Retrieving Web Pages using Content, Links, URLs and Anchors

Thijs Westerveld[1], Wessel Kraaij[2], and Djoerd Hiemstra[1]

[1] University of Twente, CTIT, P.O. Box 217, 7500 AE Enschede, The Netherlands
{hiemstra,westerve}@cs.utwente.nl
[2] TNO-TPD, P.O. Box 155, 2600 AD Delft, The Netherlands
kraaij@tpd.tno.nl

Abstract. For this year's web track, we concentrated on the entry page finding task. For the content-only runs, in both the ad-hoc task and the entry page finding task, we used an information retrieval system based on a simple unigram language model. In the Ad hoc task we experimented with alternatieve approaches to smoothing. For the entry page task, we incorporated additional information into the model. The sources of information we used in addition to the document's content are links, URLs and anchors. We found that almost every approach can improve the results of a content only run. In the end, a very basic approach, using the depth of the path of the URL as a prior, yielded by far the largest improvement over the content only results.

1 Introduction

Entry page search searching is different from general information searching, not only because entry pages differ from other web documents, but also because the goals of the tasks are different. Where in general information seeking we're interested in finding as much information as possible, for entry page searches we're looking for one specific document. Therefore, the entry page task is clearly a high precision task. Because of both the differences in the task and in the documents, for this task information sources other then the document's content can be very useful for locating the relevant entry page.

For the content-only runs, in both the ad-hoc task and the entry page finding task, we used an information retrieval system based on a simple unigram language model. This IR model, which we introduced at the TREC-7 conference [4] and which worked effectively on last year's web task, is presented in section 2. Section 3 describes how we used links, anchors and URLs to improve a content only run and section 4 lists the official results for the submitted runs as well as the results for the additional runs we did. Finally, section 5 lists our conclusions.

2 Basic IR model

All runs were carried out with an information retrieval system based on a simple statistical language model [3]. The basic idea is that documents can be represented by unigram language models. Now, if a query is more probable given a language model based on document d_1, than given e.g. a language model based on document d_2, then we hypothesise that the document d_1 is more relevant to the query than document d_2. Thus the probability of generating a certain query given a document-based language model can serve as a score to rank documents with respect to relevance.

$$P(T_1, T_2, \cdots, T_n|D_k)P(D_k) \;=\; P(D_k)\prod_{i=1}^{n}(1-\lambda)P(T_i|C) + \lambda P(T_i|D_k) \tag{1}$$

Equation 1 shows the basic idea of this approach to information retrieval, where the document-based language model is smoothed by interpolation with a background language model to compensate for sparseness. In the equation, T_i is a random variable for the query term on position i in the query ($1 \leq i \leq n$, where n is the query length), which sample space is the set $\{t^{(0)}, t^{(1)}, \cdots, t^{(m)}\}$ of all terms in the collection. The probability measure $P(T_i|C)$ defines the probability of drawing a term at random from the collection, $P(T_i|D_k)$ defines the probability of drawing a term

at random from document k; and λ is the interpolation parameter[1]. The a-priori probability of relevance $P(D_k)$ is usually taken to be a linear function of the document length, modelling the empirical fact that longer documents have a higher probability of relevance.

2.1 Combining external information

The basic ranking model is based on the content of the web pages. There is evidence that other sources of information (link structure, anchor text) play a decisive role in the ranking process of entry pages (e.g. Google[2]). The preferred way to incorporate extra information about web pages is to include this information in the model. A clean method is to incorporate this information in the prior probability of a document. A second manner is to model different types of evidence as different types of ranking models, and combine these methods via interpolation.

$$score_{\text{combi}} = \alpha score_{\text{content}} + (1-\alpha) score_{\text{features}} \tag{2}$$

Equation 2 shows how two ranking functions can be combined by interpolation. The combined score is based on a weighted function of the unigram document model and the posterior probability given the document feature set and a Bayesian classifier trained on the training set. As features we experimented with the number of inlinks and the URL form. However, for interpolation, scores have to be normalised across queries, because the interpolation scheme is query independent. Therefore, for the interpolation method we normalised the content score by the query length, the ranking models based on other document information that we applied are (discriminative) probabilities and thus need no normalisation. The interpolation method has shown to work well in cases where score normalisation is a key factor [6]. For the experiments we describe here, we have applied both methods and they yield similar results. In a context where score normalisation is not necessary, we prefer method one. We determined the document priors (document-content independent prior probabilities) using various techniques, either postulating a relationship, or learning priors from training data conditioning on e.g. the URL form. This process will be described in more detail in the Section 3.

