Discovering Missing Click-through Query Language Information for Web Search

Xing Yi and James Allan
Center for Intelligent Information Retrieval
Computer Science Department
University of Massachusetts, Amherst, MA, USA
{yixing,allan}@cs.umass.edu

ABSTRACT

The click-through information in web query logs has been widely used for web search tasks. However, it usually suffers from the data sparseness problem, known as the missing/incomplete click problems, where large volume of pages receive few or no clicks. In this paper, we adapt two language modeling based approaches to address this issue in the context of using web query logs for web search. The first approach discovers missing click-through query language features for web pages with no or few clicks from their similar pages' click-associated queries in the query logs, to help search. We further propose combining this content based approach with the random walk approach on the click graph to further reduce click-through sparseness for search. The second approach follows the query expansion method and utilizes the queries and their clicked web pages in the query logs to reconstruct a structured variant of the relevance based language models for each user-input query for search. We design experiments with one publicly available query log and two recent sets of the TREC web search tasks on the GOV2 and ClueWeb09 corpora to evaluate different approaches for handling missing click-through information for search. Our results show that using discovered semantic click-through query language features can statistically significantly improve the search performance, compared with the baseline that does not use the discovered information. The combination approach that uses discovered click-through features from both random walk and the content based approach can further improve search performance.

1. INTRODUCTION

Click-through data provide important user preference information (both individual and collective) over the returned web search results and play important roles in designing and improving web search engines. For example, click-through information can be used to derive labeled training data for optimizing web ranking functions used by web search engines [14, 26]; user clicks can be directly used as relevance judgments of the clicked URLs to generate evaluation data for comparing different retrieval approaches[28, 3]; and collective click-through features can be extracted to enhance the ranking models of search engines[30, 1].

Unfortunately, click-through data usually suffer from a data sparseness problem where large volume of queries have few or no associated clicks [9, 11]. This problem may be caused by two related user click behaviors. One is that users may only click a very limited number of pages for a query so that the clicks are not complete; the other one is that users may just browse the returned snippets to fetch some useful information while not clicking any results even they are relevant. Gao et al. [11] referred to these two situations as the *incomplete click* problem and the *missing click* problem, respectively. These problems greatly limit the possibility and reliability of using click-through features for web search. For example, click-through features cannot be extracted for pages with no clicks. To overcome the click-through sparseness, Craswell and Szummer [9] built a query-URL bipartite click graph from a web query log and then proposed a *random walk* algorithm on the graph to discover missing clicks between query nodes and URL nodes. The intuition behind this approach is to use the transitions of the semantic relation between queries and their clicked URLs on the click graph to find plausible missing clicks.

However, the random walk approach can only partially alleviate the click-through sparseness because it requires specific link structures in the click graph to discover missing clicks. For example, URLs (web pages) that have not yet received any clicks in the search history can never be associated with any previously issued queries in the query logs. To address this issue, Gao et al. [11] considered an alternative approach to compute click-through features from sparse click-through data. They introduced a Good-Turing estimator [12] based discounting method to smooth click-through features of web pages, so that pages with no clicks can have very small *non-zero* click-through features computed by discounting the average of the click-through features of all pages that receive exactly *one* click. Intuitively, their approach follows the smoothing approach of computing out-of-vocabulary (OOV) words' probabilities in statistical language models to compute missing click-through features for web pages with no clicks.

Notice that although OOV words and missing clicks of web pages can be both viewed as events unseen in training data and thus handled in a similar way, there is an important difference between the two types of unseen events: we usually have little semantic information about OOV words

CIKM 2011, Glasgow, UK.

while we normally have indexed the content of the web pages that have not received clicks yet. Gao et al.'s smoothing approach does not use any semantic information in the web page content, thus *pages that have completely different content but no clicks will obtain the same smoothed click-through features*. This is counter-intuitive and makes some smoothed features ineffective for ranking. Indeed, in the experiments they found that using the smoothed click-through features extracted from the content of a web page's *click-associated* query strings (called *query-dependent* features by them) helped little for retrieval [11].

To overcome the weaknesses of the random walk approach [9] and the Good-Turing smoothing approach [11], we propose to utilize the content similarity between web pages to address the click-through sparseness problem. Our content based approach is able to discover click-through query language model features that can properly convey semantic information in the content of pages with no clicks to help retrieval.

Specifically, we hypothesize that *web pages that are similar in content may be clicked by web searchers issuing similar queries*. Under this assumption, we introduce a language modeling based technique for discovering a target web page's plausible missing click-associated queries, using the queries that led to the clicks on the target page's *similar* pages. We then use the discovered query language features for search. We further present a way of combining the advantages of both the random walk approach and the content based approach for helping search. In addition, we adapt an alternative approach, based on the relevance-based language models (RM)[18] to address the click-through sparseness issue for search. This approach, Structured Relevance Models, has been used to handle missing fields when searching semi-structured documents[19].

We evaluate the retrieval performance of different approaches using the Microsoft Live Search 2006 search query log excerpt (MS-QLOG), which has been used in some query log studies [28, 3], and two different sets of *adhoc* web search tasks: (1) the ones in the TREC 2004-2005 Terabyte Tracks [8, 7] and (2) the ones in the TREC 2009-2010 Web Tracks [6, 15][1]. Our work has four major contributions: 1) To the best of our knowledge, we are the first to utilize web content similarity to discover missing click-through query language features for improving web search, although content similarity has been used in many other applications. 2) We adapt two language modeling based approaches to address click-through sparseness in the context of using query logs for web search; we also propose combining the content based approach and the random walk approach [9] for computing effective ranking features from query logs that have the click-through sparseness issue. 3) We empirically show that using our approaches can statistically significantly improve web search performance. 4) We empirically compare different approaches of discovering missing click-through query language features for web search and do in-depth analysis on their advantages and weaknesses.

We begin by reviewing some related work. In §3, we describe three approaches for discovering missing click-through query language features. In §4 we present how we use different discovered information for search, then compare and analyze retrieval performance of different approaches using the TREC *ad hoc* web search tasks. Then we conclude in §5.

[1] `http://plg.uwaterloo.ca/~trecweb/2010.html`

2. RELATED WORK

Previous research has encountered the data sparseness problem in click-through data, including the incomplete click problem and the missing click problem, when leveraging web query logs for helping different web search tasks [9, 1, 26]. However, there is relatively little of work directly handling the click-through sparseness. Craswell and Szummer [9] proposed a random walk approach on the query-URL click graph to discover plausible missing clicks. Gao et al. [11] have recently proposed a discounting method inspired by the Good-Turing estimator [12] to smooth click-through features for web pages that have received no clicks. Different from previous work, we propose using web content similarity to address the click-through sparseness. Recently, Seo et al. [27] proposed applying spectral graph analysis on the web content similarity graph to smooth click counts in the query logs and then using the smoothed counts for improving search. Our approach is similar to their approach in terms of using web content similarity to address click-through sparseness; however, we specifically focus on discovering click-through query language features for search.

