The Importance of Prior Probabilities for Entry Page Search

Wessel Kraaij
TNO TPD
PO BOX 155
2600 AD Delft
The Netherlands
kraaijw@acm.org

Thijs Westerveld
University of Twente
PO BOX 217
7500 AE Enschede
The Netherlands
westerve@cs.utwente.nl

Djoerd Hiemstra
University of Twente
PO BOX 217
7500 AE Enschede
The Netherlands
hiemstra@cs.utwente.nl

ABSTRACT

An important class of searches on the world-wide-web has the goal to find an entry page (homepage) of an organisation. Entry page search is quite different from Ad Hoc search. Indeed a plain Ad Hoc system performs disappointingly. We explored three non-content features of web pages: page length, number of incoming links and URL form. Especially the URL form proved to be a good predictor. Using URL form priors we found over 70% of all entry pages at rank 1, and up to 89% in the top 10. Non-content features can easily be embedded in a language model framework as a prior probability.

Categories and Subject Descriptors

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

General Terms

Experimentation

Keywords

Entry Page Search, Prior Probabilities, Links, URLs, Language Models, Parameter Estimation

1. INTRODUCTION

Entry page 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. In a general, Ad Hoc search task as defined for TREC [35, 36, 37], the goal is to find as many relevant documents as possible. The entry page (EP) task is concerned with finding the central page of an organisation, which functions as a portal for the information[1]. Since EP search has the goal to retrieve just one document, an IR system should probably be more optimised for high precision than for high recall. Search engine users typically prefer to find an EP in the first screen of results. Since queries are usually very short, finding an EP with a high initial precision is quite difficult. This paper explores ways to enhance IR systems designed for the Ad Hoc task by taking advantage of document features, which are usually ignored for the Ad Hoc task, since they were not effective. Our experiments for the EP task at TREC-2001 have shown that link structure, URLs and anchor texts are useful sources of information for locating entry pages[38].

Experiments of other groups confirmed the effectiveness of these features [6, 14, 7, 27, 32, 39]. In this paper we show how knowledge about the relationship between the non-content features of a webpage an its likelihood of being an EP can easily be incorporated in a retrieval model based on statistical language models. In Section 2 we discuss related work on tuning retrieval models to a specific task and other work on the use of information sources other than the document content. Section 3 discusses the basic language modelling approach and how priors can be used in this model. In Section 4 we discuss different sources of prior knowledge that can be used in an EP searching task. We describe our evaluation methodology in Section 5 and discuss our experiments and present results in Section 6. We conclude with a discussion of these results and a summary of our main conclusions.

2. CONTEXT AND RELATED WORK

In this section we will give argue that it is common practice to tailor IR methods to a specific search task, or even to a certain test collection, to optimize performance. It is important though, to use principled methods instead of ad hoc solutions.

2.1 Tuning IR systems

The use of proper term statistics is of the utmost importance for the quality of retrieval results on the Ad Hoc search task. All main IR models based on relevance ranking exploit these statistics and are based on at least two ingredients: the frequency of a term in a document and the distribution of a term in the document collection. IR models typically have evolved in close relationship with the availability of test collections. The older test collections were based on abstracts, so it was safe to assume that documents had about the same length. Since test collections are based on full text, this assumption does not hold anymore, and IR models have been refined to include a component which models the document length influence. Early attempts to combine the three main ingredients of an IR system were rather ad hoc and were not based on formal models. For example, Salton and Buckley [30] evaluated many combinations of term frequency statistics, document frequency statistics and document length normalisation without an explicit model for the relationship between these factors. The combine and test strategy was pursued further by Zobel and Moffat [40]. They tested 720 different term weighting strategies based on different similarity

[1] Entry pages for individual persons are usually referred to as homepages.

SIGIR'02, August 11-15, 2002, Tampere, Finland.

functions, length normalisation strategies, document term weighting etc. on 6 different query collections. They concluded that their exhaustive test strategy (4 weeks of computation time) had not resulted in any conclusive results that held across test collections. Instead of trying many ad hoc algorithms, Robertson and Walker [29] experimented with simple approximations of the 2-Poisson model[2]. The OKAPI formula is an approximation of a theoretically justified model, integrating the three components. The OKAPI formula includes tuning parameters that determine e.g. the influence of the term frequency or the influence of the document length normalisation. The parameters can be used to tune a system to the retrieval task and test collection at hand.

Based on the success of Robertson and Walker, Singhal et al. [33] showed that for the Ad Hoc search task, there is a correlation between the document length and the a-priori probability of relevance. The document length is a good example of information about a document that is not directly related to its contents, but might still be related to the possible relevance of the document: If we build a rather silly system that returns the longest document for any user query, then we actually might do better than returning a random document from the collection.

2.2 Non-content features of WEB pages

An important source of prior knowledge for web based information retrieval is the hyperlink structure. The idea behind most link structure analysis is that if there exists a link from document A to document B, this probably means that document A and B are on the same topic (*topic locality* assumption) and that the author of document A recommends document B (*recommendation* assumption). Two of the most widely known and used algorithms for link structure analysis are PageRank [4] and HITS [18]. Both methods base there calculations on the assumption that a page to which many documents link is highly recommended and therefore an authority. The authoritativeness of a document is even higher if the documents that link to it are some sort of authorities themselves. HITS and PageRank (or variants of those) are reported to be successful in numerous studies of retrieval quality [18, 3, 1, 25, 5], however when measuring relevance only, without taking quality or authoritativeness into account, HITS and PageRank based methods have not been able to improve retrieval effectiveness for Ad Hoc search tasks [15, 13, 20, 31, 8].

