Building Enriched Document Representations using Aggregated Anchor Text

Donald Metzler, Jasmine Novak, Hang Cui, and Srihari Reddy

Yahoo! Labs
4301 Great America Parkway
Santa Clara, CA. 95054

ABSTRACT

It is well known that anchor text plays a critical role in a variety of search tasks performed over hypertextual domains, including enterprise search, wiki search, and web search. It is common practice to enrich a document's standard textual representation with all of the anchor text associated with its incoming hyperlinks. However, this approach does not help match relevant pages with very few inlinks. In this paper, we propose a method for overcoming anchor text sparsity by enriching document representations with anchor text that has been aggregated across the hyperlink graph. This aggregation mechanism acts to smooth, or diffuse, anchor text within a domain. We rigorously evaluate our proposed approach on a large web search test collection. Our results show the approach significantly improves retrieval effectiveness, especially for longer, more difficult queries.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation

Keywords

anchor text, web search, link structure, term weighting

1. INTRODUCTION

One of the most unique characteristics of the web is its dynamic, human generated hypertext structure. The web has allowed millions of everyday users to publish their own content. Most web pages contain one or more hyperlinks that point to other pages. These links, referred to as *anchors*, consist of a destination URL and a short piece of text. The short piece of text, which is called *anchor text*, typically provides a description of the destination URL. For example, the anchor text associated with `http://www.acm.org/sigir` includes "sigir", "acm sigir", and "information retrieval". However, since anchor text is user generated, it may not always be a useful description of the destination URL. For example, "click here" which provides no meaningful description of the destination URL, is often used as anchor text.

It is well known that the link structure of the web can be exploited for improving web search, as evidenced by PageRank [3] and other link analysis algorithms, such as HITS [15] and SALSA [20]. While link analysis algorithms are useful for identifying authoritative pages, most of them only consider the link structure and ignore the anchor text. Despite the claimed importance of link-based algorithms, anchor text is arguably the most important piece of evidence used in web ranking functions. Anchor text is so useful because it is similar in nature to queries. Returning to the ACM SIGIR homepage example, it is easy to see that the anchor text "sigir", "acm sigir", and "information retrieval" are reasonable queries that users may enter when they are searching for the page. Therefore, the lexical gap between queries and anchor text is relatively small, whereas the gap between queries and document texts is much larger, making anchor text highly useful for matching queries to documents.

However, anchor text is only useful for ranking pages with incoming links. Previous research has shown that the distribution of the number of inlinks on the web follows a power law [4]. Thus, a small number of pages will have a large amount of anchor text associated with them, while most will have very little, or no, anchor text. We refer to this as the *anchor text sparsity problem*. The primary goal of this paper is to overcome this problem by enriching the anchor text representations of documents, especially for those documents that have little anchor text to begin with. We propose enriching anchor text representations by augmenting documents with auxiliary anchor text that is derived by *aggregating*, or *propagating*, anchor text over the web graph. Given the importance of anchor text for ranking, we hypothesize that retrieval effectiveness can be significantly improved by using these auxiliary anchor text-enriched document representations. However, the impact of our work is not limited to web search. Indeed, the enriched document representations can be used in a number of other ways, including estimating improved document models, developing advanced textual matching features, and even improving the quality of document classification algorithms.

Our work has four primary contributions. First, to the best of our knowledge, we are the first to directly formulate and address the anchor text sparsity problem. Second,

SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.

we propose a number of methods for aggregating anchor text across the web graph. Third, we propose various ways to use the aggregated anchor text to build enriched document representations. Finally, we show that our enriched document representations, when used in conjunction with a state-of-the-art ranking function, results in significant improvements in retrieval effectiveness on a very large web test collection.

The remainder of this paper is laid out as follows. First, Section 2 surveys relevant related work. Then, Section 3 provides an overview of our proposed anchor text aggregation algorithms. An evaluation of our method applied to a large-scale web search test collection is presented in Sections 4. Finally, Section 5 concludes the paper and describes various avenues for future work.

2. RELATED WORK

Anchor text has always been deemed an important source of information for relevance. Brin and Page were one of the first to note the importance of associating anchor text both with the page it occurs on, as well as the page it points to [3]. Anchor text is not only useful for enriching textual representations. Harmandas et al. showed that propagated anchor text was useful for building textual representations of images [12]. Given its importance, anchor text is extensively exploited in most, if not all, commercial search engines, and is useful for a wide range of applications.

Anchor text can be modeled in a variety of ways. Recent work by Fujii [11] describes two techniques for using anchor text within retrieval models – one using all anchor text as a surrogate of the original document and the other treating each line of anchor text independently by considering the importance of each line. We adopt a similar approach here, by representing anchor text as individually weighted lines of text.

A great deal of recent work has looked at how to effectively rank structured, or fielded, documents [25, 28, 30]. Robertson et al. explored an extension of the popular BM25 model to multiple weighted fields in documents [27, 28]. Similar work by Ogilvie and Callan [25], done within the language modeling framework for information retrieval, treats documents as a mixture of field language models, thereby achieving a similar type of ranking function. Our focus here is not to evaluate ranking functions for structured documents. Instead, our aim is to show the utility of auxiliary anchor text for building enriched document representations. Indeed, the methods described here can be used in conjunction with any retrieval model that takes document structure into account. We will return to BM25F later in this paper, in the context of our experimental evaluation.

