Using and Detecting Links in Wikipedia

Khairun Nisa Fachry[1], Jaap Kamps[1,2], Marijn Koolen[1], and Junte Zhang[1]

[1] Archives and Information Studies, Faculty of Humanities, University of Amsterdam
[2] ISLA, Faculty of Science, University of Amsterdam

Abstract. In this paper, we document our efforts at INEX 2007 where we participated in the Ad Hoc Track, the Link the Wiki Track, and the Interactive Track that continued from INEX 2006. Our main aims at INEX 2007 were the following. For the Ad Hoc Track, we investigated the effectiveness of incorporating link evidence into the model, and of a CAS filtering method exploiting the structural hints in the INEX topics. For the Link the Wiki Track, we investigated the relative effectiveness of link detection based on retrieving similar documents with the Vector Space Model, and then filter with the names of Wikipedia articles to establish a link. For the Interactive Track, we took part in the interactive experiment comparing an element retrieval system with a passage retrieval system. The main results are the following. For the Ad Hoc Track, we see that link priors improve most of our runs for the Relevant in Context and Best in Context Tasks, and that CAS pool filtering is effective for the Relevant in Context and Best in Context Tasks. For the Link the Wiki Track, the results show that detecting links with name matching works relatively well, though links were generally under-generated, which hurt the performance. For the Interactive Track, our test-persons showed a weak preference for the element retrieval system over the passage retrieval system.

1 Introduction

In this paper, we describe our participation in the INEX 2007 Ad Hoc and Link the Wiki tracks, and the INEX 2006 Interactive Track. For the Ad Hoc track, our aims were: a) to investigate the effectiveness of incorporating link evidence into the model, to rerank retrieval results and b) to compare several CAS filtering methods that exploit the structural hints in the INEX topics. Link structure has been used effectively in Web retrieval [9] for known-item finding tasks. Although the number of incoming links is not effective for general ad hoc topics on Web collections [5], Wikipedia links are of a different nature than Web links, and might be more effective for informational topics.

For the Link the Wiki Track, we investigated the relative effectiveness of link detection based on the Wikipedia article's name only, and on the matching arbitrary text segments of different pages. Information Retrieval methods have been employed to automatically construct hypertext on the Web [2], as well for specifically discovering missing links in Wikipedia [4]. The track is aimed at detecting missing links between a set of topics, and the remainder of the collection, specifically detecting links between an origin node and a destination node. To detect whether two nodes are implicitly connected, it is necessary to search the Wikipedia pages for some text segments that both nodes share.

For the Interactive Track, we took part in the interactive experiment comparing an element retrieval system with a passage retrieval system. The element retrieval system returns element of varying granularity based on the hierarchical document structure and passage retrieval returns non-overlapping passages derived by splitting the document linearly. Trotman and Geva [16] argued that, since INEX relevance assessments are not bound to XML element boundaries, retrieval systems should also not be bound to XML element boundaries. Their implicit assumption is that a system returning passages is at least as effective and useful as a system returning XML elements. Since the document structure may have additional use beyond retrieval effectiveness, think for example of browsing through a result article using a table of contents, the INEX 2006 Interactive Track set up a concerted experiment compare an element retrieval system to a passage retrieval system [11]. The INEX 2006 Interactive Track run well into INEX 2007, so we report our findings here.

The rest of the paper is organized as follows. First, Section 2 describes our retrieval approach. Then, in Section 3, we report the results for the Ad Hoc Track: the Focused Task in Section 3.1; the Relevant in Context Task in Section 3.2; and the Best in Context Task in Section 3.3. Followed by Section 4, which details our approach and results for the INEX 2007 Link the Wiki Track. In Section 5 we discuss our INEX 2006 Interactive Track experiments. Finally, in Section 6, we discuss our findings and draw some conclusions.

2 Experimental Setup

2.1 Collection, Topics, and Relevance Judgments

The document collection is based on the English Wikipedia [17]. The collection has been converted from the wiki-syntax to an XML format [3]. The XML collection has more than 650,000 documents and over 50,000,000 elements using 1,241 different tag names. However, of these, 779 tags occur only once, and only 120 of them occur more than 10 times in the entire collection. On average, documents have almost 80 elements, with an average depth of 4.82.

There have been 130 topics selected for the INEX 2007 Ad Hoc track, which are numbered 414-543. Table 1 shows some statistics on this years assessments. We have included the numbers from last years assessments for comparison. The number of relevant articles and passages is slightly higher than last year, while the number of assessed topics is lower. Last year, 114 topics were assessed, with 49.54 relevant articles and 79.68 relevant passages per topic. This year, 107 topics were assessed, with 60.66 relevant articles and 107.31 relevant passages per topic. The average number of relevant passages per relevant articles is 1.61 for the 2006 topics and 1.77 for the 2007 topics. On the other hand, the size of the relevant passages this year has decreased compared to last year. Both average (931) and

Table 1. Relevant passage statistics

	Statistics	
Description	2006	2007
# topics	114	107
# articles with relevance	5,648	6,491
# relevant passages	9,083	11,482
mean length relevant passage	1,090	931
median length relevant passage	297	272

median (272) size (in character length) are lower than last year (1,090 and 297 respectively).

