Information retrieval in structured domains

Vincent W. L. Tam and John Shepherd
School of Computer Science and Engineering
University of New South Wales,
UNSW Sydney, NSW 2052, Australia
vincetam@cse.unsw.edu.au and jas@cse.unsw.edu.au

Abstract

In this work, we investigate utilizing the structure of a website to increase the effectiveness of document retrieval within a structured domain. In particular we examine various methods to combine evidence within the website in order to improve the quality of pages returned.

Keywords: Information retrieval, Structured IR, Passage retrieval

1 Introduction

Information retrieval is a broadly studied topic. Significant research efforts have been focused on document retrieval from World Wide Web. We aim to refine document retrieval within a website by improving the quality of document relevance against queries. We achieve this by taking into consideration the evidence collected from pages that are related to the document under inspection.

Websites are normally organised according to some structure (based on an information architecture) to make it more convenient for users to navigate the site. Often, the URL structure of pages reflects this organisation. These observations raise the issue of whether we can make use of the structure/organisation to improve search. The work in this paper sets out to explore this issue by trying to answer the following questions

1. Do surrounding pages of articles carry useful information to improve the quality of results in ranking documents against queries?
2. How to define the set of related pages for the above purpose and how to define the range of this set?

Our approach to answering these questions was to conduct information retrieval experiments on websites that were known to conform to a well-defined, hierarchical structure. The goal of these experiments was to determine how to use the information in related pages to improve relevance scoring. Such experiments, of course, could prove only that the approach is effective for sites that follow this structure.

2 Related Works

In this section we review several streams of research that motivated our experiment.

2.1 URL Structure

URLs of web pages have already been used to improve retrieval results. Keywords in URLs usually provide hints for information retrieval, and this has been utilized in search engines (and by "search engine optimisers") to enhance rankings of retrieved pages. There are also known uses of URLs as evidence to categorize pages in websites (e.g. Kules, Kustanowitz and Shneiderman 2006, Shih, and Karger 2004) where the pages are organised into hierarchies of subjects within the website. Under this assumption, URLs of web pages provide information on how the pages are categorized. Clearly, not all websites follow such conventions (e.g. many of the increasing number of dynamically-generated websites). However, a sufficient number of websites are organised by URL to make it worthwhile to consider this approach.

2.2 XML Element retrieval

A major stream of research that is related to our work is information retrieval in structured documents. This research focuses mainly on text retrieval from XML documents. XML documents are well-structured articles with tags to define elements within the articles. Information retrieval from XML documents aims to retrieve elements that closely match the queries. This stream of research is inspired by the Initiative for the Evaluation of XML retrieval (INEX). Our work differs from XML retrieval in that our targeted documents are individual pages within a website instead of elements contained in articles. Elements contained in articles have a well-defined unit (the article) to draw information about the context of the elements from (e.g. Kimelfeld, Kovacs, Sagiv and Yahv 2007). On the other hand, there is no clear boundary for this part-whole relationship for web pages. The range and number of documents to be included as related pages is not well-defined. To identify such boundaries was part of our research objective. A second difference between our work and XML retrieval is that queries in XML retrieval can specify the context of the desired results via Xpath. This is the case if the schema of the XML documents is known beforehand (e.g. Beigbeder 2007, Carpineto, Romano and Caracciolo 2007). Besides, the element tags of XML documents carry additional information for retrieval in the form of element attributes and element names. This helps in

defining the function of the elements (e.g. Abstract, summary, title). In our setting, such information is not available to assist with document retrieval.

2.3 Passage Retrieval

Another stream of research that is similar to our work is passage retrieval. This research focuses mainly on enhancing search results by returning passages within documents instead of the whole documents, or enhancing document retrieval by collecting evidence of relevance from individual passages (Wilkinson 1994). It has been shown that combining evidence from this part-whole relationship helps in returning more relevant passages during retrieval (Callan 1994, Sigurbjörnsson, Kamps and Rijke 2004). Our research was inspired by this finding, and adopts a similar approach to draw evidence from surrounding pages to enhance retrieval results.

