Language Models for Searching in Web Corpora

Jaap Kamps[1,2] **Gilad Mishne**[2] **Maarten de Rijke**[2]

[1] Archives and Information Studies, Faculty of Humanities, University of Amsterdam
[2] Informatics Institute, University of Amsterdam
http://ilps.science.uva.nl/

Abstract: We describe our participation in the TREC 2004 Web and Terabyte tracks. For the web track, we employ mixture language models based on document full-text, incoming anchor-text, and documents titles, with a range of web-centric priors. We provide a detailed analysis of the effect on relevance of document length, URL structure, and link topology. The resulting web-centric priors are applied to three types of topics—distillation, home page, and named page—and improve effectiveness for all topic types, as well as for the mixed query set. For the terabyte track, we experimented with building an index just based on the document titles, or on the incoming anchor texts. Very selective indexing leads to a compact index that is effective in terms of early precision, catering for the typical web searcher behavior.

1 Introduction

At TREC 2004 we took part in the Web and Terabyte tracks; our participation in the QA track is documented elsewhere [1]. Our aim for the Web track was to investigate a range of web-centric retrieval techniques based on an analysis of non-content features, such as document length, URL structure, and link topology. Our aim for the Terabyte track was to set up an initial system based on compact document representations such as titles or incoming anchor texts, and to compare the relative effectiveness of these document surrogates.

The rest of this paper is organized as follows. In two largely self-contained sections we describe our work for the Web (§2) and Terabyte (§3) tracks. We summarize our findings in a concluding section.

2 Web Track

We experimented with a range of techniques within the language modeling framework, exploiting natural ways to incorporate multiple document representations, as well as non-content information. We use three indexes based on document-text, incoming anchor-texts, and document titles, similar to those used for our submissions to TREC 2003 [6].

2.1 Mixture Language Models

For the web tasks we use a specific mixture language model based on the following formula:

$$P(q|d) = P(d) \cdot \prod_{i=1}^{n} ((1-\lambda) \cdot P(q_i|C) + \lambda \cdot P(q_i|d)) .$$

We employ three document models:

1. $P_{\text{text}}(q_i|d)$ the estimate based on the full-text index.
2. $P_{\text{anchor}}(q_i|d)$ the estimate based on the anchortext index.
3. $P_{\text{title}}(q_i|d)$ the estimate based on the titles index.

The three models are combined as follows:

$$P(q|d) = P(d) \cdot \prod_{i=1}^{n} ((1-\lambda_1-\lambda_2-\lambda_3) \cdot P(q_i|C) + \lambda_1 \cdot P_{\text{text}}(q_i|d) + \lambda_2 \cdot P_{\text{anchor}}(q_i|d) + \lambda_3 \cdot P_{\text{title}}(q_i|d)),$$