Our approach is closely related to other similarity-based techniques, such as clustering similar documents for smoothing document language models[16, 21], smoothing documents based on document-content similarity graph [22], and using web content similarity for missing anchor text discovery [31]; however, we focus on enriching web pages' semantic click-through features for web search by using their similar pages' click-associated queries. We further consider combining web content similarity and click graph information to improve discovered missing semantic click-through features for web search. We notice that Li et al. [20] also considered combining web content features and click graph information for mitigating the click-through sparseness they experienced when classifying the web search intents of the queries in the web query logs.

There is significant research on using click-through data in the query log for enhancing web search performance: using query-page click-through pairs to derive labeled training pairs for learning web page ranking functions [14, 26], extracting click-through features and incorporating them into ranking models for web search [30, 1, 11]. The incomplete/missing click problems present major challenges for both approaches of using click-through data for web search. Our research on discovering missing click-through features can benefit the latter research direction in particular.

3. DISCOVERING MISSING CLICK INFORMATION FOR WEB PAGES

We first describe two different approaches for discovering plausible missing click-through query language information for web pages with few or no clicks. Next, we present one way to combine the advantages of the two approaches. We emphasize that in our research we are particularly interested in *obtaining click-through query language features, which can convey some semantic information of the target web page*, for search.

3.1 Finding More Click-associated Queries by Random Walk on Click Graph

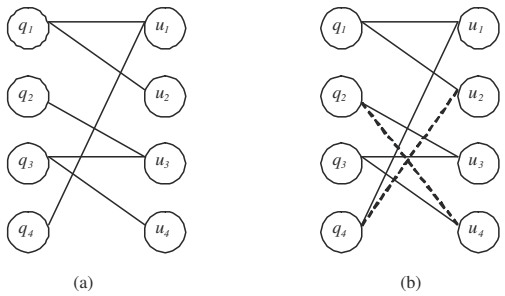


Figure 1: An illustration example of using random walk approach to discover plausible missing clicks (denoted by dashed lines): (a) the original click graph; (b) the link-enriched click graph after applying rank walk algorithm.

We start by reviewing the random walk approach that uses co-clicks in the click-through data to discover plausible missing clicks for web pages [9, 11]. This approach first builds a query-URL *bipartite* click graph from a web query log, by assigning the same query strings/URLs to the same query/URL nodes and linking them according to the click pairs in the query log; then it uses a random walk algorithm to discover plausible missing click edges. Intuitively, this approach assumes there exists close semantic relation among neighbor nodes in the click graph, and uses the transitions of the semantic relation on the graph for missing click edge discovery. For example, in Figure 1, q_1 and q_4 both lead to the clicks on u_1, thus q_1 and q_4 may be semantically related; therefore q_4 may also lead to the click on u_2, which is q_1's clicked URL. Similarly, due to the co-clicks on u_3 by q_2 and q_3, q_2 may also lead to the click on q_3's clicked URL u_4.

Formally, assume a bipartite click graph $G =< Q, U, E >$ is constructed from a set of query nodes $Q = \{q_1 \ldots q_m\}$, a set of web page URL nodes $U = \{u_1 \ldots u_n\}$ and the edges E between the query nodes and the URL nodes. $(q_i, u_j) \in E$ is an edge in G when q_i leads to at least one click on u_j, and $w(q_i, u_j)$ represents the click count associated with the edge (q_i, u_j). We can normalize the $w(q_i, u_j)$ to obtain the transition probability $p(u_j|q_i)$ on the click graph between a query q_i and each of its clicked web page u_j by:

$$p(u_j|q_i) = \frac{w(q_i, u_j)}{\sum_{k \in \{1 \ldots n\}, (q_i, u_k) \in E} w(q_i, u_k)}, \quad (1)$$

and also the transition probability $p(q_i|u_j)$ between a page u_j and each of its click-associated queries q_i by:

$$p(q_i|u_j) = \frac{w(q_i, u_j)}{\sum_{k \in \{1 \ldots m\}, (q_k, u_j) \in E} w(q_k, u_j)}. \quad (2)$$

We can use the above transition probabilities $p(u_j|q_i), p(q_i|u_j)$, $i \in \{1 \ldots m\}, j \in \{1 \ldots n\}$ to compute the probability $p^{(2t)}(q_j|q_i)$ of one query q_i transiting to another one q_j on the click graph in $2t$ steps by the following iterative equations:

# click pairs	#unique queries (query nodes)	#unique URLs (URL nodes)
12,251,067	3,545,174	4,971,990
	# click edges in the graph	
original counts	6,853,498	
random walk($t = 1$)		
$\alpha = 0$ (no noise filtering)	42,999,932	
$\alpha = 0.001$	32,240,647	
$\alpha = 0.005$	24,365,787	
$\alpha = 0.01$	20,265,365	

Table 1: Some summary statistics of the click graph built from the MS-QLOG dataset and different enriched graphs by the random walk approach with different noise filtering parameters.

$$\begin{aligned} &p^{(2t)}(q_j|q_i) = \sum_{k \in \{1 \ldots n\}, (q_j, u_k) \in E} [p(q_j|u_k) p^{(2t-1)}(u_k|q_i)], t \geq 1; \\ &p^{(2t-1)}(u_j|q_i) = \sum_{k \in \{1 \ldots m\}, (q_k, u_j) \in E} [p(u_j|q_k) p^{(2t-2)}(q_k|q_i)], t > 1; \\ &p^{(1)}(u_j|q_i) = p(u_j|q_i), i \in \{1 \ldots m\}, j \in \{1 \ldots n\}. \end{aligned} \quad (3)$$

We can see that longer transition steps can discover transitions to additional queries for a target query q_i while the discovered semantic relation between them becomes weaker and noisier. Thus for effectiveness and efficiency, we follow Gao et al.[11] to set $t = 1$ in our experiments. In order to further filter some plausible noise, we follow their approach and require that the discovered transitions for the target query q_i should satisfy $p^{(2)}(q_j|q_i) > \alpha$, where α is a controlling parameter and tuned empirically on training data for different tasks.

After discovering similar queries for each query using the random walk approach, Gao et al.[11] expanded each web page's click-associated queries with discovered similar queries. In this way, web pages are linked with more semantically related queries so that the incomplete click problem is partially mitigated. Then they used the enriched representation of the click-associated queries for each web page to extract click-through features to improve web search performance.

Table 1 shows some summary statistics of the original click graph and the enriched click graphs by the random walk approach when we use the click pairs in MS-QLOG to build the graph. The first four rows in Table 1 show some summary statistics of the original click graph, indicating the click-through information is very sparse even for the pages that received some clicks – on average, each page only received 2.5 clicks and has about 1.4 unique click-associated queries. The last four rows show the number of click edges in each enriched graph by the random walk approach using different noise filtering parameters, indicating that incomplete click problem can be partially mitigated – on average, the number of the unique click-associated queries of each web page has been raised to 6.5 when $\alpha = 0.001$.