As we will show, our proposed approach can be construed as a form of smoothing, in which we smooth the original anchor text with auxiliary anchor text obtained from the web graph. This is similar in nature to cluster-based smoothing from the language modeling framework [16, 21], except we make explicit use of the web graph, rather than clusters or induced links [17, 22, 31]. Furthermore, the language model-based approaches construct expanded probabilistic representations of the documents, whereas our approach explicitly constructs enriched document representations by adding new (weighted) text to documents.

Finally, our approach, although similar in spirit to, differs from the previously proposed approaches for spreading activation [8], link analysis [3, 15], graph regularization [9], score aggregation [29], and term frequency aggregation [26]. Although all of these approaches are branded, or framed, as different problems, they all essentially tackle the same task. The primary difference between these approaches and ours is that we aggregate (weighted) textual *representations* across the web graph, rather than scores or term frequencies. Our approach is more general, in the sense that the textual representations that we propagate can be used in a variety of ways, including building enriched document representations and computing textual features, among others.

3. AGGREGATING ANCHOR TEXT

We now describe our proposed method for aggregating anchor text over the web graph and how the aggregated text can be used to enrich existing document representations.

As described earlier, our proposed method aggregates, or propagates, anchor text along the web graph. The input to our method is a URL u and the output is a weighted set of *aggregated anchor text lines*. This is achieved in two steps. First, the aggregated anchor text lines are collected. Then, the lines are combined and weighted to produce the final result. The remainder of this section details one possible instantiation of this general framework. However, it should be noted that the idea is quite general. Indeed, the aggregated anchor text can be collected and weighted in many different ways beyond the approaches described here. We now describe our specific instantiation of the approach.

3.1 Collecting Aggregated Anchor Text

We begin by describing how we collect the aggregated anchor text. Given a URL u, we first collect all pages P, within the same site (domain), that link to u. These links are known as u's *internal inlinks*, since they come from within the same site. We then gather all anchor text A from pages that are linked to P from outside the site. This is known as the *external anchor text*, because it originates from pages outside of the site. Thus, in short, the aggregated anchor text for u is the external anchor text of the internal inlinks of u.

Figure 1 illustrates this process by way of an example. In the example, anchor text is being aggregated for the URL `http://dancing.com/lindyhop.html`. The original anchor text of the page consists of lines such as "Lindy Hop" and "swing dancing", while the aggregated anchor text lines include "Savoy Ballroom" and "dances in New York," that are not present in the original anchor text.

This particular approach is used to collect aggregated anchor text because internal inlinks typically link related pages within a given site. These links are typically created by the owner of the site, and therefore can be considered somewhat authoritative, as opposed to links originating from external sites, which may not be as purposefully generated. It is very important to notice that we do not use the anchor text associated with the internal inlinks in any way, since such anchor text tends to be navigational in nature (e.g., "home", "next page", etc.). This is why we use the external anchor text of the internal inlinks as our source of auxiliary anchor text. Such anchors are less likely to be navigational and are more likely to provide good descriptions of their destination. Therefore, since internal links connect related pages, we hypothesize that the external anchor text of these pages will also be good descriptors, by semantic transitivity, of the URL of interest.