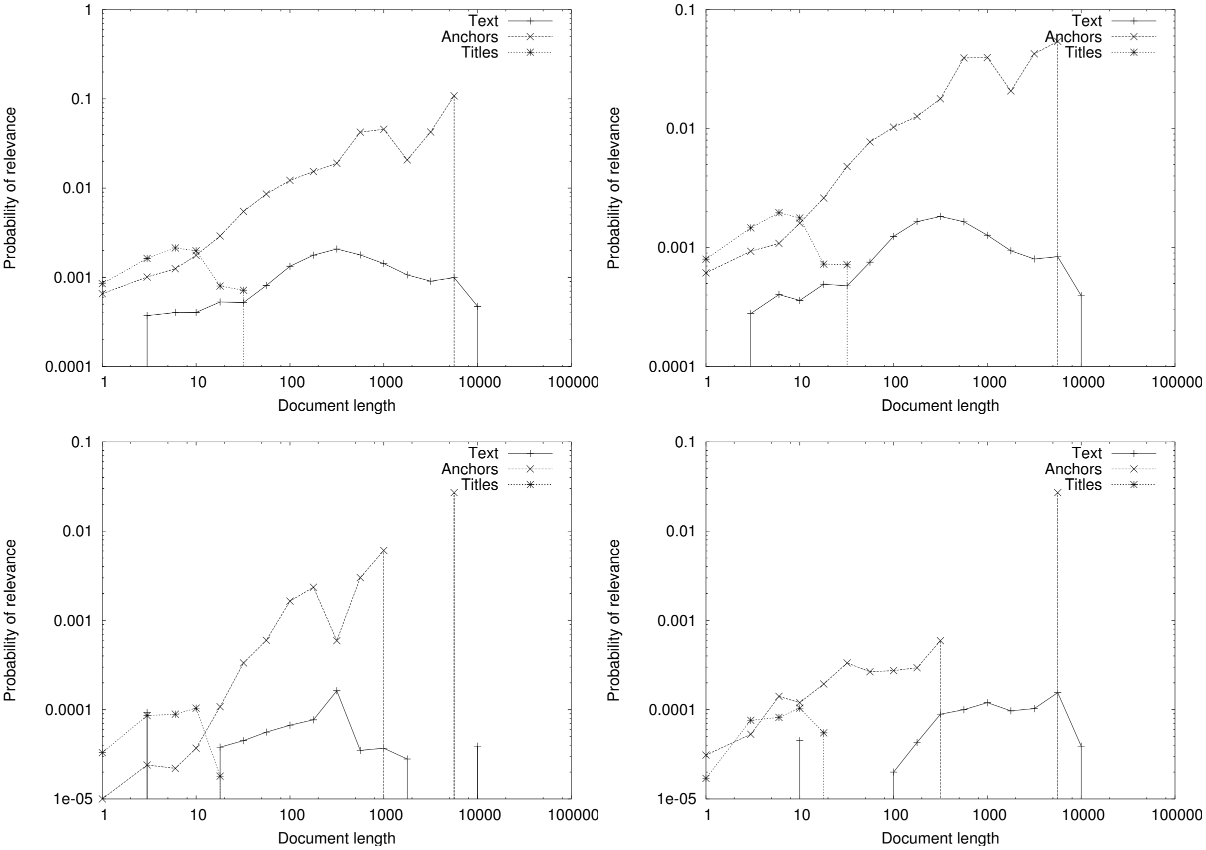

Figure 1: Document length versus relevance overall (top left), and for distillation (top right), home page (bottom left), and named page topics (bottom right).

where each of the document models is estimated using a maximum likelihood estimate. All runs on which we report below use equal weights for all three document models, that is $\lambda_1 = \lambda_2 = \lambda_3 = \lambda$.

We use the full text index as the collection model. The prior probability of a document, $P(d)$, can be used to incorporate non-content features into the scoring mechanism, as we will now explain.

2.2 Priors

We will now analyze a range of non-content features, such as the document length, the page's URL, or the link topology, and investigate their usefulness to boost retrieval effectiveness.

2.2.1 Document length

Let us focus on document length first. Figure 1 shows the prior probability of relevance against the length of a document for each of the three indexes (full-text, anchor-texts, and titles). The plot at the top left of the figure shows the prior probability of relevance of a web page for any of the mixed query topics. If we consider all mixed query topics, plotted in the figure at the top left, then the only marked length effect is for the anchor-text index.

Even though the three topic types are evenly dis-

Table 1: Number of relevant pages per topic type.

Type	Topics	# Rel	% Rel	Rel/Top
Topic distillation	75	1,600	90.8%	21.33
Home pages	75	83	4.7%	1.11
Named pages	75	80	4.5%	1.07
Mixed queries	225	1,763	100%	7.84

Table 2: Priors for the URL classes.

Class	Mixed	TD	HP	NP
Root	0.046845	0.042559	0.003990	0.000296
Subroot	0.003225	0.002894	0.000215	0.000116
Path	0.003440	0.003183	0.000167	0.000091
File	0.000786	0.000713	0.000018	0.000055

tributed, the number of relevant pages is not. Table 1 shows the number of relevant pages for each of the topic types in the TREC 2004 qrels. So, for over 90 percent the observed patterns can be attributed to the distillation topics. This is confirmed by looking at the results for the distillation topics only (top right plot in the Figure 1). As it turns out, for the other subtasks, home page finding (bottom left plot) and named page finding (bottom right plot), the results are fairly similar: the only marked length effect can be observed for the anchor-text index.

For each of the tasks the relevance of a page seems unrelated to the length of the page. It does have a relation with the length of a document in the anchor-text index. The length of the anchor-text document surrogate is directly correlating with the number of incoming links. Since the indegree of a page provides a more direct handle, we decided not to use document length as a factor for our web retrieval experiments.

2.2.2 URL

We will now focus on the uniform resource locator (URL) as a non-content feature, independent of the particular query at hand. Table 2 shows the prior probability of relevance for the familiar URL classes [7]. Note that, again, the results for the mixed queries are dominated by the distillation topics since they populate the pool of relevant documents. We break down the set of topics for the three individual topic types. The results for home page finding and named page finding are only in partial agreement with the distillation topics. There is a reversal of the relative importance for the Subroot and Path classes for the known-item topics. Also, for the named page topics, the Root class pages are only moderately more relevant, on average, than pages in the Subroot class. Although it is clear that these coarse-grained URL classes can be used as a prior for retrieval, we want to investigate more fine-grained measures of URL length.

We first normalize the URLs by removing "www" prefixes, and "index.htm(l)" postfixes. We investigate three measures of the length of the URL:

URL Slash Count Simply count the number of occurrences of "/" in the URL. For example `trec.nist.gov/act_part/act_part.html` has a slash count of 2.

URL Character Length Simply count the number of symbols in the URL. For example `trec.nist.gov/act_part/act_part.html` has a character length 36.

URL Component Length Split the URL in the *domain name* and *file path*, count the number of "." separated components in the domain name, and count the number of "/" separated components in the file path. For example `trec.nist.gov/act_part/act_part.html` will split in the domain name `trec.nist.gov` and the file path `act_part/act_part.html`. The domain name has 3 components, and the file path 2, making a component length of 5.