Davison [11] shows that the topic locality assumption holds: when pages are linked, they are likely to contain related content. This means that documents that are linked to relevant documents are potentially relevant themselves. However, exploitation of this idea by spreading retrieval scores over links has not proven successful in a general information retrieval task yet [20, 12]. On the other hand, the TREC-2001 evaluation showed that link structure analysis *does* improve content only results in an entry page search task [14, 38].

Another source of prior knowledge in web based retrieval is the URL. We are not aware of any studies in which URL knowledge was used for an Ad Hoc search. However, the TREC-2001 evaluation showed that the fact that entry pages tend to have shorter URLs than other documents can be successfully exploited by a ranking algorithm.[14, 38, 27, 32, 39].

2.3 Priors in LM based IR

Prior knowledge can be used in a standard way in the language modelling approach to information retrieval. The language modelling approach uses simple document-based unigram language models for document ranking [26]. Early TREC Ad Hoc experiments by Hiemstra and Kraaij [17] show that indeed the document length serves as a helpful prior for the Ad Hoc task, especially for short queries. Miller et al. [23] combined information in their document priors, including document length, source and average word length. In this paper we adapt the standard language modelling approach to the entry page search by including the prior probabilities in the estimation process. We show that proper use of prior probabilities results improvements of more than 100 % over the base-line language modelling system on the entry page task. Although we mainly address entry page searching here, we follow a generic approach. By following the suggested language modelling approach we can quickly adapt our standard retrieval system to other tasks or domains, e.g. automatic summarisation [19].

3. LANGUAGE MODELLING AND PRIOR PROBABILITIES

For all our experiments we used an information retrieval system based on a simple statistical language model [16]. We are interested in the probability that a document D is relevant given a query Q, which can be reformulated as follows using Bayes' rule:

$$P(D|Q) \;=\; \frac{P(D)\,P(Q|D)}{P(Q)}$$

The main motivation for applying Bayes' rule is that the probabilities on the right-hand side can be estimated more accurately than the probabilities on the left-hand side. For the purpose of document ranking, we can simply ignore $P(Q)$ since it does not depend on the documents. The probability $P(Q|D)$ can be represented by a document-based unigram language model. The standard language modelling approach would model the query Q as a sequence of n query terms $T_1, \cdots, T_n$, and use a linear combination of a unigram document model $P(T_i|D)$ and a collection model $P(T_i|C)$ for smoothing[3] as follows:

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

The probability measure $P(T_i|C)$ defines the probability of drawing a term at random from the collection, $P(T_i|D)$ defines the probability of drawing a term at random from the document and λ is the interpolation parameter, which is assumed to be some fixed constant. This leaves us with the definition of the prior probability of relevance of a document $P(D)$[4].

3.1 Estimating Priors

We distinguish two approaches to estimating the prior: direct estimation of the prior on some training data, and definition of the prior based on some general modelling assumptions.

As an example, let's have a look at the assumption that the prior probability of relevance is correlated with the document length on the Ad Hoc task. Figure 1 shows a plot of the probability of relevance against the document length for the Ad Hoc task of the TREC-2001 web track. To smooth the estimates, we divided the document

[2]However, the motivation to extend the original probabilistic model [28] with within-document term frequency and document length normalisation was probably based on empirical observations.

[3]This smoothing technique is also known as Jelinek-Mercer smoothing

[4]Relevance is not explicitly encoded in formula (1). Cf. [21] for a variant formulation of the model that includes relevance as a variable in the model. Both models lead to an equivalent term weighting function

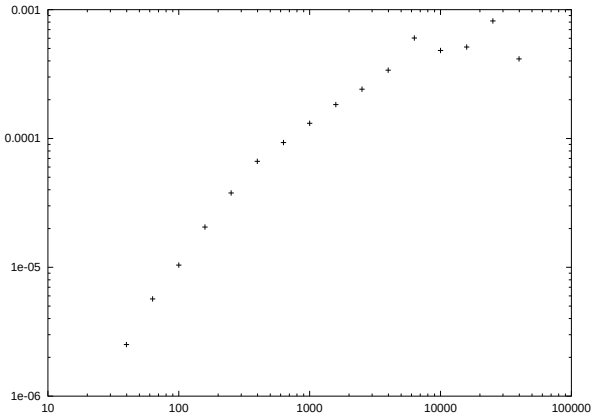

Figure 1: Prior probability of relevance given document length on the Ad Hoc task

length, which varies from documents containing only one or two words to documents containing over 10,000 words, into 16 bins on a logscale. Each point on the plot marks the probability of relevance of the documents in one of these bins. The 16 bins and the corresponding probabilities define a discrete probability measure $P(D)$ which takes one of 16 different values based on the bin in which D falls. As such, it can be used directly in Equation 1. Alternatively, looking at the plot, one could make the general modelling assumption that the a-priori probability of relevance is taken to be a linear function of the document length, so:

$$P_{doclen}(D) = P(R|D) = C \cdot doclen(D) \tag{2}$$

where R is a binary random variable 'relevance', $doclen(D)$ is the total number of words in document D, and C is a constant that can be ignored in the ranking formula. The intuition behind this assumption is that longer documents tend to cover more topics and thus have a higher probability of relevance. Indeed, the linear relationship between document length and probability of relevance is a reasonable model for different Ad Hoc test collections [20]. Including the prior especially can improve m.a.p. with up to 0.03 on an absolute scale, depending on the query length.

