A Content based Approach for Discovering Missing Anchor Text 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

Although anchor text provides very useful information for web search, a large portion of web pages have few or no incoming hyperlinks (anchors), which is known as the *anchor text sparsity problem*. In this paper, we propose a language modeling based technique for overcoming anchor text sparsity by discovering a web page's plausible missing anchor text from its similar web pages' in-link anchor text. We design experiments with two publicly available TREC web corpora (GOV2 and ClueWeb09) to evaluate different approaches for discovering missing anchor text. Experimental results show that our approach can effectively discover plausible missing anchor terms. We then use the web named page finding task in the TREC Terabyte track to explore the utility of missing anchor text information discovered by our approach for helping retrieval. Experimental results show that our approach can statistically significantly improve retrieval performance, compared with several approaches that only use anchor text aggregated over the web graph.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval–*Search process,Retrieval models*;H.3.5 [**Information Storage and Retrieval**]:Online Information Services–*Web-based services*

General Terms: Algorithms, Experimentation

Keywords: anchor text, anchor text sparsity, language models, relevance models, content similarity, web search

1. INTRODUCTION

There are rich dynamic human generated hyperlink structures on the web. Most web pages contain some hyperlinks, referred to as *anchors*, that point to other pages. Each anchor consists of a destination URL and a short piece of text, which is called *anchor text*. Anchors play an important role in helping web users conveniently navigate to their interested web information. Although some anchor text only functions as a navigational shortcut which does not have direct semantic relation to the destination URL (e.g.,"click here" and "next"), many times anchor text provides succinct description of the destination URL's content, e.g. "SIGIR 2010(Geneva, Switzerland)" is from an anchor linked to `http://www.sigir2010.org/`. Anchor text instances are usually reasonable queries that web users may issue to search for the associated URL and have been used to simulate plausible web queries relevant to the associated web pages in some web search research [15]. Therefore, anchor text is highly useful for bridging the lexical gap between user issued web queries and the relevant web pages. It is arguably the most important piece of evidence used in web ranking functions[14].

	GOV2	ClueWeb09-T09B
# of web pages	25,205,179	50,220,423
# of pages having inlinks	376,121 (1.5%)	7,640,585 (15.2%)
# of pages having original or enriched inlinks[14]	977,538 (3.9%)	19,096,359 (38.0%)

Table 1: Summary of in-link statistics on two TREC web corpora used in our study.

However, previous research has shown that the distribution of the number of inlinks on the web follows a power law [1], where a small portion of web pages have a large number of inlinks while most have few or no inlinks. Thus, most web pages do not have in-link associated anchor text, a problem originally referred to as the *anchor text sparsity problem* by Metzler *et al.* [14]. This problem presents a major obstacle for any web search algorithms that want to use anchor text to improve retrieval effectiveness. Table 1 shows the anchor text sparsity problem in two large TREC[1] web corpora (GOV2[2] and ClueWeb09-T09B[3]). To address this problem, Metzler *et al.* [14] proposed *aggregating*, or *propagating*, anchor text across the web hyperlink graph so that web pages' lack of anchor text can be *enriched* with their linked web pages' associated anchor text. Table 1 shows that the number of URLs associated with some anchor text (original or propagated) in the two TREC web corpora is significantly increased by using their linked-based anchor text enrichment approach. Nevertheless, in Table 1 we notice that large portion of web pages still do not have any associated anchor text after having been enriched. This observation motivated us to consider another possible approach, which utilizes the content similarity between web pages, to alleviate anchor text sparsity.

[1] `http://trec.nist.gov/`
[2] `http://ir.dcs.gla.ac.uk/test_collections/gov2-summary.htm`
[3] `http://boston.lti.cs.cmu.edu/Data/clueweb09/`

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.

Specifically, we hypothesize that the anchor text associated with a web page's inlinks typically has close semantic relations to the web page so that web pages that are similar in content may be pointed to by anchors having similar anchor text. Under this assumption, in this paper we propose a language modeling based technique for discovering a web page's plausible missing in-link anchor text by using its most similar web pages' in-link anchor text. We then test the effectiveness of our approach by using the discovered missing anchor text information for some TREC web search tasks. We find that even on the GOV2 data where a serious anchor text sparsity problem exists as shown in Table 1, our approach can significantly improve retrieval performance. Our content based approach can be combined with the hyperlink based approach to further reduce anchor text sparsity and benefit web search. Our enriched document and anchor text representations can also be used for many other tasks beyond web search, including estimating better document models and extracting advanced textual features for content match and document classification.

Our work has four chief contributions: 1) although content similarity has been used widely in other applications, we are the first to propose using web content similarity to address the anchor text sparsity problem. 2) We develop a language modeling based technique, which stems from ideas in one effective retrieval technique – relevance based language models [10], to effectively discover plausible missing anchor text information and use it for retrieval. 3) We empirically show that our approach performs better than Metzler *et al.*'s linked-based approach [14] in terms of discovering plausible missing anchor terms in two standard large TREC web corpora. 4) We show that our approach statistically significantly improves retrieval effectiveness, compared with several approaches that only use aggregated anchor text over the web graph, in the web named page finding task of the TREC Terabyte track [4].