2.4 Structures among web pages

Previous works on document retrieval have also explored the use of hyperlink structure. PageRank (Brin and Page 1998) and HITS (Kleinberg 1999) are two of the most widely used algorithms in this category. This work differs from our research in that it focuses mainly on the association between different websites. PageRank and HITS assigned a score of authority (together with a hub in the case of HITS) to websites. They calculate relevance of documents by incorporating this score during retrieval. This is not appropriate in our setting of retrieving pages from a single website. If we used such schemes, retrieval results would always be biased towards authority pages/page-groups, regardless of what users put in as queries. In particular we do not want to focus on ranking pages by their global importance within the site.

In addition, search engines like Google (Brin and Page 1998) utilize information propagation to enhance retrieval results. In particular, search engines propagate words from link anchors to their target pages and these anchor words play an important role during retrieval (Glover, Tsioutsiouliklis, Lawrence, Pennock and Flake 2002). This is similar to our proposal of using extra information from other pages, but instead we are looking at words from pages that surround the target page rather than words that refer to this page.

2.5 Vector space model and Cosine similarity

The vector space model (Salton 1971) and the cosine similarity algorithm are widely used to rank documents against queries. Our experiments calculated relevance based on these models. We extended the scoring mechanism of documents against queries by combining relevance score of surrounding pages returned by these models. Since the focus of this research was to examine the effectiveness of combining evidence collected from related pages in a structured domain, we did not in particular examine and compare the uses of other models e.g. BM25 (Robertson, Walker, Jones, Hancock-Beaulieu and Gatford 1994), nor any other approaches of adopting the vector space model. Given the popularity of such models we believe that this provided a reasonable and understandable platform to perform such tests.

3 Method

3.1 Structure of a website

We first describe the relationships among documents by using the URL of each page to construct a relationship graph.

Exploring page relationships by URLs is comparatively cheap in processing and is readily available. Websites often use a directory/folder hierarchy to reflect the organisation of information in the site, and this structure is reflected in the URLs. Our approach to determine relatedness of pages attempts to exploit this by considering that if a page was included in a folder it is likely that this page has a similar "context" to other pages under the same folder. Sub-folders typically contain documents which specialise the context of their parent folder. Users of such sites often exploit the hierarchical structure as a basis for navigating through the site. Figure 1 shows a scenario with a hierarchical collection of folders, along with their corresponding URLs.

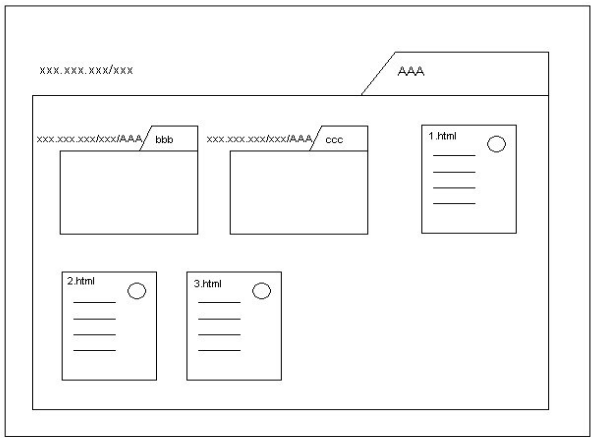

Figure 1: A folder-subfolder relationship scenario. Folders "bbb" and "ccc" are sub-folders in "AAA". The URLs of the folders reflect this hierarchy.

The first step is to establish an ancestor/descendent relationship among web pages, based on their URLs. If D is a page in a web-site, the URL(D) denotes its web address. The URLs of index pages (e.g. `index.html`) are normalised by removing the page component and treating them as a directory name. A page D_1 is defined to be an ancestor of another page D_2 if $URL(D_1)$ is a prefix of $URL(D_2)$, i.e.

$$URL(D_2) = URL(D_1)\ /*$$

Based on the ancestor/descendent relationship, we introduce a distance function $Dist(D_1,D_2)$ which measures the number of pages *d* separating the two pages along their URL paths. $Dist(D_1,D_2)$ is defined as follows:

$Dist(D_1, D_2) = 0$, if D_1 and D_2 is not related.