Figure 2 shows the prior probability of relevance for the three measures of URL length. The length of a URL has a clear reciprocal relation with relevancy: the shorter the URL, the more likely the page is to be relevant. Although all three URL length indicators can be used, pre-submission experiments on TREC 2003 data suggested that URL component length is the most promising.

In particular, we experimented with three operationalizations to the URL priors:

Linear The prior is proportional to 11 − *component_length* if the length is maximally 10, use 0.1 otherwise.

Linear Squared The prior is proportional to the square of the linear prior.

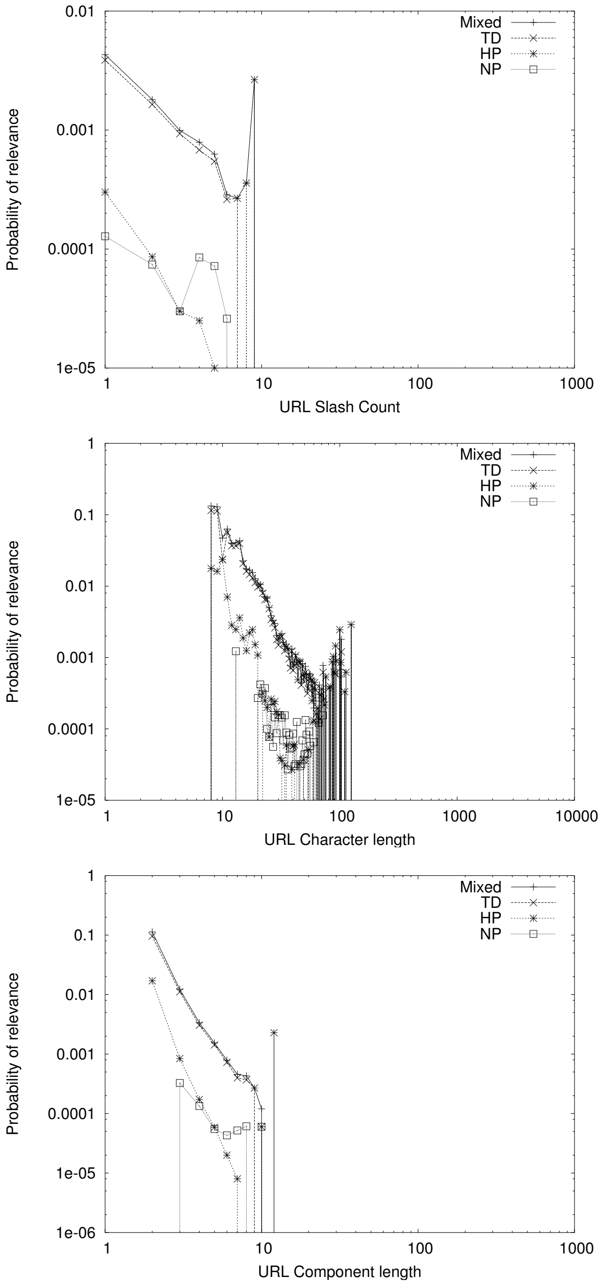

Figure 2: URL length in terms of slashes (top), characters (middle), and 'components' (bottom).

Product The prior is proportional to $\frac{1}{component\_length}$.

Product Squared The prior is proportional to $(\frac{1}{component\_length})^2$.

On pre-submission experiments using TREC 2003 data, the product squared prior proved to be the most effective, so we decided to use it for our official 2004 submissions.

2.2.3 Link Topology

Next, we focus on the link topology. We restrict our attention to the indegree and outdegree of pages:

Indegree the number of pages linking to a document, and

Outdegree the number of pages to which a document links.

Figure 3 shows the prior probability of relevance over indegree and outdegree. The degree of a page has a clear relation with relevancy: the more links a pages receives, or the more pages it links to, the more likely it is that the page is relevant. Pages with many inlinks are generally good authorities, and pages with many outlinks are generally good hubs.

We used three operationalizations of the priors.

Indegree The prior is proportional to the indegree.

Log Indegree The prior is proportional to the log of the indegree.

Outdegree The prior is proportional to the outdegree.

Log Outdegree The prior is proportional to the log of the outdegree.

Pre-submission experiments on the TREC 2003 data set gave the best results for the plain Indegree prior. So we decided to use the Indegree prior in our official 2004 submissions.