2.2 Smoothing variants

Recent experiments at CMU have shown that the particular choice of smoothing technique can have a large influence on the retrieval effectiveness. For title adhoc queries, Zhai and Lafferty [8] found Dirichlet smoothing to be more effective than linear interpolation[3] Both methods start from the idea that the probability estimate for unseen terms: $P_u(T_i|D_k)$ is modelled a constant times the collection based estimate: $P(T_i|C)$. A crucial difference between Dirichlet and Jelinek-Mercer smoothing is that the smoothing constant is dependent on the document length for Dirichlet, reflecting the fact that probability estimates are more reliable for longer documents. Equation (3) shows the weighting formula for Dirichlet smoothing, where $c(T_i|D_k)$ is the term frequency of term T_i in document D_k , $\sum_w c(T_i; D_k)$ is the length of document D_k and μ is a constant. The collection specific smoothing constant is in this case $\frac{\mu}{\sum_w c(T_i;D_k)+\mu}$, whereas the smoothing constant is $(1-\lambda)$ in the Jelinek-Mercer based model.

$$P(T_1, T_2, \cdots, T_n|D_k)P(D_k) \;=\; P(D_k)\prod_{i=1}^{n}\frac{c(T_i; D_k) + \mu P(T_i|C)}{\sum_w c(T_i; D_k) + \mu} \tag{3}$$

3 Entry page Search

To improve the results of a content only run in the entry page finding task, we experimented with various link, URL and anchor based methods. We tested several well-known and novel techniques on the set of 100 training topics provided by NIST and found that each method we tested was more or less beneficial for finding entry pages. This contrasts with last year's findings where link based techniques didn't add anything in an ad hoc search task [7]. In the following subsections, we subsequently discuss link based methods, URL based methods and anchor based methods, along with our findings on the training data.

[1] We apply a simplified version of the model developed in [3], where λ is term specific, denoting the term importance

[2] `http://www.google.com`

[3] Also called Jelinek-Mercer smoothing.

3.1 Links

One of the sources of information one can use in addition to the content is the link structure. This is the structure of hyperlinks connecting the documents on the web. We took two different approaches exploiting this structure, both relying on the fact that entry pages tend to have a different link structure than other documents.

Inlinks The first link-based approach we experimented with is based on the assumption that entry pages tend to have a higher number of inlinks than other documents (i.e. they are referenced more often). A well known example of a commercial search engine which is based on a similar assumption is Google [1]. To check whether this assumption holds, we made a of plot of $P(entrypage|\#inlinks)$ (See Figure 1). The probabilities are estimated on half of the training data. The figure shows that indeed documents with more inlinks tend to have a higher probability of being an entry page. Therefore, for an entry page task, the number of inlinks might be a good prior. In fact, as figure 2 shows, the assumption that longer documents have a higher probability of being relevant does not hold for entry page searches and a prior based on the number of inlinks might be better than one based on the length of the document.

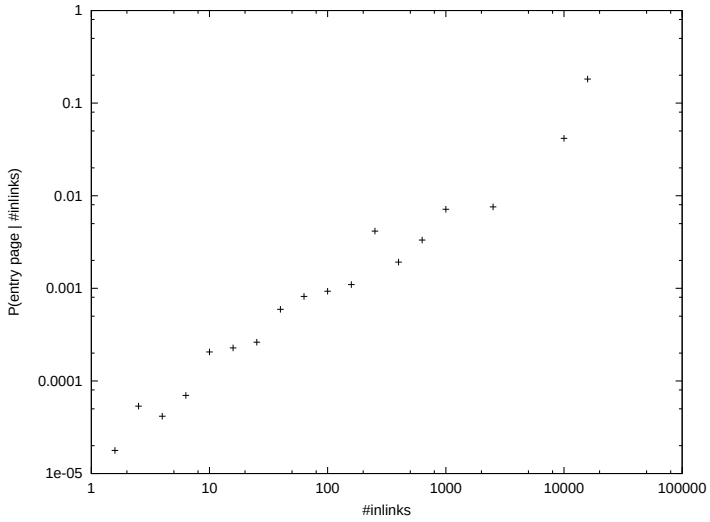

Fig. 1. $P(entrypage \mid \#inlinks)$

As a prior for ad hoc searches, we usually take a document length prior:

