The University of Amsterdam at INEX 2007

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 the Wikipedia article's name only, and on the matching arbitrary text segments of different pages. 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 name matching works best, and can still be expanded and fine-tuned to achieve better 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

Table 1. Relevant passage statistics

Description	Statistics 2006	2007
# topics	114	99
# articles with relevance	5,648	6,042
# relevant passages	9,083	10,818
mean length relevant passage	1,090	944
median length relevant passage	297	272

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. 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 concerted experiment compare an element retrieval system to a passage retrieval system [11]. 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. 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 detailing 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, 99 topics were assessed, with 60.48 relevant articles and 108.39 relevant passages per topic. The average number of relevant passages per relevant articles is 1.61 for the 2006 topics and 1.79 for the 2007 topics. On the other hand, the size of the relevant passages this year has decreased compared to last year. Both average (948) and 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.
- *Article index*: We also build an index containing all full-text articles (i.e., all wikipages) as is standard in IR.

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 a element e given a query q is calculated as:

$$P(e|q) = P(e) \cdot P(q|e) \tag{1}$$

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].

For our Link the Wiki Track runs, we use a vector-space retrieval model. Our vector space model is the default similarity measure in Lucene [10], i.e., for a collection D, document d and query q:

$$sim(q,d) = \sum_{t \in q} \frac{tf_{t,q} \cdot idf_t}{norm_q} \cdot \frac{tf_{t,d} \cdot idf_t}{norm_d} \cdot coord_{q,d} \cdot weight_t, \tag{4}$$

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 $coord_{q,d} = \frac{|q \cap d|}{|q|}$.

2.4 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:

$$P_{\text{LocGlob}}(d) \propto 1 + \frac{\mathsf{Local}_{\mathsf{In}}(d)}{1 + \mathsf{Global}_{\mathsf{In}}(d)}$$

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. 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.

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.

We used the following runs Thorough runs as base runs for the various tasks.

- `inex07_contain_beta1_thorough_cl` a standard *contain* index run, with $\beta = 1$ and $\lambda = 0.15$.
- `inex07_contain_beta1_thorough_clp_10000_cl` like the previous run, but reranked over all 10,000 results using the conservative link prior.
- `inex07_contain_beta1_thorough_cl_pool_filter` a CAS version of the standard run, where only the pool of target elements are retained.
- `inex07_contain_beta1_thorough_clp_10000_cl_pool_filter` a CAS version of the conservatively reranked run.
- `inex07_element_beta1_thorough_clp_10000_cl` a standard *element* index run, reranked using the conservative link prior.
- `inex07_element_beta1_thorough_clp_10000_cl_pool_filter` the CAS version of the previous run.

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. Results for the Ad Hoc Track Focused Task (runs in emphatic are no official submissions)

Run	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
element_beta1_focused	0.4662	**0.4126**	**0.3837**	**0.3621**	**0.2621**
element_beta1_focused_cas_pool_filter	0.4409	0.4029	0.3676	0.3476	0.2544
element_beta1_focused_clp_10000_cl	**0.4780**	0.3938	0.3236	0.2974	0.1326
element_beta1_focused_clp_10000_cl_cas_pool_filter	0.4261	0.3723	0.3108	0.2771	0.1210
contain_beta1_focused_cl	0.4505	0.3837	0.3201	0.2959	0.1324
contain_beta1_focused_cl_cas_pool_filter	0.4230	0.3779	0.3181	0.2885	0.1302
contain_beta1_focused_clp_10000_cl	0.4493	0.3865	0.3224	0.2957	0.1352
contain_beta1_focused_clp_10000_cl_cas_pool_filter	0.4225	0.3787	0.3201	0.2872	0.1325

Table 2 shows the results for the Focused Task. The *element* run scores higher than the *contain* run on all measures, which might be explained by the many smaller elements in the *element* index. The `<collectionlink>` element is by far the most frequently retrieved element throughout the result list. Since these elements contain only a few words, they add little to recall, but all relevant `<collectionlink>` elements are completely relevant, thus leading to high precision scores.

The CAS filter has a negative effect on the scores, for both the *element* and *contain* runs. 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>`. Clearly, some relevant elements are removed by the filter. Also on the link prior runs, the CAS filter has a negative effect.