$Dist(D_1, D_2) = d$, if D_2 is a descendent of D_1

$Dist(D_1, D_2) = -d$, if D_1 is a descendent of D_2

3.2 Relevance of Documents

We adopt the vector space model and cosine similarity to calculate the similarity *sim(q,d)* of a document against a query:

$$sim(q,d) = \frac{\sum (w_{t.d} \cdot w_{t.q})}{\sqrt{\sum w_{t.d}^{\ 2} \times \sum w_{t.q}^{\ 2}}}$$

where $w_{t.d}$ is the tfidf score of a term *w* in the document *d*, given by the following formula:

$$w_{t.d} = tf \times \ln(\frac{N}{n})$$

tf = term frequency of the term w in the document

N = total number of indexed document

n = total number of documents that contain w.

Documents are ranked by their *sim(q,d)* score.

3.3 Implementation

We store every term in the index. Terms are stemmed using the Porter Stemming algorithm (Rijsbergen, Robertson and Porter 1980) and stored with their term frequencies and tfidf scores. The sum of square of terms' tfidf for each page (the $w_{t.d}^2$ part in sim(q,d)) was also calculated and stored in the index for more efficient retrieval.

To obtain the distance measure $Dist(D_1,D_2)$, we compare page URLs retrieved from the index at runtime. The URLs stored in the index are pre-processed as follows: removing duplicate URLs, removing redundant '/' characters, removing index page filenames, appending '/' to the end of URLs for index pages. $Dist(D_1,D_2)$ is then calculated by counting the difference in the number of '/' characters in the URLs if one of their URLs is a prefix of the other.

3.4 Evidence from other documents

We combine the relevance score of surrounding documents with the initial score obtained from cosine similarity. We introduce a variable, factor λ, to adjust the relative weights of the surrounding evidence score and the initial similarity score. The λ factor has a value between 0 and 1; its use is shown below.

We also introduce a variable *rLimit* to adjust the definition of "related pages". To be precise, *rLimit* is the maximum distance allowed between two pages for them to be treated as related pages. For example, if *rLimit* is set to 1 we only considered pages that are one document away from the one we are looking at along the path of URL.

3.4.1 First Approach

Our first attempt accumulated the relevance score directly from related documents. The relevance score for document *d* with respect to query *q* is given by:

$$F_1(d) = (1-\lambda) \times sim(q,d) + \sum_{d_n \in R} sim(q,d_n) \times \sigma$$

$$\sigma = \lambda^{\log_2(|dist(d,d_n)|)}$$

where *R* is the set of documents with $dist(d, d_n) \neq 0$, and $|dist(d, d_n)| <= rLimit$.

Notice that in this formula, we take the absolute value of $dist(d,d_n)$ in defining the set R. We therefore do not explicitly distinguish between ancestor pages and descendent pages. The factor σ is introduced to account for the fact that the further away a document is from *d*, the smaller the effect it should be affecting *d*.

3.4.2 Second Approach

The first approach we adopted is a simple framework to combine scores of surrounding documents. There were two shortcomings in this attempt. Firstly it did not take into account sibling documents in the structure we introduced. Secondly there existed a bias to pages that had a lot of related pages. These were usually pages that were indices to folders that contained many child pages. To account for these we introduce our second set of formulae for document relevance. Note that the score is computed in two stages as described below.

Stage I

$$F_2"(d) = (1-\lambda) \times sim(q,d) + \sum_{d_n \in R} sim(q,d_n) \times \sigma$$

$$\sigma = \lambda^{\log_2(|dist(d,d_n)|)}$$

where *R* is the set of documents with $dist(d, d_n) < 0$, and $|dist(d, d_n)| <= rLimit$.