2.2 Indexing

Our indexing approach is based on our earlier work [8, 13, 14, 15].

– *Element index*: Our main index contains all retrievable elements, where we index all textual content of the element including the textual content of their descendants. This results in the "traditional" overlapping element index in the same way as we have done in the previous years [14].
– *Contain index*: We built an index based on frequently retrieved elements. Studying the distribution of retrieved elements, we found that the `<article>`, `<body>`, `<section>`, `<p>`, `<normallist>`, `<item>`, `<row>` and `<caption>` elements are the most frequently retrieved elements. Other frequently retrieved elements are `<collectionlink>`, `<outsidelink>` and `<unknownlink>` elements. However, since these links contain only a few terms at most, and say more about the relevance of another page, we didn't add them to the index.

For all indexes, stop-words were removed, but no morphological normalization such as stemming was applied. Queries are processed similar to the documents, we use either the CO query or the CAS query, and remove query operators (if present) from the CO query and the about-functions in the CAS query.

2.3 Retrieval Model

Our retrieval system is based on the Lucene engine with a number of home-grown extensions [7, 10].

For the Ad Hoc Track, we use a language model where the score for an element e given a query q is calculated as:

$$P(e|q) = P(e) \cdot P(q|e) \quad (1)$$

where $P(q|e)$ can be viewed as a query generation process—what is the chance that the query is derived from this element—and $P(e)$ an element prior that provides an elegant way to incorporate link evidence and other query independent evidence [6, 9].

We estimate $P(q|e)$ using Jelinek-Mercer smoothing against the whole collection, i.e., for a collection D, element e and query q:

$$P(q|e) = \prod_{t \in q} \left((1 - \lambda) \cdot P(t|D) + \lambda \cdot P(t|e) \right), \tag{2}$$

where $P(t|e) = \frac{\text{freq}(t,e)}{|e|}$ and $P(t|D) = \frac{\text{freq}(t,D)}{\sum_{e' \in D} |e|}$.

Finally, we assign a prior probability to an element e relative to its length in the following manner:

$$P(e) = \frac{|e|^{\beta}}{\sum_{e} |e|^{\beta}}, \tag{3}$$

where $|e|$ is the size of an element e. The β parameter introduces a length bias which is proportional to the element length with $\beta = 1$ (the default setting). For a more thorough description of our retrieval approach we refer to [15]. For comprehensive experiments on the earlier INEX data, see [12].

2.4 Link Evidence as Document Priors

One of our aims for the Ad Hoc Track this year was to investigate the effectiveness of using link evidence as an indicator of relevance. We have chosen to use the link evidence priors to rerank the retrieved elements, instead of incorporating it directly into the retrieval model.

In the official runs, we have only looked at the number of incoming links (indegree) per article. Incoming links can only be considered at the article level, hence we apply all the priors at the article level, i.e., all the retrieved elements from the same article are multiplied with the same prior score. We experimented with *global* indegree, i.e., the number of incoming links from the entire collection, and *local* indegree, i.e., the number of incoming links from within the subset of articles retrieved for one topic. Although we tried global and local indegree scores separately as priors, we limit our discussion to a weighted combination of the two degrees, as this gave the best results when we tested on the 2006 topics. We compute the link degree prior $P_{\text{LocGlob}}(d)$ for an article d as:

$$P_{\text{LocGlob}}(d) \propto 1 + \frac{\text{Indegree}_{\text{Local}}(d)}{1 + \text{Indegree}_{\text{Global}}(d)} \tag{4}$$

Since the local indegree of an article is at most equal to the global indegree (when all the articles pointing to it are in the subset of retrieved articles), $P_{\text{LocGlob}}(d)$ is a number between 1 and 2. This is a much more conservative prior than using the indegree, local or global, directly. We will, for convenience, refer to the link evidence as prior, even though we do not actually transform it into a probability distribution. Note that we can turn any prior into a probability distribution by multiplying it with a constant factor $\frac{1}{\Sigma_{d \in D} prior(d)}$, leading to the same ranking.

3 Ad Hoc Retrieval Results

This year, there was no official Thorough task. The remaining tasks were the same as last year: Focused, Relevant in Context and Best in Context. To get CAS runs, we use a filter over the CO runs, using the pool of target elements of all topics. If a tag X is a target element for a given topic, we treat it as target element for all topics. We pool the target element tags of all topics, resulting in the following tags (by decreasing frequency): `<article>`, `<section>`, `<figure>`, `<p>`, `<image>`, `<title>`, and `<body>`. Then, we filter out all other elements from the results list of each topic. In other words, a retrieved element is only retained in the list if it is a target element for at least one of the topics.