The link evidence helps in boosting relevant elements to the top ranks for the *element* run, leading to an improvement of early precision (iP[0.00]), but further down the list, precision drops rapidly. For the *contain* run, link evidence has a very small positive effect for iP[0.01], iP[0.05] and MAiP. The link prior has a clustering effect, pushing elements with a low retrieval score but with a high link indegree above elements with a higher retrieval score but a lower link indegree. The top ranked elements are often from articles with a lot of relevance, thus lower scoring elements from the same article have a high probability of containing relevance as well, leading to an improvement in early precision. But for articles with little relevance, this clustering effect might have a negative effect, since the high scoring elements of such articles contain most of the relevance and pushing up low scoring elements from those articles hurts precision.

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. Again, the standard *element* run scores better than the standard *contain* run. If we look at the different cut-offs, we see that the difference between the two runs becomes smaller. However, the *element* run also has a higher

Table 3. Results for the Ad Hoc Track Relevant in Context Task (runs in emphatic are no official submissions)

Run	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
element_beta1_ric_hse	0.2009	0.1775	0.1282	0.0951	0.0905
element_beta1_ric_hse_cas_pool_filter	**0.2227**	**0.1784**	**0.1366**	**0.1052**	**0.1003**
element_beta1_clp_10000_cl_ric_hse	0.1808	0.1508	0.1104	0.0811	0.0831
element_beta1_clp_10000_cl_cas_pool_filter_ric_hse	0.1704	0.1373	0.1000	0.0766	0.0761
contain_beta1_cl_ric_hse	0.1696	0.1440	0.1036	0.0822	0.0805
contain_beta1_cl_cas_pool_filter_ric_hse	0.1665	0.1370	0.1059	0.0801	0.0805
contain_beta1_clp_10000_cl_ric_hse	0.1732	0.1487	0.1086	0.0831	0.0860
contain_beta1_clp_10000_cl_cas_pool_filter_ric_hse	0.1683	0.1459	0.1069	0.0820	0.0846

Table 4. Results for the Ad Hoc Track Best in Context Task (runs in emphatic are no official submissions)

Run	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
element_beta1_bic_hse	0.2727	0.2623	0.2016	0.1601	0.1598
element_beta1_cas_pool_filter_bic_hse	0.3124	0.2749	0.2093	0.1647	0.1623
element_beta1_clp_10000_cl_bic_hse	0.3029	0.2690	0.2111	0.1645	0.1561
element_beta1_clp_10000_cl_cas_pool_filter_bic_hse	0.3192	0.2662	0.2026	0.1606	0.1456
contain_beta1_cl_bic_hse	0.2643	0.2552	0.1913	0.1537	0.1553
contain_beta1_cl_cas_pool_filter_bic_hse	0.3289	0.2807	0.2129	0.1647	0.1618
contain_beta1_clp_10000_cl_bic_hse	0.2816	0.2694	0.2123	0.1667	0.1684
contain_beta1_clp_10000_cl_cas_pool_filter_bic_hse	**0.3311**	**0.2906**	**0.2266**	**0.1775**	**0.1736**

MAgP score. This might be the effect of the length prior. Without the length prior, the *element* run would consist of many really small elements, which would give low recall. By adding a length prior, much larger elements, like `<article>`, `<body>` and `<section>` receive a higher score and give higher recall. However, some `<collectionlink>` elements still receive a high score, indicating that they contain many of the query terms, and can add to recall without losing precision.

For the CAS filter and link prior, we see the following. The CAS filter is effective for the standard *element* run, but not for the *contain* run. For the *element* run, the link prior has a negative effect, while on the *contain* run, it has a positive effect. The CAS filter is also not effective for the link prior runs.

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. Of the two base runs, the *element* run scores better on all measures. This is not surprising when looking at the results for the previously described tasks. The *element* scores consistently better in both the Focused and Relevant in Context tasks, although here the differences are smaller.

For the CAS filter and link prior, we see the following. The pool filter is especially effective for early precision. Where the link prior is effective for the first 50 ranks on both runs, it improves MAgP for the *contain* run, but hurts

MAgP for the *element* run. The combination of the pool filter and the link prior is less effective than the filter or link prior separately for the *element* run. For the *contain* run, the combination is more effective than the separate methods, and even outperforms the *element* runs.

4 Link the Wiki Track

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, in this case whole Wikipedia articles, and the task was to detect these links again and find the correct destination node. This year we submitted five official runs to the LTW Track, and one post-submission run. We describe our approach, our results based on the official qrels, and an analysis of the errors.

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 $T = t_1 \dots t_n$ is a string $\hat{T} = t_{i+1} \dots t_{m+i}$, where $0 \leq i$ and $m + i \leq n$. Only relevant substrings of at least 3 characters length are considered in our approach.