3.2 Combining Priors

What if we have different sources of information that might influence the a-priori probability of relevance, e.g. the length of a web page or its top level domain? The problem can be formalised as follows, we want to estimate:

$$P(R|D) = P(R|\bar{\mathbf{x}}) = P(R|x_1, x_2, \ldots, x_n) \tag{3}$$

In formula (3), R is a binary random variable 'relevance', $\bar{\mathbf{x}}$ is a shorthand for $(\bar{\mathbf{D}} = \bar{\mathbf{x}})$, i.e. $\bar{\mathbf{D}}$ is a random variable 'document' with as value a vector of document features $\bar{\mathbf{x}} = (x_1, x_2, ..., x_k)$. The problem is that we do not have a closed form solution for the prior $P(R|\bar{\mathbf{x}})$ and there is a very limited set of training data to estimate the prior for different feature values, so we have to find a smart way to estimate our prior. A possible approach is to apply the Naive Bayes assumption. Consider the Naive Bayes approach [24, 22] to classification:

$$C' = \text{argmax}_{C_k} P(C_k|\bar{\mathbf{x}}) = \frac{P(\bar{\mathbf{x}}|C_k)P(C_k)}{P(\bar{\mathbf{x}})} \tag{4}$$

Here, C' represents the class of an instance which is chosen among the set of mutually disjoint classes C_k. The decision rule chooses the C_k for which the numerator $P(\bar{\mathbf{x}}|C_k)P(C_k)$ is maximal given a certain vector of attributes $\bar{\mathbf{x}}$. Especially when there are many features, it is difficult to estimate the probabilities. A common strategy is to assume that the features are conditionally independent: $P(\bar{\mathbf{x}}|C_k) = \prod_i P(x_i|C_k)$, this is the well known *naive* Bayes assumption. This assumption results in a smaller number of probabilities that have to be estimated. The same estimation problem holds for the denominator, but this term can be ignored for classification tasks, since it is constant across the classes. In our case, there are only 2 different classes, namely relevant documents and irrelevant documents (or entry pages vs. non-entry pages). In addition, we're not only interested in the class with the highest probability, but we want to estimate the exact probability that a document is relevant given several attributes, since we want to use it for ranking. Therefore we cannot ignore the denominator since it is different for each attribute vector. So, Formula 5 should be used to estimate the priors.

$$P(C_k|\bar{\mathbf{x}}) = \frac{\prod_{j=1}^{n} P(x_j|C_k)P(C_k)}{\sum_k(\prod_{j=1}^{n} P(x_j|C_k)P(C_k))} \tag{5}$$

Indeed, the parameters in this formula are much easier to estimate. As an example, suppose we have three features, with five values each. Direct estimation of the probability of relevance would require the estimation of 125 conditional probabilities. In the Naive Bayes approach, only 30 relative frequencies would have to be computed. An alternative approach could be to reduce the number of parameters in our model, e.g. reducing the number of classes per feature to three, yielding $3^3 = 27$ parameters in order to reduce variance.

For our experiment with combined priors, we combined two features (URL form and #inlinks) each with 4 different feature values. As a simple baseline, we assumed conditional independence of the features given relevance, and approximated Formula 5 by replacing the denominator by $\prod_j^n P(x_j)$, obviating the need to build a training set for non Entry Pages. The approximated formula can be further reduced to:

$$P(C_k|\bar{\mathbf{x}}) = \frac{\prod_{j=1}^{n} P(C_k|x_j)P(C_k)}{\prod_{j=1}^{n} P(C_k)} = K \prod_{j=1}^{n} P(C_k|x_j) \tag{6}$$

which is equivalent to the product of the individual priors and a constant.

We contrasted this approximated Naive Bayes run, with a run where we directly estimated (3), but based on a reduced number of classes. We defined seven disjoint classes in such a way that there were at least 4 positive training examples in each class. This was done by merging several attribute vectors into one class. Section 6.2 describes the procedure in more detail.