$$P(D_k) = \frac{doclen(D_k)}{\sum_{j=1}^{N} doclen(D_j)} \tag{4}$$

We define the inlink prior as:

$$P(D_k) = \frac{\#inlinks(D_k)}{\sum_{j=1}^{N} \#inlinks(D_j)} \tag{5}$$

We compared the two priors of equations 4 and 5 on the training data. We normalised the content score by the query length and interpolated with the inlink prior (cf. eq. 2), the doclen prior is used conform eq. 1. Table 1 shows the mean reciprocal ranks (MRR)[4]. The interpolation parameters used in the table gave the best results. The scores show that indeed, the number of inlinks is a better prior than the length of the document.

[4] The reciprocal of the rank of the relevant entry page averaged over all queries.

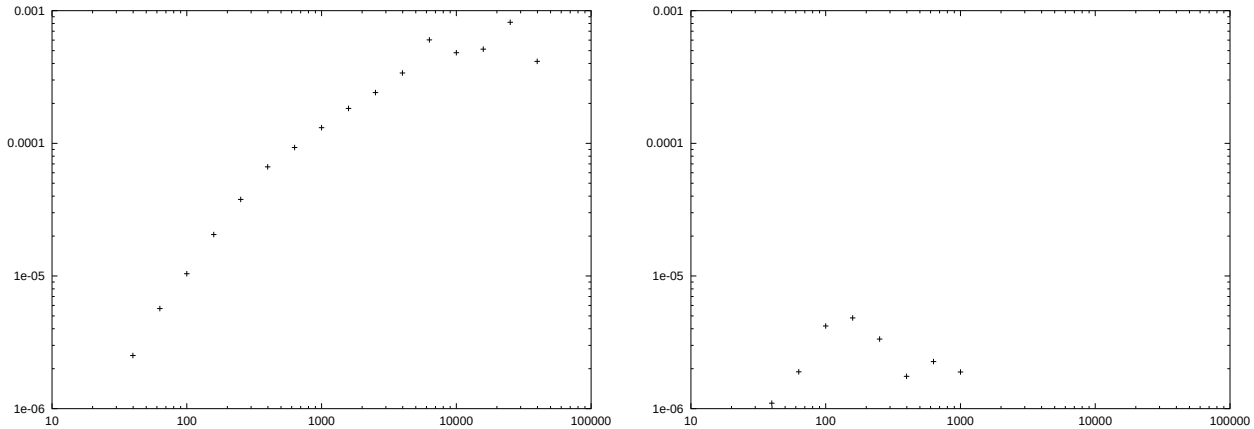

Fig. 2. $P(relevant \mid doclen)$

run	MRR
content	0.26
content + doclen prior	0.21
0.7 * content + 0.3 * inlink	0.38

Table 1. MRRs Inlink and doclen priors on training data

Kleinberg The second link-based approach we experimented with is based on Kleinberg's hub and authority algorithm [5]. This algorithm identifies authorities (important sources of information) and hubs (lists of pointers to authorities) by analysing the structure of hyperlinks. Since entry pages can be seen as authorities on a very specific subject (a certain organisation), Kleinberg's algorithm can be useful for the entry page search task. The algorithm works by iteratively assigning hub and authority scores to documents in such a way that good hubs are pages that refer to many good authorities and good authorities are referenced by many good hubs:

1. Take the top N results from the content run
2. Extend this set S with all documents that are linked to S (either through in or through outlinks)
3. Initialise all hub and authority scores in this set to 1.
4. $hub(D) = \sum_{\{i|link\ D \rightarrow i\ exists\}} auth(i)$
5. $auth(D) = \sum_{\{i|link\ i \rightarrow D\ exists\}} hub(i)$
6. normalise hub and auth scores such that $\sum_{s \in S} hub^2(s) = \sum_{s \in S} auth^2(s) = 1$
7. repeat steps 4 - 6

We computed hubs and authorities for the top N of the content only run and used the resulting authority scores to rank the documents. Table 2 shows the results for different values of N.

As the results show, taking only the top 5 or top 10 ranks from the content run and computing authority scores starting from those, is sufficient to improve the results. Apparently, if an entry page is not in the top 5 from the content run, it *is* often in the set of documents linked to these 5 documents.