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 by looking at the similarity. This similarity can be evaluated in two dimensions: global similarity between an origin node and destination node where the whole document is used, and local similarity where only text segments are compared pairwise. The local similarity is used as a precision filter. To evaluate the global similarity between an orphan page and a target page, we used Lucene's Vector Space Model on an article index (see Section 2).

Global Similarity 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 100 similar documents by using the whole document as a query against the index of the Wikipedia collection without the topic files, but filtering with the English Snowball stopwords list for efficiency reasons. We also retrieved the top 100 similar documents for a topic by using top N terms derived from a language model as query.

Local Similarity We search on the local level with text segments. Normalized (lower case, removal of punctuation trailing spaces) lines 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 of the identified anchor text. 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 can only have one link.

The INEX LTW Track focuses on structural links, which have an anchor and refers to the Best Entry Point of another page. Our Best Entry Points are paths to the closest located elements that contain substrings which match with the specified anchor text, thus the deepest node. Anchors are identified with the element path and the offset. The LTW task consists of identifying outgoing and incoming links between the 90 topics and existing Wikipedia pages. We have not focused on local links within the topics.

Incoming Links This type of link consists of a specified XPath expression (anchor) from destination nodes in the target pages to the Best Entry Point (origin node) of one of the related 90 topics. Incoming links are detected by top-down processing the relevant related pages, and for each page iteratively do (partial) line-matching with all lines of that file with the lines of the topic.

Outgoing Links A link from an anchor in the topic file to the Best Entry Point of existing related pages. We iterate over all lines of the topic file, and (partially) match the lines top-down with candidate target files until a link has been detected for that line.

In the current Wikipedia, links only point directly to entire articles, thus the beginning or name of the page. The run LTW01 is based on this observation. In this run, we extract for each topic the title enclosed with the <NAME> tag with a regular expression and match that title with (substrings of) lines in the target files to identify incoming links. To retrieve outgoing links, we extract the names of the 100 target pages and iteratively match those titles with each line (substring) of the topic file until a link has been detected or if none has been found in the file. For run LTW01 the 100 related target files are retrieved for each topic by using that full topic as query.

The runs LTW02, LTW03, and LTW04 are based on identifying the local similarity between text segments with exact line matching, effectively only accepting a local similarity of 100% to improve precision. The purpose of these runs was to test the effect of the global similarity between documents on link detection using the full topic as query by building a Vector Space Model and the top N most relevant terms derived from a language model. The top 100 target files was selected for each of the 90 topics. For run LTW03 we used the full topic (excluding Snowball stopwords) as query. The top 10 terms is selected as query for run LTW03 and the top 25 for run LTW04.

The run LTW07 was completely experimental, where we explored the use of the Longest Common Substring ($LCSS$) and WordNet as anchor text expansion. The $LCSS$ between string S and string T is the longest substring that occurs

Table 5. Results Link The Wiki: Number of Links and Time

Run	$\overline{x}$ Incoming	$\overline{x}$ Outgoing	Time (s)
LTW01	86.1	43.8	169,225
LTW02	273.6	90.0	340,473
LTW03	243.1	83.9	154,732
LTW04	280.1	88.9	179,445
LTW07	312.6	176.9	55,216
LTW03′	231.6	94.0	106,449

both in S and T denoted by $LCSS(S,T)$. The lengths and starting positions of the longest common substrings of S and T can be found with the help of a generalised suffix tree. We have built such a tree for each pair of lines. The longest common suffix ($LCSuff$) is computed as

$$LCSuff(S_{1\ldots i}, T_{1\ldots j}) = \begin{cases} LCSuff(S_{1\ldots i-1}, T_{1\ldots j-1}) + 1 & \text{if } S[i] = T[j] \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

The longest common substrings of S and T must be the maximal of these longest common suffixes of possible prefixes.

$$LCSS(S,T) = \max_{1 \le i \le m, 1 \le j \le n} LCSuff(S_{1..i}, T_{1..j}) \quad (6)$$

We also expect that anchor texts do not always exactly match with the (sub)string destination node as links can be associative. To deal with this problem, we used a Perl module that looks up synonyms for a candidate anchor text in the lexical database WordNet, thus switching to a semantically equivalent substring that is to be matched with potential destination nodes. Stopwords were filtered to avoid these being matched as the longest common substring and thus as an anchor text.

4.2 Results

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, however, for LTW02, LTW03, LTW04 and LTW07 that threshold was unintentionally set outside the line matching iteration of a target file. Table 5 shows the mean of incoming and outgoing links. The time needed to generate the runs was also recorded. For all runs there were more incoming links than outgoing links. LTW07 was generated with the least time, but also had most number of links.