3.2 Discovering Missing Click-associated Queries through Finding Similar Pages

Notice that the random walk approach needs specific click graph structures to discover plausible missing clicks, meaning it cannot handle web pages with no clicks. Therefore, we propose to adapt a content based approach, which was originally proposed by Yi and Allan for addressing the missing anchor text issue in web search [31]. Intuitively, our

approach assumes that web pages that are similar in content may receive clicks from web searchers issuing similar queries. Under this assumption, we aim to discover a query language model for each page, in order to obtain effective missing semantic click-through features to help search.

We adapt Yi and Allan's contextual translation approach of discovering missing anchor text [31] for our task. Briefly speaking, their approach first views the content of web pages as their anchor text's descriptive context and utilizes the contextual translation approach [28] to measure the semantic relation between the anchor text associated with different pages. Given any page P_i and a target page P_0, the semantic relation between their associated anchor text A_i and A_0 is measured by the contextual translation probability $t(A_i|A_0)$, computed from the Kullback-Leibler divergence (KL-div) between the document language models of P_i and P_0. Then they use $t(A_i|A_0)$ to compute a relevant anchor text language model $p(w|A_0)$ for a target page P_0 to discover P_0's plausible missing anchor terms by:

$$p(w|A_0) = \sum_{A_i \in \mathcal{A}} p(w|A_i) \times t(A_i|A_0), \quad (4)$$

where $\mathcal{A}$ denotes the complete anchor text space of all pages and $p(w|A_i)$ is a multinomial distribution of anchor terms (w) over the vocabulary $\mathcal{V}_{\mathcal{A}}$.

Similarly, we first view each page P_i's content as the descriptive context of the page's click-associated queries Q_i and use P_i's document language model, $p_i = \{p(w|P_i)\}$, as Q_i's *contextual* language model, which is computed by applying Dirichlet smoothing [17] on the original un-smoothed document language model:

$$p(w|P_i) = \frac{N_{P_i}}{N_{P_i} + \mu} p_{ML}(w|P_i) + \frac{\mu}{N_{P_i} + \mu} p(w|\mathcal{C}), \quad (5)$$

where $p_{ML}(w|P_i)$ is the maximum likelihood (ML) estimate of observing a word w in the page, $p(w|\mathcal{C})$ is w's probability in the collection $\mathcal{C}$, N_{P_i} is the length of P_i's content and μ is the Dirichlet smoothing parameter.

Then given any page P_i and a target page P_0, we measure the semantic relation between their click associated queries Q_i and Q_0 by their contextual translation probability $t(Q_i|Q_0)$, computed from the KL-div $Div(\cdot||\cdot)$ between their contextual models p_0 and p_i:

$$t(Q_i|Q_0) = \frac{\exp(-Div(p_0||p_i))}{\sum_i \exp(-Div(p_0||p_i))} \propto \prod_w p(w|P_i)^{p(w|P_0)}. \quad (6)$$

The end of Equation 6 is the likelihood of generating Q_0's context P_0 from the smoothed language model of Q_i's context P_i, being normalized by Q_0's context length.

After that, for each given target page P_0, we calculate a relevant (click-associated) query language model (**RQLM**) $p(w|Q_0)$ to discover P_0's plausible click-associated query terms by:

$$p(w|Q_0) = \sum_{Q_i \in \mathcal{Q}} p(w|Q_i) \times t(Q_i|Q_0), \quad (7)$$

where Q_i denotes all the queries that may lead to the clicks on P_i but may be *incomplete* or *missing*, $\mathcal{Q}$ denotes the complete textual space of the click-associated queries of all pages, $p(w|Q_i)$ is a multinomial distribution of query terms (w) over the click-associated query language vocabulary $\mathcal{V}_{\mathcal{Q}}$.

To compute the RQLM $p(w|Q_0)$ in Equation 7, we use each page P_i's click-associated queries originally *observed* in the query log to estimate a query language model $p_{obs}(w|Q_i)$ to approximate $p(w|Q_i)$, which should be estimated ideally from some unknown *complete* set of P_i's all plausible click-associated queries in the query log[2]. In practice, for effectiveness and efficiency we compute the RQLM of the target page P_0 using the click-associated queries of P_0's top-k most similar pages in the query log. This choice is due to two reasons: (1) $t(Q_i|Q_0)$ is very small for other pages thus has less impact on the RQLM; (2) increasing k can increase the number of query samples for better estimating RQLM but also may introduce more noise to degrade the quality of the estimated RQLM. We tune k's value on the training data for each different retrieval task.

3.3 Combining Random Walk Approach and Finding Similar Approach

We can take advantages of both the random walk approach in §3.1 and our content based approach to further reduce the click-through sparseness and calculate better semantic click-through features for search. Here we present one language modeling based way to combine the advantages of two approaches.

We first employ the random walk approach to enrich the original bipartite click graph and discover more click-associated queries for each web page. Then we estimate a query language model $p(w|Q_{aug})$ for each web page from the new added click-associated queries, which we call *augmented* queries, of the page. We also estimate a query language model $p(w|Q_{orig})$ for each page from its click-associated queries originally observed in the query log that has not been enriched by the random walk approach. Next, we employ the mixture model approach [23, 24] to combine two query language models $p(w|Q_{orig})$ and $p(w|Q_{aug})$, and compute a better smoothed query language model $\tilde{p}(w|Q)$ by:

$$\tilde{p}(w|Q) = \gamma p(w|Q_{orig}) + (1-\gamma) p(w|Q_{aug}), \quad (8)$$

where γ is a meta-parameter to control the mixture weight (or prior probability) of each component and can be tuned on training data for different tasks. Then we use the updated query language model $\tilde{p}(w|Q)$ of each page to better approximate the $p(w|Q_i)$ in Equation 7 so that we can better estimate the RQLM $p(w|Q_0)$ of each page to help retrieval.

Next, we describe how we utilize missing click-through query language information discovered by these different approaches to help improve search performance.

4. USING DISCOVERED CLICK-THROUGH INFORMATION FOR WEB SEARCH

We have described several ways to infer click-through information when in situations where a page or query has few clicks. In this section we consider how to use the inferred information and measure its utility. We first present how we utilize discovered missing click-through query language information for retrieval in §4.1, following the language modeling based retrieval framework[25]. Then in §4.2 we consider an approach that expands the query with additional terms, also inferred from the click-through log. For the convenience of discussing different retrieval models and baselines, we start by briefly describing the data and methodology we used for evaluating different approaches.

[2]We will use this fact in §3.3 to combine the random walk approach and the content based approach for discovering missing click-through features.

Mainly due to privacy and security concerns, there are very limited *publicly available* query log data for research. Here we use the MS-QLOG dataset which contains about 12m click-through events and also information of about 15m additional user-issued web queries that received no clicks, sampled from the query log of Microsoft's web search engine during 05/01/2006 to 05/31/2006. We only use the click-through records in this dataset for our experiments.

