRM on two TREC-8 topics.

With Manual Query Variants (UQV data for TREC Robust). Table 5 shows that with a small number of query variants, the

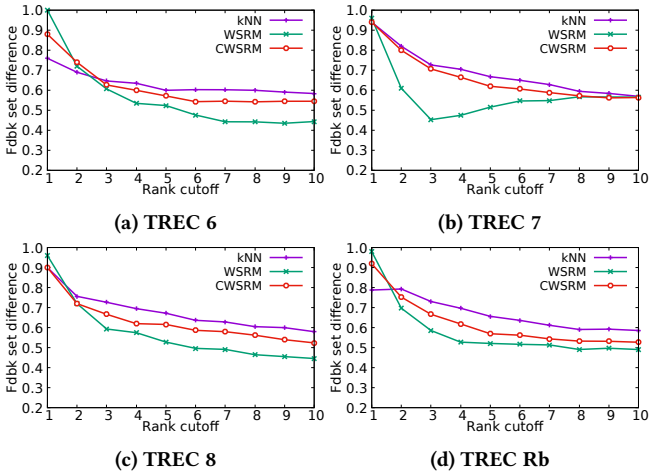
Table 5: Comparisons between WSRM and AriRM/MultiRM on the UQV manual query variants. Significance of WSRM (t-test with $p = 0.05$) is shown with * (AriRM) and † (MultiRM).

Dataset	Method	Parameters	P@5	P@10	MAP
TREC 6	AriRM	$M = 3, T = 160$	0.5840	0.5160	**0.2882**
	MultiRM	$M = 3, T = 160$	0.5760	0.4920	0.2823
	WSRM	$M' = 3, T = 60$	**0.6000**†	**0.5220**†	0.2757
TREC 7	AriRM	$M = 3, T = 160$	**0.6680**	**0.5760**	**0.3000**
	MultiRM	$M = 3, T = 160$	0.6640	0.5660	0.2939
	WSRM	$M' = 3, T = 60$	0.6400	0.5620	0.2666
TREC 8	AriRM	$M = 3, T = 160$	0.6360	0.5800	**0.3279**
	MultiRM	$M = 3, T = 160$	0.6280	0.5840	0.3233
	WSRM	$M' = 3, T = 60$	**0.6600***†	**0.5920**	0.3170
TREC Rb	AriRM	$M = 3, T = 160$	**0.6828**	**0.5848**	**0.4237**
	MultiRM	$M = 3, T = 160$	0.6707	0.5727	0.4110
	WSRM	$M' = 3, T = 60$	0.6586	0.5556	0.3901

fusion-based approaches usually work well in practice. We failed to notice any consistent trends in the results from Table 5. A reason for this could be the fact that since manual variants are *good quality* alternate representations of an information need, the results achieved by the fusion based models exhibit a *saturation effect* in the results making it difficult to further improve them with automated processing. Despite this saturation effect, some of the results show improvements in a couple of cases, e.g. we notice that WSRM leads to an improvement in the precision at top ranks on TREC-6 and TREC-8 topic sets. As a point of note, it is worth noting that the experiments reported in Table 5 represent a rather unrealistic situation for ad-hoc IR, because it unlikely for a user to enter a number of synonymous representations of his information need.

Feedback-Document Set Analysis. Existing literature has shown that it is necessarily true that either a well filtered set of top-M documents or the set of true relevant documents are the most effective to improve retrieval effectiveness [6, 18, 28]. An interesting question then is to investigate how many new (yet effective) documents, on an average, is a document selection strategy able to bring within the top M' ranks which eventually leads to the improvements in retrieval effectiveness as demonstrated by the WSRM results in Table 3. In other words, as per our terminology, the question becomes - what is the difference between the sets $\mathcal{M}$ and $\mathcal{M}'$? A high value of this difference indicates that a feedback document selection algorithm is able to leverage information even from outside the initial set of top-M documents thus attributing the reasons for improvements to this difference.

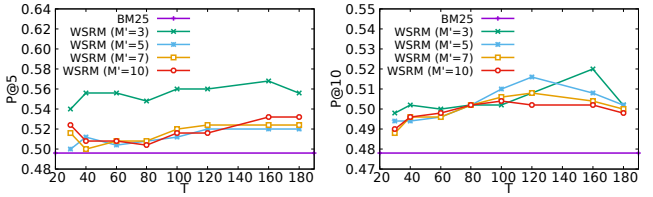
Figure 3 shows the differences between the two sets $\mathcal{M}$ (top-M of initial retrieval, in our case, BM25) and $\mathcal{M}'$ (top-M' after document re-scoring) as obtained by the three feedback document selection methods, namely kNN, WSRM and CWSRM. These differences are measured at a number of different rank cut-off points. The results show that both kNN and (C)WSRM are able to retrieve a fair number of new feedback documents (outside the initial top-M). However, the better MAP values of (C)WSRM (Table 3) indicates that both WSRM and CWSRM achieve the desired trade-off between exploration (leveraging information from new documents) and exploitation (making use of the top-M set). The method, kNN, on the other hand, leads to a more aggressive exploration, which eventually yields lower P@5 and MAP values (as seen from Table 3).



