The Impact of Crawl Policy on Web Search Effectiveness

Dennis Fetterly
Microsoft Research
Mountain View, CA USA
fetterly@microsoft.com

Nick Craswell
Microsoft Research
Cambridge, UK
nickcr@microsoft.com

Vishwa Vinay
Microsoft Research
Cambridge, UK
vvinay@microsoft.com

ABSTRACT

Crawl selection policy has a direct influence on Web search effectiveness, because a useful page that is not selected for crawling will also be absent from search results. Yet there has been little or no work on measuring this effect. We introduce an evaluation framework, based on relevance judgments pooled from multiple search engines, measuring the maximum potential NDCG that is achievable using a particular crawl. This allows us to evaluate different crawl policies and investigate important scenarios like selection stability over multiple iterations. We conduct two sets of crawling experiments at the scale of 1 billion and 100 million pages respectively. These show that crawl selection based on PageRank, indegree and trans-domain indegree all allow better retrieval effectiveness than a simple breadth-first crawl of the same size. PageRank is the most reliable and effective method. Trans-domain indegree can outperform PageRank, but over multiple crawl iterations it is less effective and more unstable. Finally we experiment with combinations of crawl selection methods and per-domain page limits, which yield crawls with greater potential NDCG than PageRank.

Categories and Subject Descriptors

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

General Terms

Measurement, Experimentation

1. INTRODUCTION

A useful Web search result will only be seen by users if it is crawled by the search engine, indexed correctly, found in the index when matched with a query and ranked highly in the search result listing. It only takes one failure in this chain of events for the useful (relevant) result to be lost. If such failures happen often, users will perceive a drop in the quality of search results. Therefore, to optimize user satisfaction, it is important to avoid failure at every stage.

Success at the crawling stage depends on the size of the crawl and the crawl selection policy. For example, the policy of preferring pages with highest PageRank [7] and a size limit of N leads to selecting a set of N high-PageRank pages. When searches are carried out, the quality of search results will sometimes be reduced because pages that would have been relevant and retrievable were not selected for crawling. One way to reduce such failures is to increase the size N of the crawl. Another approach is to improve the selection policy.

Although well-known methods exist for evaluating search relevance, such as NDCG [13], we are not aware of any published experiments that compare the relevance achievable by different crawl policies. Acting as a barrier to experimentation are the large communication and computational costs of conducting multiple crawls, creating multiple indices and processing queries. Our framework ameliorates this via a crawl sandbox and an evaluation metric that only requires the set of selected URLs. The sandbox is simply a cache, to avoid crawling URLs more than once if selected by multiple policies or iterations. The metric, maxNDCG, is the best potential NDCG that could be achieved based on the presence or absence of relevant pages in a crawl. maxNDCG is proportional to NDCG but may be calculated without indexing and retrieval. It may even be calculated for a selected set of pages without attempting to crawl them, estimating the NDCG that would be achievable by a perfect ranker if all selected pages were successfully crawled.

These efficiency techniques allow us to run a large number of experiments comparing crawl policies. We focus on policies for selecting a new crawl based on the link graph of a previous crawl [6]. This is a common scenario, allowing an engine to shift its selection towards pages that are preferred according to some link-based policy (such as PageRank) but not yet included in the crawl.

In Section 2, we discuss different aspects involved in the evaluation of crawling methods. We provide motivation for our experimental setup, and where appropriate, we provide references to related work. We then present experiments in Section 3 and conclusions in Section 4.

2. CRAWLING AND EVALUATION

Search engines are the primary discovery mechanism for pages on the Web, and the Web has an effectively infinite set of pages that might be indexed. A search engine must be selective in which pages it indexes, to make the best use of its finite indexing resources. Search engines use crawlers to download copies of web pages, over which indices are built. Starting from a set of seeds, a crawler can download pages and extract links. It records which URLs have been seen and adds any unseen URLs to a structure called the frontier.

After an initial crawl, there is the problem of maintaining the corpus, since pages are continually updated [11], created [16] and deleted [4]. One option would be to start crawling again from seeds, although completely throwing away the old pages and link graph

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

information may not be the most efficient (or effective [5]) way to proceed.

The alternative is to perform an incremental recrawl [6]. Recrawling has two aspects that can be decoupled: (a) Refreshing URLs to discover changes/deletions and (b) Adjusting the selection of URLs. On the dynamic web, requiring an index update due to (a) is self-evident. Successive iterations of an incremental crawl potentially provide different estimates of a page's importance, using updated information could potentially lead to a previously important page being down-weighted and therefore dropped in favor of another page. Such an update could lead to changes in the crawled corpus. This paper focuses on evaluating crawl selection policies in the context of incremental recrawling, including experiments with multiple recrawling iterations.

We consider three popularity-based crawl ordering policies, selecting pages with: (1) the greatest number of incoming links (indegree), (2) the greatest number of incoming links from other Web domains (trans-domain indegree)and (3) the highest PageRank. It is assumed that an initial baseline crawl from seeds has been done (a breadth-first crawl in our case), and we compare corpora generated by these three methods against the baseline breadth-first. We also consider multiple iterations of crawls generated by these three policies and trace the behavior of each method across generations. All these comparisons are from the specific viewpoint of using the corpora generated to form the basis of a search engine's index.

2.1 Evaluating a crawl

There are multiple perspectives on what makes a good crawl or good crawling software. One is efficiency; for example, the IRLbot [14] authors consider politeness, queuing, data structures and budgeting issues to crawl 6 billion pages on a single machine. We do not consider these efficiency issues, although our experiments consider the tradeoff between crawl size (efficiency) and retrieval effectiveness.

In the Web search literature, a good crawl is one that contains the 'important' pages. Specifically, the general consensus has been that link-based connectivity metrics like PageRank (and variants) allow us to measure the quality of a crawl. Given a goodness score for each page, the corpus generated has a total goodness that has been referred to as its 'RankMass' [8, 12, 3].