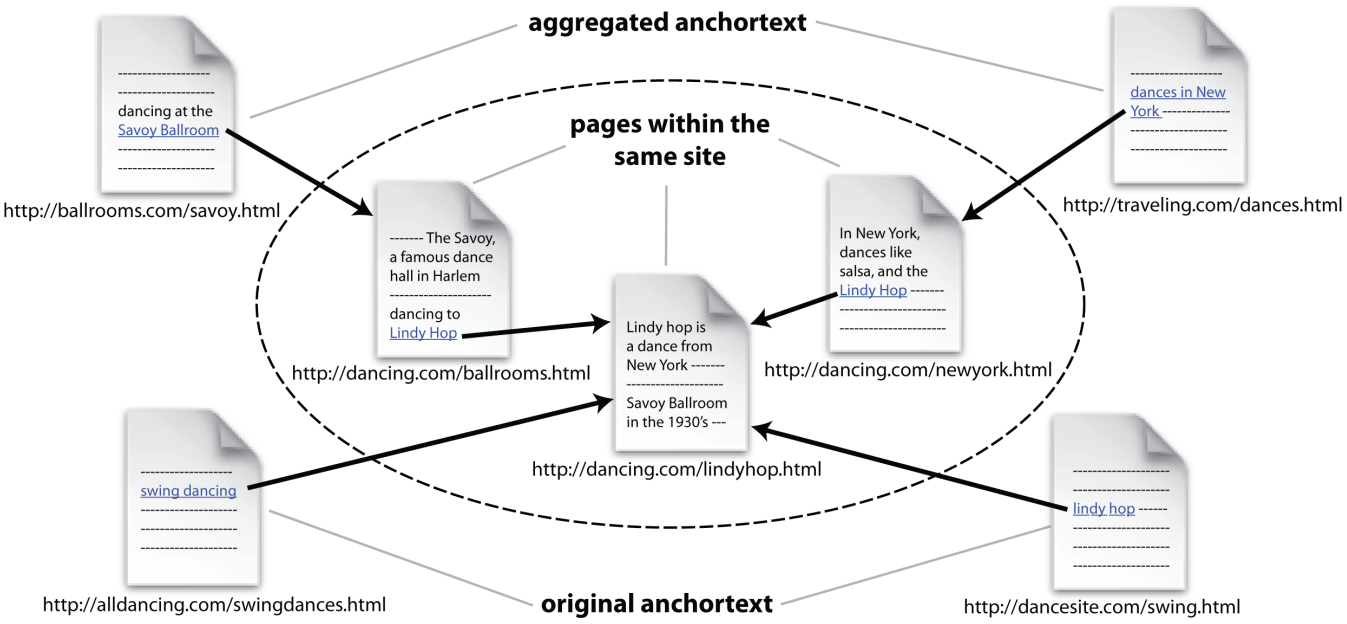

Figure 1: Illustration of how anchor text is aggregated across the web graph. In this example, anchor text is being aggregated for the URL http://dancing.com/lindyhop.html.

3.2 Weighting Aggregated Anchor Text

Next, we describe how the aggregated anchor text lines are combined and weighted. We assume that every line l of anchor text associated with a URL u has some weight $wt(l,u)$ assigned to it. Since lines are aggregated from multiple sources (internal inlinks), it is possible that the same line of aggregated anchor text may originate from multiple URLs, each with a potentially different weight. Since we require one weight $wt(l,u)$ per distinct line of anchor text, we must combine the weights of lines originating from multiple sources in some way. If we treat the weighted lines of anchor text as result lists, then we can easily apply standard result set fusion techniques to combine the weights [10, 18].

We use the six following weight aggregation functions, inspired by result set fusion techniques, to weight the aggregated lines of anchor text:

$$wt_{Min}(l,u) = \min_{u' \in \mathcal{N}(u)} wt(l,u')$$

$$wt_{Max}(l,u) = \max_{u' \in \mathcal{N}(u)} wt(l,u')$$

$$wt_{Mean}(l,u) = \frac{1}{|\mathcal{N}(u)|} \sum_{u' \in \mathcal{N}(u)} wt(l,u')$$

$$wt_{MeanMNZ}(l,u) = \frac{|u' \in \mathcal{N}(u) : wt(l,u') > 0|}{|\mathcal{N}(u)|} \sum_{u' \in \mathcal{N}(u)} wt(l,u')$$

$$wt_{Sum}(l,u) = \sum_{u' \in \mathcal{N}(u)} wt(l,u')$$

$$wt_{SumMNZ}(l,u) = |u' \in \mathcal{N}(u) : wt(l,u') > 0| \sum_{u' \in \mathcal{N}(u)} wt(l,u')$$

where $\mathcal{N}(u)$ is the set of internal inlinks and $wt(l,u')$ is the original weight of anchor text line l for URL u'. Notice that if some line of aggregated anchor text originates from a single URL u', then the aggregated weight will equal $wt(l,u')$ regardless of the aggregation function chosen. However, when a line originates from multiple URLs, each of the aggregation functions computes the weight differently.

Before moving on, we note that there is no canonical way to compute the original anchor text line weights (i.e., $wt(l,u')$), which we assume are calculated *a priori*. These weights will be computed differently for every search engine implementation. For our purposes, original lines of anchor text are weighted as follows:

$$wt(l,u) = \sum_{s \in \mathcal{S}(u)} \frac{\delta(l,u,s)}{|anchors(u,s)|}$$

where $\mathcal{S}(u)$ is the set of external sites that link to u, $\delta(l,u,s)$ is 1 if and only if anchor text l links to u from some page within site s, and $|anchors(u,s)|$ is the total number of unique anchors originating from site s that link to u.

3.3 Enriched Document Representations

Now that we have described how to collect and weight aggregated anchor text, we explain various ways that we can use the output to build enriched document representations. Aggregated anchor text-enriched document representations may be useful for various information retrieval and natural language processing tasks including web search, contextual advertising, text classification, and summarization, to name just a few. The best representation will depend on the task. For completeness, we now briefly describe four possible representations.

The first representation is the *flat representation*. Here, all document structure, such as fields, formatting, and metadata, are ignored. The aggregated anchor text weights are

discarded and only the raw text itself is added to the original document. We do not expect this representation to be particularly useful for most tasks, but is one very simple possibility.

A more reasonable representation is the *combined representation*, which preserves the document structure, and augments the original anchor text lines with the aggregated anchor text lines. The aggregated anchor text weights may also be used here, as long as the search engine's indexing architecture supports it.

One issue with the combined representation is that there may be a great deal of overlap between the original and aggregated anchor text lines. The aggregated anchor text lines may add noise to a set of high quality original anchor text lines. To overcome this issue, the *backoff representation* only adds aggregated anchor text to documents that do not originally have any anchor text lines associated with them.

Finally, the *new field representation* adds the aggregated anchor text as a completely new field to every document. Unlike the combined and backoff representations that add the aggregated anchor text to the original anchor text field, the new field representation treats the new lines of anchor text as a new source of evidence. This may be useful for textual features, such as BM25F, that weight the importance of each field separately. In this representation, the original and aggregated anchor text fields can be weighted differently, which may be useful.