2.2.4 Implementing the Priors

For the implementation of the prior probability of the documents, we face a choice of methods:

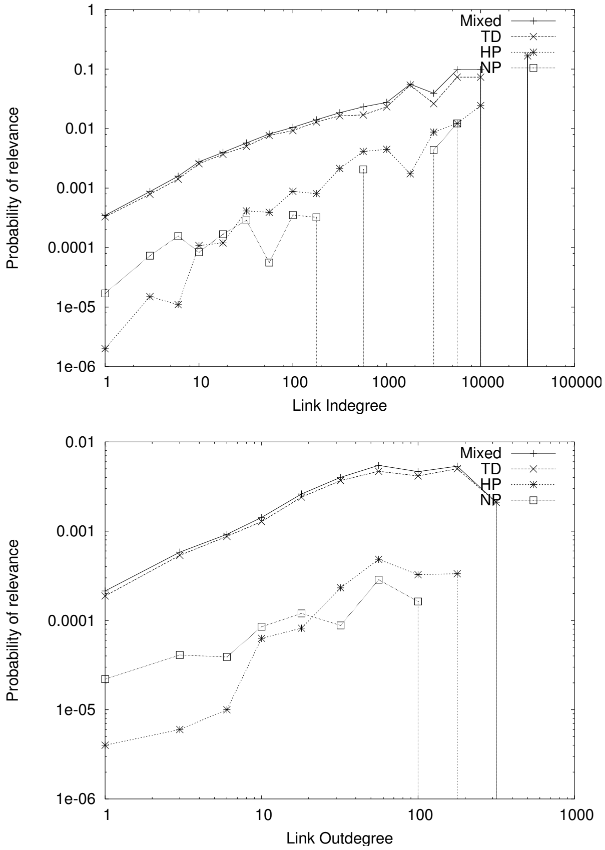

Figure 3: Link indegree (top) and link outdegree (bottom).

Within the Language Model An elegant way to implement the prior is directly in the language modeling scoring formula (see §2.1). This implies that the corresponding prior for all documents in the collection needs to be calculated, and is being fed into the language model. The result set consists of the 1,000 documents with the highest final score, based on both the content and the prior.

Reranking Prior Alternatively, one may argue that the prior should not influence what pages are returned, but only influence the relative ranking of pages returned because of their content. This can be realized in the following way: a content-based run is produced not using the prior, and the score is recalculated by multiplying the content-based score with the prior probability. The result set now consists of the 1,000 documents with the highest content-based score, reranked according to the final score.

For some priors, the reranking implementation is much more effective. Consider, for example, the case of an indegree prior. Here, the indegree can be fairly large number (ranging from 1 to 44,499), causing the infiltration of pages with a very low content-score, but a very high indegree. For our official runs, we used the priors as a reranking of an original, content-based result set.

2.3 Query Operations

In addition to our language modeling experiments, we conducted experiments to measure the effect of phrase and proximity query operations in the context of web retrieval; for this, we tested a variety of query-rewrite heuristics using phrases and proximity terms, in the vector space model.

The usage of phrases and proximity operators for adhoc retrieval has been studied extensively. Reports on their contribution are mixed, and it is generally accepted now that with a good basic ranking formula, the effectiveness of phrases is negligible or even negative [10], while recent evaluations of the use of automatically generated proximity terms suggest that term proximity may improve retrieval effectiveness especially at the top documents retrieved [13]. However, these evaluations use traditional ad-hoc test-sets as their data; web retrieval is different both in document structure and in query characteristics. Because of the nature of HTML, documents may be represented by using different sections of the HTML source (as we did for our language modeling experiments). Some of these representations, such as those based on anchor text and title, tend to be very short, phrase-like text. Queries are also different: they are shorter and more focused than ad-hoc queries (even when taking only the "title" of a topic). Indeed, several participants report improvements based on proximity information, spans, and phrases in various ways. We systematically investigated the effect of phrase and proximity operators on web retrieval, aiming to see whether it differs from the effect in non-web collections.

2.3.1 Query reformulation

We use a straightforward query rewrite mechanism which adds phrase or proximity terms to the topic. Identifying phrases, or words to be included in a proximity term, is traditionally done with statistical, syntactical, or lexical methods [2, 3, 10, 12]. We use s simple, shallower way; in our approach, every word n-gram from the query, of any length, is a phrase (or a proximity term). For example, for topic WT04-58 from, "automobile emissions vehicle pollution," it seems that in addition to the linguistically and statistically motivated phrases "automobile emissions" and "vehicle pollution", viewing "emissions vehicle" as a phrase may also be beneficial (after stopping and stemming, it matches "emissions from a vehicle" or "emitted by vehicles").

For our ranking, we use the default similarity measure in Lucene [8], i.e., for a collection D, document d and query q containing terms t_i:

$$sim(q,d) = \sum_{t \in q} \frac{tf_{t,q} \cdot idf_t}{norm_q} \cdot \frac{tf_{t,d} \cdot idf_t}{norm_d} \cdot coord_{q,d} \cdot weight_t,$$

where

$$\begin{aligned} tf_{t,X} &= \sqrt{\mathrm{freq}(t,X)} \\ idf_t &= 1 + \log \frac{|D|}{\mathrm{freq}(t,D)} \\ norm_d &= \sqrt{|d|} \\ coord_{q,d} &= \frac{|q \cap d|}{|q|} \\ norm_q &= \sqrt{\sum_{t \in q} tf_{t,q} \cdot {idf_t}^2} \end{aligned}$$

The *idf* of phrase or proximity terms is estimated by using the minimal *idf* of the words in the term.