RankMass would normally be calculated based on some final 'ground truth' PageRank, using a large link graph. If the crawl selection policy also has the full graph information, it can achieve optimal RankMass at any given crawl size N by selecting the top-N pages according to PageRank. Under RankMass evaluation, typical experiments use a crawl selection policy that has a much smaller graph and must select pages for crawling based on incomplete information [7, 5].

Next we consider evaluation based on retrieval NDCG. We prefer this form of evaluation to PageRank RankMass because it is specific to our goal of effective Web retrieval.

2.2 Comparing crawl policies

Crawling is a topic that has been well studied in the past; as a result there is no shortage of proposed crawl policies. Each crawling method is designed for a particular purpose, and can be evaluated in that narrow context. Conducting a head-to-head comparison between alternative crawling methods is complicated by the dynamic nature of the Web. An experiment would ideally involve separate crawls to be run and evaluated. But pages and sites undergo continuous change, as a result, any such changes that transpire between two separate crawls might down-weight the reliability of any inference we reach from the comparison.

Apart from the changes to the pages, there are other factors that make comparing multiple crawls difficult. Real-time events are sometimes non-reproducible; for example, time-out events on remote hosts and other network related issues. Two crawls that are started from the exact same set of seeds and following the same link exploration policy could still end up with quite different corpora.

In order to experiment with a variety of corpus selection policies in a way that would not be adversely impacted by the changes on the Web, we propose performing a large baseline crawl that effectively defines a sandbox. All subsequent crawls, which correspond to the policies that are being compared, will be simulated as being within this sandbox. Such a setup provides us with a practical method to compare multiple crawl policies, reducing random variation and requesting each URL at most once.

Baeza-Yates et al compare [3] several page ordering strategies to determine which strategy crawled important pages early in the crawl, where page importance is measured with PageRank. To do so, they use a methodology similar to ours. They judge the crawling methods on the ability to find important pages first; most of their tested strategies achieve this objective.

In the current paper, we not only compare different crawling methods against a baseline crawl, we also consider multiple iterations. For iterative crawling, at every iteration, when a crawl policy picks a URL that is not present in our universe, we crawl it and add it to the set of known pages. This not only allows us to evaluate different policies for the same iteration, but we can also track a single method through iterations.

An important factor while evaluating a crawled corpus is its size, and therefore to simplify comparisons across corpora generated by different policies, we might want to compile sets of pages that are roughly the same size. From an ordered list of pages, where the ordering is on the basis of the crawl policy, if we attempt to select a precise number of pages, we might find that a number of pages share the same score. Some crawling policies are more prone to having such ties than others, therefore while making comparisons, we have to employ a suitable tie-breaking strategy that does not give advantage to one method over the others. Tie-breaking at random is a simple solution. In this paper, the experiments described in Section 3.4 required such a mechanism.

2.3 Evaluating and comparing crawl policies for search

Our evaluation of crawl policies is based on a set of relevance judgments on the pooled results from three major search engines. The judgments would be sufficient to compare the effectiveness of the three engines. However, instead we use the judgments to evaluate crawl policies. Unlike most papers which hold the corpus constant and vary the retrieval methods, in this paper we ignore the retrieval method and vary the corpus.

Our judgments are on a five-level relevance scale, and we use Normalized Discounted Cumulative Gain (or NDCG [13]) to represent effectiveness. Given the goal of maximizing retrieval effectiveness, one way of evaluating a crawl selection would be to index its documents, run a set of test queries and report mean NDCG. In previous work [10], we considered such an evaluation, highlighting the various issues involved, one of which is that such an experiment would require a large computational effort for indexing and searching. We also wish to obtain a measurement that is purely a function of the crawling strategy, rather than one convoluted by the choice of retrieval function.

To reduce computational effort, eliminating any dependency on indexing and searching methods, we use a new metric that we call *maxNDCG*. It is the maximum potential NDCG that could be

achieved by a perfect indexing and ranking system, limited only by what pages are present in the crawl. Under maxNDCG the optimal crawl selection is one which contains the relevant URLs for each of a large set of test queries. maxNDCG will be lower if high-gain URLs are missing from the crawl selection. The highest-gain URLs tend to be answers that will be most missed by users, such as the result `ebay.com` for the query 'ebay', so it is appropriate to penalize a crawl selection policy that tends to miss high-gain URLs. As mentioned before, the known relevant URLs were identified by pooling the top-10 results from three search engines. Therefore, our metric rewards a crawl policy for identifying relevant and retrievable URLs.

While relevance is certainly critical for any search engine, there are other desirable characteristics that can be traced back to the corpus. For example, response time to queries is seen by users as an important quality for a search engine, and we know that time taken to retrieve a result set is dependent on the size of the collection. Given the diverse nature of user interests, search engines have to be able to construct a diverse corpus that still allows efficient retrieval. In this paper, we will be contrasting potential user satisfaction (maxNDCG) with size of the corpus to reflect this tradeoff.

Related work in this context is a recent paper by Pandey and Olston [17] that addresses the problem of corpus construction for a search engine. Their approach is to identify 'needy queries' from the workload of a search engine, where it is likely that useful results are missing from the crawl. They use the queries to identify pages in the frontier, based on anchor text and URL matches, that would not have been selected based on popularity alone. This work differs from ours in that their objective is to design a crawling technique based on relevance information of queries, while we use the relevance information as one of the criterion on the basis of which we evaluate standard popularity-based crawling methods.

Our focus is on retrieval effectiveness and thus our use of relevance judgments, but to add support for our conclusions we consider a second source of relevance information. We obtained a list of URLs that had been displayed on the result page of the Microsoft Live Search engine. Each URL had associated with it a count indicating the number of times it had been clicked on. This click information provides an indicator of potential desirability of this page for an end user. The utility of a corpus is then calculated as the sum over the utility of its constituent pages. It is worth noting that this is a click-based RankMass evaluation, where the desirability of each page is defined according to click count. It is also related to impact-based evaluation [17] except we use the frequency of click rather than the frequency of retrieval, and we do not weight by rank. The latter was justified in [17] because lower ranks are less visible, but in click data all clicked results are visible and we count clicks in all ranks as equally important.