For the Focused Task, no overlapping elements may be returned. For the Relevant in Context Task, all retrieved elements must be grouped per article, and for the Best in Context Task only one element or article offset may be returned indicating the best point to start reading. However, since both our indexes contain overlapping elements, the initials runs might contain overlapping results.

The link degrees in the official runs where erroneous, so we report on updated versions of the official runs, where only the degrees are different. We used the following Thorough runs as base runs for the various tasks:

– `element`: a standard *element* index run, with $\beta = 1$ and $\lambda = 0.15$.
– `contain`: a standard *contain* index run, with $\beta = 1$ and $\lambda = 0.15$.

where

– `+link` means the elements of the top 100 articles are reranked using the link prior.
– `+pool` means the run is filtered on the pool of target elements from the CAS queries.

3.1 Focused Task

To ensure the Focused run has no overlap, it is post-processed by a straightforward list-based removal strategy. We traverse the list top-down, and simply remove any element that is an ancestor or descendant of an element seen earlier in the list. For example, if the first result from an article is the article itself, we will not include any further element from this article.

Table 2 shows the results for the Focused Task. Looking at the two base runs first, we see that the *element* run scores better on very early precision, but loses out on the *contain* run at higher recall levels. With many smaller elements in the index it finds many relevant `<collectionlink>` elements which, due to their small size add little to recall, but are wholly relevant, thus leading to high precision. If a relevant `<collectionlink>` element is retrieved, any relevant ancestor nodes are not allowed in the results list, making it hard to improve recall with other element from that article. The *element+link* run scores best on very early

Table 2. Results for the Ad Hoc Track Focused Task

Run	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
element	0.5672	0.4599	0.3137	0.2339	0.0707
element+link	**0.5999**	0.4745	0.3321	0.2753	0.0850
element+pool	0.5287	0.4705	0.3547	0.2729	0.0916
element+pool+link	0.5337	0.4779	0.3624	02938	0.1048
contain	0.5371	0.4728	0.3545	0.2952	0.0956
contain+link	0.5541	**0.4949**	0.3746	0.3156	0.1117
contain+pool	0.5289	0.4774	**0.3749**	0.2974	0.1011
contain+pool+link	0.5309	0.4821	0.3734	**0.3173**	**0.1157**

precision. The link prior clearly moves relevant elements to the top of the results list and shows a consistent improvement over the base run. For the *element* run, the pool filter has a huge impact, filtering out all the `<collectionlink>` and many other small elements, so that after the subsequent list based overlap removal, the relevant ancestors of these small elements are retained. The pool of target elements is very small. The only elements that are mentioned as target elements in this years CAS topics are `<article>`, `<body>`, `<section>`, `<p>`, `<figure>`, `<image>` and `<title>`. The *contain* index has only the larger elements and `<name>` elements, making the pool filter much less effective, although it still has a positive effect on overall precision. The combination of link evidence and structural hints improves matters further. Although not effective at the highest ranks (iP[0.00]), it consistently improves on all three runs; base, link and pool, further down the results list.

3.2 Relevant in Context Task

For the Relevant in Context task, we use the Focused runs and cluster all elements belonging to the same article together, and order the article clusters by the highest scoring element. Table 3 shows the results for the Relevant in Context Task. Comparing the two base runs, the elements in the *contain* run match the relevant text within articles much better than those in the *element* run. Given the better early precision of the *element* run, the larger elements in the *contain* run have more relevant text. The link prior here improves the article

Table 3. Results for the Ad Hoc Track Relevant in Context Task

Run	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
element	0.1805	0.1566	0.1232	0.0891	0.0770
element+link	0.1838	0.1584	0.1216	0.0860	0.0814
element+pool	0.2373	0.2037	0.1523	0.1197	0.1117
element+pool+link	0.2336	0.2048	0.1529	0.1221	0.1125
contain	0.2156	0.1882	0.1484	0.1181	0.1066
contain+link	0.2315	0.1966	0.1504	0.1174	0.1085
contain+pool	**0.2497**	0.2069	0.1576	0.1239	0.1177
contain+pool+link	0.2456	**0.2144**	**0.1584**	**0.1271**	**0.1191**