We begin by reviewing related work in §2. Next, we describe different approaches of discovering missing anchor text to enrich document representations in §3. Then we describe the experimental setup and results of evaluating different approaches for anchor text discovery in §4. After that, we present how to use discovered anchor text information for retrieval in a language modeling approach and report the experimental results in §5. We conclude in §6.

2. RELATED WORK

Metzler *et al.* [14] first directly addressed the anchor text sparsity problem by using the web hyperlink graph and propagating anchor text over the web graph. Our work also addresses the same problem but using a different approach, which is based on the content similarity between web pages. Our approach is similar in nature to other similarity based techniques, such as cluster-based smoothing from the language modeling framework[8, 9, 11], except we focus on enriching web documents' anchor text representation by using their similar documents' associated anchor text.

Anchor text can be modeled in many different ways. Westerveld *et al.* [20] and Nallapati *et al.* [15] model anchor text in the language modeling approach [17] and calculate an associated anchor language model to update the original document model for retrieval. Fujii [6] further considers differently weighting each line of anchor text associated with the same page thus obtaining a more robust anchor language model. Here, we also adopt the language modeling approach but focus on discovering a plausible associated anchor language model for web pages with no or few inlinks. Our approach can be easily used together with any language model based retrieval model (e.g., Ogilvie and Callan's model [16]) that takes document structure into account.

Our approach of overcoming anchor text sparsity stems from ideas in the relevance based language models(RMs), proposed by Lavrenko and Croft [10]. Their original work introduces the RMs to find plausible useful terms missing in the original query for query expansion. Here we adapt the RMs to compute a web content dependent associated anchor language model for discovering missing anchor terms and using anchor text for retrieval. Thus, our approach, although similar in spirit to, differs from document expansion [18] and graph-based document smoothing[13].

3. DISCOVERING MISSING ANCHOR TEXT

We now describe three different approaches for discovering plausible missing anchor text for web pages with few or no inlinks. The goal of each is to produce a ranked list of plausible anchor text terms for a page.

3.1 Aggregating Anchor Text

To overcome anchor text sparsity, Metzler *et al.*[14] originally proposed to augment web pages with *auxiliary anchor text* (denoted as A_{aux}) that is derived by aggregating anchor text over the web graph. We first briefly review the procedure they have used to build A_{aux}, which is very important for our discussions and comparisons in this research. Given a web page P_0's URL u_{P_0}, the procedure first collects all pages $P_{In}(P_0)$, within the same site (domain), that link to u_{P_0}. These links are known as u_{P_0}'s *internal inlinks*. Then the procedure collects all anchor text A from pages (denoted as $P_{Aux}(P_0)$) that are linked to any page in $P_{In}(P_0)$ from outside the site. The anchor text set A is known as *external anchor text* and is used as A_{aux} for u_{P_0}.

Figure 1 illustrates the procedure by using a real-world example from the TREC GOV2 collection. We collect the auxiliary anchor text A_{aux} for the page P_0. P_0's *original anchor text* (denoted as A_{orig}), which comes from all pages (denoted as $P_{Orig}(P_0)$) that are *directly* linked to P_0 from outside the site, consists of lines including "Optima National Wildlife Refuge" and "Optima NWR". P_0's A_{aux} consists of lines including "Oklahoma Refuge Websites" and "Oklahoma National Wildlife Refuges".

Note that the above procedure does not use any anchor text associated with internal inlinks, because internal inlinks are typically generated by the owner of the site for navigational purposes and their associating anchor text tends to be navigational in nature (e.g., "home","next page", etc.; refer to [14] for more discussions on this issue). We emphasize that in the remainder part of this paper we follow Metzler *et al.* and do not use the anchor text associated with internal inlinks in any way.

In this paper we are specifically interested in the effectiveness of using A_{aux} to serve as a surrogate for possibly missing original anchor text. In other words, we consider how effectively we may use A_{aux} to discover plausibly missing original anchor text of the URL of the interest so that anchor text sparsity can be effectively reduced. Therefore, we focus on the discovered anchor terms themselves in the A_{aux}. We use two typical methods to rank the relative importance of each anchor term w. The first method, denoted as **AUX-TF**, is to use each term w's term frequency $tf_{aux}(w)$ in the A_{aux}.