We experimented with a range of approaches to query modifications, including measuring the effect on different document representations and different weighting schemes for terms. One of these experiments, a linear combination of a proximity term run and a phrase term run will be discussed below. A detailed description of all experiments is given in [9].

2.4 Runs

We created two "base" runs using the mixture language model (see §2.1) on either the three stemmed indexes, or the three non-stemmed indexes:

`UAmsT04MW` Mixture language models on the non-stemmed indexes (Full-text, Anchors, Titles), $\lambda = 0.3$

`UAmsT04MS` Mixture language models on the Snowball [14] stemmed indexes (Full-text, Anchors, Titles), $\lambda = 0.1$

The word-based run is geared toward precision, hence the higher value of the smoothing parameter.

These two base runs were reranked with either an

Indegree prior the prior probability of a document is proportional to *indegree*, or an

URL-length prior the prior probability of a document is proportional to $(\frac{1}{component\_length})^2$.

This resulted in the following four runs:

`UAmsT04MWind` Mixture language models on the non-stemmed indexes (Full-text, Anchors, Titles), $\lambda = 0.3$, using an indegree prior.

`UAmsT04MWurl` Mixture Language models on the non-stemmed indexes (Full-text, Anchors, Titles), $\lambda = 0.3$, using an URL prior.

`UAmsT04MSind` Mixture language models on the stemmed indexes (Full-text, Anchors, Titles), $\lambda = 0.1$, using an indegree prior.

`UAmsT04MSurl` Mixture language models on the stemmed indexes (Full-text, Anchors, Titles), $\lambda = 0.1$, using a URL prior.

The run labeled `UAmsT04MSind` was one of our official 2004 submissions.

The same URL-length prior has been applied to the indegree prior runs:

`UAmsT04MWinu` Mixture language models on the non-stemmed indexes (Full-text, Anchors, Titles), $\lambda = 0.3$, using an indegree prior, and an URL prior.

UAmsT04MSinu Mixture language models on the stemmed indexes (Full-text, Anchors, Titles), $\lambda = 0.1$, using an indegree prior, and an URL prior.

The runs labeled `UAmsT04MWinu` and `UAmsT04MSinu` were both part of our official 2004 submissions.

These two resulting runs were combined using CombMNZ on the non-normalized scores [4]:

UAmsT04MWScb CombMNZ (non-normalized, non-weighted) of runs `UAmsT04MWinu` and `UAmsT04MSinu`.

We also submitted the run labelled `UAmsT04MWScb` as an official run for 2004.

There is one further run experimenting with methods for boosting early precision in the vector space model:

UAmsT04LnuNG A linear combination of a proximity term run and a phrase term run, with equal weights assigned to both runs, based on the query rewrite approach discussed in section 2.3. We use indegree and URL length reranking in the same manner as they are employed in our language model runs.

Run `UAmsT04LnuNG` completes the set of official runs for TREC 2004.

Table 3: Results for topic distillation.

Run identifier	MAP	S@1	S@5	S@10
`UAmsT04MW`	0.0980	0.1733	0.3867	0.5600
`UAmsT04MS`	0.0973	0.1733	0.4133	0.5333
`UAmsT04MWurl`	0.1118	0.1867	0.4133	0.6133
`UAmsT04MSurl`	0.1169	0.1867	0.4667	0.6400
`UAmsT04MWind`	0.1310	0.3067	0.6400	0.7333
`UAmsT04MSind`	0.1328	0.2933	0.6533	0.7600
`UAmsT04MWinu`	0.1418	0.3467	0.6533	0.7733
`UAmsT04MSinu`	**0.1462**	0.3733	**0.7200**	**0.7867**
`UAmsT04MWScb`	**0.1462**	0.3600	0.6667	0.7600
`UAmsT04LnuNG`	0.1447	**0.4267**	0.6667	0.7467

Table 4: Results for home page finding.

Run identifier	MRR	S@1	S@5	S@10
`UAmsT04MW`	0.4265	0.2933	0.6133	0.7200
`UAmsT04MS`	0.4438	0.3200	0.6000	0.7200
`UAmsT04MWurl`	0.5744	0.4667	0.6933	0.7867
`UAmsT04MSurl`	0.5895	0.4800	0.7067	0.7600
`UAmsT04MWind`	0.6415	0.5467	0.7333	0.7867
`UAmsT04MSind`	0.6575	**0.5600**	0.7467	0.8267
`UAmsT04MWinu`	0.6402	0.5200	0.7733	0.8267
`UAmsT04MSinu`	**0.6586**	**0.5600**	0.7600	0.8267
`UAmsT04MWScb`	0.6451	0.5200	**0.7867**	**0.8400**
`UAmsT04LnuNG`	0.5858	0.5333	0.6400	0.6800

2.5 Results

Before we discuss our results for the mixed query task, we present the results for a breakdown of the set of topics into the three subtasks, i.e., topic distillation, home page finding, and named page finding.