We use the queries and the relevance judgments in two different sets of the TREC web search tasks to design retrieval experiments. The first set consists of the *adhoc* web search tasks in the TREC 2004-2005 Terabyte Tracks [8, 7], where the GOV2 collection (a TREC web collection crawled from government web sites during early 2004) was used for search; the second set consists of the *adhoc* web search tasks in the TREC 2009 Web Track [6, 15] and the TREC 2010 Web Track[3], where the search was originally performed on the ClueWeb-09 Dataset[4](a larger TREC web collection recently crawled during 01/06/2009 to 02/27/2009 from all domains of the Web).

Because our approach depends on web page content similarity, we crawl the web pages of all the *clicked* URLs in MS-QLOG and use the crawled pages and their click-associated queries in MS-QLOG as the training data for extracting semantic click-through features. The GOV2 collection and the TREC category B subset of the ClueWeb09 web collection (or ClueWeb09-T09B dataset), are used as the searched targets in our experiments. Each ClueWeb09 or GOV2 web page can be viewed as a page whose click information is completely missing[5], thus we need to handle the click-through sparseness problem in both the training pages and the searched collections.

More details about the data and methodology used for evaluating the retrieval performance of different approaches will be described in §4.3.1. Then we will discuss the experimental results in §4.3.2 and §4.3.3.

4.1 Document Smoothing Approach

The first baseline is a query likelihood baseline following the typical language modeling based retrieval approach[25]. This baseline does not use any click-through features and ranks each web page P for a query Q by the likelihood of the page P's document language model $p(w|P)$ generating the query Q:

$$p(Q|P) = \prod_{w \in Q} p(w|P). \tag{9}$$

We use Dirichlet smoothing [17] to compute the document language model $p(w|P)$ used in the above equation and denote this query likelihood baseline **QL** here. We tune the Dirichlet parameter μ for QL to achieve the best retrieval performance for different tasks. Note that μ is fixed to 2500 when using Equation 5 to compute the document models of the crawled clicked pages for estimating RQLMs (relevant click-associated query language models described in §3.2).

We follow the mixture model approach [23, 24] to use the discovered click-through query language model features to help search. After we estimate the RQLM $p(w|Q_0)$ for each page, we mix a web page P's document language model $p(w|P)$ with the RQLM to obtain a better document language model $\tilde{p}(w|P)$ by:

$$\tilde{p}(w|P) = \beta p(w|P) + (1-\beta)p(w|Q_0), \tag{10}$$

where $p(w|P)$ is the original smoothed document model in the QL baseline and β is the meta-parameter controlling the mixture weights of the component distributions. Then we can use the updated document language model $\tilde{p}(w|P)$ and Equation 9 for retrieval.

We have described three different approaches of discovering semantic missing click-through features in §3. We point out *because the searched items here are ClueWeb09 or GOV2 web pages with no click information, only using the random walk approach cannot discover any click-associated queries for them.* Therefore, we do not use the sparse click count (which is *zero* for almost all pages and not helpful for retrieval) in our experiments, but use our content based approach and the combination approach for improving search performance. In the combination approach, we first discover plausible missing links in the click graph (built from MS-QLOG) by the random walk approach and then use the enriched click graph to estimate better RQLMs for the ClueWeb09 or GOV2 pages as described in §3.3. We denote the retrieval baseline that employs our content based approach to update document models for search as **RQLM**, and the baseline that uses combination approach for search as **RW+RQLM** in later discussions.

4.2 Query Expansion Approach

Besides document smoothing approaches, we are also interested in exploring some *query-side* alternative approaches of handling missing click-through information for search. Here we adapt a structured variant of the relevance based language models [18], which was proposed by Lavrenko et al.[19] and called Structured Relevance Models (SRM), for discovering useful click-through query information to reconstruct queries. The SRM technique was originally developed to search *semi-structured* documents with incomplete/missing fields; thus, here we introduce field structure for the queries and pages, represent click-through information using this structure and then utilize the SRM approach for search.

Formally, we view each web page as a semi-structured document containing two fields: (1) the *PageContent* field (denoted by $\mathbf{w}_p$) which contains the original page content and (2) the *QueryContent* field (denoted by $\mathbf{w}_q$) which contains all the click-associated queries of the page in the web query log. Then for each unstructured query q, we generate a semi-structured query $\mathbf{q} = \{\mathbf{w}_p, \mathbf{w}_q\}$ that has the same semi-structure as the web page document by duplicating the query string in both fields, i.e. $\mathbf{w}_p = \mathbf{w}_q = q$: intuitively, *the query is searching for pages that match the query in content and/or their click-associated queries.* We assume that both fields are incomplete and then use the SRM approach to estimate plausible missing field values in $\mathbf{q}$ based on the observed $\{\mathbf{w}_p, \mathbf{w}_q\}$. We use our crawled pages of the clicked URLs in MS-QLOG and their click-associated queries in MS-QLOG to form the *training* semi-structured document collection $\mathcal{W}$.

We then use the training collection to calculate the SRM $\{R_p(\cdot), R_q(\cdot)\}$ for $\mathbf{q}$, where each relevance model $R_i(w)$ specifies how plausible it is the word w would occur in the field i ($i \in \{p, q\}$) of $\mathbf{q}$ given the observed $\mathbf{q} = \{\mathbf{w}_p, \mathbf{w}_q\}$, i.e.

$$R_i(w) = P(w \circ \mathbf{w}_i | \mathbf{q}) = P(w \circ \mathbf{w}_i | \mathbf{w}_p, \mathbf{w}_q), i \in \{p, q\}, w \in \mathcal{V}_i, \tag{11}$$

[3] http://plg.uwaterloo.ca/~trecweb/2010.html

[4] http://boston.lti.cs.cmu.edu/Data/clueweb09/

[5] Some research showed that there is very small overlap between the clicked URLs in MS-QLOG and the GOV2 collection[3].

where $w \circ \mathbf{w}_i$ denotes appending word w to the string $\mathbf{w}_i$ and $\mathcal{V}_i$ denotes the vocabulary of the field i. Using the training web page documents $\mathbf{w}' \in \mathcal{W}$ and Equation 11, $R_i(w)$ can be further calculated by:

$$R_i(w) = \sum_{\mathbf{w}' \in \mathcal{W}} p(w|\mathbf{w}'_i) \times P(\mathbf{w}'|\mathbf{q}), i \in \{p, q\}, w \in \mathcal{V}_i. \quad (12)$$

To calculate the posterior probability $P(\mathbf{w}'|\mathbf{q})$,we use the following equations:

$$P(\mathbf{w}'|\mathbf{q}) \propto P(\mathbf{q}|\mathbf{w}') * P(\mathbf{w}'), \\ P(\mathbf{q}|\mathbf{w}') = P(\mathbf{w}_p|\mathbf{w}'_p)^{\beta_p} * P(\mathbf{w}_q|\mathbf{w}'_q)^{\beta_q}, \quad (13)$$

where $P(\mathbf{w}')$ is assumed to be a uniform distribution, the meta-parameters β_p, β_q are used to control the impact of each field on the posterior probability and tuned with the training queries. When computing $P(\mathbf{w}_i|\mathbf{w}'_i), i \in \{p, q\}$ in Equation 13, we fix the Dirichlet smoothing parameter $\mu_p = 50, \mu_q = 1$ for the *PageContent* and *QueryContent* fields, respectively.[6]