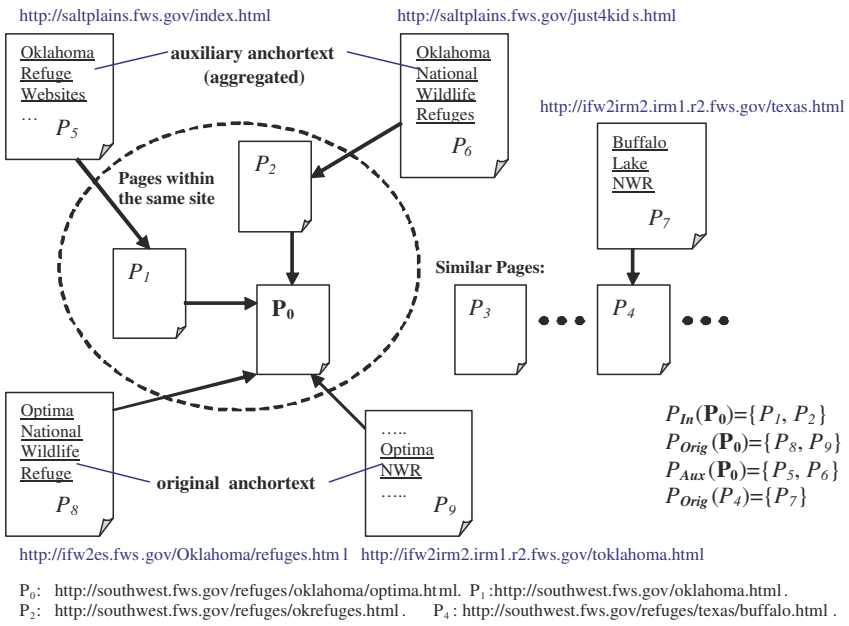

Figure 1: Illustration of how to aggregate anchor text over the web graph or use similar web pages' anchor text for discovering more anchor text for a web page (P_0 in this example). The page P_0 is a GOV2 web page, whose DocID is GX010-01-9459902 and URL is http://southwest.fws.gov/refuges/oklahoma/optima.html.

The second method, denoted as **AUX-TFIDF**, is to use each term w's $tf_{aux} \cdot idf(w)$ score, computed by multiplying $tf_{aux}(w)$ with w's idf score in the web collection. The quality of the discovered anchor term rank lists produced from these two link based approaches implies the effectiveness of using auxiliary anchor text as a surrogate of missing original anchor text. We will compare these two approaches with our content based approach in §4.

3.2 Discovering Anchor Text through Finding Similar Web Pages

Note that in the link based approach, a web page P_0 still cannot obtain the auxiliary anchor text if it has no internal inlinks or if all pages in its $P_{In}(P_0)$ have no external anchor text. Indeed, Metzler *et al.* reported only 38% anchor text sparsity reduction on a web sample with the link based approach[14]. Therefore, we propose a content based approach, which does not have specific link structure requirements on the target web page, to discover its plausible missing anchor text. Intuitively, our approach assumes that web pages that are similar in content may be described by similar associated anchor text. For example, in Figure 1, the target page P_0, which is about Optima national wildlife refuge, is similar in content with the page P_4, which is about Buffalo Lake national wildlife refuge. We observe that the anchor term "NWR", which appears in P_0's and P_4's A_{orig} but not in P_0's A_{aux}, can be used to partially describe both P_0 and P_4 although two pages are concerned about different places.

We consider a language modeling approach to better use document similarity and anchor text information, based on ideas from the relevance-based language models (**RM**)[10]. In brief, given a query q, RM first calculates the posterior $p(D_i|q)$ of each document D_i in the collection $\mathcal{C}$ generating the query q, then calculates a query dependent language model $p(w|q)$:

$$p(w|q) = \sum_{D_i \in \mathcal{C}} p(w|D_i) \times p(D_i|q), \tag{1}$$

where w is a word from the vocabulary $\mathcal{V}$ of $\mathcal{C}$. Similarly, given an target page P_0, our approach aims to calculate a relevant anchor text language model (**RALM**) $p(w|A_0)$ by:

$$p(w|A_0) = \sum_{A_i \in \mathcal{A}} p(w|A_i) \times p(A_i|A_0), \tag{2}$$

where A_i denotes the complete original anchor text that should be associated with P_i but may be *missing*, $\mathcal{A}$ denotes the complete original anchor text space for all pages, $p(w|A_i)$ is a multinomial distribution over the anchor text vocabulary $\mathcal{V}_{\mathcal{A}}$. To compute $p(A_i|A_0)$ in Equation 2 where A_0 and A_i information may be missing, we view each page P_i's content as its anchor text A_i's context and use P_i's document language model $p_i = \{p(w|P_i)\}$ as A_i's contextual model. Then we can calculate a translation model $t(A_i|A_0)$ by using A_0 and A_i's contextual models and use $t(A_i|A_0)$ to approximate $p(A_i|A_0)$. This contextual translation approach is also used in Wang and Zhai's work [19].

When calculating a page P_i's document language model $\{p(w|P_i)\}$, we employ Dirichlet smoothing on the maximum likelihood (ML) estimate of observing a word w in the page ($p_{ML}(w|P_i)$) with the word's collection probability $p(w|\mathcal{C})$:

$$p(w|P_i) = \frac{N_{P_i}}{N_{P_i} + \mu} p_{ML}(w|P_i) + \frac{\mu}{N_{P_i} + \mu} p(w|\mathcal{C}), \tag{3}$$

where N_{P_i} is the length of P_i's content and μ is the Dirichlet smoothing parameter ($\mu = 2500$ in our experiments). Then given two pages P_0 and P_i, we use the Kullback-Leibler divergence (KL) $Div(\cdot||\cdot)$ between their document models p_0 and p_i to measure their similarity and view that as the contextual similarity between the associated anchor text A_0 and A_i. Then the contextually based translation probability $t(A_i|A_0)$ is calculated by:

$$t(A_i|A_0) = \frac{\exp(-Div(p_0||p_i))}{\sum_i \exp(-Div(p_0||p_i))}. \tag{4}$$

This $t(A_i|A_0)$ is then used to approximate $p(A_i|A_0)$ in Equation 2 to get:

$$p(w|A_0) \approx \sum_{A_i \in \mathcal{A}} p(w|A_i) \times t(A_i|A_0). \quad (5)$$

A few transformations of Equation 4 can obtain:

$$t(A_i|A_0) \propto \prod_w p(w|P_i)^{p(w|P_0)}, \quad (6)$$

which is the likelihood of generating A_0's context P_0 from A_i's context P_i's smoothed language model and being normalized by A_0's context length. This likelihood can be easily obtained by issuing P_0 as a long query to any language model based search engine. In addition, we use the observed *incomplete* original anchor text language model $p_{obs}(w|A_i)$ associated with P_i to approximate $p(w|A_i)$ in Equation 5, and let $p_{obs}(w|A_i) = 0$ if P_i has no A_{orig}. In this way, the RALM $p(w|A_0)$ can be computed.