Table 4. Results for the Ad Hoc Track Best in Context Task

Run	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
element	0.2089	0.2048	0.1673	0.1291	0.1194
element+link	0.2334	**0.2283**	**0.1804**	0.1348	**0.1316**
element+pool	0.2373	0.2193	0.1684	0.1323	0.1232
element+pool+link	**0.2423**	0.2218	0.1712	0.1364	0.1238
contain	0.2075	0.2060	0.1700	0.1356	0.1243
contain+link	0.2319	0.2212	0.1710	0.1356	0.1273
contain+pool	0.2304	0.2140	0.1693	0.1360	0.1283
contain+pool+link	0.2343	0.2246	0.1729	**0.1388**	0.1297

ranking of the early ranks, but after 25 articles (50 in the *contain* run) the base run is better. Recalling that the link prior showed consistent improvement in precision, it seems that it pushes the articles with less relevant text up in the ranking. The pool filtered runs show consistent improvement over the base runs, especially for the *element* runs and for the first 5 retrieved articles. By Filtering out the smaller elements, the *element* run retains much more relevant text after overlap removal. Reranking the pool filtered run using the link prior further boosts scores. In contrast to the effect of the link prior on the base runs, for the pool filtered CAS runs, after rank 5, the article ranking of both runs improves. This could be explained by the pool filtered runs having a better article ranking than the base runs, and thus more relevant articles in the local link graph.

3.3 Best in Context Task

The aim of the Best in Context task is to return a single result per article, which gives best access to the relevant elements. Table 4 shows the results for the Best in Context Task. The two base runs show similar performance. The link prior has a huge impact on the article ranking—the link prior only affects the article ranking of runs—of both base runs up to rank 10. After that, it still has a positive effect on the *element* run, but almost no effect on the *contain* run. We see a similar effect with the pool filtered CAS runs. The biggest impact is on the first 10 results. Combining the pool filter and the link prior leads to a further improvement in early precision and seems to be more effective for the *contain* run than the *element* run. To summarise, link evidence and structural hints are both effective for improving early precision for both base runs and are complementary to some extent.

4 Link Detection Experiments

In this section, we discuss our participation in the Link The Wiki (LTW) track. LTW is aimed at detecting missing links between a set of topics, and the remainder of the collection, specifically detecting links between an origin node and a destination node. Existing links in origin nodes were removed from the 90 topics, making these articles 'orphans.' The task was to detect these links again and

Table 5. Statistics of Types of Links in the 90 un-orphaned LTW articles

			All		**Link in Article**	
Type	**Uniq**	**Total**	**1×**	**Max**	**1×**	**Max**
`<collectionlink>`	5,786	8,868	4,275	51	5,781	15
`<unknownlink>`	1,308	1,458	1,201	14	1,271	7
`<outsidelink>`	807	851	772	5	778	5
`<imagelink>`	197	212	195	15	197	15
`<languagelink>`	79	1,147	12	66	1,147	1
`<wikipedialink>`	59	60	58	2	58	1
`<weblink>`	27	28	26	2	26	2
Total	8,263	12,624	6,513	-	9,232	-

find the correct destination node ('fosters'), thus detecting links both on element and article level.

There are several types of links in the topics. These links have been implemented in the Wikipedia collection using XLink. An overview of the occurrence of these types of links in the un-orphaned (original) topics is presented in Table 5. For example, if we regard all the links as one distribution, then the `<languagelink>` has 79 different types (appearing once), but the same types are used 1147 times, of which the single link `<languagelink lang="de">` is used as often as 66 times, which means the same language links are reused in the articles. When we look at each file separately, then a language link appears only once in a file.

For the LTW task, three type of links are used for detection: `<collection-link>`, `<wikipedialink>`, and `<unknownlink>`. The `<collectionlink>` comprises of the bulk of the links in the orphaned articles (70.0%). When looking at all orphaned articles, there are 5,786 unique type of collection links, out of the total of 8,868. The number of collection links that only occurs once is 4,275, which is 73.9% of the different types of collection links, and 48.2% out of all collection links. The collection link to article 35524.xml is occurring most often: 51 times, but it surprisingly does not to exist in the 2007 collection that we used. When we look at the links in the files separately, then 5,781 of the 8,868 collection links appear only once (65.2%), an outlier is the collection link 10829.xml (*"Florida"*), which is occurring 15 times in the topic 150340.xml (*"Miss Universe"*). On average, there are 98.5 *outgoing* collection links per topic, of which 64.3 per topic are unique, thus occurring once.

We also found that there is a significant strong positive relationship between the length of a Wikipedia article (excluding structure) and the number of links appearing in that article (Spearman's rho = 0.85, $p < 0.01$), i.e. longer articles have more links than shorter articles. Moreover, the average length of an anchor text is 12.3 characters, only 62 (0.7%) collection links are 3 characters or shorter.