We show the scores for the runs in Table 6: (a) incoming links, (b) outgoing links, and (c) a combined score. The run LTW01 performed best overall, and LTW07 performed poorly. There is little difference between LTW02, LTW03, and LTW04. We have one post-submission LTW03′, which is the same as LTW03 but corrects the approach for incoming links set to reduce duplicated article-to-article links, and hence improves the result. However, the results show that restricting the partial line-matching to the names of Wikipedia pages performs best as expected.

Table 6. Results for the Link The Wiki Track

(a) Incoming links

Run	MAP	R-Prec	P5	P10	P20	P30	P50
LTW01	0.2264	0.2583	0.7022	0.6622	0.5767	0.5051	0.3920
LTW02	0.1085	0.1648	0.6600	0.5167	0.3267	0.2411	0.1571
LTW03	0.1096	0.1437	0.6222	0.5133	0.3644	0.2770	0.1827
LTW04	0.0927	0.1418	0.6400	0.4889	0.3317	0.2441	0.1591
LTW07	0.0039	0.0196	0.2378	0.1667	0.0883	0.0596	0.0358
LTW03′	0.1282	0.1755	0.6867	0.5978	0.4667	0.3767	0.2591

(b) Outgoing Links

Run	MAP	R-Prec	P5	P10	P20	P30	P50
LTW01	0.1377	0.1739	0.7844	0.6844	0.4844	0.3437	0.2073
LTW02	0.0803	0.1538	0.4667	0.4344	0.3517	0.2885	0.1958
LTW03	0.0733	0.1410	0.4778	0.4211	0.3472	0.2767	0.1789
LTW04	0.0806	0.1494	0.4978	0.4278	0.3517	0.2870	0.1882
LTW07	0.0671	0.1273	0.5000	0.4256	0.3206	0.2467	0.1500
LTW03′	0.0744	0.1467	0.4911	0.4122	0.3489	0.2867	0.1873

(c) Combined with F-Score

Run	MAP	R-Prec	P5	P10	P20	P30	P50
LTW01	0.1712	0.2079	0.7411	0.6731	0.5265	0.4091	0.2712
LTW02	0.0924	0.1591	0.5467	0.4720	0.3387	0.2627	0.1743
LTW03	0.0878	0.1423	0.5405	0.4626	0.3556	0.2769	0.1808
LTW04	0.0862	0.1455	0.5600	0.4563	0.3414	0.2638	0.1724
LTW07	0.0075	0.0339	0.3223	0.2395	0.1385	0.0960	0.0578
LTW03′	0.0941	0.1598	0.5727	0.4879	0.3993	0.3256	0.2175

4.3 Link the Wiki Track Findings

Our incoming links performed poorly. This year's evaluation is based on article-to-article links. We over-generated incoming links, while at the same time setting the threshold of incoming links at 250. Moreover, since we generated links as Best Entry Points into the target pages, we created too many duplicated article-to-article links, which hurt our performance. The exact line-matching (LTW02, LTW03, LTW04) does not perform well. The post-submission run improved the incoming links, but the results are still not satisfactory.

Our assumption that pages that link to each other are related or similar in content may not necessarily hold, thus reducing the pool of relevant pages that can be linked. The granularity of text segments as lines could work well, however, more context may be required to properly detect the local similarity between two nodes. LTW07 was technically most complicated, and performed worst. The reason was that the local similarity matching was not discriminative enough, a candidate link was too easily accepted, and thus both incoming and outgoing links were over-generated.

In summary, the results show that of our different approaches to detect links, name matching works best, and that this run should be expanded and fine-tuned to achieve better performance.

Table 7. 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?

5 Interactive Experiments

In this section, we discuss out 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 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. 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.

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 Experiment 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. Table 7 shows the post task questionnaire. Table 8 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. The fact that

Table 8. 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)

Table 9. Post-experiment responses on ease of use and learn: mean scores and standard deviations (in brackets)

	Ease of learning		Ease of use	
System 1: Element	4.29	(0.49)	4.14	(0.38)
System 2: Passage	3.86	(0.90)	3.86	(0.69)

element retrieval system is rated less pleasing then the passage retrieval while it is regarded as a more effective system (Q3, Q5) is rather surprising.

5.2 Post Experiment Questionnaire

After each completed task, the test persons filled in a post-experiment questionnaire. Table 9 shows the responses to questions on ease of using, and easy of learning. The answer categories used a 5-point scale with 1=not at all, 3=somewhat, and 5=extremely. With respects to ease of learning and ease of use of the systems, we found out that the test persons gave higher scores to element system than to passage system.