4. EXPERIMENTS

We evaluate our proposed anchor text aggregation methodology and document representation schemes using web search as our task. Web search is a natural choice due to its practical importance, and because our proposed methods are particularly amenable to the task.

4.1 Data and Methodology

Our experiments make use of a large-scale web search test collection consisting of 22,822 queries. For each query, we have judgments for an average of 23 URLs, resulting in 524,418 judged query/URL pairs. Each URL has been judged by a human editor and given a rating as to how relevant it is for the corresponding query. This rating is one of Perfect, Excellent, Good, Fair, or Bad.

We chose to use this test collection, rather than one of the standard TREC Web [7], Terabyte [6], or Million Query [1] Track test collections, because it is more characteristic of real web search. In particular, the number of queries and judgments is substantially larger than the TREC web collections. Furthermore, the query and document population reflects real systems better, as well, since the queries are randomly sampled and the documents are retrieved from the entire web, rather than a small snapshot or single top level domain, as in the WT10G, GOV, and GOV2 collections. Finally, and perhaps most importantly, the anchor text and link structure used in our experiments covers the crawlable web, rather than just a subset, which can skew or bias results.

Since we have graded relevance judgments, we make use of discounted cumulative gain (DCG)-based evaluation metrics [13]. The DCG at rank K is computed as:

$$DCG\text{-}K(Q) = \sum_{i=1}^{K} \frac{g(i)}{\log(1+i)}$$

where $g(i)$ is the gain associated with the rating of result at rank i and K is maximum depth result to consider. We report DCG-1 and DCG-5 in our experiments, which are commonly used to evaluate web search effectiveness.

In addition to these two metrics, we also report normalized discounted gain (NDCG), which is a normalized version of DCG, which can be computed as:

$$DCG(Q) = \sum_{i=1}^{N(Q)} \frac{g(i)}{\log(1+i)}$$

where $N(Q)$ is the number of URLs ranked for query Q. From this, NDCG is calculated as:

$$NDCG(Q) = \frac{DCG(Q)}{IDCG(Q)}$$

where $IDCG(Q)$ is the "ideal DCG" achieved if the results for Q were ranked perfectly. Therefore, $NDCG(Q) = 1$ indicates the best possible ranking. In this paper, we use gains of 10, 7, 3, 0.5, and 0, respectively, corresponding to the ratings described above.

4.2 Ranking Function

For ranking, we use BM25F, which is an extension of the popular BM25 model that takes into account structured (fielded) documents and been shown to be highly effective for web search in the past [33]. Our original, unenriched documents contain several fields, including `title`, `body`, and `anchor`, that we will use in combination with our aggregated anchor text, to rank documents using BM25F.

The BM25F model first computes term field weights for every field f in the document u and every term t in the query. If the field f is the original or aggregated anchor text (when using the new field representation), then the term's field weight is computed as:

$$wt(t,f,u) = \sum_{l \in \mathcal{L}(u)} wt(l,u) \cdot tf(t,l) \cdot \alpha^{x(q,l)} \cdot \beta^{m(q,l)}$$

where $\mathcal{L}(u)$ is the set of anchor text lines associated with u, $wt(l,u)$ is the weight associated with the line of anchor text, $tf(t,l)$ is the number of times that term t occurs in line l, $x(q,l)$ is the number of terms in line l that do not occur in the query, $m(q,l)$ is the number of query terms missing from line l, and α and β are free parameters in the range $(0,1]$ that penalize missing and extra terms. Thus, the term's anchor text field weight depends on the weight of the line, the number of times the term occurs in the line, and how closely the line matches the query ($\alpha^{x(q,l)} \cdot \beta^{m(q,l)}$). This final component, which is a simple measure of the lexical similarity between the query and the line of anchor text, is added because it was shown to be useful in previous research [2].

If f is any other field, such as the body, then the term field weight is computed as follows:

$$wt(t,f,u) = tf(t,f,u)$$

where $tf(t,f,u)$ is simply the number of times that term t occurs in field f in document u.

Given a term field weight, BM25F first normalizes the weight with respect to the field length as follows:

$$\hat{wt}(t,f,u) = \frac{wt(t,f,u)}{1 + b_f \left(\frac{l(f,u)}{\bar{l}(f)} - 1 \right)}$$

where $wt(t, f, u)$ is the original, non-normalized term field weight, $l(f, u)$ is the length of field f in document u, $\bar{l}(f)$ is the average length of field f across all documents, and b_f is a free parameter that controls the amount of document length normalization. The weight for a term t in a document u is then simply the weighted summation of the length normalized term field weights, which is computed as:

$$\hat{wt}(t, u) = \sum_f wt(f) \cdot \hat{wt}(t, f, u)$$

where each $wt(f)$ is a free parameter that determines how much each field contributes to the overall term weight.

The final BM25F score for a query q with respect to document u is then computed as follows:

$$BM25F(q, u) = \sum_{t \in q \cap u} \frac{\hat{wt}(t, u)}{k_1 + \hat{wt}(t, u)} \log \frac{N - df_t + 0.5}{N + 0.5}$$

where N is the total number of documents in the collection, df_t is the total number of documents that term t occurs in, and k_1 is a free parameter that controls term frequency saturation.