2.5.1 Topic Distillation

The results for the topic distillation subtask are shown in Table 3 (best scores in boldface). The second column gives the mean average precision score, the three remaining columns the percentage of topics with at least one relevant document in the top 1, top 5, or top 10. For topic distillation, we make the following observations. First, all priors (URL, indegree, and combined prior) pay off, leading to impressive improvements over the content-based scores. In particular, the indegree prior makes a substantial difference. This is true both for our language modeling runs and for our vector space run. Second, the differences between the stemmed and non-stemmed indexes are not very large, with the stemmed indexes slightly superior for most of the scores. Finally, the run using query word n-grams tailoring for precision received, with distance, the best score for success at 1.

2.5.2 Home Page Finding

The results for the home page finding subtask are shown in Table 4 (best scores in boldface). The second column here gives the mean reciprocal rank. In case there is only a single document judged relevant, MAP and MRR will coincide. The basic idea of known-item search is that there is a single target page. However, due to duplicates in the collection, there may be more than one page judged relevant (see also Table 1). Hence, the mean reciprocal rank score better reflects the underlying navigational task, but can be straightforwardly combined with mean average precision scores for the informational topics.

For this task, we find the following. Firstly, as with

Table 5: Results for named page finding.

Run identifier	MRR	S@1	S@5	S@10
UAmsT04MW	0.6656	0.5733	0.8000	0.8667
UAmsT04MS	0.6595	0.5467	**0.8133**	0.8667
UAmsT04MWurl	0.6736	0.5733	**0.8133**	0.8667
UAmsT04MSurl	**0.6865**	**0.6000**	0.7867	0.8533
UAmsT04MWind	0.6451	0.4933	**0.8133**	**0.8800**
UAmsT04MSind	0.6398	0.5067	0.8000	0.8667
UAmsT04MWinu	0.6123	0.4533	0.8000	0.8667
UAmsT04MSinu	0.6045	0.4533	0.7600	0.8400
UAmsT04MWScb	0.6240	0.4667	**0.8133**	0.8667
UAmsT04LnuNG	0.4283	0.3067	0.5867	0.6533

Table 6: Results for mixed queries.

Run identifier	MAP/RR	S@1	S@5	S@10
UAmsT04MW	0.3967	0.3467	0.6000	0.7156
UAmsT04MS	0.4002	0.3467	0.6089	0.7067
UAmsT04MWurl	0.4533	0.4089	0.6400	0.7556
UAmsT04MSurl	0.4643	0.4222	0.6533	0.7511
UAmsT04MWind	0.4725	0.4489	0.7289	0.8000
UAmsT04MSind	**0.4767**	0.4533	0.7333	0.8178
UAmsT04MWinu	0.4648	0.4400	0.7422	**0.8222**
UAmsT04MSinu	0.4698	**0.4622**	0.7467	0.8178
UAmsT04MWScb	0.4718	0.4489	**0.7556**	**0.8222**
UAmsT04LnuNG	0.3863	0.4222	0.6311	0.6933

the earlier topic distillation task, for this task the priors pay off as well. There is a substantial improvement for both the URL and indegree prior. The best MRR score is for the combined prior, although the result is very close to the result of the indegree prior only. Secondly, the runs on the stemmed indexes are generally somewhat better than those on the non-stemmed indexes. Finally, the scores obtained here are, in an absolute sense, much higher than for the distillation topics. This implies that the home page topics will have a larger impact on the MRR score over all mixed queries.

2.5.3 Named Page Finding

The results for the named page finding subtask are shown in Table 5 (best scores in boldface). For the named page finding task, we see the following. First, the performance of the plain mixture model runs (with a uniform prior) is impressive with over 80 percent of the topics in the top 5. The performance is much higher than the plain mixture model runs for the other known-item search task, home page finding. Second, the priors are much less effective than for the distillation and home page finding topics. The results for the priors are mixed at best: the URL prior leads still to a slight gain in performance, but indegree and combined prior lead to a loss of performance. Thirdly, although the differences are small, the runs on the non-stemmed indexes are generally somewhat superior to the stemmed indexes. Finally, also the scores for the second known-item task are, in an absolute sense, much higher than for the distillation topics. This implies that the home page finding and named page finding topics will dominate the score over all mixed queries.

2.5.4 Mixed Query Task

We now discuss the results of the whole set of mixed query topics. The results are shown in Table 6 (best scores in boldface). The second column here gives the mean of average precision (topic distillation topics) and reciprocal ranks (known-item topics). For the entire set of mixed query topics, we see the following. First of all, the priors help to improve retrieval effectiveness. The indegree only prior is the most effective and gets the highest MAP/RR score. The combined priors get a slightly lower MAP/RR score, but slightly higher success at 1, 5, and 10 scores. Second, the stemmed indexes are slightly superior to the non-stemmed indexes, although the differences are small. Finally, the overall performance of the retrieval system is impressive with an MAP/RR of close to 0.5, and over 80% of the topics with at least one relevant page in the top 10.