We can see that there is a tendency to favor the element retrieval system. This also shown by the answers of the post experiment questionnaire where the users were more positive for the element retrieval system. Furthermore, we also asked the test persons opinion about what they like and dislike about the search systems. In both systems all of the test persons appreciated the table of content. The table of content was detailed enough and gave a good overview of the document. They also think that detailed information on the result list, links to other document, term and paragraph highlighting, and document back and forward functions helped them during searching tasks. Almost all of the test persons complain about the performance of the system. They also claim that the result list sometimes gave to many irrelevant documents. In comparison between the two systems, the element system seemed to give a more complete table of content compare to the passage 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 document.

5.3 Interactive Track Findings

From the result of the experiment, we mainly focus on the comparison of element and passage retrieval systems. Although the users appreciated both systems positively, there is a tendency that the users prefer the element retrieval system

to the passage retrieval system. From the user tasks questionnaires we discovered that the element retrieval is considered more effective then the passage retrieval system. Furthermore, from the post experiment questionnaires we found that element retrieval system seems to provider a clearer overview of the document. 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 none of the users find any striking difference between element and passage system. Several users did not even notice the differences at all. In addition, table of content was found the most useful feature of the system. The table of content for both element and passage retrieval were rated positively by the users. They argue that the content of table gave them a good overview of the document. The least appreciated feature of the system was related terms. From the comment we found out that the related terms did not help the users because they are too long and often off-topics.

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 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, 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. Our results show that name matching works best, and can still be expanded and fine-tuned to achieve better performance. It is too early to conclude that more sophisticated approaches are ineffective, since the current evaluation was restricted to article-to-article links.

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, our small study does not warrant a general conclusion on the usefulness of passage-based approaches in XML retrieval. The technique may still be immature, or the system's responsive may be improved.

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.

Bibliography

[1] M. Agosti, F. Crestani, and M. Melucci. On the use of information retrieval techniques for the automatic construction of hypertext. *Information Processing and Management*, 33:133–144, 1997.

[2] J. Allan. Building hypertext using information retrieval. *Information Processing and Management*, 33:145–159, 1997.

[3] L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 40: 64–69, 2006.

[4] S. Fissaha Adafre and M. de Rijke. Discovering missing links in wikipedia. In *LinkKDD '05: Proceedings of the 3rd international workshop on Link discovery*, pages 90–97. ACM Press, New York NY, USA, 2005.

[5] D. Hawking and N. Craswell. Very large scale retrieval and web search. In *TREC: Experiment and Evaluation in Information Retrieval*, chapter 9, pages 199–231. MIT Press, 2005.

[6] D. Hiemstra. *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] J. Kamps, M. Koolen, and B. Sigurbjörnsson. Filtering and clustering XML retrieval results. In *Comparative Evaluation of XML Information Retrieval Systems: Fifth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2006)*, volume 4518 of *Lecture Notes in Computer Science*, pages 121–136. Springer Verlag, Heidelberg, 2007.

[9] W. Kraaij, T. Westerveld, and D. Hiemstra. 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*, pages 27–34. ACM Press, New York NY, USA, 2002.

[10] Lucene. The Lucene search engine, 2007. http://lucene.apache.org/.

[11] S. Malik, A. Tombros, and B. Larsen. The interactive track at INEX 2006. In *Comparative Evaluation of XML Information Retrieval Systems: Fifth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2006)*, volume 4518 of *Lecture Notes in Computer Science*, pages 387–399. Springer Verlag, Heidelberg, 2007.

[12] B. Sigurbjörnsson. *Focused Information Access using XML Element Retrieval*. SIKS dissertation series 2006-28, University of Amsterdam, 2006.

[13] B. Sigurbjörnsson and J. Kamps. The effect of structured queries and selective indexing on XML retrieval. In *Advances in XML Information Retrieval and Evaluation: INEX 2005*, volume 3977 of *LNCS*, pages 104–118, 2006.

[14] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. An Element-Based Approach to XML Retrieval. In *INEX 2003 Workshop Proceedings*, pages 19–26, 2004.

[15] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. Mixture models, overlap, and structural hints in XML element retreival. In *Advances in XML Information Retrieval: INEX 2004*, volume 3493 of *LNCS 3493*, pages 196–210, 2005.

[16] A. Trotman and S. Geva. Passage retrieval and other XML-retrieval tasks. In *Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology*, pages 43–50. University of Otago, Dunedin New Zealand, 2006.

[17] Wikipedia. The free encyclopedia, 2006. http://en.wikipedia.org/.