3.2 URLs

Apart from content and links, a third source of information are the document's URLs. Entry page URLs often contain the name or acronym of the corresponding organisation. Therefore, an obvious way of exploiting URL information is

N	MRR
content	0.26
1	0.18
5	0.33
10	0.32
50	0.30

Table 2. MRRs Kleinberg@10 results on training data

trying to match query terms and URL terms. Our URL approach however, is based on the observation that entry page URLs tend to be higher in a server's document tree than other web pages, i.e. the number of slashes ('/') in an entry page URL tends to be relatively small.

We define 4 different types of URLs:

- root: a domain name, optionally followed by 'index.html' (e.g. `http://trec.nist.gov`)
- subroot: a domain name, followed by a single directory, optionally followed by 'index.html' name (e.g. `http://trec.nist.gov/pubs/`)
- path: a domain name, followed by an arbitrarily deep path, but not ending in a file name other than 'index.html' (e.g. `http://trec.nist.gov/pubs/trec9/papers/`)
- file: anything ending in a filename other than 'index.html' (e.g. `http://trec.nist.gov/pubs/trec9/t9_proceedings.html`)

We analysed WT10g and the relevant entry pages for half of the training documents to see how entry pages and other documents are distributed over these URL types. Table 3 shows the statistics.

URL type	#entry pages		#WT10g	
root	38	(71.7%)	11680	(0.6%)
subroot	7	(13.2%)	37959	(2.2%)
path	3	(5.7%)	83734	(4.9%)
file	3	(5.7%)	1557719	(92.1%)

Table 3. Distributions of entry pages and WT10g over URL types

From these statistics, we estimated prior probabilities of being an entry page on the basis of the URL type $P(entrypage|URLtype = t)$ for all URL types t. We then interpolated these priors with the normalised content only scores (cf. eq. 2) and tested this on the other 50 entry page search topics of the training data. This gave a major improvement on the content only results (see table 4).

run	MRR
content only	0.26
0.7 * content + 0.3 * URL prior	0.79

Table 4. URL prior results

3.3 Anchors

The fourth source of information is provided by the anchor texts of outlinks. These anchor texts are the underlined and highlighted texts of hyperlinks in web pages. We gathered all anchor texts of the outlinks, combined all texts pointing

to the same document to form a new textual representation of that document, and built a separate index on these texts. The texts include the so-called ALT-tags of images as well as the words occurring in the URL.

Note that the score provided by an anchor run is not a document prior. The anchor texts and the body texts ('content-only') provide two very different textual representations of the documents. The information retrieval language models are particularly well-suited for combining several document representations [3]. Our preferred way of combining two representations would be by the following, revised ranking formula.

$$\text{score} = \log(P_{\text{prior}}(D_k)) + \sum_{i=1}^{n} \log((1-\lambda-\mu)P(T_t|C) + \lambda P_{\text{content}}(T_i|D_k) + \mu P_{\text{anchor}}(T_i|D_k)) \quad (6)$$

So, the combination of the anchor run with the content run would be done on a 'query term by query term' basis, whereas the document prior (provided by inlinks or URLs) is added separately. Unfortunately, the current implementation of the retrieval system does not support combining document representations like this. Instead, the anchor runs were done separately from the content runs, their document scores being combined afterwards.

run	MRR
content only	0.26
anchor only	0.29
0.9 * content + 0.1 * anchor	0.36
0.63 * content + 0.07 * anchor + 0.3 * url	0.82

Table 5. MRRs of anchor runs on training data

Table 5 shows the MRRs on half of the training topics. Surprisingly, the anchor-only run slightly outperforms the content-only run. Apparently, search engines do not actually have to see entry pages to provide some useful retrieval functionality. Combining the two approaches leads to improved results. Anchors still seem to provide additional information if they are combined with the successful URL priors.