So far, we have discussed crawl corpus construction as a one-time activity. For a web search engine, this is certainly not the case, and keeping the corpus up-to-date is an important challenge. A search engine can maintain a fixed index size, but selectively drop some pages and incorporate others; the mechanism for knowing pages outside the index is the link graph. For example, uncrawled pages with high link popularity might be incorporated into the index, replacing some pages of lesser popularity. An update mechanism for dropping/including uncrawled pages, along with fixing an index size, defines a collection of web pages that are selected by this combination that will be the new corpus.

One difference between single iteration crawling and multiple iteration crawling is in the overall goal. When picking the set of pages for the current iteration, it has to be ensured that not only do we have good pages for retrieval (maxNDCG), we should also have pages that will be helpful in selecting good pages for the next iteration.

To measure the efficiency of the incremental update, we consider changes to the corpus that occur between generations. For example, we later calculate the Jaccard coefficient, a measure of set similarity, between sets of crawled pages. A low Jaccard indicates an expensive update, since the intersection of corpora generated by two successive iterations is smaller than the union, many new pages were crawled. Also, a policy that drops a page when going from iteration i to iteration $i + 1$, only to include it in the corpus again at $i + 2$ is deemed unstable. We term this alternating behavior as *churn*, and contrast it to the core of pages that the crawling method keeps stable across iterations. Measuring the relative sizes of these two sets is a useful indicator of crawling behavior. These numbers can not only be calculated on the total crawl, but also on just the set of relevant pages, i.e., those that have been considered useful by our relevance assessors.

Stability can also be motivated from a user perspective. A user trying to re-find a page [19] through the search engine might not find it if the page had been dropped in an update. Result-set predictability is driven by a stable corpus, and this factor should be considered while evaluating a crawl policy for search.

The axes along which we evaluate corpora in this paper are by no means exhaustive. Additional factors that affect how users perceive the effectiveness of a search engine include the amount of spam in the corpus, as well as the freshness of the pages. This highlights the multi-faceted nature of search engine evaluation, establishing the quality of the corpus through well-designed measures is very important. Our experiments in this paper are an initial step in this direction, and we leave more thorough comparisons and evaluations for future work.

3. EXPERIMENTS

As described in earlier sections, we are interested in the task of evaluating crawl policies in the context of search effectiveness. We consider three policies in this paper:

1. Inlink count (IDG) - each page is associated with the number of links pointing towards this URL that have been encountered up until that stage in the crawl.

2. Trans-domain inlink count (TD) - this method is similar to the indegree calculation described above, but only considers links between different domains, for example a link from `http://www.aa.com` to `http://www.aacareers.com` is a trans-domain link, while a link from `http://www.aa.com` to `http://fly.aa.com` is not.

3. PageRank score (PR) - using PageRank to order pages during a crawl has a long history (e.g.[8]).

These are known techniques - our contribution in this paper is the novel framework and methodology used to compare them. We are not aware of previous work that evaluates the corpora generated by different crawling methods in terms of potential search effectiveness. Additionally, we also consider how these three algorithms behave over multiple generations of incremental crawling. We also considered using OPIC [1] as a page selection method, however the advantages of OPIC for crawl-time page selection stem from the online nature of the algorithm, which is not relevant in our setting.

As described in Section 2.1, to make this large scale experiment possible, we constructed a sandbox of URLs from a baseline breadth-first crawl starting from the homepage of the Open Directory Project as the single seed. Over the course of this crawl we

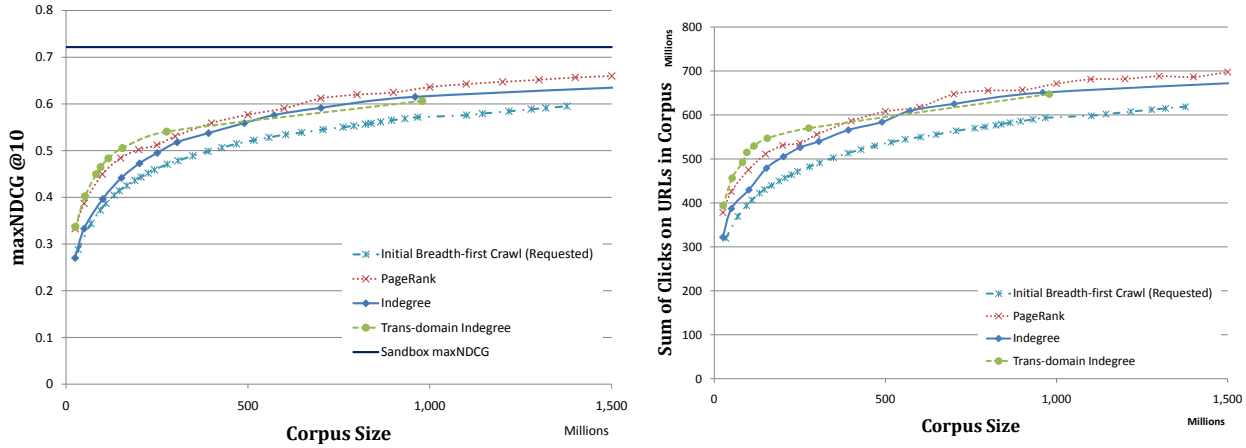

Figure 1: Comparison of the three crawling methods. Left: maxNDCG. Right: Sum of clicks

made 1,376,880,495 HTTP requests which resulted in 930,320,010 `text/html` documents and HTTP redirects, which we consider as documents with exactly one link. We employed a filtering mechanism that filtered out any HTTPS URLs or URLs from a small set of domains containing crawler traps.