In practice, for efficiency the RALM of the target page P_0 is computed from P_0's top-k most similar pages' A_{orig} (original anchor text) because $t(A_i|A_0)$ in Equation 4 is very small for the other pages. Due to the anchor text sparsity, we set $k = 2000$ in our experiments. Because some of these similar pages do not have associated A_{orig}, we use another parameter m to denote the number of most similar pages whose associated original anchor text is not missing and contributes information in the RALM, and we tune m in the experiments. Intuitively, increasing m can increase the number of anchor text samples to better estimate RALM but may also introduce more noise when the sample size is large.

The probability $p(w|A_0)$ of an anchor term w in the RALM directly reflects the goodness of the term w used as original anchor text for the page P_0, thus we use the anchor terms that have the largest probabilities $p(w|A_0)$ in the RALM to evaluate the effectiveness of our content based approach. Theoretically our approach can associate any web page with some anchor term distribution information if there is some anchor text in the corpus, thus it can further reduce the anchor text sparsity.

3.3 Using Keywords as Anchor Text

The keyword based approaches come from the intuition that important keywords in a web page may be good description terms for the page, thus may be arguably used as anchor text. We use three typical term weighting schemes to identify the keywords and rank the words in a web page's content. The first method, denoted as **DOC-TF**, uses each word w's term frequency $tf_{P_0}(w)$ in the page P_0 for term weighting. The second method, denoted as **DOC-TFIDF**, uses each word w's $tf_{P_0} \cdot idf(w)$ score, computed by multiplying $tf_{P_0}(w)$ with w's idf score in the web collection. The third method, denoted as **DOC-OKAPI**, uses each word w's Okapi BM25 score $BM25_{P_0}(w)$, computed by:

$$BM25(w) = \frac{tf_{P_0}(w) \cdot (k_1+1)}{tf_{P_0}(w) + k_1 \cdot (1 - b + b \cdot \frac{|P_0|}{avgdl}} \cdot idf(w), \quad (7)$$

where $avgdl$ is the average document length of the pages in the collection. We use the typical setting $k_1 = 2, b = 0.75$ in Equation 7 in our experiments.

The top ranked terms in a page P_0 by three methods are used as the possible missing original anchor terms for P_0. We will use three keyword based methods as baselines in §4.

4. EVALUATING DISCOVERY

We now compare the capability of discovering missing anchor text by different approaches described in §3, including two link based approaches (AUX-TF and AUX-TFIDF), our content based approach (RALM), and three keyword based approaches (DOC-TF, DOC-TFIDF and DOC-OKAPI).

4.1 Data and Methodology

We use two publicly available large TREC web collections (**GOV2** and **ClueWeb09-T09B**). GOV2 is a standard TREC web collection [4] crawled from government web sites during early 2004. The ClueWeb09 collection is a much larger and more recent web crawl, which contains over 1 billion pages. ClueWeb09-T09B is a subset of ClueWeb09 and contains about 50 million English web pages. Compared with GOV2 crawled only from the gov domain, ClueWeb09-T09B is crawled from the general web thus is a less biased web sample; in another aspect, GOV2 contains relatively high quality government web pages thus having less noise than ClueWeb09-T09B. Thus we use both GOV2 and ClueWeb09-T09B in our experiments to show how different approaches perform in web collections that have different characteristics. The Indri Search Engine[4] was used to index both collections by removing a standard list of 418 INQUERY [2] stopwords and applying Krovetz stemmer. In a separate process, we run Indri Search Engine's harvestlinks utility on the two collections to collect web page inlinks and raw anchor text information where we do not perform stopping or stemming.

To evaluate the quality of discovered anchor text for a web page P_0, we utilize the original anchor text A_{orig} associated with all inlinks of P_0. Specifically, we first hide the page P_0's A_{orig}, apply different anchor text discovery approaches on P_0, then compare the discovered anchor text with P_0's A_{orig}. This procedure can be run automatically so that we can leverage large volumes of web pages to evaluate the performance of different approaches with no human labeling effort. More specifically, we consider each anchor term in a page P_0's A_{orig} as a good description term, or a *relevant* term, for P_0 while terms not in A_{orig} as *non-relevant* ones; in this way, we can generate term relevance judgments for P_0. Then we employ each different approach to discover a ranked list of plausible missing anchor terms for P_0 and then use the relevant judgments to evaluate the ranked anchor term list. Note that for fair comparison P_0's A_{orig} is not used in Equation 2 for calculating RALM in our approach. In the experiments, we perform slight stopping on the raw anchor text by removing a short list of 39 stopwords, which includes 25 common stopwords[12, pp.26] and 14 additional anchor terms[5] that are either common navigational purposed words or part of URLs – it is common that anchor text contains some URL.