Therefore, BM25F has a number of free parameters that need to be tuned to be effective. In our experiments, we use 2-fold cross validation for this purpose. In all cases, we optimize the various BM25F parameter values to maximize the NDCG on our training set using the optimization procedure described in [33]. The parameters learned on the training set are then applied to the test set for evaluation purposes.

4.3 Results

This section describes the results of our experimental evaluation. We begin by showing how our proposed method helps overcome the anchor text sparsity problem. The remainder of the section details how aggregated anchor text can consistently and significantly improve retrieval effectiveness for web search.

4.3.1 Anchor Text Sparsity

Earlier, we posed the anchor text sparsity problem, which is based on the fact that many URLs have very few lines of anchor text associated with them. Given the importance of anchor text for ranking, this phenomenon may result in suboptimal ranking, especially for highly relevant pages with little, or no, anchor text that adequately matches the query.

Figure 2 plots the distribution of lines of anchor text per URL for the original document representation (solid line) and the combined representation (dashed line) over the URLs in our test collection. As the graph shows, 211,565 URLs originally have no anchor text lines associated with them. This is consistent with previous observations that the distribution of inlinks, and hence anchor text, follows a power law distribution [4]. However, when aggregated anchor text is included in the combined representation, only 152,911 URLs have zero lines of anchor text associated with them, which is a 38% reduction.

As Figure 2 shows, the combined representation has many more URLs with 1 line of anchor than the original representation. This trend, although barely visible in the graph, continues. In fact, the average number of lines of aggregated anchor text per document is 34 and the maximum is 997.

The URLs in our test collection, which were collected using pooling, are strongly biased, and do not represent a random sample of URLs. For this reason, we also computed various anchor text measures over a random sample of 1 million URLs. Our results showed that 32,715 of the URLs had original anchor text, while 50,127 have aggregated anchor text. Of these 50,127 URLs, 43,841 did not have any original anchor text. Therefore, by aggregating anchor text we can more than double the number of URLs that have some type of anchor text associated with them. Furthermore, our analysis showed that the average number of original lines of anchor text associated with the URLs was 1, while the average number of lines of aggregated anchor text was 11.

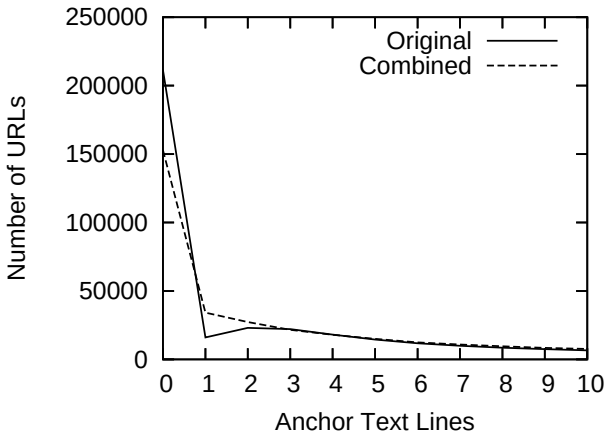

Figure 2: Distribution of the number of anchor text lines in the original (solid line) and combined (dashed line) document representations.

Therefore, these results show that our approach is effective at overcoming the anchor text sparsity problem. The method not only significantly reduces the number of URLs with no anchor text, but also adds new anchor text to URLs that have few original anchor text lines.

4.3.2 Representations and Aggregation Functions

We have just shown that our proposed method helps overcome the anchor text sparsity problem. However, this does not guarantee that the new document representations will be useful in practice. Therefore, we evaluate how useful our aggregated anchor text enriched document representations are in the context of web search.

Our first experiment investigates the usefulness of different combinations of weight aggregation functions (Section 3.2) and document representations (Section 3.3). We do not perform any experiments with the flat representation, since HTML documents are inherently structured, and for this reason, we expect the representation to give poor results.

The outcome of our experiments are shown in Table 1. Our baseline, which corresponds to the last row in the table, uses the original document representation and does not perform any anchor text aggregation. The results show that various combinations of weight aggregation functions and document representations, across all metrics, yield statistically significant improvements over the baseline.

In terms of document representations, the backoff representation generally performs the worst. The new section and combined representations perform comparably. How-

Representation	Agg. Func.	DCG-1	DCG-5	NDCG
Combined	Min	**3.279**‡	**7.596**‡	**.8262**‡
Combined	Max	3.279‡	7.596‡	.8261‡
Combined	Mean	3.267	7.583	.8256
Combined	Mean-MNZ	3.260↓	7.568↓	.8253
Combined	Sum	3.266	7.583	.8256
Combined	Sum-MNZ	3.232↓	7.543↓	.8247↓
Backoff	Min	3.260	7.579‡	.8253†
Backoff	Max	3.273†	7.589‡	.8257‡
Backoff	Mean	3.266	7.573	..8253‡
Backoff	Mean-MNZ	3.272	7.582	.8258
Backoff	Sum	**3.277**‡	**7.589**‡	**.8259**‡
Backoff	Sum-MNZ	3.273‡	7.583‡	.8257‡
New Section	Min	3.271‡	7.584‡	.8258‡
New Section	Max	**3.274**‡	**7.595**‡	**.8260**‡
New Section	Mean	3.270‡	7.583‡	.8258‡
New Section	Mean-MNZ	3.261	7.577†	.8254‡
New Section	Sum	3.270	7.592‡	.8257‡
New Section	Sum-MNZ	3.269	7.585‡	.8259‡
Original	N/A	3.266	7.576	.8254