Stage II (final score used)

$$F_2(d) = (1-\lambda) \times F_2"(d) + \sum_{d_n \in R} F_2"(d_n) \times \sigma$$

$$\sigma = \lambda^{\log_2(dist(d,d_n))}$$

where *R* is the set of documents with $dist(d, d_n) > 0$, and $dist(d, d_n) <= rLimit$.

We split the scoring process into two stages. In the first stage we accumulate cosine similarity scores from

descendents of pages. In the second run, we do the reverse, accumulating similarity scores returned in the first run from ancestors. The motivation for doing so was that in our setting in defining website structure, a document could have more than one child page, while each document belongs to a single parent page. In other words, a document belongs to one and only one folder, but a folder usually consists of more than one document. Therefore splitting the accumulation of score into two stages ensures that the child pages can share the biased scores from their parents returned in the first run.

Another advantage of this formula is that it captures the effect of both the parent-child relationship and the effects of sibling pages in the folder. This is because in obtaining the results during the first stage, the parent's score has been affected by all of its children. Therefore in the second run, when we combine evidence collected from parent pages with their updated similarity scores, the effect of sibling pages is propagated to every child page of the parent. Figure 2 shows an example of this.

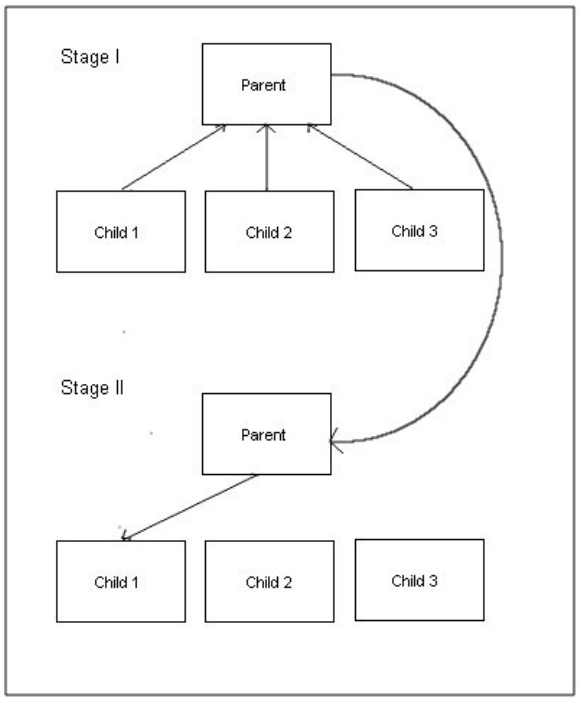

Figure 2: An example illustrating how evidence from sibling pages is propagated to "Child 1".

3.4.3 Third Approach

The first two attempts in our experiments accumulated the *sim(q,d)* score of surrounding documents as evidence to refine the relevance of the documents during retrieval. In our third formula we also wanted to take into account the length of each document explicitly. Not only does this give us a closer approximation to the original cosine similarity approach, but it also accounts for the bias to pages with more children mentioned above, without the need to split the process into two runs. This is done by dividing the sum of the dot product for each related page, after taking the factor λ into account, by the length of all documents that have been included in the calculation:

$$F_3(d) = \frac{\sum (w_{t.d} \cdot w_{t.q}) \times \sigma + \sum_R \sum_{d_n} \frac{w_{t.d_n} \cdot w_{t.q}}{\alpha}}{\sqrt{\left[\sum {w_{t.d}}^2 + \sum_R \sum_{d_n} \left(\frac{w_{t.d_n}}{\alpha} \right)^2 \right] \times \sum {w_{t.q}}^2}}$$

$$\sigma = (1 - \lambda)$$

$$\alpha = \log_2(|dist(d, d_n)|)$$

where R is the set of documents with dist(d, d_n) ≠0, and |dist(d, d_n)| <= *rLimit*.