(a) TREC 6 (b) TREC 7

(c) TREC 8 (d) TREC Rb

Figure 3: Comparison of set differences in top ranked documents for 3 different feedback document scoring methods - kNN, WSRM and CWSRM, at specific rank cutoffs, i=1,. . .,10, shown on the x-axis. The set difference values (y-axis) are computed as $(\mathcal{M}'_i - \mathcal{M}_i)/i$, where $\mathcal{M}_i$ ($\mathcal{M}'_i$) represents the set of top-i documents before (after) document re-scoring.



(a) P@5 on TREC 8 (b) P@10 on TREC 8

Figure 4: Effect of precision at top ranks (P@5, P@10) with respect to changes in #feedback documents (M') and #expansion terms (T) used in WSRM estimation.

Sensitivity Analysis. Figure 4 shows the parameter sensitivity effects of WSRM on precision at top ranks for the development set, i.e. TREC-8 topics. We observe that selecting a small number of feedback documents after re-scoring helps achieve the best results, which in turn shows that our approach of document selection by rank aggregation over query variants is effectively able to filter out useful information for relevance feedback at the very top ranks.

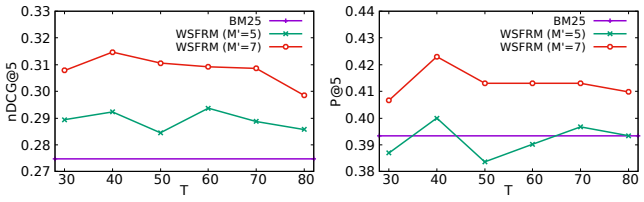
6.2 Contextual Recommendation Experiments

Similar to the observations for the ad-hoc task, Table 6 shows that our approach improves the POI retrieval effectiveness (particularly, precision at top-ranks) for the contextual recommendation task. It can be seen that our proposed approach, WSRM, and its factor-based variant, WSFRM, outperform both RM and FRM. This

Table 6: Comparisons between our proposed approaches (WSRM and WSFRM) and the baselines on the TREC-CS data. Significance between the differences between WSFRM and FRM is denoted by '*' (t-test with $p = 0.05$)

Method	Optimal Params.	nDCG@5	nDCG	P@5	MAP
BM25	$k = 1.1, b = 0.3$	0.2747	0.2889	0.3934	0.1326
RM	$M = 5, T = 25$	0.2615	0.3091	0.3574	0.1437
FRM	$M = 5, T = 25$	0.2919	0.3418	0.3934	0.1616
WSRM	$M' = 5, T = 30$	0.2746	0.3214	0.3738	0.1520
WSFRM	$M' = 7, T = 40$	**0.3147***	**0.3576***	**0.4230***	**0.1727***

indicates that automatic generation of query variants and then using the retrievability measure on them to construct the feedback set works better in the presence of true prior beliefs about term-level relevance. Being a precision oriented task (because real-life use-case requires that results are to be displayed on mobile devices with limited UI resources), it is particularly interesting to observe the improvement of precision for POI recommendation at the top-ranks ($nDCG@5$) with WSFRM. Figure 5 shows that the trends for parameter sensitivity effects are similar to that of Figure 4.



(a) nDCG@5 on TREC-CS **(b) P@5 on TREC-CS**

Figure 5: Effect of precision at top ranks (nDCG@5, P@5) with respect to changes in #feedback documents (M') and #expansion terms (T) used in WSFRM estimation.

7 CONCLUSIONS

We proposed a concept of weakly supervised relevance models by using the notion of retrievability from automatically constructed query variants to improve the quality of relevance feedback. We observed that our approach consistently improves precision at top ranks in two different tasks, namely TREC ad-hoc and the contextual POI recommendation. As a future exercise, we would like to investigate how effectively can one generate the query variants for verbose queries, and how effective will these verbose variants be for improving retrieval effectiveness.

ACKNOWLEDGMENTS

This work was supported by the ADAPT Centre for Digital Content Technology, funded under the Science Foundation Ireland Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