2.5.5 Conclusions

Two web-centric techniques, the use of URL structure and the use of web topology, were shown to be effective for the mixed query task. The break down of the task in topic distillation, home page finding and named page finding, revealed that these techniques are particularly helpful for distillation and home page topics, but give mixed results for the named page topics. In terms of mean average precision, topic distillation is a much harder task than the known-item searches. This implies that the MAP for the known-item topics will also dominate the mixed queries score, and that a system tuned for known-item search may easily outcompete a generic web retrieval system. For the success at *n* measures, all topic types contribute equally; hence, for the mixed queries the success at *n* scores seem to be the best performance indicators for this task.

For our query operation experiments, we conclude that usage of query operations such as phrases is more beneficial where there are multiple representations of documents—particularly when some of these representations tend to be short and phrase-like (such as title or anchor text, in the web retrieval case). The query operations provide a better performance gain for distillation topics than for known item topics.

3 Terabyte Track

We performed some initial experiments for the Terabyte track, aiming to test the scalability of some of the techniques proven effective for the smaller web collections.

3.1 Indexes

For the .GOV2 collection, we built the following two indexes:

Titles Snowball stemmed index of all ⟨title⟩ fields. The index contains all 25,205,179 documents, although only 20,919,902 have text (after removing stopwords). Thus, the index covers 83% of the total collection.

The indexing proper took 240 minutes, preprocessing took ± 5 days to extract the titles from the collection. The total size of the index is 1,406 MB. An exhaustive run takes 17 minutes and 21 seconds for all 50 title-only topics.

Anchors Snowball stemmed index of all incoming anchor-texts, only considering fully specified URLs, i.e., `http://xxx.yyy/zzz`. We only index the anchor-text (if present, some links are on non-text), and ignore the ALT fields. We only index a single occurrence of repeated anchor-texts.

These are all between-site links plus only verbose within-site links; most within-site links are ignored. Contains in total 1,643,078 documents, although only 1,507,499 have text (after stopping). Thus, this covers in total 6% of the total collection.

The indexing proper took 23 minutes, preprocessing took ± 5 days for anchor-text extraction, and ± 10 hours on generating the propagated anchor-text documents. The total size of the index is 105.6 MB. An exhaustive run takes 33 seconds for the 50 title-only topics.

Based on the extracted anchor-texts (non-sorted), we calculated the within-collection indegree. This indegree can be used as a prior in the following way. As with the Web Track experiments, we use a prior that is proportional to the indegree. However, since the indegree can be fairly large number (ranging from 1 to 1,834,555), this may cause the infiltration of pages with a very low content-score, but a very high indegree. Thus, we decided to apply the prior as a reranking post-processor (see §2.2.4). Since reranking the top 10,000 documents will effectively allow the infiltration of almost any page with a very low content-score, we decide to only “rerank” the top 100 documents. Since we calculate the actual probabilities in the mixture model (as detailed in §2.1), we can simply multiply by the degree (without dividing with the sum of all degrees). Since we now multiply with a number that is larger or equal than one, we will never get a lower similarity score by applying the prior. Now, we’ll only apply the length prior to the 100 documents with the highest content-based similarity score. At ranks 101 through 10,000, the documents remain ranked according to the content-score only.

3.2 Runs

We submitted the following five runs, all using only the title field of the topics:

`UAmsT04TBtit` Language model run on the stemmed titles with $\lambda = 0.7$ and length-prior.

`UAmsT04TBanc` Language model run on the stemmed anchors with $\lambda = 0.7$, and length-prior.

`UAmsT04TBm1` We use a mixture language model (see §2.1) run on the stemmed titles and anchors, with $\lambda = 0.1$ and no length-prior. We use the titles index as the collection model.

Table 7: Results for the Terabyte track.

Run identifier	MAP	MRR	P@5	P@10
UAmsT04TBtit	0.0388	0.5250	0.2980	0.2306
UAmsT04TBanc	0.0132	0.4043	0.2367	0.1918
UAmsT04TBm1	**0.0435**	**0.5587**	0.3102	**0.2816**
UAmsT04TBm3	0.0432	0.5351	**0.3265**	0.2755
UAmsT04TBm1p	0.0431	0.5271	0.3184	0.2673

UAmsT04TBm3 Mixture language model run on the stemmed titles and anchors, with $\lambda = 0.3$ and no length prior.

UAmsT04TBm1p Mixture language model run on the stemmed titles and anchors, with $\lambda = 0.1$ and no length prior, using an indegree prior on the top 100 documents per topic.