For efficiency and effectiveness we use $\mathbf{q}$'s top-k most similar documents instead of all $\mathbf{w}' \in \mathcal{W}$ to calculate $R_i(w)$. k is tuned with the training queries. Because the click information is completely missing in our two searched target collections $\mathcal{W}''$ (ClueWeb09-T09B and GOV2), the *QueryContent* field is missing there. Therefore, we only use the relevance model $R_p(w)$ of the estimated SRM in the *PageContent* field to search each target collection. We interpolate it with the original query language model to obtain a better relevance model for retrieval:

$$R'_p(w) = \lambda * (p(w|\mathbf{w}_p)) + (1 - \lambda) * R_p(w), \quad (14)$$

which is similar as in the Relevance Model 3 [10]. We use the parameter λ to control the impact of the original query language model on the updated relevance model and tune it with the training queries. Then the searched documents $\mathbf{w}'' \in \mathcal{W}''$ are ranked by their weighted cross-entropy [17] based similarity to $R'_p(w)$:

$$H(R'_p; \mathbf{w}''_p) = \sum_{w \in \mathcal{V}_p} R'_p(w) \log p(w|\mathbf{w}''_p) \quad (15)$$

We denote this query expansion retrieval baseline as **SRM** in our experiments.

For comparison, we also provide the typical highly effective language modeling based query expansion baseline – Relevance Model [18] – in our experiments. We use the version of Relevance Model 3 [10] and denote it as **RM**. Note that different from SRM, here RM *does not use any click-through information* for search: it builds a relevance model from the top results of the QL baseline, which is obtained by running the original query directly against the search target (ClueWeb09-T09B or GOV2); then it mixes the built relevance model with the original query language model (similar as in Equation 14) and ranks the searched pages again using the updated model.

In addition, we consider an approach that combines the advantages of both the RM approach and the combination approach (RW+RQLM in §4.1) for further improving search performance. This approach first uses discovered click-through query information from RW+RQLM to get a better query-likelihood ranked list of pages for a given query, and then uses the top ranked pages to compute a plausibly better relevance model for query expansion and re-retrieval. This approach is similar in spirit to previous research that combines document expansion techniques and RM for further improving search [29]. We denote this approach as **RW+RQLM+RM** in the experiments.

[6]When using some sampled queries in MS-QLOG to search their clicked URLs in our crawled web collection, we found that these smoothing parameters can perform the best, if the user click is directly used as the relevance indicator of a web page. Note that only very sparse, biased and incomplete relevance judgments may be obtained in this way.

4.3 IR Experiments

4.3.1 Data and Methodology

As described at the beginning of §4, we consider two set of retrieval tasks. The first one is performed on GOV2 which contains about 25m U.S. government web pages; the second one is performed on ClueWeb09-T09B which contains about 50m English web pages. We use the Indri Search Engine[7] to index each collection by removing a standard list of 418 INQUERY [4] stopwords and applying Krovetz stemmer.

For the first retrieval task, we use 50 *adhoc* queries (topic id:701-750,title-only) in the TREC 2004 Terabyte Track [7] for train and 50 *adhoc* queries (topic id:751-800,title-only) in the TREC 2005 Terabyte Tracks[8] for test. For the second retrieval task, we use 50 *adhoc* queries (title-only) in the TREC 2009 Web Track [6, 15] for train and 50 queries (title-only) in the TREC 2010 Web Track for test. Moreover, instead of using the whole ClueWeb09 collection as the search target as in the TREC 2010 Web Track, we only use the ClueWeb09-T09B subset here; thus only relevant pages in this subset collection are used for evaluation.

We crawled the web pages of the clicked URLs in the MS-QLOG during June 2010 and use the HTML pages downloadable during that time period and their click-associated queries in the MS-QLOG as the training collection for our experiments. Originally there are about 5m unique clicked URLs in this query log, as shown in Table 1; we successfully crawled about 3m HTML pages of the clicked URLs and indexed them using the Indri Search Engine. We remove 418 INQUERY stopwords and apply Krovetz stemmer during the indexing and call the indexed collection as MS-QLOG-Web, which contains about 21.5 million unique words and 4.1 billion word postings. These training pages are then used to discover click-through query language features for the GOV2 or ClueWeb09 pages. We also preprocess the queries in the MS-QLOG using the same stopwords removing and stemming procedure.

To evaluate the retrieval performance, we calculate typical IR evaluation measurements including Mean Average Precision (MAP), Precision at position k (P@k), Normalized Discounted Cumulative Gain (NDCG) [13]. For the TREC 2009 Web Track queries, we report two additional measurements: *statMAP* and *MPC(30)*, which were used by the TREC community for that track [6] and computed by the TREC evaluation tool *statAP_MQ_eval_v3.pl* [8]; thus, we can compare our results with other researchers' published results on the same query set. Intuitively, both statMAP and MPC(30) measurements are used for addressing the incomplete judgment issue [2]: the former one is a statistical

[7]`http://www.lemurproject.org/indri/`

[8]It is downloadable at: `http://trec.nist.gov/data/web09.html`

	MAP	P@10	P@30	NDCG
QL	0.2617	0.5102	0.4694	0.4829
SRM	$0.2777^{\ddagger}$	$0.5551^{\ddagger}$	$0.5020^{\ddagger}$	0.4945
RQLM	0.2688	$0.5388^{\ddagger}$	0.4796	$0.4927^{\ddagger}$
RW+RQLM	$0.2691^{\dagger}$	$0.5347^{\ddagger}$	$0.4823^{\dagger}$	$0.4933^{\ddagger}$
RM	$0.2824^{\ddagger}$	$0.5449^{\ddagger}$	$0.4966^{\ddagger}$	0.4928
RW+RQLM+RM	$0.2875^{\ddagger}$	$0.5612^{\ddagger}$	$0.5061^{\ddagger}$	$0.4974^{\ddagger}$
	Optimal Parameters:			
QL	$\mu = 1000$			
SRM	$k = 10, N = 50, \lambda = 0.3, \beta_p = 0.99, \beta_q = 0.01$			
RQLM	$k = 100, \beta = 0.95$			
RW+RQLM	$k = 100, \beta = 0.95, \alpha = 0.01, \gamma = 0.6$			
RM	$k = 50, N = 50, \lambda' = 0.2$			
RW+RQLM+RM	$k' = 50, N = 50, \lambda' = 0.2$			

Table 2: Retrieval performance and tuned parameters on the TREC 2004 Terabyte Track queries (train).

version of the MAP measurement and the latter one is a statistical version of the measurement P@30.