3.3 Combining Content Models

Just as we might have different sources of information to estimate prior probabilities, we might as well have different sources of information to estimate content models. As an example, consider the possibility to estimate a document language model on the anchor texts of all hyperlinks pointing to the document. Note that the score provided by a query on the anchor texts is not a document prior. The anchor texts and the body texts ('content-only') provide two very different textual representations of the documents. The IR language models can accommodate several document representations in a straightforward manner, by defining a mixture model. Our preferred way of combining two representations would be by

the following, revised ranking formula.

$$P(D|T_1,\cdots,T_n) \propto P(D)\prod_{i=1}^{n}\big((1-\lambda-\mu)P(T_t|C)+ \lambda P_{\text{content}}(T_i|D)+\mu P_{\text{anchor}}(T_i|D)\big) \quad (7)$$

So, the combination of the anchor run with the content run would be done on a 'query term by query term' basis, whereas the combination with the document prior is done separately. Unfortunately, the current implementation of our retrieval system does not support combining document representations like this. Instead, the anchor experiments described in the next sections were done separately from the content runs, their document scores being combined afterwards. In a document summarisation setting we showed the potential of similar mixture models for ranking [19].

4. PRIOR PROBABILITIES IN ENTRY PAGE SEARCH

In an entry page search task, a user is searching for the main entry page of a specific organisation or group within an organisation. This is the main entry point for information about that group on the WWW. From here, one can usually navigate to the other web pages of the same group. For example the main entry page of SIGIR 2002 is `http://www.sigir2002.org`.

A query in an entry page search task consists of nothing more than the name of the organisation or group whose entry page we're after. In general there are only a few correct answers: the URL of the entry page of the requested organisation and perhaps some alternate URLs pointing to the same page and a few mirror sites.

Entry page searching is similar to known item searching. In both types of searches, a user is looking for one specific piece of information. However, in known item search the user has already seen the information needed, whereas in entry page search this is not necessarily the case.

Since entry page search is a different task than Ad Hoc search, different prior probabilities might be needed. We investigated which sources of information are useful priors for entry page searching and whether priors are needed at all for such a task.

We looked at three different sources of information: i) The length of a document ; ii) The number of links pointing to a document (inlinks) ; iii) The depth of a document's URL.

Features we didn't investigate, but which might be useful for estimating whether a page is an entry page include: the number of occurrences of certain cue words (e.g. 'Welcome', 'home'), the number of links in a document (outlinks) and the number of times a page is visited.

4.1 Document Length

Section 3.1 showed that document length priors are useful in an Ad Hoc search task. Here we investigate whether the length of a document is also a useful indicator of the probability that a document is an entry page. Figure 2 shows a plot of the probability of relevance versus page length, calculated on the training data provided for the entry page search task of TREC-2001's web track.

Indeed document length can predict the relevance of a page, since the distribution is not uniform. Pages with a medium length (60-1000 words) have a higher probability, with a maximum around 100-200 words. However, the differences are much less marked as for Ad Hoc search. Although it is clear that the dependence of the probability of relevance on document length is quite different from the Ad Hoc task, we ran an experiment where we used our best performing IR model for the Ad Hoc task (language model plus document length prior as defined in (2)) on the EP task, to show that models that perform well on Ad Hoc do not necessarily perform well on the Entry Page task.

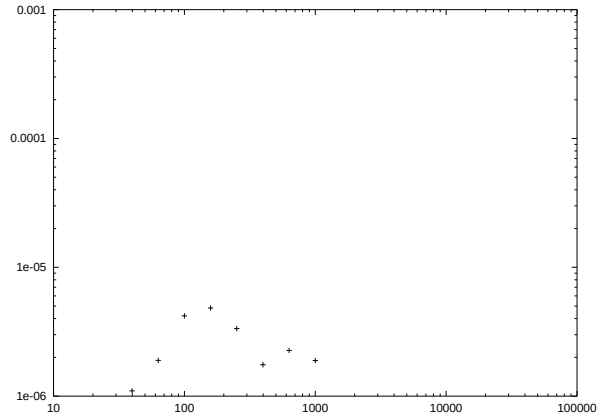

Figure 2: Prior probability of relevance given document length on the Entry Page task ($P(entry\ page|doclen)$)

4.2 Number of InLinks

In a hypertext collection like the WWW, documents are connected through hyperlinks. A hyperlink is a connection between a source and a target document. Our inlinks based prior is based on the observation that entry pages tend to have a higher number of inlinks than other documents (i.e. they are referenced more often). The plot of the probability of being an entry page against the number of inlinks in Figure 3 shows this correlation for the training data for the entry page task of TREC-2001's web track. Since the

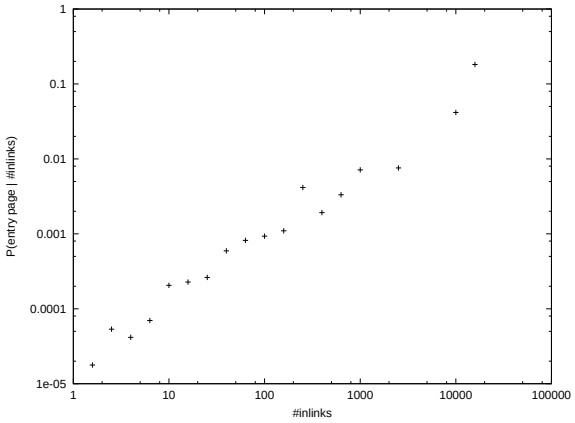

Figure 3: Prior probability of relevance given number of inlinks on the Entry Page task ($P(entry\ page|inlinkCount)$)

plot suggests a linear relationship, we define the inlink prior as:

$$P_{inlink}(D) = P(R|D) = C \cdot inlinkCount(D) \quad (8)$$

where R is a binary random variable 'relevance', $inlinkcount(D)$ is the count of the number of inlinks in document D and C is a constant.