Notice that this formula is similar to the cosine similarity function, the difference being that we have factored down the effect of related documents by their distance from the inspecting document. The formula also treats every document individually in calculating the vector length instead of adding all the vectors before calculating the length. Given the sparsely distributed vector space property of text in a retrieval system we believe this is a reasonable approximation, and avoids the need to re-calculate variable document length at runtime. As mentioned in section 3.3, the sum of squares of $w_{t.d}$ was pre-calculated and stored in the index and therefore was readily available.

4 Experiments

We tested our algorithms on the websites of computing courses at the University of New South Wales. Our sample websites contained on average more than 1500 pages each. A set of queries was derived from questions that were asked on the course forums, so should represent realistic requests for information from the course websites.

In the evaluation we manually determined if a retrieved page was relevant, relevant but not helpful, or irrelevant from a number of subjects. We then assigned a score of 2, 1 and 0 respectively to the retrieved documents and took the average from the manual scores. We evaluated the effectiveness of different approaches by picking the first five ranked documents for each approach and calculating the total relevance score for these documents. The final result of each run was expressed in a precision ratio calculated from the abovementioned method. The tests were carried out by varying the factor λ in each formula. We also obtained two runs for each formula by setting *rLimit* to 1 and to infinity respectively.

The larger λ is, the more we weight evidence collected from related documents. When *rLimit* was equal to 1, the set of related pages were limited to pages that are directly related along the URL path. In other words, only those that surround the pages were used. On the other hand with *rLimit* set to infinity we took into consideration all pages along the path, with the effect of pages being factored by the distance from the inspecting page as described in our formulae. We compared each run with the baseline method of cosine similarity.

5 Results and Discussion

The precision ratios for our example queries are presented as follows.

Query	Baseline	F_1 (λ =0.25)	F_1 (λ =0.5)	F_1 (λ =0.75)
1	0.3	0.2	0.2	0.2
2	0.3	0.4	0.4	0.4
3	0.3	0.3	0.1	0.1
4	0.7	0.4	0.3	0.3
5	0.2	0.4	0.4	0.4
6	0.1	0.3	0.3	0.3
7	0.1	0.3	0.3	0.3
8	0.2	0.2	0.2	0.2
9	0.3	0.1	0.3	0.4
10	0.3	0.4	0.2	0.1
11	0.2	0.2	0.1	0.2
12	0	0.1	0	0.1
13	0.2	0.3	0.3	0.1
14	0.6	0.8	0.8	0.8
15	0.4	0.3	0.2	0
Average	**0.28**	**0.31**	**0.27**	**0.26**

Table 1: F1 with *rLimit* = 1

Query	Baseline	F_1 (λ =0.25)	F_1 (λ =0.5)	F_1 (λ =0.75)
1	0.3	0.2	0.2	0.2
2	0.3	0.3	0.3	0.3
3	0.3	0.1	0.1	0.1
4	0.7	0.4	0.3	0.3
5	0.2	0.4	0.4	0.4
6	0.1	0.3	0.3	0.3
7	0.1	0.2	0.2	0.2
8	0.2	0	0.2	0.2
9	0.3	0.1	0.3	0.4
10	0.3	0.2	0.1	0.1
11	0.2	0.1	0.1	0.1
12	0	0.1	0	0.1
13	0.2	0.1	0.1	0.1
14	0.6	0.8	0.8	0.8
15	0.4	0	0	0
Average	**0.28**	**0.22**	**0.23**	**0.24**

Table 2: F1 with no *rLimit*

Table 1 and Table 2 compare the results of retrieval using Formula 1 against the baseline method, which was cosine similarity. While table 1 showed the ability to obtain a similar result to the baseline method, table 2 showed inferior results. As we had earlier point out, Formula 1 would bias to pages that contained many child pages and we found that this was exactly the case when we looked in details into the pages returned by the algorithm. Nevertheless, by setting *rLimit* = 1 the results had been better than that without *rLimit*. In particular we obtained better results than the baseline when *rLimit* = 1 and λ = 0.25. This showed that under certain circumstances, we did benefit from collecting context evidence from surrounding documents. The implication of having better results with $\lambda < 0.5$ was that while we took into account of related documents, we shouldn't forget the importance of the initial similarity score of the documents themselves. In other words, the initial score of documents should still be the major deciding factor, yet we could benefit from taking into account other evidence with comparatively less weight.