In each retrieval task, we first tune the Dirichlet smoothing parameter μ in Equation 5 to obtain the best QL baseline that can achieve the highest MAP with training queries on each searched target collection (GOV2 or ClueWeb09-T09B). Then for both the RQLM baseline (using our content based approach) and the RW+RQLM baseline (using the combination approach), we follow the *reranking* scheme, where we use the updated document language model by each approach to recompute the query likelihood scores of the top-1000 web pages returned by the QL baseline for each query and then rerank the pages. For the RQLM baseline, we tune these two parameters: the number (k) of the similar pages whose click-associated queries are used to compute the RQLM and the mixture weight β in Equation 10. For the RW+RQLM baseline, we tune two additional parameters: the transition probability threshold α (discussed in §3.1) and the query language model updating weight γ in Equation 8. For the SRM baseline, as described in §4.2, we tune the number of the similar pages (k) used to build SRM, the number of terms (N) in each field of the built SRM, the meta-parameters λ in Equation 14 and β_p, β_q in Equation 13. For the RM baseline, we tune the number of top ranked pages (k), the number of terms (N) used to build a relevance model and the mixture weight λ' between the relevance model and the original query model. For the RW+RQLM+RM baseline, we first use the tuned RW+RQLM baseline to obtain a best query likelihood ranked list of pages, then we use this best ranked list to build a relevance model and tune the number of top ranked pages (k'), the number of terms (N) and the mixture weight λ' similarly as in the RM baseline.

In each retrieval task, we tune the parameters of each approach with the training queries, and then test their performance on the test queries.

4.3.2 Results

Table 2 and 3 show the retrieval performance of different approaches with the training and testing queries, respectively, in the first retrieval task. Table 4 and 5 show the retrieval performance of different approaches with the training and testing queries, respectively, in the second retrieval task. The ‡ and † in these tables indicate *statistically* significant improvement over of the QL baseline based on one-sided t-test with $p < 0.05$ and $p < 0.1$,respectively. The * in these tables indicates *statistically* significant improvement over of the highly effective RM baseline based on one-sided t-test with $p < 0.05$. Table 2 and 4 also show the corresponding tuned parameters of each approach in the first and second retrieval task, respectively. We can see from these tables that using click-through query information discovered by different approaches can help to improve web search performance, although their performance is affected in different degree by different query sets. We have the following main observations:

	MAP	P@10	P@30	NDCG
QL	0.3043	0.5560	0.4980	0.5475
SRM	0.3110	0.5700	0.5060	0.5502
RQLM	$0.3161^{\ddagger}$	$0.5960^{\ddagger}$	0.5120	$0.5601^{\ddagger}$
RW+RQLM	$0.3132^{\dagger}$	$0.5840^{\ddagger}$	0.5067	$0.5579^{\ddagger}$
RM	$0.3540^{\ddagger}$	$0.5800^{\ddagger}$	$0.5440^{\ddagger}$	$0.5797^{\ddagger}$
RW+RQLM+RM	$0.3617^{\ddagger *}$	$0.6080^{\ddagger *}$	$0.5580^{\ddagger}$	$0.5866^{\ddagger *}$

Table 3: Retrieval performance on the TREC 2005 Terabyte Track queries (test).

1. Using click-through query language features from MS-QLOG benefit more for the web search tasks on the ClueWeb09 data than the ones on the GOV2 data. This is not surprising because the TREC *adhoc* search tasks on the ClueWeb09 data are, in nature, more similar to real-world web search scenarios as those recorded in MS-QLOG: (1) the ClueWeb09 dataset were crawled from the general web while the GOV2 data was crawled only from government web sites; (2) the TREC Web Tracks queries were created to closely simulate the real-world web search scenarios, while the TREC Terabyte Track queries targeted government web pages in order to have some relevant pages in the GOV2 data.

2. On the test query sets in both retrieval tasks, (a) both RQLM and RW+RQLM performed statistically significantly better than QL in terms of MAP,P@10 and NDCG; (b) RW+RQLM+RM performed statistically significantly better than RM in terms of MAP and NDCG. This result demonstrates that using click-through query language model features discovered by our content based approach can help to improve the web search performance significantly, even compared with a highly effective typical query expansion baseline. This also indicates that our content based approach can effectively alleviate the click-through sparseness problem. In addition, RW+RQLM performed slightly better than RQLM on the training query sets in both retrieval tasks and the test query set in the ClueWeb09 retrieval task, indicating that the combination of our content based approach and the click-graph based random walk approach can further reduce the click-through sparseness and refine the discovered missing click-through features for search.

3. The structured query expansion approach (SRM) achieved very good performance (3^{rd} on GOV2 and 2^{nd} on ClueWeb data) on the training query sets in both retrieval tasks. This shows when the model parameters are carefully tuned, SRM can use click-through information to discover missing query language information to improve the search effectiveness. $\beta_p = 0.99, \beta_q = 0.01$ in the first retrieval task implies that the reconstructed query field content mainly comes from the content of the MS-QLOG-Web pages that have the highest likelihoods of generating the original query.

	MAP	P@10	P@30	statMAP	MPC(30)
QL	0.1951	0.3408	0.3354	0.1732	0.3636
SRM	$0.2258^{\ddagger}$	$0.4388^{\ddagger}$	$0.3959^{\ddagger}$	0.2069	0.4661
RQLM	$0.2107^{\ddagger}$	$0.3714^{\dagger}$	$0.3694^{\ddagger}$	0.1916	0.4215
RW+RQLM	$0.2123^{\ddagger}$	$0.3796^{\ddagger}$	$0.3728^{\ddagger}$	0.1908	0.4359
RM	$0.2113^{\dagger}$	$0.3939^{\ddagger}$	$0.3646^{\ddagger}$	0.1993	0.3970
RW+RQLM +RM	$0.2284^{\ddagger *}$	$0.4408^{\ddagger *}$	$0.3891^{\ddagger *}$	0.2213	0.4299
	Optimal Parameters:				
QL	$\mu = 1000$				
SRM	$k = 5, N = 100, \lambda = 0.4, \beta_p = 0.01, \beta_q = 0.99$				
RQLM	$k = 25, \beta = 0.9$				
RW+RQLM	$k = 25, \beta = 0.9, \alpha = 0.01, \gamma = 0.5$				
RM	$k = 50, N = 200, \lambda' = 0.3$				
RW+RQLM +RW	$k' = 50, N = 200, \lambda' = 0.3$				

Table 4: Retrieval performance and tuned parameters on the TREC 2009 Web Track queries (train).

	MAP	P@10	P@30	NDCG
QL	0.1761	0.2292	0.2451	0.3347
SRM	0.1808	0.2354	0.2556	0.3264
RQLM	$0.1925^{\ddagger}$	$0.2646^{\dagger}$	$0.2708^{\dagger}$	$0.3509^{\ddagger}$
RW+RQLM	$0.1995^{\ddagger}$	$0.2688^{\ddagger}$	$0.2715^{\dagger}$	$0.3526^{\ddagger}$
RM	0.1751	$0.2729^{\ddagger}$	0.2535	0.3233
RW+RQLM+RW	$0.1979^{\ddagger *}$	$0.3125^{\ddagger}$	$0.2958^{\ddagger *}$	0.3401^{*}

Table 5: Retrieval performance on the TREC 2010 Web Track queries (test).