4 Results

4.1 Ad hoc task

For the Ad Hoc task, we submitted two runs, based on a Jelinek-Mercer smoothing scheme. We did some post hoc runs, based on the Dirichlet smoothing method and were impressed by their superior performance. All runs used only the title field of the topics.

run	description	m.a.p.
tnout10t1	JM smoothing ($\lambda = 0.8$) without doclen prior	0.1652
tnout10t2	JM smoothing ($\lambda = 0.5$) with doclen prior	0.1891
tnout10t3	Dirichlet smoothing ($\mu = 1000$) without doclen prior	0.2039

Table 6. Results of the Ad Hoc runs

Table 4.1 gives the results (mean average precision) for the title only adhoc runs (offical runs in bold font). We know from previous experiments that the Jelinek-Mercer smoothing scheme as such works better on full queries than on title queries. For title queries, the system has too much preference for shorter documents in comparison with the ideal line depicted by $P(rel|dlen)$ (see fig. 3). This can be "compensated" by assuming a prior probability which is linearly dependent on the document length. However, a strict linear relationship will favour long documents too much. An alternative is to use Dirichlet smoothing, such a system yields a $P(ret|dlen)$ curve which has the same shape and

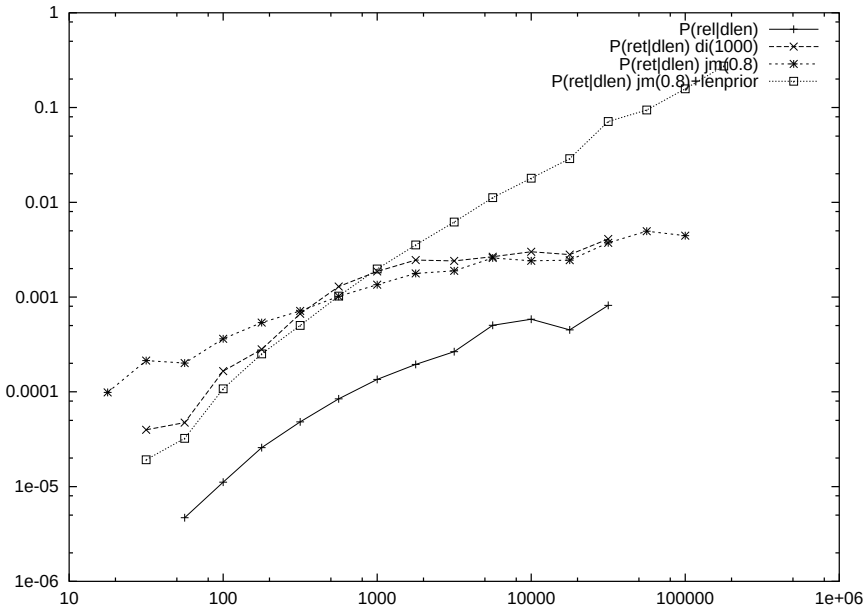

Fig. 3. Probability of relevance and probability of being retrieved as a function of document length

orientation as the ideal $P(rel|dlen)$ curve (fig. 3). The Dirichlet smoothing scheme is less sensitive to query length [8], and the preference for longer documents is inherent, since less smoothing is applied to longer documents.

Figure 3 illustrates the effect. The Dirichlet run follows the shape of the $P(rel|dlen)$ line more closely than the runs based on Jelinek-Mercer smoothing. The JM run based on a document length dependent prior indeed follows the ideal curve better in the lower ranges of document lengths, but overcompensates for the higher document length ranges.

4.2 Entry page task

For the entry page task, we submitted four runs: a content only run, a anchor only run, a content run with URL prior[5] and a run with content, anchors and URL priors. We did some additional runs to have results for all sensible combinations of content, anchors and priors, as well as an inlinkprior run and a Kleinberg run. The mean reciprocal ranks for all runs are shown in table 7 (official runs in bold face). Figure 4 shows the success rate at N for all runs[6] (on a logarithmic scale to emphasise high precision).

The first thing that should be noted from the results is that each combination of content and another source of information outperforms the content only run. The same holds for combinations with the anchor run. However, the improvement when adding URL information is for the anchor run less impressive than for the content run. This is probably due to the differences in the two runs. Although these runs have similar scores (MRR around 0.33), they have different characteristics. The anchor run is a high precision run, whereas the content run also has a reasonable recall. Therefore, it is hard to improve the anchor run since the entry pages that are retrieved are already in the top ranks and the other entry pages are simply not retrieved at all. Figure 4 shows the differences between the two runs: the anchor run has a slightly higher success rate for the lower ranks, but as the ranks get higher, the content run takes over.

As mentioned in section 2.1, our preferred way of combining sources of information when normalisation is not necessary, is to incorporate the additional information in the prior probability of a document. However, in the runs listed in table 7 we interpolated URL priors and inlink priors with the content scores. We did additional runs in which we used the priors exactly as in equation 1; Table 8 shows the results.