Note that, although we didn't differentiate between links from the same host and links from other hosts, doing so might be interesting. These different types of links have different functions. Links from other hosts often imply a form of recommendation, whereas

links from the same host mainly serve navigational purposes. However, both types of links probably point more often to entry pages than to other pages. Recommendation links usually point to the main entry point of a site and sites are often organised in such a way that each page links to the entry page.

4.3 URL Depth

A third source of prior information is the URL of a document. A URL defines the unique location of a document on the WWW. It consists of the name of the server, a directory path and a filename. A first observation is that documents that are right at the top level of a specific server (i.e. the documents of which the URL is no more than a server name) are often entry pages. Also, as we descend deeper into the directory tree, the relative amount of entry pages decreases. Thus the probability of being an entry page seems to be inversely related to the depth of the path in the corresponding URL.

We define four types of URLs:

root: a domain name, optionally followed by 'index.html' (e.g. `http://www.sigir.org`)

subroot: a domain name, followed by a single directory, optionally followed by 'index.html' name (e.g. `http://www.sigir.org/sigirlist/`)

path: a domain name, followed by an arbitrarily deep path, but not ending in a file name other than 'index.html' (e.g. `http://www.sigir.org/sigirlist/issues/`)

file: anything ending in a filename other than 'index.html' (e.g. `http://www.sigir.org/resources.html`)

Analysis of both WT10g, the collection used in TREC-2001's web track, and the correct answers to the training topics for the track's entry page task (i.e. the correct entry pages), shows that entry pages have a very different distribution over the four URL-types (cf. Table 1). As suspected, entry pages seem to be mainly of type

URL type	Entry pages		WT10g	
root	79	(73.1%)	12258	(0.7%)
subroot	15	(13.9%)	37959	(2.2%)
path	8	(7.4%)	83734	(4.9%)
file	6	(5.6%)	1557719	(92.1%)

Table 1: Distributions of entry pages and WT10g over URL-types

root while most documents in the collection are of type file. For inlinks, we were able to define a model for the prior probability of relevance given the number of inlinks, which is a good approximation of the real data. For the URL prior it is less straightforward to define such an analytical model. Therefore, we estimated the URL based prior directly on the training collection:

$$P_{URL}(D) = P(EP|URLtype(D) = t_i) = \frac{c(EP, t_i)}{c(t_i)}, \quad (9)$$

where $URLtype(D)$ is the type of the URL of document D, $c(EP, t_i)$ is the number of documents that is both an entry page and of type t_i and $c(t_i)$ is the number of documents of type t_i. The resulting probabilities for the different URL-types are listed in Table 2. Note that the prior probabilities differ several orders of magnitude; a root page has an almost 2000 times larger probability of being an entry page than any other page.

$P(Entry\ page \mid root)$	$6.44 \cdot 10^{-03}$
$P(Entry\ page \mid subroot)$	$3.95 \cdot 10^{-04}$
$P(Entry\ page \mid path)$	$9.55 \cdot 10^{-05}$
$P(Entry\ page \mid file)$	$3.85 \cdot 10^{-06}$

Table 2: Prior probabilities for different URL-types, estimated on the training data

5. EVALUATION

We compared the different priors discussed in the previous section using collections and relevance judgements from the entry page finding task of TREC-2001's web track. The Text Retrieval Conferences (TREC) [35, 36, 37], are a series of workshops aimed at large-scale testing of retrieval technology. The entry page finding task was designed to evaluate methods for locating entry pages. In this task a search topic consists of a single phrase stating the name of a specific group or organisation, for some examples cf. Table 3. A

Worldnet Africa
Hunt Memorial Library
Haas Business School
University of Wisconsin Lidar Group

Table 3: Example topics for entry page search task

system should retrieve the entry page for the group or organisation described in the topic. E.g. when a topic is *University of Wisconsin Lidar Group*, the entry page for this specific group should be retrieved. Retrieving the homepage for the whole University or for a subgroup of the Lidar group would be counted incorrect. Topics for the entry page finding task were created by randomly selecting a document from the document collection and following links from there until they got to a document they considered to be the homepage of a site that contained the document they started with. This way, 145 topics were constructed for this task. In addition a set of 100 topics was provided for training purposes. We used the latter set for making the plots in sections 4.1 and 4.2 from which we inferred the document length and inlink priors. The same set was used for estimating the URL priors as described in section 4.3. The experiments reported in the next section are all conducted using the set of 145 test topics.

The collection to be searched for this task is the WT10g collection [2], a 10 gigabytes subset of the VLC2 collection which in turn is a subset of a 1997 crawl of the WWW done by the Internet Archive [5]. WT10g is designed to have a relatively high density of inter-server hyperlinks.

For each combination of document prior and language model, we ranked our document collection and returned the top 100 results. Runs were evaluated by the mean reciprocal rank (MRR) measure. For each topic the reciprocal value ($1/n$) of the rank (n) of the highest ranked correct entry page is computed, subsequently these values are averaged. If no correct entry page was found for a specific topic the reciprocal of the rank for this topic is defined 0.