REFERENCES

[1] Mohammad Aliannejadi and Fabio Crestani. 2018. Personalized Context-Aware Point of Interest Recommendation. *ACM Trans. Inf. Syst.* 36, 4 (2018), 45:1–45:28.
[2] Mohammad Aliannejadi, Ida Mele, and Fabio Crestani. 2017. A Cross-Platform Collection for Contextual Suggestion. In *Proc. of SIGIR '17*. 1269–1272.
[3] Leif Azzopardi and Vishwa Vinay. 2008. Retrievability: An Evaluation Measure for Higher Order Information Access Tasks. In *Proc. of CIKM '08*. 561–570.
[4] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2016. UQV100: A Test Collection with Query Variability. In *In Proc. of SIGIR '16*. 725–728.
[5] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2017. Retrieval Consistency in the Presence of Query Variations. In *Proc. of SIGIR '17*. 395–404.
[6] Shariq Bashir and Andreas Rauber. 2009. Improving Retrievability of Patents with Cluster-Based Pseudo-Relevance Feedback Documents Selection. In *Proc. of CIKM '09*. 1863–1866.
[7] Rodger Benham, J Shane Culpepper, Luke Gallagher, Xiaolu Lu, and Joel Mackenzie. 2018. Towards Efficient and Effective Query Variant Generation. In *Proc. of DESIRES '18*. 62–67.
[8] Rodger Benham, Joel Mackenzie, Alistair Moffat, and J. Shane Culpepper. 2019. Boosting Search Performance Using Query Variations. *ACM Trans. Inf. Syst.* 37, 4 (2019), 41:1–41:25.
[9] Bodo Billerbeck and Justin Zobel. 2004. Questioning Query Expansion: An Examination of Behaviour and Parameters. In *Proceedings of the 15th Australasian Database Conference - Volume 27 (ADC '04)*. 69–76.
[10] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting Good Expansion Terms for Pseudo-Relevance Feedback. In *SIGIR '08*. 243–250.
[11] Ben Carterette, Evangelos Kanoulas, Mark M. Hall, and Paul D. Clough. 2014. Overview of the TREC 2014 Session Track. In *Proc. of TREC 2014*.
[12] Anirban Chakraborty, Debasis Ganguly, Annalina Caputo, and Séamus Lawless. 2019. A Factored Relevance Model for Contextual Point-of-Interest Recommendation. In *Proc. of ICTIR '19*. ACM, New York, NY, USA, 157–164.
[13] Anirban Chakraborty, Debasis Ganguly, and Owen Conlan. 2020. Relevance Models for Multi-Contextual Appropriateness in Point-of-Interest Recommendation. In *Proc. of SIGIR '20*. ACM, New York, NY, USA, 1981–1984.
[14] Seyyed Hadi Hashemi, Charles LA Clarke, Jaap Kamps, Julia Kiseleva, and Ellen M Voorhees. 2016. Overview of the TREC 2016 contextual suggestion track. In *Proc. of TREC '16*.
[15] Ben He and Iadh Ounis. 2009. Finding Good Feedback Documents. In *Proc. of CIKM '09*. 2011–2014.
[16] Nasreen Abdul Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah S. Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. In *TREC 2004*.
[17] Victor Lavrenko and W. Bruce Croft. 2001. Relevance Based Language Models. In *Proc. of SIGIR '01*. ACM, New York, NY, USA, 120–127.
[18] Kyung Soon Lee, W. Bruce Croft, and James Allan. 2008. A Cluster-Based Resampling Method for Pseudo-Relevance Feedback. In *Proc. of SIGIR '08*. 235–242.
[19] Binsheng Liu, Nick Craswell, Xiaolu Lu, Oren Kurland, and J. Shane Culpepper. 2019. A Comparative Analysis of Human and Automatic Query Variants. In *Proc. of ICTIR '19*. 47–50.
[20] Xiaolu Lu, Oren Kurland, J. Shane Culpepper, Nick Craswell, and Ofri Rom. 2019. Relevance Modeling with Multiple Query Variations. In *Proc. of ICTIR '19*. 27–34.
[21] Yuanhua Lv and ChengXiang Zhai. 2009. A Comparative Study of Methods for Estimating Query Language Models with Pseudo Feedback. In *Proc. of CIKM '09*. 1895–1898.
[22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *NIPS '13*. 3111–3119.
[23] Ali Montazeralghaem, Hamed Zamani, and James Allan. 2020. A Reinforcement Learning Framework for Relevance Feedback. In *SIGIR*. ACM, 59–68.
[24] Paul Ogilvie, Ellen Voorhees, and Jamie Callan. 2009. On the number of terms used in automatic query expansion. *Information Retrieval* 12 (12 2009), 666–679.
[25] S.E. Robertson, S. Walker, M.M. Beaulieu, M. Gatford, and A. Payne. 1996. Okapi at TREC-4.
[26] Dwaipayan Roy, Debasis Ganguly, Mandar Mitra, and Gareth J.F. Jones. 2016. Word Vector Compositionality Based Relevance Feedback Using Kernel Density Estimation. In *Proc. of CIKM '16*. 1281–1290.
[27] Procheta Sen, Debasis Ganguly, and Gareth J. F. Jones. 2019. Word-Node2Vec: Improving Word Embedding with Document-Level Non-Local Word Co-occurrences. In *NAACL-HLT (1)*. Association for Computational Linguistics, 1041–1051.
[28] Egidio Terra and Robert Warren. 2005. Poison Pills: Harmful Relevant Documents in Feedback. In *Proc. of CIKM '05*. 319–320.
[29] Hamed Zamani and W. Bruce Croft. 2017. Relevance-Based Word Embedding. In *Proc. of SIGIR '17*. ACM, New York, NY, USA, 505–514.
[30] Hamed Zamani, Javid Dadashkarimi, Azadeh Shakery, and W. Bruce Croft. 2016. Pseudo-Relevance Feedback Based on Matrix Factorization. In *Proc. of CIKM '16*. ACM, 1483–1492.
[31] Oleg Zendel, Anna Shtok, Fiana Raiber, Oren Kurland, and J. Shane Culpepper. 2019. Information Needs, Queries, and Query Performance Prediction. In *Proc. of SIGIR '19*. 395–404.