3.3 Results

The results for the Terabyte track are shown in Table 7 (best scores in boldface). The second column gives the mean average precision, the second column gives the mean reciprocal rank, and the remaining two columns the precision at 5 and 10 respectively. Our findings are the following. First, and unsurprisingly, the compact indexes result in low mean average precision values. Second, taking the small size of the indexes into account, the early precision scores are impressive. Third, the mixture model runs improve substantially over the individual indexes. Fourth, the use of an indegree prior doesn't lead to improvement, only for precision at 5 the run using the prior scores superior.

4 Conclusions

In this paper we have described our participation in the TREC 2004 Web and Terabyte tracks.

For the Web track, our findings highlighted that web retrieval is unlike standard ad hoc retrieval. Whereas document-length is a useful indicator for relevance in the general ad hoc case, it is not for the case of web retrieval. Specific webcentric techniques, such as using the URL structure or using the link topology, turned out to be useful indicators of relevance for the mixed query task. Our findings extend on earlier results on the effectiveness of language model priors for web retrieval. Kraaij et al. [7] established the effectiveness of webcentric priors for the home-page finding task. Ogilvie and Callan [11] extended these results to the other navigational task of named-page finding. Our findings extend these results to the informational task of topic distillation. In our experiments, the web-centric techniques were particularly useful for topic distillation and home page finding, and less benificial for named page finding. This can be easily explained by the task definition that required returning home pages of sites for both topic distillation and named page finding.

For the Terabyte track, our findings showed the relative effectiveness of very selective indexing. Based on just indexing the page's title field, or just the incoming anchor-texts, we created compact indexes. Where the incomplete indexes result in poor MAP scores, the early precision scores are impressive. That is, the compact indexes cater for the average web searcher, who doesn't look beyond the first handful of pages. Of course, finding at least one or a few relevant pages becomes easier if the size of the collection increases—a fact well exploited by Internet search engines. As a result, early precision scores tend to increase with the size of the collection [5]. In this light, we expect that a full-text index results in even better scoring for early precision than our compact indexes. However, at least in theory, we could build such a compact for collections far beyond the size of .GOV2, which may, in turn, again result in superior scoring.

Acknowledgments

Thank you to Börkur Sigurbjörnsson for useful suggestions and discussion.

This research was supported by the Netherlands Organization for Scientific Research (NWO) under project numbers 017.001.190, 220-80-001, 264-70-050, 612-13-001, 612.000.106, 612.000.207, 612.066.302, and 612.069.006.

References

[1] D. Ahn, V. Jijkoun, G. Mishne, K. Müller, M. de Rijke, and S. Schlobach. Using Wikipedia in the TREC QA Track. In *This volume*, 2005.

[2] A. Arampatzis, T. van der Weide, C. Koster, and P. van Bommel. An Evaluation of Linguistically-motivated Indexing Schemes. In *Proceedings of the 22nd BCS-IRSG Colloquium on IR Research*, 2000.

[3] J. Fagan. Experiments in automatic phrase indexing for document retrieval: A comparison of syntactic and non-syntactic methods. Technical report, Cornell University, 1987.

[4] E. Fox and J. Shaw. Combination of multiple searches. In D. Harman, editor, *The Second Text REtrieval Conference (TREC-2)*, pages 243–252. National Institute for Standards and Technology. NIST Special Publication 500-215, 1994.

[5] D. Hawking and S. Robertson. On collection size and retrieval effectiveness. *Information Retrieval*, 6: 99–150, 2003.

[6] J. Kamps, C. Monz, M. de Rijke, and B. Sigurbjörnsson. Approaches to robust and web retrieval. In E. M. Voorhees and L. P. Buckland, editors, *The Twelfth Text REtrieval Conference (TREC 2003)*, pages 594–599. National Institute of Standards and Technology. NIST Special Publication 500-255, 2004.

[7] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In K. Järvelin, M. Beaulieu, R. Baeza-Yates, and S. H. Myaeng, editors, *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 27–34. ACM Press, New York NY, USA, 2002.

[8] Lucene. The Lucene search engine, 2005. `http://jakarta.apache.org/lucene/`.

[9] G. Mishne and M. de Rijke. Boosting web retrieval through query operations. In *Proceedings ECIR 2005*, 2005.

[10] M. Mitra, C. Buckley, A. Singhal, and C. Cardie. An analysis of statistical and syntactic phrases. In *Proceedings of RIAO-97*, 1997.

[11] P. Ogilvie and J. Callan. Combining document representations for known-item search. In C. Clarke, G. Cormack, J. Callan, D. Hawking, and A. Smeaton, editors, *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 143–150. ACM Press, New York NY, USA, 2003.

[12] J. Pickens and W. Croft. An exploratory analysis of phrases in text retrieval. In *Proceedings of RIAO-2000*, 2000.

[13] Y. Rasolofo and J. Savoy. Term proximity scoring for keyword-based retrieval systems. In *Proceedings 25th European Conference on IR Research (ECIR 2003)*, pages 207–218, 2003.

[14] Snowball. Stemming algorithms for use in information retrieval, 2005. `http://www.snowball.tartarus.org/`.