6. EXPERIMENTS

For our experiments we used the IR model described in section 3. The prior probability in our model, $P(D)$ was instantiated by one of the priors defined in section 4 or left out (i.e. uniform priors). The language models ($P(Q|D)$) were either estimated on

[5] `http://www.archive.org`

the document collection (WT10g) or on the collection of anchor based pseudo documents. This anchor-based collection was generated from the original WT10g data by gathering all anchor texts from links pointing to the same document in one pseudo document. This pseudo document was then used as a description of the original document.[6]. Again, we didn't differentiate between links from the same server and links from other servers. Documents were preprocessed as follows

1. remove HTML comment and script document fragments
2. remove HTML markup tags
3. map HTML entities to ISO-Latin1 characters
4. stop terms
5. stem terms using the Porter algorithm

Table 4 shows the mean reciprocal rank (MRR) for runs on both the web page and the anchor text collection in combination with different priors. Both inlinks and URL form help to increase search effectiveness, especially the URL prior is highly effective. Further discussion is deferred until Section 7.

Ranking method	Content ($\lambda = 0.9$)	Anchors($\lambda = 0.9$)
$P(Q\|D)$	0.3375	0.4188
$P(Q\|D)P_{doclen}(D)$	0.2634	0.5600
$P(Q\|D)P_{URL}(D)$	**0.7705**	0.6301
$P(Q\|D)P_{inlink}(D)$	0.4974	0.5365

Table 4: Results for different priors

6.1 Combining content models

We combined page content and anchor runs by a linear interpolation of the result files: $RSV' = \alpha RSV_{page} + (1-\alpha)RSV_{anchor}$ Table 5 shows that although the combination strategy is not optimal

α	MRR
0.5	0.3978
0.7	0.4703
0.8	**0.4920**
0.9	0.4797

Table 5: Combining web page text and anchor text

(cf. Section 3.3), combining both content representation is useful. Best result is achieved for $\alpha = 0.8$: MRR=0.4920 (anchor text only run: MRR = 0.4188). We also investigated the effect of priors on these combined runs (Table 6).

Ranking method	Content+Anchors ($\alpha = 0.8$)
$P(Q\|D)$	0.4920
$P(Q\|D)P_{URL}(D)$	**0.7748**
$P(Q\|D)P_{inlink}(D)$	0.5963

Table 6: Results for different priors(content+anchor)

6.2 Combining Priors

In section 4 we saw that both the number of inlinks and the URL can give important information about whether a page is an entry page or not. The results in Table 4 confirm this. A combination of these two sources of information might also be useful. In section 3.2 we discussed different approaches for combining priors. Table 8 shows the results for both combination approaches. All runs are based on the web page corpus and do not regard anchor text. We have argued that the main problem of combining features is that it is difficult to estimate $P(EP|\bar{\mathbf{x}})$ directly, since the training collection is small. A simple approach is to multiply the individual priors, which is an approximation of a model based on the conditional independence assumption. This baseline combination run yielded an MMR value of 0.7504, which is lower than a run based on just the URL prior.

We conjecture that this disappointing result is due to the fact that there is conditional dependence between the features or that the approximation of the denominator is too crude. We investigated a variant run, without indepence assumptions, where we directly estimated $P(EP|\bar{\mathbf{x}})$, but based on a model with a reduced number of parameters.

For this approach we could not work with Formula (8), instead we disretized the #inlinks feature space in four bins on a logscale. Subsequently, we defined seven disjoint document classes. Table 7 shows their statistics. Since there are just a few entry pages in the training set of category non-root, we only divided the root class in four separate bins based on the number of inlinks. From these

Document type t_i	#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%)

Table 7: Distribution entry pages and WT10g over different document types

statistics we computed a combined inlink-URL prior, using:

$$P_{inlink-URL}(EP|dt(D) = t_i) = \frac{c(EP, t_i)}{c(t_i)}, \qquad (10)$$

where $dt(D)$ is the document type of D and $c(.)$ is as defined in equation 9.

Table 8 summarises results for the combination experiments. In addition to the already tabulated single prior results, it shows the result for a run where the inlink prior was estimated on the training data instead of being modeled by Formula (2). The MRR of this run is lower than the run based on the analytical inlink prior, which probably means that the number of bins is too small. The combination runs show a marked difference: the run based on a naive multiplication of priors scores scores worse then the run based on the URL prior. However, the combination run which directly estimates $P(EP|\bar{\mathbf{x}})$, based on seven disjoint classes does improves upon the individual prior runs, so combination of priors performs better than a single prior.

7. DISCUSSION

The entry page task is significantly different from the Ad Hoc task. The main difference is that just one page is relevant, so recall enhancement techniques do not play a role, on the contrary, the goal is to boost initial precision. We experimented with content-only runs, based on either the web page content or its related anchor

[6]The experiments in this paper are based on the CSIRO anchor collection [6, 7], whereas the experiments reported in [38] are based on an anchor collection we built ourselves

Ranking method	Content	description
$P(Q\|D)P_{URL}(D)$	0.7705	
$P(Q\|D)P_{inlink}(D)$	0.4974	analytical prior
$P(Q\|D)P_{inlink}(D)$	0.4752	4 bins
$P(Q\|D)P_{URL+inlink}(D)$	0.7504	baseline: product of priors (analytical inlink prior)
$P(Q\|D)P_{URL+inlink}(D)$	**0.7749**	direct estimation (7 disjoint classes)

Table 8: Combining priors

text, and with runs which also exploited other page characteristics, like number of links and URL form.