4.1 Approach

Information Retrieval methods have been employed to automatically construct hypertext on the Web [1, 2], as well for specifically discovering missing links in Wikipedia [4]. To detect whether two nodes are implicitly connected, it is necessary to search the Wikipedia pages for some text segments that both nodes share. Usually it is only one specific and extract string [1]. Our approach is mostly based on this assumption, where we defined one text segment as a single line, and a string that both nodes share is a relevant substring. A substring of a string $T = t_1 \ldots t_n$ is a string $\hat{T} = t_{i+1} \ldots t_{m+i}$, where $0 \leq i$ and $m + i \leq n$. Only relevant substrings of at least 3 characters length are considered in our approach, because anchor texts of 3 characters or less do not occur frequently, and to prevent detecting too many false positives.

We also assume that pages that link to each other are somehow related in text content. We adopt a *breadth m–depth n* technique for automatic text structuring for identifying candidate anchors and text node, i.e. a fixed number of documents accepted in response to a query and fixed number of iterative searches. So the similarity on the document level and text segment level is used as evidence. The latter is used as a precision filter. So our approach consisted of two steps:

1. First, we detect links on the article level. We focus on the global similarity by collecting a set of similar or related pages using the set of topics. We search in the collection by retrieving the top N similar documents by using the whole document (including stopwords, stemmed with Porter stemmer, no XML structure) as a query against the index of the Wikipedia collection. We use the Vector Space Model (VSM) to retrieve related documents (articles). Our vector space model is the default similarity measure in Lucene [10], i.e., for a collection D, document d and query q:

$$sim(q,d) = \sum_{t \in q} \frac{tf_{t,q} \cdot idf_t}{norm_q} \cdot \frac{tf_{t,d} \cdot idf_t}{norm_d} \cdot coord_{q,d} \cdot weight_t, \tag{5}$$

   where $tf_{t,X} = \sqrt{\text{freq}(t,X)}$; $idf_t = 1 + \log \frac{|D|}{\text{freq}(t,D)}$; $norm_q = \sqrt{\sum_{t \in q} tf_{t,q} \cdot idf_t{}^2}$; $norm_d = \sqrt{|d|}$; and $coord_{q,d} = \frac{|q \cap d|}{|q|}$.
2. Second, we detect links on the element level. We search on the local level with text segments. Normalized lines (lower case, removal of punctuation and trailing spaces) are matched with string processing. At the same time we parse the XML and keep track of the absolute path for each text node and calculate the starting and end position (offset) of the identified anchor text by looking up the index of the string. For all our official runs, we blindly select the first instance of a matching line, and continue with the next line so an anchor text only has one link.

INEX LTW Task focuses on structural links, which have an anchor and refers to the Best Entry Point of another page (on the element level). Our Best Entry Point for both incoming and outgoing links was the start of an article, or */article[1]/name[1]* element, because in the current Wikipedia, links often point directly to entire articles or sections of these articles as logical units.

We do not assume that links are reciprocal, so we have different approaches for detecting outgoing and incoming links, though we set a threshold of 250 for both type of links and do not allow duplicated links as requested in the LTW task specification.

Detecting Outgoing Links. This is a link from an anchor text in the topic file to the Best Entry Point of existing related articles, which in our case was always the text-node of the */article[1]/name[1]* element. There is an outgoing link for topic t, when $S_{1 \cdots n} = T_{q \cdots r}$, where S is the title of a foster article, and T is a line in a orphan article.

Detecting Incoming Links. This type of link consists of a specified path expression (anchor) from text nodes in the target articles to the */article[1]/name[1]* node of one of the 90 topics. There is an incoming link for topic t, when $T_{1 \cdots n} = S_{q \cdots r}$, where $T_{1 \cdots n}$ is the title of t, and S is a line in a foster article.

Links also appear locally within an article to improve navigation on that page, but this was outside the scope of the LTW track. We extract for each topic the title enclosed with the `<name>` tag with a regular expression and store that in a hash-table for substring matching. We do not apply case-folding, but we do remove any existing disambiguation information put between brackets behind the title, e.g. *"What's Love Got to Do with It (film)"* becomes the substring *"What's Love Got to Do with It."*

4.2 Link the Wiki Track Findings

For the evaluation, only article-to-article links are considered in the scores. The threshold for the number of incoming and outgoing links was each set to 250 for each topic. Table 6 shows the mean number and range of incoming and outgoing links. For all runs there were more incoming links than outgoing links. Compared to the frequencies of the original articles as depicted in Table 5, we seem to have under-generated the number of outgoing links. This is a limitation of the Vector Space Model, as links in Wikipedia do not always relate to textually related or similar documents. It also shows that as we increase the pool of candidate target pages retrieved with the VSM (top 100, 200, 250, 300, 400), the number of detected links is also increased for incoming and outgoing links. However, this does not mean necessarily that retrieval performance is also improved as Table 7 shows. We achieved best performance by setting the threshold of the result list to the top 300 ($\text{MAP}_{in} = 0.3713$). The run name400 stands for the top 400 of the hit list retrieved with the VSM, and with name-matching post-processing. It shows that while the recall improves, which has slight positive effect on the performance for the outgoing links, the precision drops and thus the fallout also increases for the incoming links.