Query	Baseline	F_2 (λ =0.25)	F_2 (λ =0.5)	F_2 (λ =0.75)
1	0.3	0.6	0.6	0.6
2	0.3	0.4	0.3	0.3
3	0.3	0.2	0.2	0.2
4	0.7	0.9	0.8	0.8
5	0.2	0.4	0.2	0.2
6	0.1	0.3	0.1	0.1
7	0.1	0.3	0.2	0.2
8	0.2	0	0	0
9	0.3	0.5	0.6	0.6
10	0.3	0.6	0.3	0.3
11	0.2	0.1	0.4	0.3
12	0	0	0	0
13	0.2	0.3	0	0
14	0.6	0.8	0.6	0.6
15	0.4	0.2	0	0
Average	**0.28**	**0.37**	**0.29**	**0.28**

Table 3: F2 with *rLimit* = 1

Query	Baseline	F_2 (λ =0.25)	F_2 (λ =0.5)	F_2 (λ =0.75)
1	0.3	0.6	0.6	0.6
2	0.3	0.3	0.4	0.3
3	0.3	0.2	0.2	0.2
4	0.7	0.9	0.8	0.8
5	0.2	0.4	0.2	0.2
6	0.1	0.3	0.1	0.1
7	0.1	0.1	0.2	0.2
8	0.2	0	0	0
9	0.3	0.5	0.6	0.6
10	0.3	0.2	0.2	0.3
11	0.2	0.3	0.4	0.3
12	0	0	0	0
13	0.2	0.1	0	0
14	0.6	0.8	0.6	0.6
15	0.4	0.2	0.2	0.2
Average	**0.28**	**0.33**	**0.30**	**0.29**

Table 4: F2 with no *rLimit*

Table 3 and Table 4 compare the results of retrieval using Formula 2 against the baseline method. This formula was designed with the aim to alleviate the bias towards the

parent-child relationship presented with our proposed way to define website structure. In addition to that, the formula also took advantage of capturing the effect of sibling pages. The effect of this was obvious. As seen in the tables, we succeeded in improving retrieval results from that of Formula 1 in all cases.

From Table 4 we see that we obtained a much improved results when comparing the results to Table 2. With no *rLimit*, the bias in F1 we mentioned above would propagate along the path to every level in the hierarchy and therefore deteriorate document relevance. Although we had factored down the effect of pages that are further apart by the σ factor, the accumulation of scores from a group of pages might have too large an effect and therefore pages were still heavily affected. On the other hand having alleviated the bias with F2 we could see from Table 4 that the retrieval quality benefited from taking context along the path. The best result was obtained when $\lambda = 0.25$.

Similarly, from table 3, we had the best retrieval results when $\lambda = 0.25$. When comparing Table 3 and Table 4 we found improvement in results when *rLimit* = 1 and $\lambda = 0.25$. This is similar to the findings with Formula 1.

Query	Baseline	F_3 (λ =0.25)	F_3 (λ =0.5)	F_3 (λ =0.75)
1	0.3	0.4	0.4	0.2
2	0.3	0.4	0.4	0.5
3	0.3	0.2	0.2	0.2
4	0.7	0.4	0.4	0.4
5	0.2	0.2	0.2	0.1
6	0.1	0.3	0.3	0.3
7	0.1	0.3	0.4	0.2
8	0.2	0	0	0
9	0.3	0.1	0.2	0.2
10	0.3	0.7	0.7	0.5
11	0.2	0.2	0.2	0.2
12	0	0	0.1	0.1
13	0.2	0.3	0.3	0.3
14	0.6	0.6	0.6	0.6
15	0.4	0.4	0.4	0.4
Average	**0.28**	**0.30**	**0.32**	**0.28**