Interestingly, a run based on just the anchor text outperformed the basic content run by far, indicating that anchor text linking to entry pages often contains the name of the organisation's entry page. Combining both textual representations by simple interpolation proved effective: MRR = 0.4920 (+17%).

For content runs, an (exact) match of the query is not sufficiently discriminating to find entry pages, since a lot of web pages will have exact matches. Therefore, other features have to be used. We tested three different features: document length, number of inlinks and the URL form. A linear document length prior did significantly decrease MRR for the content run, which we had expected, since such a prior did not fit the training data at all. Nevertheless it is interesting to see that choosing a wrong prior can harm results. The linear document length prior seems to help the anchor run though, but that is an illusion. The length of an anchor text document consists of a concatenation of the anchor texts of the links pointing to a particular web page. Thus, the length of an anchor text 'document' is linearly related to the number of inlinks. Indeed, the anchor runs based on an inlinks prior or a document length prior have rather equivalent performance gains. There seems to be an additional gain of the document length prior though; apparently more verbose anchor texts are correlated with entry pages. The best results are achieved by the runs with a URL form based prior (the content run with a URL prior found over 70% of the entrypages at rank 1, and 87% in the top 10[7]). Surprisingly, the better result was achieved by the URL prior run based on the web page text. This can be explained by the fact that this run has a larger recall than the anchor text run. The URL prior is a powerful precision enhancing device which combines well with the web page content run, which has good recall. Figure 4 illustrates this: while the anchor run has a high initial success rate (30.3% of entry pages were found at rank 1; 22.8% for content), the content run in the end finds more homepages(81.4% found in top 100; 71.0% for anchors).

We also investigated whether we could combine URL and inlink features. A simple multiplication of priors (assuming conditional and statistical independence) deteriorated results. Another strategy, bases on a reduced number of classes, but without any independence assumptions yielded a small gain w.r.t. the URL prior run. We think that we could improve upon these results. With more training data we might be able to estimate the joint conditional probabilities of the two features in a more accurate way. The fact that the training data based inlink prior performed worse than the analytical prior (8) indicated that the number of bins is probably too small. The small number of bins might also have contributed to the disappointing results of the combination runs that use the inlink binning approach. We tested the latter hypothesis, by a further refinement of the URL root category in 9 bins (based on #inlinks classes) instead of 4 bins. This resulted in a further improvement (MRR= 0.7832; 72% at 1; 89% in top 10), confirming our hypothesis. We also reran the inlink run based on 9 bins instead of 4 bins: MRR 0.5064, which is even better than the analytical inlink prior. It is thus safe to conclude that the number of classes is critical for the performance of the training data based priors.

[7]The corresponding content only run only returned 22% of the entry pages at rank 1, and 59% in the top 10.

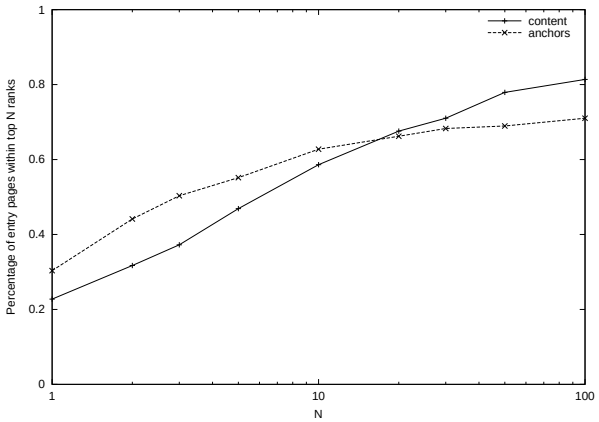

Figure 4: Succes at N for content and anchor runs

8. CONCLUSIONS

There are several web page features which can help to improve the effectiveness of an entry page retrieval system, i.e. the number of inlinks and the form of the URL. We have shown that an IR system based on unigram language models can accommodate these types of information in an almost trivial way, by estimating the posterior probability of a page given its features and using this probability as a prior in the model. The URL form feature proved to have the strongest predictive power, yielding over 70% of entry pages at rank 1, and up to 89% in the top 10. Initial experiments with combining both features showed a further improvement. The performance of the combined prior model is highly sensitive to the number of model parameters since the training collection was very small (100 topics).

Acknowledgements

This research was funded by the DRUID project of the Telematics Institute, The Netherlands. We would like to thank David Hawking and Nick Craswell of CSIRO, for making their collection of anchor text available.

9. REFERENCES

[1] B. Amento, L. Terveen, and W. Hill. Does "authority" mean quality? predicting expert quality ratings of web documents. In Croft et al. [9], pages 296–303.

[2] P. Bailey, N. Craswell, and D. Hawking. Engineering a multi-purpose test collection for web retrieval experiments. *Information Processing and Management*, In press.