We calculate some typical TREC style evaluation measurements including Mean Average Precision (**MAP**), Mean Reciprocal Rank(**MRR**), Precision at the number of relevant terms(**R-Prec**), Precision at K (**P@**k) and also normalized discounted cumulative gain (**NDCG**) [7]. In the experiments, we are specifically interested in the quality of top ranked discovered anchor terms; thus, we only use the

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

[5] The additional terms are: *http, https, www, gov, com, org, edu, net, html, htm, click, here, next, home.*

	MAP	NDCG	MRR	P@2	P@5	P@10	P@20	R-Prec	Discovered Rel.
DOC-TF	0.3162	0.4585	0.5441	0.3833	0.2800	0.2060	0.1333	0.2716	400
DOC-TFIDF	0.2936	0.4348	0.5400	0.3700	0.2613	0.1827	0.1240	0.2530	372
DOC-OKAPI	0.2936	0.4348	0.5400	0.3700	0.2613	0.1827	0.1240	0.2530	372
AUX-TF	0.1969	0.2598	0.3707	0.2833	0.1773	0.1153	0.0643	0.1643	193
AUX-TFIDF	0.1716	0.2423	0.3442	0.2433	0.1720	0.1140	0.0647	0.1428	194
RALM	0.3183	0.4275	0.5050	0.3467	0.2840	0.1860	0.1140	0.3051	342

Table 2: Performances on the GOV2 collection. There are 708 relevant anchor terms overall. Column 10 shows overall relevant anchor terms discovered by each different approach. RALM performs statistically significantly better than AUX-TF and AUX-TFIDF by each measurement in columns 2–9 according to the one-sided t-test ($p < 0.005$). There exists no statistically significant difference between each pair of RALM, DOC-TF, DOC-TFIDF and DOC-OKAPI by each measurement according to the one-sided t-test ($p < 0.05$).

	MAP	NDCG	MRR	P@2	P@5	P@10	P@20	R-Prec	Discovered Rel.
DOC-TF	0.3517	0.4891	0.5588	0.3467	0.2373	0.1360	0.1090	0.2990	327
DOC-TFIDF	0.3107	0.4388	0.5145	0.3133	0.2213	0.1173	0.0983	0.2608	295
DOC-OKAPI	0.3107	0.4388	0.5145	0.3133	0.2213	0.1173	0.0983	0.2608	295
AUX-TF	0.1840	0.2507	0.3309	0.2248	0.1463	0.0729	0.0577	0.1675	172
AUX-TFIDF	0.1634	0.2347	0.3116	0.2047	0.1383	0.0676	0.0560	0.1402	167
RALM	0.2612	0.3615	0.4630	0.2833	0.1733	0.0911	0.0770	0.2398	231

Table 3: Performances on the ClueWeb09-T09B collection. There are 582 relevant anchor terms overall. Column 10 shows overall relevant anchor terms discovered by each different approach. DOC-TF performs statistically significantly better than both RALM and AUX-TF by each measurement in columns 2–9 according to the one-sided t-test ($p < 0.05$). RALM performs statistically significantly better than AUX-TF and AUX-TFIDF by each measurement in columns 2–9 according to the one-sided t-test ($p < 0.05$).

top-20 terms in the discovered term rank lists by different approaches to calculate the measurements.

Note that web pages that can be used in our evaluation procedure need to satisfy two requirements: (1) they need to have some associated A_{orig} and (2) they can collect some auxiliary anchor text from the web graph as described in §3.1. Thus, for each of two collections, we randomly sample 150 pages satisfying the two requirements for training and another 150 pages for testing. On both training sets, RALM's parameter $m = 15$ described in §3.2 achieves the highest MAPs.

4.2 Results and Analysis

The performance of discovering original anchor text by different approaches on the testing set of GOV2 and ClueWeb-09-T09B are shown in Table 2 and Table 3, respectively. The results show that our approach (RALM) can effectively discover missing original anchor terms. On both collections RALM performs *statistically* significantly better than two link based approaches (AUX-TF and AUX-TFIDF). This indicates that, for discovering a page's missing anchor text, the anchor text associated with the similar pages provides more useful information than that associated with the linked web neighbors. The numbers of discovered relevant anchor terms by different approaches, shown in the last column of two tables, also indicate that only using auxiliary anchor text misses more original anchor text information than our content based approach.

Another observation is that RALM performs worse on ClueWeb09-T09B and not *statistically* significantly better on GOV2 than the keyword based approaches. This indicates that words having high IR utility like tf or $tf \cdot idf$ scores are often good description terms for the page and used by human being as the anchor text. Removing a long list of stopwords from web page content has also helped the keyword based approaches to effectively select good description words from the web content. One plausible reason that RALM performs relatively poorly on ClueWeb09-T09B is that, compared with the high quality GOV2 pages, ClueWeb pages are crawled from the general web, where the inlinks and anchor text may be generated in a more noisy way (e.g. spam), degrading RALM's performance. To better understand the performance of different approaches, in Table 5 and Table 6 we show the top-10 words of the anchor term rank lists discovered by different approaches for one evaluation web page in GOV2 and ClueWeb09-T09B, respectively.