Table 1: Comparison of results for different document representations and weight aggregation functions. The † and ‡ symbols represent statistically significant improvements versus the baseline at the $p < 0.1$ and $p < 0.05$ level, respectively. The ↓ represents a statistically significant decrease at the $p < 0.05$ level. Bold values indicate the aggregation function that produces the best effectiveness for each pair of metrics and representations.

ever, the combined representation appears to be consistently, yet only slightly, better than the new section representation. One may expect the new section representation to perform consistently better than the combined representation, because the combined representation uses a single BM25F field weight for the field that consists of the combined original and aggregated anchor text, whereas the new section representation uses separate BM25F field weights for the original and aggregated anchor text sections. This extra degree of freedom, one may suppose, would improve effectiveness. However, our experiments show that the BM25F field weights learned for the original and aggregated anchor text sections are almost always equal. Therefore, tying the two weights together, which the combined representation essentially does, is a reasonable thing to do. The new section representation, which has an extra degree of freedom, may actually be slightly overfitting to the training data, thereby resulting in a small loss in effectiveness versus the combined representation.

One interesting observation is that the best weight aggregation function differs for each representation. The best functions for the combined, backoff, and new section representations are min, sum, and max, respectively. This suggests that there is no "one size fits all" weight aggregation function, and that the best choice will depend on how the anchor text is originally weighted, how the anchor text is aggregated, and how the resulting document is represented. The max function, however, seems to be a rather robust choice, as it achieves statistically significant improvements over the baseline across all representations and metrics.

One may notice that the relative improvements in the various metrics are rather small. For example, the best DCG-5 of 7.596 is achieved using the combined representation with the min weight aggregation function. This improvement, which is highly statistically significant, is only a 0.26% relative improvement over the baseline. While this may seem small, especially compared to studies on TREC collections that achieve improvements over 10%, this improvement is actually quite large for this particular test collection. Indeed, relative improvements in DCG-5 greater than 0.1% are regarded as "substantial" for this test collection. One of the primary reasons why it is difficult to obtain large improvements on this data set stems from the fact that the queries were randomly sampled, meaning that a large proportion of the queries are short, navigational queries or long, very difficult queries. Most of the short, navigational queries achieve the maximum possible DCG-5 using the original document representation, because the original anchor texts in these cases provide overwhelming evidence in favor of relevance. On the other hand, the long, very difficult queries tend to achieve a DCG-5 of near 0 for both the baseline and enriched document representations. Therefore, substantially smaller improvements in retrieval effectiveness are to be expected.

Baseline NDCG	Queries	NDCG	DCG-1	DCG-5
0.00 – 0.25	29	+0.02%	0.00%	0.00%
0.25 – 0.50	765	+1.73%‡	+28.8%‡	+17.3%‡
0.50 – 0.75	6256	+1.23%‡	+16.6%‡	+3.22%‡
0.75 – 1.00	15772	-0.24%‡	-0.97%‡	-0.35%‡

Table 2: Relative improvements in retrieval effectiveness across query difficulty levels when using aggregated anchor text. The † and ‡ symbols indicate statistically significant improvements over the baseline at the $p < 0.1$ and $p < 0.05$ level, respectively.

4.3.3 *Deeper Analysis*

We just hypothesized that our method did little to improve the retrieval effectiveness of short, navigational queries and long, difficult queries. We can test this hypothesis by performing a stratified evaluation of the queries according to difficulty and length.

In Table 2, we stratify queries according to their difficulty, where difficulty is quantified as the NDCG achieved by the baseline system. The table shows the number of queries for each strata and the relative improvements in NDCG, DCG-1, and DCG-5, respectively. The results shown are for the combined representation using the min aggregation function, which achieved the best overall NDCG. The results for other document representations show similar trends.

This detailed breakdown of the results reveals that aggregated anchor text does little to improve the most difficult queries (0.0 – 0.25 range), but significantly improves queries of medium difficulty. Indeed DCG-1 and DCG-5 for queries in the 0.25 – 0.50 range are improved by well over 10%. This supports our hypothesis that our aggregated anchor text improves difficult (but not the most difficult), non-navigational queries. However, the results also show that aggregated anchor text significantly decreases the retrieval effectiveness for easy queries in the range 0.75 – 1.0. These queries make up most of the queries, which is why the overall improvements reported in the previous section were on the order of

Representation	Agg. Func.	Length 1	Length 2	Length 3	Length 4+
Combined	Min	9.254 (+0.10%)	8.709 (+0.25%)†	7.241 (+0.36%)	5.417 (+0.33%)‡
Backoff	Sum	9.257 (+0.13%)†	8.704 (+0.20%)‡	7.212(-0.04%)	5.420 (+0.39%)‡
New Section	Max	9.252 (+0.08%)	8.701 (+0.16%)†	7.246 (+0.43%)‡	5.419 (+0.37%)‡
Original	N/A	9.245	8.687	7.215	5.399

Table 3: Comparison of DCG-5 across query lengths. The values in parentheses express the relative difference in effectiveness versus the baseline. The † and ‡ symbols indicate statistically significant improvements over the baseline at the $p < 0.1$ and $p < 0.05$ level, respectively.