In summary, we experimented with the Vector Space Model and substring match for detecting missing links in Wikipedia. We used entire orphaned articles as query. We showed that exact substring matching improves the performance as compared to generating plain article-to-article links. This approach worked well, especially for the early precision. Our assumption that pages that link to each

Table 6. Results Link The Wiki: Number of Outgoing and Incoming Links

	Outgoing					Incoming				
Run	**Mean**	(SD)	Min	Median	Max	**Mean**	(SD)	Min	Median	Max
Name100	12.80	(7.58)	2	11	35	38.40	(34.67)	0	24.5	100
Name200	16.63	(10.39)	2	14	55	62.09	(65.49)	0	33.5	200
Name250	17.83	(11.40)	2	14.5	65	72.10	(79.84)	0	35.5	250
Name300	19.08	(12.52)	2	16	74	77.77	(85.97)	0	38.5	250
Name400	22.72	(14.78)	3	19	83	82.34	(90.26)	0	37.5	250

Table 7. Results for the Link The Wiki Track

	Outgoing				Incoming			
Run	**MAP**		**R-Prec**	**P@5**	**MAP**		**R-Prec**	**P@5**
Article100	0.1518		0.2277	0.5711	0.2646		0.3062	0.7311
Name100	0.1533	+1.0%	0.1781	0.7489	0.2906	+9.8%	0.3134	0.8000
Article200	0.1629		0.2389	0.5711	0.3075		0.3529	0.7311
Name200	0.1739	+6.8%	0.2073	0.7356	0.3471	+12.9%	0.3835	0.8044
Article250	0.1658		0.2406	0.5711	0.3193		0.3628	0.7311
Name250	0.1783	+4.9%	0.2147	0.7267	0.3618	+13.3%	0.3998	0.8044
Article300	0.1678		0.2407	0.5711	0.3274		0.3691	0.7311
Name300	0.1825	+8.8%	0.2233	0.7178	**0.3713**	+13.4%	0.4101	0.8044
Name400	**0.1836**		0.2405	0.6844	0.3117		0.3757	0.6067

other are related or similar in content may not necessarily hold, thus reducing the pool of relevant pages that can be linked. Our experiments focused on exact string matching, but we have not explored yet techniques with best matching of substrings, e.g. semantic clustering of words, which could further improve the performance.

5 Interactive Experiments

In this section, we discuss the interactive experiments of the INEX 2006 Interactive Track (which has run well into INEX 2007). For details about the track and set-up we refer to [11]. For the interactive track, we conducted an experiment where we took part in the concerted effort of Task A, in which we compare element and passage retrieval systems. We reported the result of the track based on the users responses on their searching experience for each task and comparative evaluation on the element and passage retrieval systems. The element and passage retrieval systems evaluated are developed in a java-based retrieval system built within the Daffodil framework by the track organizers.

We participated in task A with nine test persons in which seven of them completed the experiment. Two persons failed to continue the experiment due to systems down time. Each test person worked with four simulated tasks in the Wikipedia collection. Two tasks were based on the element retrieval and the other two tasks were based on the passage retrieval. The track organizer provided a multi-faceted set of 12 tasks in which the test person can choose from. The 12 tasks consist of three task types (decision making, fact finding and information gathering) which further slit into two structural kinds (hierarchical and parallel). The experiment was conducted in accordance with the track guideline.

5.1 Post Task Questionnaire

For each task, each test person filled in questionnaires before and after each tasks, and before and after the experiment, resulting in 70 completed questionnaires. The questionnaire focuses on the users' searching experience with the systems and the usefulness of system features. Table 8 shows the post task questionnaire.

Table 9 shows the responses for the post-task questionnaire. First, we look at the result for all tasks. We found that the test persons were positive regarding both systems. Next, we look at responses for the element and passage system, without considering the task types and structures. We found that the element system is rated higher in terms of the amount of time used (Q2), certainty of completing the task (Q3), easiness of task (Q4), and satisfaction (Q5). As for the experience rate (Q1) and the usefulness of presentation (Q6), the passage retrieval system is rated higher.

Furthermore, we asked the test persons to rate the usefulness of system features. The answer categories used a 5-point scale with 1=not at all useful, 3=somewhat, and 5=extremely useful. When asked what helped the test persons in their searching tasks, five persons mentioned the table of content. The usefulness of table of content was rated at 3.90. The reasons were the table of content was detailed and it gave a good overview of the document. As one person noted, "the table of content was useful to get the overview of the document

Table 8. Post-task questionnaire

Q1 How would you rate this experience?
(1=frustrating, 3=neutral, 5=pleasing)
Q2 How would you rate the amount of time available to do this task?
(1=much more needed, 3=just right, 5=a lot more than necessary)
Q3 How certain are you that you completed the task correctly?
(For Q3 until Q6, 1=not at all, 3=somewhat, 5=extremely)
Q4 How easy was it to do the task?
Q5 How satisfied are you with the information you found?
Q6 To what extent did you find the presentation format (interface) useful?