	GOV2	ClueWeb09-T09B
pct(AUX-TF, DOC-TF)	30.5%	26.0%
pct(AUX-TF, RALM)	47.6%	46.3%
pct(RALM, DOC-TF)	26.0%	22.3%

Table 4: The average percentage $pct(X, Y)$ of the terms discovered by the X approach appearing in the ones discovered by the Y approach.

Although using keyword information can discover some good anchor terms, the content-generated anchor terms do not help bridging the lexical gap between a web page and varied queries that attempt to search the page. Indeed, human generated anchor text is highly useful for reducing the word mismatch problem because the lexical gap between anchor text and queries is relatively small[14]. Here, we do some lexical gap analysis to show that our approach can also discover anchor terms similar in nature to human-generated ones but different from content-generated ones.

For each web page i in the testing set, we calculate the percentage $pct_i(X, Y)$ of the terms discovered by the X approach also appearing in the ones discovered by the Y approach, then compute the average percent $pct(X, Y)$ with all the pages. We use the outputs from the keyword based DOC-TF, the link based AUX-TF, and the RALM in this analysis. Table 4 shows three average percentages $pct(X, Y)$

which we have specific interest in. We observe that AUX-TF's discovered terms have much higher average per query overlap ratio with RALM's than with DOC-TF's. Moreover, RALM's discovered anchor terms have small overlap with DOC-TF's.

5. USING DISCOVERED ANCHOR TEXT FOR WEB SEARCH

We now describe how we use the discovered anchor text by different approaches for retrieval in a language modeling approach [17]. We point out that our focus here is not to evaluate different schemes to aggregate or combine anchor text [14]; instead, we focus on comparing the utility of RALM and auxiliary anchor text for helping retrieval.

5.1 Retrieval Models

We follow the typical language modeling based retrieval approach[17] and score 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:

$$p(Q|P) = \prod_{w \in Q} p(w|P). \tag{8}$$

When using Dirichlet smoothing, the document language model $p(w|P)$ can be calculated by Equation 3 and then used in Equation 8 for retrieval. We call this baseline **QL**. We only fix $\mu = 2500$ in Equation 3 for the document models used to calculate RALM, but tune the μ for QL to achieve the best retrieval performance in our experiments in §5.2.

We follow the mixture model approach [15, 16] to use the discovered anchor text information for helping retrieval. In this approach, a web page P's document language model is assumed to be a mixture of multiple component distributions where each component is associated with a prior probability, or a mixture weight. Therefore, we can estimate a language model $p(w|A)$ from anchor text discovered by each different approach for the page P and use $p(w|A)$ as a component of P's document model thus obtaining a better document language model $\tilde{p}(w|P)$:

$$\tilde{p}(w|P) = \alpha p(w|P) + (1-\alpha)p(w|A), \tag{9}$$

where $p(w|P)$ is the original smoothed document model in the QL baseline. Then we can plug $\tilde{p}(w|P)$ into equation 8 for retrieval. We compare the retrieval performance of document language models updated by different discovered anchor text information.

We consider three different anchor text sources to update a web page P's document model: (1) the observed original anchor text A_{orig} associated with P, (2) the auxiliary anchor text A_{aux} of P, and (3) the RALM computed by our approach for P. We estimate the anchor text language model $p(w|A_{orig})$ and $p(w|A_{aux})$ by using the ML estimate of observing each word w in A_{orig} and A_{aux}, respectively. Here, we design the following five retrieval methods that use the above three anchor text sources:

1. M-ORG, which only uses the observed original anchor text language $p(w|A_{orig})$.

2. M-AUX, which only uses the auxiliary anchor text language $p(w|A_{aux})$.

3. M-ORG-AUX, which uses both $p(w|A_{orig})$ and $p(w|A_{aux})$ to update the document model $p(w|P)$ by:

$$\begin{aligned}\tilde{p}(w|P) = \beta(\alpha p(w|P) + (1-\alpha)p(w|A_{orig})) \\ +(1-\beta)p(w|A_{aux}).\end{aligned} \tag{10}$$

	MRR	%Top10	Opt. Param.
QL	0.3132	49.7	
M-ORG	0.3696	57.5	$\alpha = 0.95$
M-AUX	0.3187	50.8	$\alpha = 0.99$
M-ORG-AUX	0.3711	57.5	$\alpha = 0.95, \beta = 0.99$
M-RALM	$0.3388^{\triangle}$	53.6	$m = 20, \alpha = 0.95$
M-ORG-RALM	$0.3975^{\star\triangle}$	59.7	$\alpha, \beta = 0.95, m = 20$

Table 7: Retrieval performance of different approaches with TREC 2006 NP queries. The star indicates statistically significant improvement over MRRs of M-ORG and M-ORG-AUX by one-sided t-test ($p < 0.05$). The triangle indicates statistically significant improvement over MRRs of QL and M-AUX by one-sided t-test ($p < 0.05$).

4. M-RALM, which only uses the RALM $p(w|A_0)$ in Equation 2. The original anchor text of P_0 is not used in Equation 2 for calculating RALM.

5. M-ORG-RALM, which uses both $p(w|A_{orig})$ and the RALM $p(w|A_0)$ in Equation 2 by:

$$\begin{aligned}\tilde{p}(w|P) = \beta(\alpha p(w|P) + (1-\alpha)p(w|A_{orig})) \\ +(1-\beta)p(w|A_0).\end{aligned} \tag{11}$$

The original anchor text of P_0 is not used in Equation 2 for calculating RALM.