In contrast, $\beta_p = 0.01, \beta_q = 0.99$ in the second retrieval task implies that the reconstructed query field content mainly determined by each query's similar queries' clicked pages, thus the query log information is more helpful for the search task on the ClueWeb09 data. However, we observe that on the test queries SRM achieved little improvement over the QL baseline. We will do more analysis on this issue in §4.3.3 to investigate some plausible causes, such as the sensitivity of the performance of SRM is to the change of its model parameters and irrelevant noise in the training data across different query sets.

4. The typical query expansion approach (RM) failed (i.e. performed worse than QL) on the TREC 2010 Web Track web search task (in Table 5), but leveraging the click-through query information discovered from random walk and our content based approach can make RM more resistent to irrelevant noise in the searched collection and effectively reduce the risk of *topic-drifting*. Indeed, on both training and test queries across different searched collections and retrieval tasks, RW+RQLM+RM approach performed robustly and very well: it achieved the best MAP on 3 of 4 query sets and the second best MAP on the remained query set.

	statMAP	MPC(30)
QL	0.1442	0.3079
Anchor	0.0567	0.5558
Mix	0.1643	0.4812
UDWAxQEWeb	0.1999	0.5010
uogTrdphCEwP	0.2072	0.4966
ICTNETADRun4	0.1746	0.4368

Table 6: Retrieval performance of some published results on the TREC 2009 Web Track *ad hoc* queries.

It is worthwhile to compare our results with some very recently published results on the TREC Web Track *ad hoc* web search tasks on the ClueWeb09-T09B collection (our second retrieval task). Table 6 shows some results on the TREC 2009 Web Track *ad hoc* search task from some participants [15]. The 2nd-4th rows of the table show Koolen and Kamps' results[15] on the same retrieval task when they examined the potential of using existing anchor text in large scale web corpora for helping search. One major difference between their QL baseline and ours is that they used linear smoothing approach while we used Dirichlet smoothing. The 5th-7th rows of Table 6 show the top3 best official TREC submissions for the same retrieval task from other participants. Comparing Table 6 with our results in Table 4, our retrieval approaches that use discovered click-through query language information from sparse click-through data achieved similar or better performance, compared to these top-performing TREC submissions.

To summarize, our content based approach can effectively discover missing click-through query language information to help improving retrieval performance. The content based approach can be combined with the random walk approach to further improve the quality of the discovered language model information from click-through data. The query-side approach of handling click-through sparseness performs very well on some query sets but not on other query sets. The discovered click-through query information can be combined with the typical relevance model to further improve web search performance.

4.3.3 More Analysis

We are concerned about how sensitive different approaches' performance is to the change of their retrieval model parameters. Specifically, for the content based approach (RQLM), we are interested in the number of similar pages needed to build RQLMs for each page to be able to perform reasonably well and how the change of this number will affect the retrieval performance; for the combination approach (RW+RQLM), we are further interested in the impact of the augmented queries discovered by the random walk approach on the performance. For the SRM based approach, we are concerned about the impact of different number of feedback pages used to build SRM and the mixture weight between the original query language and the built SRM on the retrieval performance. As discussed in the previous section, our second retrieval task on the ClueWeb09-T09B collection better simulates the real-world web search scenarios; therefore, here we use this task for our investigation.

Figure 2(a) and (b) depict the model parameters' impact for RW+RQLM with training/testing queries in this retrieval task, respectively, where we fix $\alpha = 0.01, \beta = 0.9$ while varying γ and k. Figure 3(a) and (b) depict the model parameters' impact for SRM with training/testing queries, respectively, where we fix $\beta_p = 0.01, \beta_q = 0.99, N = 100$ while varying λ and k.

From Figure 2, we have the following major observations:

1. Using click-associated queries from about $25 \sim 35$ most similar pages to build RQLM for each page can achieve near optimal retrieval performance on both training/test query sets. Increasing k beyond 35 brings little additional benefit to (or even hurt) the retrieval performance, and only changes the performance very slowly. This property means that in real-world use, for efficiency we need only index click-

through information from a small number of similar pages of each page, without sacrificing the retrieval effectiveness.
2. Using augmented queries discovered by the random walk approach from the click graph can slightly help the retrieval effectiveness. The mixture weight γ's value can be selected between $0.4 \sim 0.6$ across different query sets and the change of this value among this range has little impact on the retrieval performance. This also indicates that the augmented queries discovered by the random walk approach are at least as useful as the click-through query information discovered from the content based approach for search.

Figure 3 shows that the performance of SRM mainly depends on whether the training web pages (the MS-QLOG-Web collection here) contains useful content for helping to search the target web collection (the ClueWeb09-T09B here). For example, on the training queries, using only top-5 feedback pages for re-constructing the *PageContent* query field can achieve very good performance (even better than the RW+RQLM approach); in contrast, on the test queries, the performance improvement is very little. Furthermore, the mixture weight λ also affects SRM's retrieval performance significantly, both within the same query set and across different query sets. λ's choice indicates the quality of the built SRM: the higher quality is the SRM, the smaller λ and less information from the original query language model are needed for better performance.

5. CONCLUSIONS

In this paper, we adapt two language modeling based approaches to address the click-through sparseness problem in the context of using web query logs for helping web search.

Our first approach stems from the contextual translation approaches [28, 31] and uses web content similarity for discovering missing click-through query language model information for web pages with no or few clicks, in order to help search. This approach computes a relevant (click-associated) query language model, called RQLM, for each web page from the click-associated queries of its similar pages in the web query log, and then uses the RQLM to smooth the original document language model of each page for achieving better retrieval performance. Compared with the random walk approach [9], this content based approach does not need to use specific click graph structure to discover missing clicks thus can further mitigate the click-through sparseness. Furthermore, we present a combination approach that takes advantage of both random walk and the content based approach to further improve search.

Our second approach follows the query expansion approach and utilizes the semantic relation between the queries and the content of their clicked URLs in the web query log to reconstruct a structured variant of the relevance based language models, called Structured Relevance Models (SRM)[18, 19], for each user-input query, to help search.

We then demonstrated the effectiveness and compared the performance of different approaches of handling the click-through sparseness problem for web search with two recent sets of TREC *ad hoc* web search tasks. The results showed that discovering missing click-through query language information from click-through data can statistically significantly improve search performance, compared with two retrieval baselines (QL and RM) that did not use the discovered information. The document smoothing approaches (RQLM and RW+RQLM), performed well across different query sets while the query expansion approach (SRM) of using sparse click-through information was more sensitive to model parameter selection and irrelevant noise in the click-through data. The random walk approach complemented the content approach for addressing the click-through sparseness problem: the combination approach performed slightly better than the content-only approach in three of the four query sets in our experiments. In addition, the most complex approach (RW+RQLM+RM) that combines to use the typical query expansion approach (RM) and the discovered click-through query language information from RW+RQLM can achieve significantly better search performance in different search tasks.