Table 9. Post-task responses on searching experience: mean scores and standard deviations (in brackets)

	Q1	Q2	Q3	Q4	Q5	Q6
All tasks	3.11 (1.45)	3.63 (1.28)	3.30 (1.32)	3.30 (0.99)	3.33 (1.21)	3.48 (0.70)
Element	2.93 (1.44)	3.64 (1.22)	3.43 (1.22)	3.36 (1.01)	3.36 (1.22)	3.43 (0.76)
Passage	3.31 (1.49)	3.62 (1.39)	3.15 (1.46)	3.23 (1.01)	3.31 (1.25)	3.54 (0.66)

and it helped me to get back to where I want (by clicking on it)." However, the table of content was not so appreciated in short documents. For example, one person noted, "the table of content is useful, but to go through document I was only scrolling, because the text was not too long." Another reason why the table content was not useful in short documents is because in passage retrieval often the table of content only consisted of one item, thus the table seemed to be useless.

Four test persons mentioned result list helped them in their searching tasks and the test persons rated the usefulness of result presentation at an average of 3.90. The result list provided the test persons with detailed information of relevant paragraphs. However, we found out that result list was not sufficient enough because in some cases it returned too many irrelevant document. As one person who noticed the problem noted, "the ranking was poor, but I could immediately reject non-relevant result based on the shown information."

Paragraph highlighting was rated at an average of 3.04. Two test persons mentioned that paragraph highlighting was useful while two other test persons mentioned it as not useful. The reason of the mix-answers was because sometimes the relevant information was not highlighted by the system. As one person noted, "I did not find the paragraph highlighting useful since I found the relevant information at the non-highlighted passages."

Links in the document were appreciated by the test persons. It is observed that sometimes the system did not return relevant documents, thus the test persons just clicked the available links and found the relevant information through the links in the document. As mentioned by one person, "the most relevant pages (the general description of castle and fortress page) were not on the result list, but I just found them by clicking the links in the document."

Related terms function was rated the least with an average of 0.8. Six test persons commented that they did not use the related terms at all. We found out that the related terms provided by the system were not useful for the tasks because they were too long and not relevant.

5.2 Post Experiment Questionnaire

After each completed task, the test persons filled in a post-experiment questionnaire on ease of use and ease of learning of element and passage systems. The answer categories used a 5-point scale with 1=not at all, 3=somewhat, and 5=extremely. With respects to ease of use, element retrieval is rated higher ($\underline{M}$= 4.29, $\underline{SD}$=0.488) then passage retrieval ($\underline{M}$=3.86, $\underline{SD}$=0.90). Also with respect to ease of learning, element retrieval is rated higher ($\underline{M}$=4.14, $\underline{SD}$= 0.378) then passage retrieval ($\underline{M}$= 3.86, $\underline{SD}$=0.69).

We can see that there is a tendency to favor element retrieval system. This also shown by the answers of the post experiment questionnaire where the test persons were more positive for the element retrieval system. In comparison between the two systems, the element retrieval system seemed to give a more complete table of content compare to the passage retrieval system, resulting a better overview to see the relations between sections. Furthermore, the result list in the passage system seemed to give a poorer result in the result list and in some cases it missed the relevant documents.

5.3 Interactive Track Findings

From the result of the experiment, we focus on the comparison of element and passage retrieval systems and the usefulness of system features. From the quantitative result, we discovered that test persons appreciated both systems positively and found only small difference between element and passage retrieval systems. Passage retrieval seemed favorable in post-task questionnaire but element retrieval was rated higher in the comparative questions. However, it is too early to conclude that element retrieval is better then passage retrieval on this experiment. Because our finding is based on a small user test that only involved seven test persons. Furthermore, the system performance was slow and we think that this might influence our result. Over the whole experiment, perhaps the most striking result is that in the beginning of the post-experiment questionnaire, two test persons did not notice the differences between element and passage systems at all. They started to notice the differences after they were presented screenshots of both systems. In addition, table of content and result list were found to be the most useful features of the system. The test persons argued that the content of table gave them a good overview for long documents and the result list provided them with detailed information about the document. The least appreciated feature of the system was the related terms feature. From the comments we found out that the related terms did not help the test persons because the terms were too long and not relevant.