[5] We recomputed the priors on the whole set of training data.

[6] The number of entry pages retrieved within the top N documents returned

run	scores	description	MRR
tnout10epC	$contentscore$	Content only run	0.3375
tnout10epA	$anchorscore$	Anchor only run	0.3306
tnout10epCU	$0.7 \cdot contentscore + 0.3 \cdot urlprior$	Content run combined with URL priors	0.7716
tnout10epAU	$0.7 \cdot anchorscore + 0.3 \cdot urlpriors$	Anchor run combined with URL priors	0.4798
tnout10epCA	$0.9 \cdot contentscore + 0.1 \cdot anchorscore$	Interpolation of Content and Anchor runs	0.4500
tnout10epCAU	$0.63 \cdot contentscore + 0.07 \cdot anchorscore + 0.3 \cdot urlpriors$	Interpolation of Content and Anchor runs combined with URL priors	0.7745
tnout10epInlinks	$0.7 \cdot contentscore + 0.3 inlinkprior$	Content run combined with Inlink priors	0.4872
tnout10epKlein10	$Kleinberg's\ auth.score$ @ 10	Authority scores after Kleinberg algorithm on top 10 ranks from Content run	0.3548

Table 7. Entry Page results

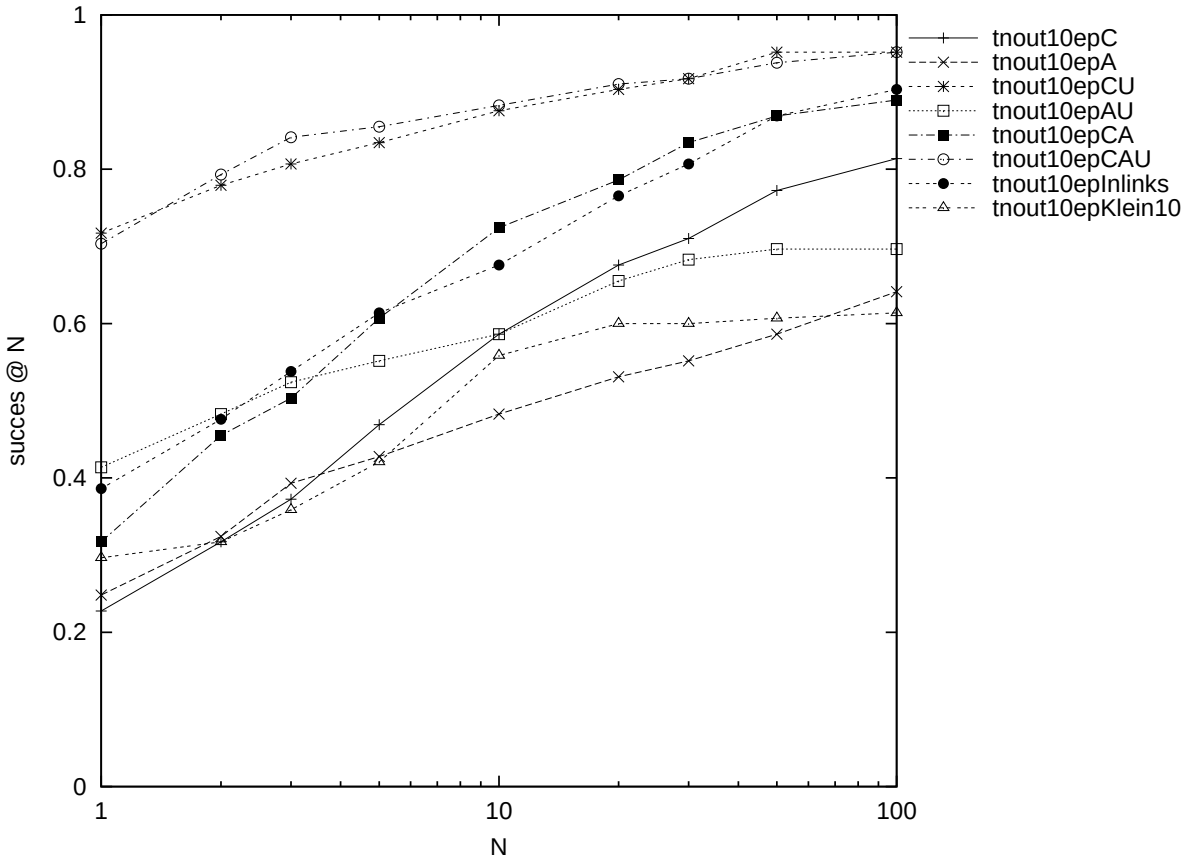

Fig. 4. Entry Page results: success @ N

run	MRR
content only	0.3375
content * URL prior	0.7743
content * inlink prior	0.4251
content * inlink prior * URL prior	0.5440
content * combiprior	0.7746