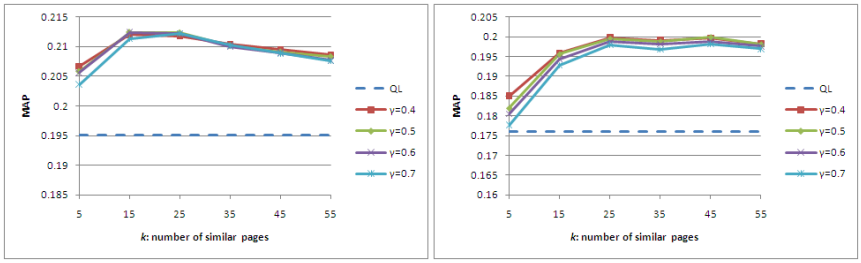
There are several interesting directions of future work. It is worthwhile to explore using the discovered missing click-through query language features beyond the language modeling based retrieval framework, for example, using the discovered features in the learning-to-rank retrieval approach [5], so that we can combine different approaches described here with the Good-Turing based smoothing approach [11] to see whether the retrieval performance can be further improved. Moreover, here we have only explored using the contextual translation probability $p(P_i|P_0)$ between web pages to discover useful missing click-through query language features; theoretically, we may also use this probability to compute an expected feature $E(f_{P_0}) = \sum_{f_{P_i} \in \mathcal{F}} f_{P_i} \times p(P_i|P_0)$ for a page P_0 for any selected click-through feature from the same click-through features f_{P_i} of the page's similar pages P_i. Intuitively, this approach aims to smooth the click-through features for web pages with no clicks as in Gao et al.'s approach, but leverages the web content similarity during the smoothing. We would like to explore the utility of these smoothed click-through features for web search.

6. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF CLUE IIS-0844226, and in part by NSF grant #IIS-0910884. Any opinions, findings and conclusions or recommendations expressed in this material are the author's and do not necessarily reflect those of the sponsor.

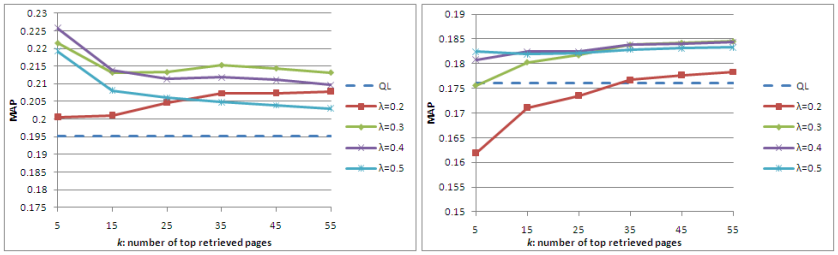
7. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR*, pages 19–26, 2006.
[2] J. A. Aslam and V. Pavlu. A practical sampling strategy for efficient retrieval evaluation. Technical report, 2007.
[3] M. Bendersky and W. B. Croft. Analysis of long queries in a large scale search log. In *Workshop on Web Search Click Data (WSCD 2009)*, pages 8–14, 2009.
[4] J. Broglio, J. P. Callan, and W. B. Croft. An overview of the INQUERY system as used for the TIPSTER project. Technical report, Amherst, MA, USA, 1993.
[5] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of ICML*, pages 89–96, 2005.
[6] C. A. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2009 web track. In *TREC*, 2009.
[7] C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2004 terabyte track. In *TREC*, 2004.
[8] C. L. A. Clarke, F. Scholer, and I. Soboroff. The TREC 2005 Terabyte Track. In *TREC*, 2005.



(a) training (b) testing

Figure 2: The impact of choosing different number (k) of most similar pages and mixture weight γ (between the original click-associated query language model and the language model from the augmented queries discovered by the random walk approach) on RW+RQLM 's retrieval effectiveness. (a) with the training queries; (b) with the test queries.



(a) training (b) testing

Figure 3: The impact of choosing different number (k) of retrieved pages to build SRM and mixture weight λ (between the built SRM and the original query language model) on SRM's retrieval effectiveness. (a) with the training queries; (b) with the test queries.

[9] N. Craswell and M. Szummer. Random walks on the click graph. In *SIGIR*, pages 239–246, 2007.

[10] F. Diaz and D. Metzler. Improving the estimation of relevance models using large external corpora. In *SIGIR*, pages 154–161. ACM, 2006.

[11] J. Gao, W. Yuan, X. Li, K. Deng, and J.-Y. Nie. Smoothing clickthrough data for web search ranking. In *SIGIR*, pages 355–362, 2009.

[12] I. Good. The population frequencies of species and the estimation of population parameters. *Biomerika*, 40(3):237–264, 1953.

[13] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[14] T. Joachims. Optimizing search engines using clickthrough data. In *SIG KDD*, pages 133–142, 2002.

[15] M. Koolen and J. Kamps. The importance of anchor text for ad hoc search revisited. In *SIGIR*, pages 122–129, 2010.

[16] O. Kurland and L. Lee. Respect my authority!: Hits without hyperlinks, utilizing cluster-based language models. In *SIGIR*, pages 83–90, 2006.

[17] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR*, pages 111–119, 2001.

[18] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR*, pages 120–127, 2001.

[19] V. Lavrenko, X. Yi, and J. Allan. Information retrieval on empty fields. In *Proceedings of NAACL-HLT*, pages 89–96, 2007.

[20] X. Li, Y. Wang, and A. Acero. Learning query intent from regularized click graphs. In *SIGIR*, pages 339–346, 2008.

[21] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *SIGIR*, pages 186–193, 2004.

[22] Q. Mei, D. Zhang, and C. Zhai. A general optimization framework for smoothing language models on graph structures. In *SIGIR*, pages 611–618, 2008.

[23] R. Nallapati, W. B. Croft, and J. Allan. Relevant query feedback in statistical language modeling. In *CIKM*, pages 560–563, 2003.

[24] P. Ogilvie and J. Callan. Combining document representations for known-item search. In *SIGIR*, pages 143–150, 2003.

[25] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281, 1998.

[26] F. Radlinski and T. Joachims. Active exploration for learning rankings from clickthrough data. In *ACM KDD*, pages 570–579, 2007.

[27] J. Seo, W. B. Croft, K. Kim, and J. Lee. Smoothing click counts for aggregated vertical search. In *Proceedings of 33st European Conference on Information Retrieval, to appear*, 2011.

[28] X. Wang and C. Zhai. Mining term association patterns from search logs for effective query reformulation. In *CIKM*, pages 479–488, 2008.

[29] X. Wei and W. B. Croft. Lda based document models for ad hoc retrieval. In *Proceedings of ACM SIGIR*, pages 178–185, 2006.

[30] G.-R. Xue, H.-J. Zeng, Z. Chen, Y. Yu, W.-Y. Ma, W. Xi, and W. Fan. Optimizing web search using web click-through data. In *CIKM*, pages 118–126, 2004.

[31] X. Yi and J. Allan. A content based approach for discovering missing anchor text for web search. In *SIGIR*, pages 427–434, 2010.