We wrote the crawl state (consistently across all crawling nodes) to stable storage at intervals during the crawl, which we refer to as checkpoints. We have a total of 36 checkpoints over the breadth-first crawl. All the source pages in the initial link graph either had a mime-type of `text/html` or contained a HTTP redirect.

Using the link graph given by the initial breadth-first crawl, we carry out one iteration of crawl selection. Evaluation is based on the maxNDCG metric as defined in the previous section, indicating the ceiling on NDCG performance imposed by a particular crawl. The retrieval effectiveness experiments described in this paper were evaluated over a set of 10,570 queries sampled from the workload of Microsoft Live Search. Each of these queries had associated with them URLs whose relevance with respect to those queries had been obtained from human judges. Candidates for judgment were chosen from the top-ranking results generated by multiple engines. These relevance assessments were gathered independent from our current task, and therefore there could be some judged URLs missing from our various corpora.

There were 1,433,308 relevance judgments in total, each judging a page to either be "Bad", "Fair", "Good", "Excellent" or "Perfect" for a given query. In our set of assessments, 17.2% of our query-URL pairs were judged to be Good, Perfect, or Excellent. Since these queries were uniformly sampled from set of all queries, rare queries are represented as well as popular queries yielding a full picture of user satisfaction even for a query that only occurred once in our logs.

The five relevance categories were associated with gain values 0, 3, 7, 15 and 31 respectively. If the gain of the document at rank j is $G(j)$ then the Discounted Cumulative Gain at rank cutoff K is: $\sum_{j=1}^{K} \frac{G(j)}{\log(1+j)}$. The NDCG@K score for a query is calculated by dividing the DCG@K by the maximum DCG@K score that can be obtained for that query if all documents whose relevance is known were ranked in descending order of Gain. We note that this includes documents that were not in the crawl, and may not have been in any of the crawls. In this paper we always use a cutoff of $K = 10$, a standard setting in Web IR, and so omit the suffix $@K$. We also experimented with deeper cutoffs such as $K = 100$, but were concerned that this rewards a policy for crawling a large number of mediocre URLs, such as those with gain of 3 (i.e., a "Fair"). Crawling many mediocre URLs is not as important as crawling a few excellent URLs. We also perform these measurements using a linear gain vector without any change in the relative effectiveness of the selection policies.

In the first set of experiments, we evaluate the three crawling policies (indegree, trans-domain indegree, PageRank) on the basis of available human relevance assessments and then in terms of click information, to confirm maxNDCG as a reliable metric. In further maxNDCG experiments, we consider simple combinations of these single strategies, by taking the union of two selections. Thereafter, we evaluate crawl policies that place per-domain limits on the PageRank score for URL ordering. Our final set of experiments look at the task of incremental crawling, and the three methods are evaluated with respect to the corpora that are each of size 100 million, selected over multiple iterations.

3.1 Individual crawl ordering methods

The baseline for our first experiment is the breadth-first crawl of $930,320,010$ `text/html` source pages and redirects. The link graph induced by these crawled pages contains $16,124,409,514$ vertices in total, where every known page is a vertex and a link will connect the source and destination vertices. The 16 billion counts both the pages that were successfully fetched as well as the discovered, but un-crawled, pages that populate the crawl frontier. The 14.7 billion pages that were discovered but not crawled appear as vertices with one or more inlinks but no outlinks.

Figure 1 depicts the evaluation of each of the corpus selection techniques using the maxNDCG metric. We plot each selection method at various cutoff points, to get different points in the size-vs-maxNDCG tradeoff. The horizontal bar at the top of the figure is the maxNDCG (=0.7216) obtained if the complete set of 16,124,409,514 known URLs were successfully crawled. Please note that this line is not at 1 because our sandbox did not include all the judged relevant pages.

When calculating maxNDCG we can either use the entire set of selected URLs or the subset that were successfully crawled. In our crawls, around 70% of requests were successful. In Figure 1 we plot the *requested* maxNDCG, based on the selected pages without attempting to crawl them. For reasons of fair comparison, we also show the requested maxNDCG for the breadth-first crawl, rewarding the presence of a requested page regardless of whether the request was successful.

As can be seen from the figure, the breadth-first selection did not perform as well as any further selection method we experimented with. The amount of potential improvement is significant. The

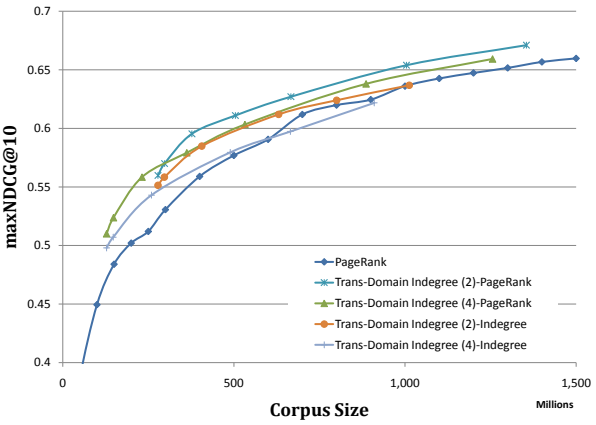

Figure 2: maxNDCG for combined selection criteria.

gap between the best performing selection at any size and the best possible individual corpus selection method is 0.0619. While the PageRank selection has a significant advantage over the indegree selection when selecting corpora in the 100 million to 200 million page range, the two curves track each other for larger corpus sizes. Considering selections of at least 300 million documents, the PageRank based selections consistently outperform the indegree based selections by 1 to 2 NDCG points.

PageRank has at least two advantages over indegree, despite being more computationally expensive since it involves repeated calculations on a large matrix. The first is retrieval performance as indicated by maxNDCG. The second is discriminatory ability. PageRank has a large number of unique values and therefore a simple threshold allows a wide range of crawl sizes to be selected. By contrast, indegree-based methods have a large number of pages with the same score (i.e., number of incoming links), due to their Zipfian distribution. The current graph has 960 million pages with indegree 3 and 2 billion pages with indegree 2, and we can only set a cutoff in the middle of such a range if we employ a tie breaker.