Table 5: F3 with *rLimit* = 1

Query	Baseline	F_3 (λ=0.25)	F_3 (λ =0.5)	F_3 (λ =0.75)
1	0.3	0.4	0.4	0.2
2	0.3	0.3	0.3	0.4
3	0.3	0.2	0.2	0.2
4	0.7	0.4	0.4	0.4
5	0.2	0.2	0.2	0.1
6	0.1	0.3	0.2	0.2
7	0.1	0.2	0.2	0.2
8	0.2	0	0	0
9	0.3	0.1	0.2	0.2
10	0.3	0.6	0.6	0.5
11	0.2	0.1	0.1	0.2
12	0	0	0.1	0.1
13	0.2	0.3	0.3	0.3
14	0.6	0.6	0.6	0.6
15	0.4	0.2	0.4	0.4
Average	**0.28**	**0.26**	**0.28**	**0.27**

Table 6: F2 with no *rLimit*

Table 5 and Table 6 show the results of applying Formula 3 to the retrieval. While we managed to obtain slightly better results with the case when setting *rLimit* = 1, the results with no *rLimit* is less obvious. Formula 3 incorporated distance into the calculation of documents scores. We obtained the best results with $\lambda = 0.5$ and *rLimit* = 1. When comparing the approach of Formula 3 with the other formulae, Formula 3 was designed so that we did not have to worry about the effect of having very many related documents, which was the cause of the poor result cases when using Formula 1. In comparing results of Formula 1 and Formula 3 we found that this has been successful. However Formula 2 is superior to Formula 3. We believe that splitting the process into two runs, not only alleviates the problem of having too many related documents, but also more effectively takes into account the similarity score of sibling pages.

To sum up, the best results were obtained from Table 3 when Formula 2 was used, with $\lambda = 0.25$ and *rLimit*=1. The worst results were obtained when we used Formula 1 without setting *rLimit*. This reinforced our belief that the use of evidence without distinguishing pages as ancestors or descendents would be inferior in our setting because there is a 1-to-m relationship, whereas a parent could have more than one child page and therefore the accumulation of evidences from them resulted in bias to these pages. Our two attempts to solve the bias, namely by splitting the runs into two separate single direction accumulation of score along the paths of related pages so as to allow the other pages to share this bias, and to take document length into account, have been successful. Apart from Formula 1, our results suggest that retrieval is more effective when context from related pages is taken into consideration.

In any case, we observed that we always obtained better results if used only the immediate surrounding pages as related pages (i.e. *rLimit* = 1). This suggests an answer to our second research question of "how to define the ranges of related pages to assist in improving the results of information retrieval?" We believe the reason for this was that the further away an ancestor is from a page, the more general context it has (and thus is less directly relevant to the query). On the other hand the further away a descendent is from a page, the more specific context it carries (and this context may be irrelevant to the query). Nonetheless in all of our approaches we observe that retrieval results do benefit from taking surrounding context into calculation.

Another point that is worth noting is that the first two formulae we used depend on the cosine similarity that we are comparing to. In other words, these aimed to improve the retrieval results among a pool of already retrieved

documents. This can be observed by the fact that the individual score of each run on each query did not vary much. If the cosine similarity method was not able to draw the relevant documents set from the index then Formula 1 and Formula 2 can at most improve slightly on the results but wouldn't had a much better results returned. On the other hand Formula 3 is less dependent on the original results set given by cosine similarity and we therefore observed cases that either improve much or vice versa. Nevertheless, from our observation, with the right settings of environment variable λ and *rLimit*, the retrieval results were improved.

6 Conclusion and Future works

In this work, we have conducted preliminary experiments to show that for websites where the underlying domain structure is reflected in the URLs of the documents, retrieval results can be improved by taking into account evidence collected from related articles. We utilized the URLs in order to explore the hierarchy of the website and draw related pages from this hierarchy. We also showed that it is most effective when only immediately surrounding documents were used instead of taking into account every document along the related path. Although we are yet to perform further experiments on approaches other than re-ranking the documents returned by cosine similarity, our experiment has shown that it is worthwhile to draw evidence of context from other documents in the website.