[3] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In Croft et al. [10], pages 104–111.

[4] 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.

[5] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In *Proceedings of the 17th International Conference on Machine Learning*, 2000.

[6] N. Craswell, D. Hawking, and S. Robertson. Effective site finding using link anchor information. In Croft et al. [9], pages 250–7.

[7] N. Craswell, D. Hawking, R. Wilkinson, and M. Wu. TREC10 web and interactive tracks at CSIRO. In Voorhees and Harman [37], pages 261–268.

[8] F. Crivellari and M. Melucci. Web document retrieval using passage retrieval, connectivity information, and automatic link weighting - TREC-9 report. In Voorhees and Harman [36], pages 611–620.

[9] W. B. Croft, D. J. Harper, D. H. Kraft, and J. Zobel, editors. *Proceedings of the 23rd Annual International Conference on Research and Development in Information Retrieval*, 2000.

[10] W. B. Croft, A. Moffat, C. van Rijsbergen, R. Wilkinson, and J. Zobel, editors. *Proceedings of the 21st Annual International Conference on Research and Development in Information Retrieval*, 1998.

[11] B. D. Davison. Topical locality in the web. In Croft et al. [9], pages 272–9.

[12] C. Gurrin and A. F. Smeaton. Dublin city university experiments in connectivity analysis for TREC-9. In Voorhees and Harman [36], pages 179–188.

[13] D. Hawking. Overview of the TREC-9 web track. In Voorhees and Harman [36], pages 87–102.

[14] D. Hawking and N. Craswell. Overview of the TREC-2001 web track. In Voorhees and Harman [37], pages 25–31.

[15] D. Hawking, E. voorhees, N. Craswell, and P. Bailey. Overview of the TREC-8 web track. In Voorhees and Harman [35], pages 131–148.

[16] D. Hiemstra. *Using language models for information retrieval*. PhD thesis, Centre for Telematics and Information Technology, University of Twente, 2001.

[17] D. Hiemstra and W. Kraaij. Twenty-One at TREC-7: Ad-hoc and cross-language track. In Voorhees and Harman [34], pages 227–238.

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

[19] W. Kraaij, M. Spitters, and M. van der Heijden. Combining a mixture language model and naive bayes for multi-document summarisation. In *Working notes of the DUC2001 workshop (SIGIR2001), New Orleans*, 2001.

[20] W. Kraaij and T. Westerveld. TNO/UT at TREC-9: How different are web documents? In Voorhees and Harman [36], pages 665–671.

[21] J. Lafferty and C. Zhai. Probabilistic IR models based on document and query generation. In J. Callan, B. Croft, and J. Lafferty, editors, *Proceedings of the workshop on Language Modeling and Informati on Retrieval*, 2001.

[22] D. D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398 in Lecture Notes in Computer Science, pages 4–15, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[23] D. R. H. Miller, T. Leek, and R. M. Schwartz. BBN at TREC-7: using hidden markov models for information retrieval. In Voorhees and Harman [34], pages 133–142.

[24] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[25] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In Croft et al. [9], pages 258–266.

[26] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In Croft et al. [10], pages 275–281.

[27] D.-Y. Ra, E.-K. Park, and J.-S. Jang. Yonsei/etri at TREC-10: Utilizing web document properties. In Voorhees and Harman [37], pages 643–650.

[28] S. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

[29] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th ACM Conference on Research and Development in Information Retrieval (SIGIR'94)*, pages 232–241, 1994.

[30] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[31] J. Savoy and Y. Rasolofo. Report on the TREC-9 experiment: Link-based retrieval and distributed collections. In Voorhees and Harman [36], pages 579–588.

[32] J. Savoy and Y. Rasolofo. Report on the TREC-10 experiment: Distributed collections and entrypage searching. In Voorhees and Harman [37], pages 578–590.

[33] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 19th ACM Conference on Research and Development in Information Retrieval (SIGIR'96)*, pages 21–29, 1996.

[34] E. M. Voorhees and D. K. Harman, editors. *The Seventh Text Retrieval Conference (TREC7)*, volume 7. National Institute of Standards and Technology, NIST, 1999.

[35] E. M. Voorhees and D. K. Harman, editors. *The Eighth Text Retrieval Conference (TREC8)*, volume 8. National Institute of Standards and Technology, NIST, 2000.

[36] E. M. Voorhees and D. K. Harman, editors. *The Ninth Text Retrieval Conference (TREC9)*, volume 9. National Institute of Standards and Technology, NIST, 2001.

[37] E. M. Voorhees and D. K. Harman, editors. *The Tenth Text Retrieval Conference (TREC-2001)*, volume 10. National Institute of Standards and Technology, NIST, 2002.

[38] T. Westerveld, W. Kraaij, and D. Hiemstra. Retrieving web pages using content, links, URL's and anchors. In Voorhees and Harman [37], pages 52–61.

[39] W. Xi and E. A. Fox. Machine learning approaches for homepage finding tasks at TREC-10. In Voorhees and Harman [37], pages 633–642.

[40] J. Zobel and A. Moffat. Exploring the similarity space. *SIGIR FORUM*, 32(1):18–34, 1998.