Trans-domain indegree has been shown to be useful for ranking web results [15]. We have used it here as a corpus selection method. In Figure 1, we observe that considering only trans-domain links provides a strong, valid signal, but this signal is of limited utility because our corpus does not have enough trans-domain links to perform selection of large collections. Trans-domain indegree outperforms PageRank for several crawl selections of less than 500 million documents. We observe issues similar to indegree with the discriminative power of the method. In our breadth-first crawl there are 276 million URLs with a trans-domain indegree greater than 1, and 979 million URLs with a trans-domain indegree of 1.

The results for our click based evaluation metric are plotted on the right side of Figure 1. The curves for sum of clicks closely track the maxNDCG values. For example, the trans-domain indegree and PageRank initiate very close to one point, and the breadth-first crawl and indegree selections start very close to a second point, just as they do for maxNDCG. At lower corpus sizes the trans-domain link method beats the others on both maxNDCG and sum of clicks. However, the PageRank and indegree selections soon catch-up and eventually are better on both measures.

We wish to point out that the available click information could have been used in a number of ways, e.g., it can motivate a crawl policy that prefers highly clicked pages/domains. Towards our aim of a search effectiveness based evaluation of corpus selection methods, human relevance judgments provide explicit evidence of document desirability. By keeping the implicit click-based relevance evaluation independent from our selection policies, we are able to reinforce the validity of using the maxNDCG metric to evaluate crawl corpora.

3.2 Combinations of crawl selection methods

The three selection methods outperformed the baseline breadth-first crawl, but so far we do not have any indication of how similar the different crawl selections are. We investigated the overlap between the sets of pages selected by the various selection policies as well as the overlap when only considering pages labeled Good or better. In both cases, the overlap was found to be low. While the indegree and PageRank selections had a large overlap, trans-domain indegree has a large number of distinct URLs, including distinct URLs labeled Good or better.

Based on this analysis, it seems worthwhile to consider combinations of our basic crawl policies. Here we implement a hybrid policy simply as the union of two basic-policy selections. The combined selection will be larger than its composite selections, but we would hope to see an increase in maxNDCG that compensates for this increased requirement for resources.

Figure 2 presents some examples of hybrid selection policies that outperform PageRank. They add the set of trans-domain indegree pages with cutoff 2 or cutoff 4 to PageRank and to indegree. We note that trans-domain indegree in combination can outperform the PageRank policy while being computationally much less expensive.

The best maxNDCG is achieved by combining trans-domain indegree with PageRank. Depending on the size of the desired corpus, we can choose a suitable threshold for the number of trans-domain links. An interesting observation from the figure is the utility of URLs with low numbers of TD links. In Figure 2, we see that combining the PR selection with the set of URLs having more than 2 trans-domain inlinks leads to a substantially higher maxNDCG than using a threshold of 4 for the number of TD links. Even if we were to factor in the fact that many more pages get included into the corpus, due to the more lenient requirement of the passing condition, the increased effectiveness more than makes up for the increase in corpus size. It is quite likely that pages that have low numbers of TD inlinks would have been ignored by a pure PR selection method, possibly due to low PageRank values, but a combination method helps extract the best of both individual policies.

3.3 Domain limits for crawl selection

So far we have considered policies that treat pages independently, such that the selection of one page does not depend on the selection of another. However, it may also be desirable to instate a 'budget' or limit on the number of pages crawled from each domain [2, 14]. This could potentially avoid the situation where a certain domain is over-crawled, selecting a large number of pages that are unlikely to be good search results.

The question is how best to set the limit. One solution would be to crawl more pages from a domain that has links from a larger number of other domains, as in the case of IRLbot [14]. Crawling more pages from a popular site would tend to be a good idea, but there are also potential counter-examples, such as the download site for a piece of software that has links from many domains but only has a few pages that are of interest as a search result.

An alternate scheme is to consider the number of entry points in a domain, being the number of unique URLs that have a trans-domain inlink or a search engine click. This way a download site would correctly be detected for shallow crawling, whereas a large site with many individual pages of technical information or archival material would be considered for deep crawling as long as it has

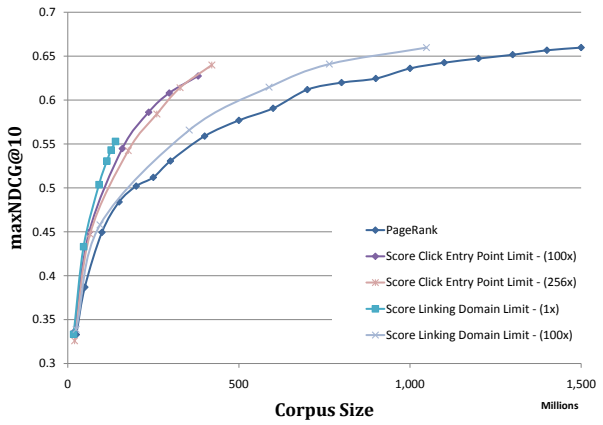

Figure 3: Domain limits on PageRank based corpus selections.

a large number of individual pages that are entry points. We also considered a static scheme, but found it to be less effective than the popularity or entry point-based schemes.

In addition to the source of the signal used to set the limit, there are also multiple ways to calculate the limit value. In [14], Lee et al propose a rank based method that assigns a default limit of ten pages per domain and then assigns an extra limit to the 10k highest ranked domains. These top domains have a limit that is linearly interpolated between 10k and 10 pages. In our experiments we also apply a scale factor, multiplying the baseline limit by 1, 10, 100, 256 and 512. We further evaluate score based methods for setting domain limits. Instead of considering the rank of the top 10k domains, we utilize the score from the chosen budgeting scheme for all domains, yielding greater discriminative power for the nearly 20 million domains in our corpus that are not in the top 10k.