6 Discussion and Conclusions

In this paper, we documented our efforts at INEX 2007 where we participated in the Ad hoc Track, the Link the Wiki Track, and the Interactive Track that continued from INEX 2006.

For the Ad Hoc Track, we investigated the effectiveness of incorporating link evidence into the model, and of a CAS filtering method exploiting the structural hints in the INEX topics. We found that link priors improve our base runs, especially in early precision, for all tasks. The CAS pool filtering method is effective for all three tasks as well, showing more consistent improvement throughout the results lists. Combining the two methods improves performance further.

For the Link the Wiki Track, we investigated the relative effectiveness of link detection based on the VSM by using an entire article as a query. First, we established article-to-article links. We continued by detecting links on the element level by filtering with the names of Wikipedia articles. We show that name-filtering the results obtained with the VSM improves the precision. We achieved best performance by setting the threshold to 300.

For the Interactive Track, we took part in the interactive experiment comparing an element retrieval system with a passage retrieval system. Our test-persons showed a weak preference for the element retrieval system over the passage retrieval system. Of course, due to its small scale the study warrant general conclusions on the usefulness of passage-based approaches in XML retrieval.

Acknowledgments. Jaap Kamps was supported by the Netherlands Organization for Scientific Research (NWO, grants # 612.066.302, 612.066.513, 639.072.601, and 640.001.501), and by the E.U.'s 6th FP for RTD (project MultiMATCH contract IST-033104). Marijn Koolen was supported by NWO under grant # 640.001.501. Khairun Nisa Fachry and Junte Zhang were supported by NWO under grant # 639.072.601.

References

[1] Agosti, M., Crestani, F., Melucci, M.: On the use of information retrieval techniques for the automatic construction of hypertext. Information Processing and Management 33, 133–144 (1997)
[2] Allan, J.: Building hypertext using information retrieval. Information Processing and Management 33, 145–159 (1997)
[3] Denoyer, L., Gallinari, P.: The Wikipedia XML Corpus. SIGIR Forum. 40, 64–69 (2006)
[4] Fissaha Adafre, S., de Rijke, M.: Discovering missing links in wikipedia. In: LinkKDD 2005: Proceedings of the 3rd international workshop on Link discovery, pp. 90–97. ACM Press, New York (2005)
[5] Hawking, D., Craswell, N.: Very large scale retrieval and web search. In: TREC: Experiment and Evaluation in Information Retrieval, ch. 9, pp. 199–231. MIT Press, Cambridge (2005)
[6] Hiemstra, D.: Using Language Models for Information Retrieval. PhD thesis, Center for Telematics and Information Technology, University of Twente (2001)
[7] ILPS. The ILPS extension of the Lucene search engine (2007), `http://ilps.science.uva.nl/Resources/`
[8] Kamps, J., Koolen, M., Sigurbjörnsson, B.: Filtering and clustering XML retrieval results. In: Fuhr, N., Lalmas, M., Trotman, A. (eds.) INEX 2006. LNCS, vol. 4518, pp. 121–136. Springer, Heidelberg (2007)
[9] Kraaij, W., Westerveld, T., Hiemstra, D.: The importance of prior probabilities for entry page search. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 27–34. ACM Press, New York (2002)
[10] Lucene. The Lucene search engine (2007), `http://lucene.apache.org/`
[11] Malik, S., Tombros, A., Larsen, B.: The interactive track at INEX 2006. In: Fuhr, N., Lalmas, M., Trotman, A. (eds.) INEX 2006. LNCS, vol. 4518, pp. 387–399. Springer, Heidelberg (2007)
[12] Sigurbjörnsson, B.: Focused Information Access using XML Element Retrieval. SIKS dissertation series 2006-28, University of Amsterdam (2006)
[13] Sigurbjörnsson, B., Kamps, J.: The effect of structured queries and selective indexing on XML retrieval. In: Fuhr, N., Lalmas, M., Malik, S., Kazai, G. (eds.) INEX 2005. LNCS, vol. 3977, pp. 104–118. Springer, Heidelberg (2006)

[14] Sigurbjörnsson, B., Kamps, J., de Rijke, M.: An Element-Based Approach to XML Retrieval. In: INEX 2003 Workshop Proceedings, pp. 19–26 (2004)
[15] Sigurbjörnsson, B., Kamps, J., de Rijke, M.: Mixture models, overlap, and structural hints in XML element retreival. In: Fuhr, N., Lalmas, M., Malik, S., Szlávik, Z. (eds.) INEX 2004. LNCS, vol. 3493, pp. 196–210. Springer, Heidelberg (2005)
[16] Trotman, A., Geva, S.: Passage retrieval and other XML-retrieval tasks. In: Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology, pp. 43–50. University of Otago, Dunedin New Zealand (2006)
[17] Wikipedia. The free encyclopedia (2006), `http://en.wikipedia.org/`