0.2%. This result is somewhat unexpected, as we hypothesized that aggregated anchor text would not considerably hurt the easy queries. One possible explanation for this behavior is that the aggregated anchor text brings in noisy lines of anchor text for navigational web pages, thereby leading to decreased retrieval effectiveness for these easy queries.

In addition, Table 3 shows the DCG-5 results for the best weight aggregation function and document representation combinations stratified by query length. The results show that aggregated anchor text improves longer queries more than shorter queries. This supports our claim that shorter queries, which are mostly navigational, are not helped by aggregated anchor text.

Thus, our proposed method is better at improving longer, more difficult queries, which are more often informational. This means that anchor text sparsity is less of a problem for short, navigational queries, since relevant documents can easily be found using the original anchor text. However, for longer, informational queries, relevant documents tend to be those that have little, or no, original anchor text. By augmenting these documents with aggregated anchor text, we are able to establish better, more effective rankings for such queries. These findings suggest that aggregated anchor text is likely to be most beneficial when used in conjunction with a navigational/informational query classifier [19] or a query difficulty predictor [34] in order to avoid using it for easy queries.

4.3.4 Aggregated Anchor Text Pruning

Finally, we address a practical concern of our methodology. By aggregating anchor text across the web graph, we may introduce spam, unrelated, or simply junk anchor text into a document's representation. Furthermore, as we described before, our approach adds, on average, 11 lines of aggregated anchor text to affected URLs. This may seem like a small amount of space overhead, but can quickly become problematic, especially for indexes that contain millions, or even billions, of documents. Therefore, we investigate the effect on effectiveness of limiting the number of lines of anchor text added to each document.

The plot in Figure 3 shows how DCG-5 varies with respect to the maximum number of lines of aggregated anchor text allowed to be added to each document. In this experiment, we only keep, at most, the k highest weighted lines of aggregated anchor text. The rest of the aggregated anchor text lines are discarded.

It is important to note that the plot in Figure 3 begins with one aggregated anchor text line, not zero. The zero case, which is the baseline system, corresponds to DCG-5 = 0.7576. As the plot indicates, no matter how many lines of aggregated anchor text we add, the result is still better than the baseline. This is encouraging, because, at the very minimum, we can add a single line of aggregated anchor text to each document, and still achieve considerable gains in retrieval effectiveness. The plot indicates that allowing a maximum of 100 lines of aggregated anchor text is the optimal policy. Interestingly, this policy is even better than the policy of using all of the aggregated anchor text lines. One possible explanation of this behavior is that spam, unrelated, or junk lines start to occur somewhere around the 100th line of anchor text.

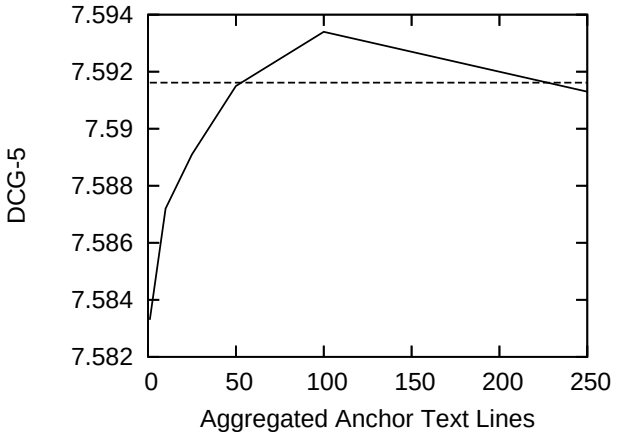

Figure 3: Effect of aggregated anchor text pruning. The plot shows DCG-5 versus the maximum number of aggregated anchor text lines allowed per URL. The dashed line indicates the DCG-5 obtained from using all of the anchor text.

Therefore, by limiting the maximum number of lines of aggregated anchor text per document, we can not only save space by dropping some lines of aggregated anchor text, but we can also improve retrieval effectiveness at the same time.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we framed the anchor text sparsity problem, which highlights the importance of anchor text for representing documents. We argue that document representations enriched with auxiliary anchor text can be useful for a variety of information retrieval and natural language processing tasks, including web search, contextual advertising, document classification, and summarization.

To address the problem, we proposed a method of aggregating, or propagating, anchor text across the web graph. Our approach defines the aggregated anchor text for a given page as the external anchor text of the internal inlinks. We also proposed six different weight aggregation schemes, inspired by result set fusion techniques, that can be used to

combine anchor text weights from multiple sources. In addition, we described various ways that document representations can be enriched with the aggregated anchor text, including the flat, combined, backoff, and new section representations. Finally, experimental evaluations were carried on a large-scale, real world web test collection. The results showed that our proposed approach reduces anchor text sparsity by 38% and consistently and significantly improves retrieval effectiveness, especially for longer, informational queries, which exhibited DCG-1 and DCG-5 improvements of over 10%.