Table 8. Results with clean (non-interpolated) priors

Table 8 shows that also when we use priors in the clean way(cf. eq 1, they improve our results. Comparing these results to the ones in table 7, we see no difference in performance between the interpolated inlinks and the clean inlinks. The interpolated URL priors are slightly better than the clean ones.

When we take a combination of inlink and URL information as a prior, by simply multiplying the two priors, our results drop (see table 8). This indicates that the two sources of information are not independent. We therefore dropped the independence assumption and had another look at the training data. Just like with the estimation of the URL priors, we subdivided the collection into different categories and estimated prior probabilities of being an entry page given a certain category. As a basis for the categories, we took the 4 URL types defined in section 3.2, then we subdivided the root type into categories on the basis of the number of inlinks. Again we counted the number of entry pages from the training data and the number of documents from WT10g that fell into each category and estimated the prior probabilities from that. We took the categories from the URL types as a starting point and subdivided the root type into 4 subtypes on the basis of the number of inlinks. Table 9 shows the statistics for the different categories.

Document type	#entry pages	#WT10g
root with 1-10 inlinks	39 (36.1%)	8938 (0.5%)
root with 11-100 inlinks	25 (23.1%)	2905 (0.2%)
root with 101-1000 inlinks	11 (10.2%)	377 (0.0%)
root with 1000+ inlinks	4 (3.7%))	38 (0.0%)
subroot	15 (13.9%)	37959 (2.2%)
path	8 (7.4%)	83734 (4.9%)
file	6 (5.6%)	1557719 (92.1%)

Table 9. Distribution entry pages and WT10g over different document types

As can be seen in table 8, this proper combination of URL and inlink information (i.e. without the independence assumption) performs as good as or better than the two separate priors.

5 Conclusion

Post hoc runs show that the Dirichlet smoothing technique yields superior performance for title ad hoc queries on the web collection. This is probably due to the document length dependent smoothing constant, but further investigation is needed.

The Entry page finding task turns out to be very different from an ad hoc task. In previous web tracks link information didn't seem to help for general searches [7] [2]. This year, we found that in addition to content, other sources of information can be very useful for identifying entry pages. We described two different ways of combining different sources of information into our unigram language model: either as a proper prior or by interpolating results from different ranking models. We used both methods successfully when combining content information with other sources as diverse as inlinks, URLs and anchors. URL info gives the best prior info. Adding inlinks yields marginal improvement.

References

1. S. Brin and L. Page *The Anatomy of a Large-Scale Hypertextual Web Search Engine* Proceedings of WWW98, Brisbane, 1998
2. D. Hawking *Overview of the TREC-9 Web Track.* Proceedings of the 9th Text Retrieval Conference (TREC-9), National Institute for Standards and Technology, 2001
3. D. Hiemstra. *Using language models for information retrieval.* PhD thesis, Centre for Telematics and Information Technology, University of Twente, 2001. htpp://www.cs.utwente.nl/˜hiemstra/paper/thesis.pdf
4. D. Hiemstra and W. Kraaij. *Twenty-One at TREC7: Ad-hoc and Cross-Language track.* Proceedings of the 7th Text Retrieval Conference (TREC-7), National Institute for Standards and Technology, pages 227-238, 2001
5. J.M. Kleinberg. *Authorative Sources in a Hyperlinked Environment.* Proceedings of 9th ACM-SIAM Symposium on Discrete Algorithms, pages 668-377, 1998
6. W. Kraaij and M. Spitters and M. van der Heijden *Combining a mixture language model and Naive Bayes for multi-document summarisation* Working notes of the DUC2001 workshop (SIGIR2001), New Orleans, 2001
7. W. Kraaij and T. Westerveld. *TNO/UT at TREC-9: How different are Web documents?* Proceedings of the 9th Text Retrieval Conference (TREC-9), National Institute for Standards and Technology, 2001
8. C. Zhai and J. Lafferty *A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval* Proceedings of the 2001 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01), 2001