Note that we can update each page's document model offline, thus this computationally expensive procedure has little impact on the online query processing time. Moreover, different from experiments in §4.1, we use all anchor terms instead of the top-20 most important terms discovered by different approaches.

5.2 Experiments

We use the TREC web named page finding tasks in Terabyte Track[4, 5] to evaluate the performance of different retrieval methods described in §5.1. The objective of the named page (NP) finding task is to find a particular page in the GOV2 collection, given a topic that describes it. We use the NP topics and their relevance judgments for our experiments. In this experiment, we used Porter stemmer and did not remove stopwords when indexing the GOV2 collection.

For each NP query, we first run it against the GOV2 collection to obtain the QL baseline; then we use five retrieval methods described in §5.1 to rerank the top-100 web pages returned by QL. The reranked lists are evaluated by two TREC measurements previously used for the task [5]: **MRR** which is the mean reciprocal rank of the first correct answer and the **%Top10** which is the proportion of queries for which a correct answer was found in the first 10 search results. We use the TREC 2005 NP topics (NP601-872) for training and the TREC 2006 NP topics (NP901-1081) for testing. We first tune the Dirichlet parameter $\mu = 500$ for QL to achieve the highest MRR on the training set and obtain QL's top-100 web pages for reranking. We then fix $\mu = 500$ to calculate the smoothed document model component $p(w|P)$ in the five retrieval methods but tune the mixture parameters α and β for them to achieve the highest MRRs with the training queries. For the two approaches that use RALM, the parameter m of RALM is also tuned. After that, we run different methods on the testing set.

Table 7 shows the retrieval performance of different methods and the tuned parameters in each method. We observe: (1) M-ORG-RALM performs statistically significantly bet-

"Optima National Wildlife Refuge", "Optima NWR", "Washita Optima National Wildlife Refuge near Butler OK"							
DOC-TF	$tf_{P_0}(w)$	**DOC-TFIDF**	$tf_{P_0}idf(w)$	**DOC-OKAPI**	$BM25_{P_0}(w)$	**AUX-TF**	$tf_{aux}(w)$
refuge	15	refuge	79.69	refuge	153.76	oklahoma	6
wildlife	10	optima	74.30	optima	143.37	wildlife	2
oklahoma	10	hardesty	47.48	hardesty	91.63	refuge	2
optima	8	hawk	36.20	hawk	69.86	website	1
species	6	oklahoma	36.03	oklahoma	69.53	u	1
hawk	6	wildlife	31.98	wildlife	61.71	service	1
habitat	6	guymon	29.35	guymon	56.63	s	1
area	6	habitat	26.42	habitat	50.98	office	1
prairie	5	species	23.70	species	45.73	national	1
national	5	quail	21.74	quail	41.95	fish	1
AUX-TFIDF	$tf_{aux}idf(w)$	**RALM**	$P(w\|A_0)$	**Rel.**			
oklahoma	21.62	nwr	0.1164	**butler**			
refuge	10.62	wildlife	0.0834	**national**			
wildlife	6.40	refuge	0.0834	**near**			
fish	3.11	national	0.0834	**nwr**			
u	3.03	general	0.0657	**optima**			
website	2.36	brochure	0.0657	**refuge**			
office	1.54	kansas	0.0601	**washita**			
s	1.29	lake	0.0522	**wildlife**			
national	1.22	tear	0.0308				
service	1.09	sheet	0.0308				

Table 5: Discovered missing anchor terms and their term weights by applying different approaches on one GOV2 web page (TREC DocID in GOV2: GX010-01-9459902) . The first row shows the original three pieces of anchor text associated with the page. The Rel column in bold font shows the term relevance judgments extracted from the first row. RALM can discover some term like "NWR", which may not appear in both the page and the auxiliary anchor text, thus may help to bridge the lexical gap between pages and web queries as using the original anchor text does.

"Weight Loss Resolutions", "Weight Loss New Year's Resolution to Lose Weight","Resolve to Lose Weight"							
DOC-TF	$tf_{P_0}(w)$	**DOC-TFIDF**	$tf_{P_0}idf(w)$	**DOC-OKAPI**	$BM25_{P_0}(w)$	**AUX-TF**	$tf_{aux}(w)$
weight	46	weight	96.38	weight	112.53	weight	709
loss	26	loss	78.65	loss	91.83	loss	705
lose	20	lose	64.47	lose	75.28	diet	32
new	17	resolution	46.57	resolution	54.38	weightloss	21
year	15	diet	34.27	diet	40.02	guide	20
resolution	13	goal	26.01	goal	30.37	scott	8
time	12	eat	25.61	eat	29.90	jennifer	8
make	10	year	23.90	year	27.90	contact	8
goal	9	calorie	15.73	calorie	18.36	site	6
diet	9	pound	15.34	pound	17.91	s	4
AUX-TFIDF	$tf_{aux}idf(w)$	**RALM**	$P(w\|A_0)$	**Rel.**			
loss	2132.63	weight	0.2245	**lose**			
weight	1485.49	loss	0.1737	**loss**			
weightloss	157.70	diet	0.0550	**new**			
diet	121.86	easy	0.0436	**resolution**			
guide	37.26	lose	0.0422	**resolve**			
jennifer	33.96	way	0.0412	**s**			
scott	28.52	myth	0.0396	**weight**			
guidesite	22.04	warn	0.0232				
em	13.15	ppa	0.0232				
mlibrary	11.37	fda	0.0232				