We evaluate the effectiveness of the three different budgeting schemes previously mentioned: static limits, dynamic click-entrypoint based limits, and dynamic linking-domain based limits. The click entrypoint limits are calculated using the number of unique URLs, belonging to the domains, that were clicked on in search result pages. The linking domain budgets are calculated using the number of domains that link to a particular domain. For the two dynamic limits, we consider budgets using both rank and score based methods. We find that the score based methods outperform the rank based methods in terms of the quality to size ratio.

Figure 3 depicts the impact of a subset of these dynamic limits using several different constants for each type of limit. The limits are applied to corpus selections performed on the large breadth-first crawl using PageRank as the selection method. These limits were applied to PageRank selections of six different sizes: 25 million, 100 million, 400 million, 700 million, 1 billion, and 1.5 billion documents. The maxNDCG value achieved using the baseline PageRank corpus selection method to select 1.5 billion documents is 0.6597. The budget with the best performance is the click entry point with a scale factor of 256, which has a maxNDCG of 0.6398 and a corpus size of 421 million documents, which is a relatively slight reduction in quality for an index that is 28% the size. If we compare this to Figure 1 we see that this is a substantial improvement over the individual selection methods, as well as the combined methods shown in Figure 2. Such a crawl corpus selection method therefore allows us to construct a collection of pages that has a lower resource cost. In addition to being able to use these limits to choose a corpus, it would also be possible to use this to stratify a search engine's index in a multi-tier architecture such as the one proposed in [18].

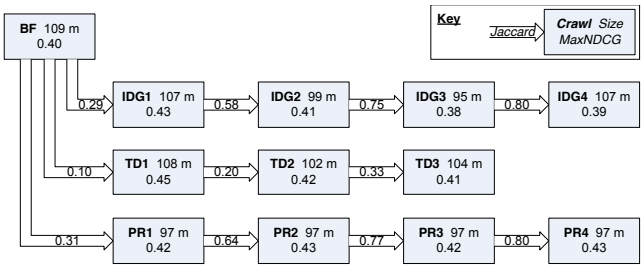

Figure 4: Iterative selection experiments, showing the size and maxNDCG of requested selections, and the Jaccard similarity between iterations.

3.4 Iterative crawl selection

As we have discussed previously, our motivation in this paper is the selection of corpora for a search engine, and the task of evaluating crawling methods to this end. In this context, keeping the index up-to-date, through periodic iterative crawls is clearly an important issue. So far, we have evaluated how our three crawling methods (indegree, trans-domain indegree, PageRank) behave on their first iteration.

We attempted to perform selections of multiple 100 million page corpora from the large breadth-first crawl, but found our sandbox unsuitable for this task due to a significant number of newly selected pages that were not requested during the construction of the original sandbox. Given the substantial time interval between our initial breadth-first crawl and this iteration experiment and the rapid rate of change on the web, we chose not to expand the initial sandbox by crawling the missing pages. As a result, for the selections described in this experiment, we constructed another collection of pages. We seeded this collection by performing another breadth-first crawl in a manner very similar to that described at the beginning of this section. This crawl, which occurred in October, 2008, requested 147,484,129 documents, of which 108,908,962 were successfully fetched as either a `text/html` document or an HTTP redirect. This breadth-first collection, hereafter called BF, was used to perform new corpus selections. For each URL in these new selections that does not exist in the set of URLs in the sandbox, crawl it, and record the result of the request in the collection. Therefore, we will only attempt to crawl any document once, and will dynamically expand the sandbox as needed. At the end of our iteration experiments the sandbox size was 411,491,513 pages.

Figure 4 summarizes our iterative experiments. In the first iteration the highest maxNDCG was achieved by trans-domain indegree (TD1), which marginally outperformed PR1 and IDG1. This confirms some results from Figure 1, the superiority of TD selection at small crawl size and the improvement of all three approaches over the baseline breadth-first crawl. PageRank becomes the most successful method on the second iteration with IDG and TD dropping dramatically in later iterations. The mean maxNDCG of actual selections was usually 93-95% of the requested mean maxNDCG.

We next consider how the corpora selected by the methods change over multiple iterations. For each policy-iteration pair, we have a set of URLs that defines the corpus, and using the set of relevance judgments we can calculate maxNDCG for this corpus.

As previously noted, high overlap between successive crawl iterations is desirable because it reduces crawling effort and maintains greater stability of search results for end users. To measure this we calculate the Jaccard coefficient between corpora generated by the different selection methods. The Jaccard coefficient between two corpora C_i and C_j would be given by: $\frac{|C_i \cap C_j|}{|C_i \cup C_j|}$. Figure 4 illustrates the Jaccard overlap between successive iterations, showing

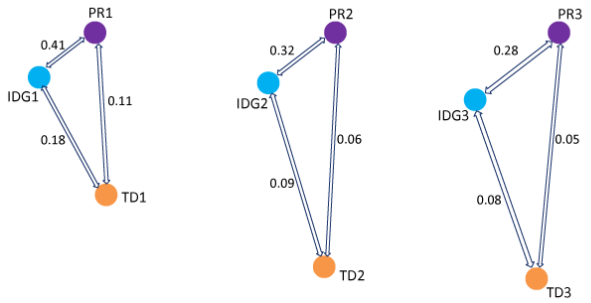

Figure 5: Jaccard similarity between corpora selected by different policies at different iterations.

that PageRank and indegree selections both stabilize with Jaccard of around 0.80 at the last iteration. The TD selections were relatively much less stable. Due to limited crawling resources, we stopped the TD crawling one iteration earlier, to avoid the massive cost of hitting so many new pages per iteration.

We can also calculate the similarity of crawls with each other. This is depicted in Figure 5. We find that the corpora generated by each crawling method become less and less similar with each iteration. In combination, Figures 4 and 5 indicate that policies stabilize on iteration, but do so towards quite disparate crawl selections.

In order to make comparisons across crawl policies easier (as in Figure 4), we wished to select corpora of roughly similar sizes. For TD, this meant having to use a random tie-breaking strategy. The inclusion of additional pages increased the maxNDCG, but also led to an decrease of 1 point in the Jaccard similarities.