There are several interesting directions of future work. This paper proposed a general framework for overcoming anchor text sparsity. However, in the future, it would be useful to extend this framework by exploring novel methods for collecting anchor text, weight aggregation functions, and enriched document representations. It would also be worthwhile to explore beyond simple BM25F scoring. For example, use the enriched document representations in conjunction with term proximity models [23, 32] or as a feature within a machine learned ranking function [5, 24, 14]. Finally, we would like to apply our framework to aggregated other textual representations, including document titles, abstracts, or social media tags.

6. REFERENCES

[1] J. Allan, B. Carterette, J. A. Aslam, V. Pavlu, B. Dachev, and E. Kanoulas. Million query track 2007 overview. In *Proc. 16th Text REtrieval Conference*, 2007.

[2] J. Bai, Y. Chang, H. Cui, Z. Zheng, G. Sun, and X. Li. Investigation of partial query proximity in web search. In *Proc. 17th. Intl. Conf. on World Wide Web*, pages 1183–1184, New York, NY, USA, 2008. ACM.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[4] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Comput. Netw.*, 33(1-6):309–320, 2000.

[5] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proc. 22nd Proc. Intl. Conference on Machine Learning*, pages 89–96, 2005.

[6] C. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2004 terabyte track. In *Proc. 13th Text REtrieval Conference*, 2004.

[7] N. Craswell and D. Hawking. Overview of the TREC 2003 web track. In *Proc. 12th Text REtrieval Conference*, 2003.

[8] F. Crestani. Application of spreading activation techniques in information retrieval. *Artif. Intell. Rev.*, 11(6):453–482, 1997.

[9] F. Diaz. Regularizing ad hoc retrieval scores. In *Proc. 14th Intl. Conf. on Information and Knowledge Management*, pages 672–679, 2005.

[10] E. Fox and J. Shaw. Combination of multiple searches. In *Proc. 2nd Text REtrieval Conference*, 1994.

[11] A. Fujii. Modeling anchor text and classifying queries to enhance web document retrieval. In *Proc. 17th Intl. Conf. on World Wide Web*, pages 337–346, New York, NY, USA, 2008. ACM.

[12] V. Harmandas, M. Sanderson, and M. D. Dunlop. Image retrieval by hypertext links. In *Proc. 20th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 296–303, New York, NY, USA, 1997. ACM.

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

[14] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. 8th Ann. Intl. ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, pages 133–142, 2002.

[15] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[16] O. Kurland and L. Lee. Corpus structure, language models, and ad hoc information retrieval. In *Proc. 27th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 194–201, 2004.

[17] O. Kurland and L. Lee. Respect my authority!: Hits without hyperlinks, utilizing cluster-based language models. In *Proc. 29th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 83–90, New York, NY, USA, 2006. ACM.

[18] J. H. Lee. Analyses of multiple evidence combination. In *Proc. 20th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 267–276, New York, NY, USA, 1997. ACM.

[19] U. Lee, Z. Liu, and J. Cho. Automatic identification of user goals in web search. In *Proc. 14th Intl. Conf. on World Wide Web*, pages 391–400, New York, NY, USA, 2005. ACM.

[20] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (salsa) and the tkc effect. *Comput. Netw.*, 33(1-6):387–401, 2000.

[21] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *Proc. 27th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 186–193, 2004.

[22] Q. Mei, D. Zhang, and C. Zhai. A general optimization framework for smoothing language models on graph structures. In *Proc. 31st Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 611–618, New York, NY, USA, 2008. ACM.

[23] D. Metzler and W. B. Croft. A Markov random field model for term dependencies. In *Proc. 28th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 472–479, 2005.

[24] R. Nallapati. Discriminative models for information retrieval. In *Proc. 27th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 64–71, 2004.

[25] P. Ogilvie and J. Callan. Combining document representations for known-item search. In *Proc. 26th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 143–150, 2003.

[26] T. Qin, T.-Y. Liu, X.-D. Zhang, Z. Chen, and W.-Y. Ma. A study of relevance propagation for web search. In *Proc. 28th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 408–415, New York, NY, USA, 2005. ACM.

[27] S. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proc. 17th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 232–241, 1994.

[28] S. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. In *Proc. 13th Intl. Conf. on Information and Knowledge Management*, pages 42–49, 2004.

[29] A. Shakery and C. Zhai. A probabilistic relevance propagation model for hypertext retrieval. In *Proc. 15th Intl. Conf. on Information and Knowledge Management*, pages 550–558, New York, NY, USA, 2006. ACM.

[30] K. Spärck Jones. Wearing proper combinations. Technical report, University of Cambridge, 2005.

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

[32] T. Tao and C. Zhai. An exploration of proximity measures in information retrieval. In *Proc. 30th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 295–302, New York, NY, USA, 2007. ACM.

[33] H. Zaragoza, N. Craswell, M. Taylor, S. Saria, and S. Robertson. Microsoft Cambridge at TREC 13: Web and hard tracks. In *Proc. 13th Text REtrieval Conference*, 2004.

[34] Y. Zhou and W. B. Croft. Query performance prediction in web search environments. In *Proc. 30th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 543–550, New York, NY, USA, 2007. ACM.