Table 6: Discovered missing anchor terms and their term weights by applying different approaches on one ClueWeb09 web page (ClueWeb09 RecordID: clueweb09-en0004-60-01628). The first row shows the original three pieces of anchor text associated with the page. The Rel column in bold font shows the term relevance judgments extracted from the first row. The keyword approaches discovered "new year resolution", which may be hard to be discovered by using the page's web-graph neighbor pages' anchor text or using the page's similar pages' anchor text.

ter than M-ORG. This indicates that missing anchor text discovered by RALM provides additional information not in the original anchor text so that combining them can further improve the retrieval performance. (2) M-ORG-RALM and M-RALM performs statistically significantly better than M-ORG-AUX and M-AUX, respectively. This indicates that in GOV2 missing anchor text information discovered by our content based approach helps retrieval more effectively than the auxiliary anchor text.[6]

In Table 7, we observe that the auxiliary anchor text helps the performance very little in this task. There are two plausible reasons: first, TREC NP queries are short queries and Metzler *et al.* observed that auxiliary anchor text does not help or even hurts the performance of short navigational web queries[14]; second, the anchor text sparsity problem is serious on the GOV2, thus very small percentage of pages can collect some auxiliary anchor text as shown in Table 1 to benefit the search task. However, even when serious anchor text sparsity exists and queries are short, our content based approach still helps improving retrieval effectiveness.

We expect our technique can enhance the retrieval performance of general web search engines where there are large portion of short navigational queries. As is well known, in the general web search environment there are many low-quality web pages and spam; thus, we need to address issues about web page quality and noise filtering for better benefitting general web search. We leave this as future work.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a language modeling based technique to overcome the anchor text sparsity problem by using web content similarity. Our approach computes a relevant anchor text language model, called RALM, from its similar web pages' associated anchor text to discover its plausible missing anchor text. Compared with a link based approach [14], our content based approach has no specific link structure requirements on the web page of interest and thus can further reduce anchor text sparsity.

We designed experiments with two TREC web corpora to evaluate the effectiveness of discovering missing anchor terms by three different approaches: the link based approach, the RALM approach, and the keyword based approach. Experimental results show that the RALM approach can effectively discover missing original anchor text and performs statistically significantly better than the two link based approaches on both collections. Moreover, RALM's discovered anchor text is similar in nature to auxiliary anchor text while different from the keywords in the web page.

By using the mixture model[15, 16], we used different discovered anchor text information within the language modeling framework for retrieval. We evaluated using different approaches for improving retrieval effectiveness with the TREC named page finding task. The results show that (1) RALM helps retrieval more than using the auxiliary anchor text collected over the web graph and (2) combining RALM and the original anchor text can statistically significantly improve the retrieval performance of only using the original anchor text. Furthermore, RALM can help improving retrieval effectiveness for short navigational queries even when serious anchor text sparsity exists. This makes RALM a promising technique for improving general web search engines.

[6]Our goal is not to compare ranking schemes, but to show the utility of the discovered anchor text. However, we note that these scores match or beat top-performing approaches [4].

There are several interesting directions of future work. Metzler *et al.* found that auxiliary anchor text can effectively help longer, informational queries [14]; we will explore how well RALM can help long informational queries. We also want to explore using RALM's discovered missing anchor text information beyond the language modeling based retrieval framework, e.g. using it to extract useful features for learning-to-rank retrieval approaches [3].

7. ACKNOWLEDGMENTS

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

8. REFERENCES

[1] A. Broder et al. Graph structure in the web. *Comput. Netw.*, 33(1-6):309–320, 2000.

[2] 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.

[3] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proc. of ICML*, pp. 89–96, 2005.

[4] S. Büttcher, C. L. A. Clarke, and I. Soboroff. The TREC 2006 Terabyte Track. In *TREC*, 2006.

[5] C. L. A. Clarke, F. Scholer, and I. Soboroff. The TREC 2005 Terabyte Track. In *TREC*, 2005.

[6] A. Fujii. Modeling anchor text and classifying queries to enhance web document retrieval. In *Proc. of WWW*, pp. 337–346, 2008.

[7] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[8] O. Kurland and L. Lee. Corpus structure, language models, and ad hoc information retrieval. In *SIGIR*, pp. 194–201, 2004.

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

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

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

[12] C. D. Manning, P. Raghavan, and H. Schütze. Introduction to Information Retrieval. Cambridge Univ. Press. 2008.

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

[14] D. Metzler, J. Novak, H. Cui, and S. Reddy. Building enriched document representations using aggregated anchor text. In *SIGIR*, pp. 219–226, 2009.

[15] R. Nallapati, B. Croft, and J. Allan. Relevant query feedback in statistical language modeling. In *Proc. of CIKM*, pp. 560–563, 2003.

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

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

[18] T. Tao, X. Wang, Q. Mei, and C. Zhai. Language model information retrieval with document expansion. In *Proc. of NAACL-HLT*, pp. 407–414, 2006.

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

[20] T. Westerveld, W. Kraaij, and D. Hiemstra. Retrieving web pages using content, links, urls and anchors. In *Proc. of TREC*, pp. 663–672, 2001.