Returning to our original objective of wanting to evaluate crawling methods with respect to search effectiveness, if we had concentrated on just the first iteration, using the trans-domain indegree would have seemed like the best policy. However, it soon was overtaken by both indegree and PageRank methods. Following the observation that successive generations of TD corpora have little in common, this method was deemed unstable.

To further analyze the instability we assign a three bit code to each URL according to whether it was present in iterations 1, 2 and 3. For example, a page present in all iterations is 111 and one which was present at first but disappeared in the next round and did not return is 100. Newly found URLs, i.e., those that were not in the original sandbox but would have been chosen by this method were given the labels 011 or 001 depending on the iteration at which they were found. Lost URLs are 100 or 110, because they were present in the original corpus and would have been dropped in an update. Instability or *churn* is indicated by 010 or 101.

Figure 6 shows what might be suspected, that a TD policy sees very few URLs which are stable across all three iterations, around 20 million. PageRank and indegree have relatively very little churn. Considering only relevant pages labeled Perfect, Excellent or Good, we note that TD looks more stable and sees a larger set of relevant URLs overall, but its set of stable (111) URLs is lesser than those of the other methods.

Our iterative experiments suggest that careful consideration is necessary when choosing a policy for repeated iterative selection. While trans-domain indegree was the best method on first iteration, PageRank is superior in later iterations. One explanation is that a good iterative crawl selection method will not only optimize for maxNDCG, but will also select pages that will be useful for selection during the next iteration. Perhaps trans-domain indegree is less effective at retaining such 'helpful' pages. Overall when moving from iteration $i-1$ to iteration i, we have to balance the greedy objective of finding as many desirable pages for NDCG on the current iteration, with that of including those pages that are likely to be helpful in picking the corpus for iteration $i+1$ [9].

To analyze this we consider the presence of 'helpful' URLs at each crawl iteration, being URLs that *link to* pages with a "Perfect", "Excellent" or "Good" rating. In Figure 7 we compare the number of helpful pages in iteration $i-1$ with the maxNDCG in iteration i, for iterations $i = 1, 2, 3, 4$ for PageRank and indegree and for $i = 1, 2, 3$ for trans-domain indegree.

Iteration zero in all cases is the breadth-first crawl, so the rightmost point for all three curves is fixed by the number of 'helpful' pages in the baseline BF crawl. In the figure, we note that the number of helpful pages available drops more quickly for TD selection than it does for other methods. For both TD and IDG, fewer helpful pages at iteration $i-1$ tends to be associated with lower maxNDCG at iteration i. However, PageRank manages to achieve a stable level of maxNDCG despite losing helpful pages more quickly than IDG. A possible explanation is that PageRank loses the *right* helpful pages. Pages that are more helpful for PageRank calculation will tend to have a higher PageRank weight, and therefore are more likely to be retained on iteration. When choosing between a large number of potentially helpful pages that have the same indegree, and thus were indistinguishable to degree-based methods, PageRank can choose to retain those with a higher weight.

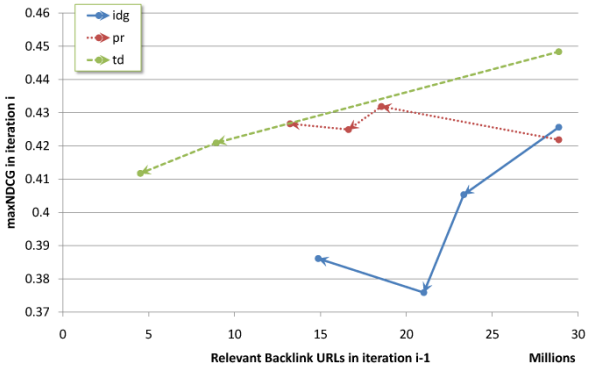

Figure 7: Importance of having pointers to relevant content for iterative recrawls.

4. CONCLUSIONS AND FUTURE WORK

The overall approach presented in this paper was to explore the tradeoff between crawl size and potential retrieval effectiveness. This was done in the context of a search engine that is required to perform an iterative update over its current corpus. This is important because any Web IR system faces size constraints, imposed by financial or technical limits. We have shown that the quality of search results achievable at a given crawl size is dependent on the crawl policy, and demonstrated an approach for analyzing a policy's size-vs-effectiveness tradeoff. Advocating the use of maxNDCG, measured in terms of potential retrieval effectiveness based on real relevance judgments, as a way to evaluate the user-centric utility of a crawled corpus is a major contribution of the paper. To make a large scale comparison of crawl policies possible, we used a sandbox-based method that decreases the complexity of an experiment of this scale. This coupled with our main choice of metric, maxNDCG, makes it tractable to evaluate a broad range of very large crawls.

The individual crawl ordering methods we tried were the use of inlink count, trans-domain inlink count and PageRank score. Our experiments indicated that while all three were better than the baseline

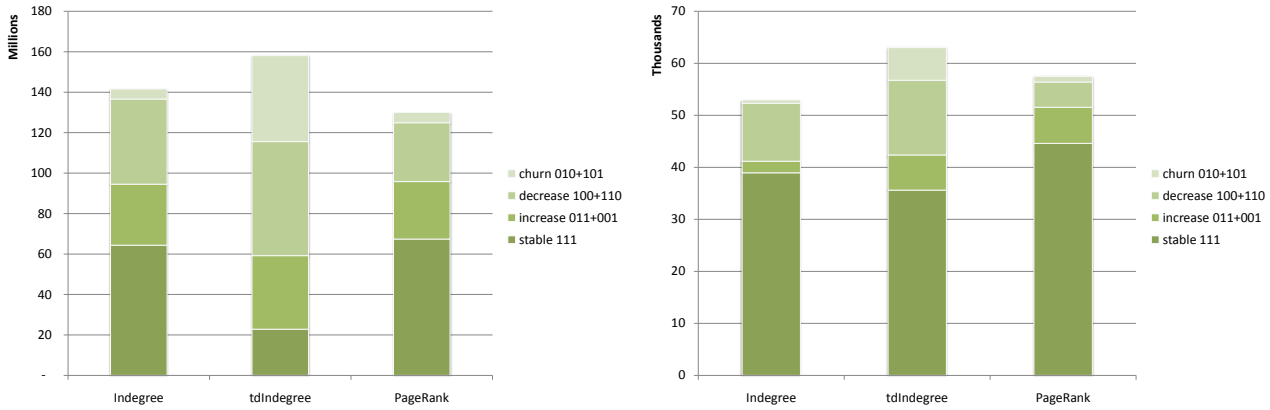

Figure 6: Stability of crawl policies, over three iterations. Left: All URLs. Right: Relevant URLs