Having gained encouraging results from our tests, the next stage is to perform more tests (more queries, more webistes) to provide a stronger base of evidence for such effects. In addition we aim to improve our formula in calculating similarities of documents and queries. In particular, as mentioned in the previous section, Formula 1 and Formula 2 rely on the initial scores obtained by cosine similarity. We therefore would be interested to look for better algorithms, so that our retrieval system would not only to make improvement based on the cosine similarity score, but also look for relevant documents that might be missed by it.

In addition we would try on retrieval algorithm other than cosine similarity to test on the effect of our approach to draw evidence from related page. We could then combine various scoring methods to enhance retrieval results further.

Besides, we would like to perform tests to see if the order of modifying scores according to the hierarchy is important. We have already carried out similar tests with Formula 2, in which we split the formula into two runs. In future research, we would examine the order of applying score to ancestors and descendents in each run.

Finally apart from using URLs as hints to categorize pages, we would also like to examine the use of other structure, including linkages among pages within a website, to test the effectiveness of drawing evidence from related pages that utilizes other structure within the website. This might enable us to deal with websites, such as Wikis and CMSs, where the domain structure of the site is not directly reflected in the URL structure.

7 References

Beigbeder, M. (2007): Structured content-only information retrieval using term proximity and propagation of title terms. In Proceedings of INEX 2006, page 200-212.

Brin, S. and Page, L (1998): The anatomy of a large-scale hypertextual web search engine. In Proceedings of the 7th International World Wide Web Conference, pages 107-117.

Callan, J. (1994): Passage-level evidence in document retrieval. In Proceedings of the 17th ACM-SIGIR Conference on Research and Development in information retrieval, pages 302-310.

Carpineto, C., Romano, G., Caracciolo, C. (2007): Information Theoretic Retrieval with structured Queries and Documents. In Proceedings of INEX 2006, pages 178-184.

Glover, E., Tsioutsiouliklis, K., Lawerence, S., Pennock, D. M. and Flake G. W. (2002): Using web structure for classifying and describing web pages. In Proceedings of the 11th international conference on World Wide Web. Pages 562-569.

INEX: Initiative for the Evaluation of XML Retrieval, 2007

Kimelfed, B., Kovacs, E., Sagiv, Y. and Yahav, D. (2007): Using Langauge models and the HITS Algorithm for XML Retrieval, Proceedings of INEX 2006, pages 253-360.

Kleinberg, J (1999): Authoritative sources in a hyperlinked environment. Journal of the ACM, 46:604-632.

Kules B., Kustanowitz J. and Shneiderman, B. (2006): Categorizing web search results into meaningful and stable categories using fast-feature techniques. In Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. and Gatford, Mike. (1994): Okapi at TREC-3. NIST Special Publication 500-226, Overview of the Third Text Retrieval Conference (TREC-3).

Salton, G. (1971): The SMART Retrieval System – Experiments in Automatic Document Processing, Prentice-Hall, Inc., Upper Saddle River, NJ, 1971.

Shih, L. K. and Karger D. R. (2004): Using URLs and table layout for web classification tasks, In Proceedings of the 13th international conference on World Wide Web, pages 193-202.

Sigurbjörnsson, B., Kamps, J. and Rijke, M. (2004): An Element-Based Approach to XML Retrieval. In INEX 2003 Workshop Proceedings, pages 19-26.

Wilkinson, R. (1994): Effective retrieval of structured documents. In Proceedings of the 17th ACM-SIGIR Conference on Research and Development in information retrieval, pages 311-317.

Van Rijsbergen, C. J., Robertson, S. E. and Porter, M. F. (1980): New models in probabilistic information retrieval. London: British Library. (British Library Research and Development Report, no. 5587).