of a breadth-first crawl strategy, PageRank provided the best performance. We observed that the aggregate performance of each method was roughly in the same range, but the corpora they generate were very diverse. In an attempt to exploit this observation, we showed that simple combinations, such as the use of trans-domain inlinks in association with PageRank, can lead to substantial improvements.

We also showed experimental results for crawl policies that combine PageRank with a limit on the number of pages crawled from a single domain. We used the number of domains that link into this domain and the number of click entry points, to set the per-domain limits. These methods provided significant improvements over using PageRank alone. Investigating different domain limiting criterion, and coupling them with the appropriate crawl selection method, is an important and interesting avenue for future work. Other potential policies include selecting pages to a pre-specified depth, as well as using the number of known pages in the domain as the basis for the domain's allowance. It remains to be seen if these policies continue to provide benefits when used in association with strategies other than PageRank. Some of the policies have been explored in previous work, but a retrieval-based evaluation would be novel. Based on combination experiments so far, if these policies are effective in isolation, incorporating them in hybrid selection policies may be fruitful.

While retrieval effectiveness was the main criterion when comparing corpora generated by each crawling method, we showed that multiple iteration incremental re-crawling needs to be studied in the context of a broader set of considerations. These include the efficiency of the update, in terms of the fraction of pages that are dropped between generations. The resulting notion of stability of a corpus led us to conclude that amongst the three methods tried, PageRank is the best incremental crawling strategy.

Of general interest is the need to define measures of corpus quality. In the context of web search engines, some desirable properties of the corpus are intuitively obvious, e.g. we would want to be robust with respect to manipulations that increase the fraction of spam documents in our corpus. Other application specific properties of corpora also follow by definition, e.g. a news search engine would want fresh pages. Focusing on generic web search, tracing back from what users would perceive as positive characteristics of search results, all the way towards designing a crawl policy that ensures the inclusion of such pages into the corpus is therefore of great importance.

5. REFERENCES

[1] S. Abiteboul, M. Preda, and G. Cobena. Adaptive on-line page importance computation. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 280–290, New York, NY, USA, 2003. ACM.

[2] R. Baeza-Yates and C. Castillo. Crawling the infinite web. *Journal of Web Engineering*, 6(1):49–72, 2007.

[3] R. Baeza-Yates, C. Castillo, M. Marin, and A. Rodriguez. Crawling a country: better strategies than breadth-first for web page ordering. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 864–872, 2005.

[4] Z. Bar-Yossef, A. Z. Broder, R. Kumar, and A. Tomkins. Sic transit gloria telae: towards an understanding of the web's decay. In *Proceedings of WWW*, pages 328–337, 2004.

[5] P. Boldi, and M. Santini, and S. Vigna. Paradoxical effects in pagerank incremental computations. *Internet Mathematics*, 2(3):387–404, 2005.

[6] J. Cho and H. Garcia-Molina. The evolution of the web and implications for an incremental crawler. In *Proceedings of VLDB*, pages 200–209, 2000.

[7] J. Cho, H. Garcia-Molina, and L. Page. Efficient crawling through URL ordering. *Computer Networks and ISDN Systems*, 30(1-7):161–172, 1998.

[8] J. Cho and U. Schonfeld. Rankmass crawler: a crawler with high personalized PageRank coverage guarantee. In *Proceedings of VLDB*, pages 375–386, 2007.

[9] A. Dasgupta, A. Ghosh, R. Kumar, C. Olston, S. Pandey, and A. Tomkins. The discoverability of the web. In *Proceedings of WWW '07*, pages 421–430, 2007.

[10] D. Fetterly, N. Craswell, and V. Vinay. Search effectiveness with a breadth-first crawl. In *Proceedings of 31st European Conference on Information Retrieval (ECIR)*, 2009.

[11] D. Fetterly, M. Manasse, M. Najork, and J. Wiener. A large-scale study of the evolution of web pages. In *Proceedings of WWW*, pages 669–678, 2003.

[12] M. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. Measuring index quality using random walks on the Web. *COMPUT. NETWORKS*, 31(11):1291–1303, 1999.

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

[14] H.-T. Lee, D. Leonard, X. Wang, and D. Loguinov. IRLbot: scaling to 6 billion pages and beyond. In *Proceedings of WWW 2008*, pages 427–436, 2008.

[15] M. A. Najork, H. Zaragoza, and M. J. Taylor. Hits on the web: how does it compare? In *Proceedings of SIGIR*, pages 471–478, 2007.

[16] A. Ntoulas, J. Cho, and C. Olston. What's new on the web?: the evolution of the web from a search engine perspective. In *Proceedings of WWW*, pages 1–12, 2004.

[17] S. Pandey and C. Olston. Crawl ordering by search impact. In *Proceedings of WSDM*, pages 3–14, 2008.

[18] K. M. Risvik, Y. Aasheim, and M. Lidal. Multi-tier architecture for web search engines. *la-web*, 00:132, 2003.

[19] J. Teevan, E. Adar, R. Jones, and M. A. S. Potts. Information re-retrieval: repeat queries in yahoo's logs. In *Proceedings of SIGIR*, pages 151–158, 2007.