University of Amsterdam at INEX 2009: Ad hoc, Book and Entity Ranking Tracks

Marijn Koolen[1], Rianne Kaptein[1], and Jaap Kamps[1,2]

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

Abstract. In this paper we describe our participation in INEX 2009 in the Ad Hoc Track, the Book Track, and the Entity Ranking Track. In the Ad Hoc track we investigate focused link evidence, using only links from retrieved sections. The new collection is not only annotated with Wikipedia categories, but also with YAGO/WordNet categories. We explore how we can use both types of category information, in the Ad Hoc Track as well as in the Entity Ranking Track. Results in the Ad Hoc Track show Wikipedia categories are more effective than WordNet categories, and Wikipedia categories in combination with relevance feedback lead to the best results.

1 Introduction

In this paper, we describe our participation in the INEX 2009 Ad Hoc, Book, and Entity Ranking Tracks. Our aims for this year were to familiarize ourselves with the new Wikipedia collection, to continue the work from previous years, and to explore the opportunities of using category information, which can be in the form of Wikipedia's categories, or the enriched YAGO/WordNet categories.

The rest of the paper is organized as follows. First, Section 2 describes the collection and the indexes we use. Then, in Section 3, we report our runs and results for the Ad Hoc Track. Section 4 briefly discusses our Book Track experiments. In Section 5, we present our approach to the Entity Ranking Track. Finally, in Section 6, we discuss our findings and draw preliminary conclusions.

2 Indexing the Wikipedia Collection

In this section we describe the index that is used for our runs in the adhoc and the entity ranking track, as well as the category structure of the collection. The collection is based, again, on the Wikipedia but substantially larger and with longer articles. The original Wiki-syntax is transformed into XML, and each article is annotated using "semantic" categories based on YAGO/Wikipedia. We used Indri [14] for indexing and retrieval.

2.1 Indexing

Our indexing approach is based on our earlier work [1, 3, 5, 11, 12, 13].

- *Section index*: We used the `<section>` element to cut up each article in sections and indexed each section as a retrievable unit. Some articles have a leading paragraph not contained in any `<section>` element. These leading paragraphs, contained in `<p>` elements are also indexed as retrievable units. The resulting index contains no overlapping elements.
- *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, and terms were stemmed using the Krovetz stemmer. Queries are processed similar to the documents. In the ad hoc track 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.

2.2 Category Structure

A new feature in the new Wikipedia collection is the assignment of WordNet labels to documents [10]. The WordNet categories are derived from Wikipedia categories, but are designed to be conceptual. Categories for administrative purposes, such as 'Article with unsourced statements', categories yielding non-conceptual information, such as '1979 births' and categories that indicate merely thematic vicinity, such as 'Physics', are not used for the generation of WordNet labels, but are excluded by hand and some shallow linguistic parsing of the category names. WordNet concepts are matched with category names and the category is linked to the most common concept among the WordNet concepts. It is claimed this simple heuristic yields the correct link in the overwhelming majority of cases.

A second method which is used to generate WordNet labels, is based on the basis of information in lists. For example, If all links but one in a list point to pages belonging to a certain category, this category is also assigned to the page that was not labelled with this category. This is likely to improve the consistency of annotation, since annotation in Wikipedia is largely a manual effort.

3 Ad Hoc Track

For the INEX 2009 Ad Hoc Track we aim to investigate:

- Focused link evidence. Use local link degrees as evidence of topical relevance. Instead of looking at all local links between the top 100 retrieved articles, we consider only the links occurring in the retrieved elements. A link from article A to article B occurring in a section of article A that is not retrieved is ignored. This link evidence is more focused on the search topic and possibly leads to less infiltration.

– Wikipedia and WordNet categories. The new INEX Wikipedia collection has markup in the form of YAGO elements including WordNet categories. Most Wikipedia articles are manually categorised by the Wikipedia contributors. The category structure can be used to generate category models to promote articles that belong to categories that best match the query. We aim to directly compare the effectiveness of category models based on the Wikipedia and WordNet categorisations for improving retrieval effectiveness.

We will first describe our approach and the official runs, and finally per task, we present and discuss our results.

3.1 Approach

We have four baseline runs based on the indexes described in the previous section:

Article : run on the article index with linear length prior and linear smoothing $\lambda = 0.15$.
Section : run on the section index with linear length prior and linear smoothing $\lambda = 0.15$.
Article RF : run on the article index with blind relevance feedback, using 50 terms from the top 10 results.
Section RF : run on the section index with blind relevance feedback, using 50 terms from the top 10 results.

All our official runs for all four tasks are based on these runs. To improve these baselines, we explore the following options.

Category distance : We determine two target categories for a query based on the top 20 results. We select the two most frequent categories to which the top 20 results are assigned and compute a category distance score using parsimonious language models of each category. This technique was successfully employed on the INEX 2007 Ad hoc topics by Kaptein et al. [7]. In the new collection, there are two sets of category labels. One based on the *Wikipedia* category structure and one based on the *WordNet* category labels.
CAS filter : For the CAS queries we extracted from the CAS title all semantic target elements, identified all returned results that contain a target element in the xpath and ranked them before all other results by adding a constant c to the score per matching target element. Other than that, we keep the ranking in tact. A result that matches two target elements gets $2c$ added to its score, while a result matching one target element gets $1c$ added to its score. In this way, results matching n target elements are ranked above results matching $n-1$ target elements. This is somewhat similar to coordination level ranking of content-only queries. Syntactic target elements like `<article>`, `<sec>`, `<p>` and `<category>` are ignored.
Link degrees : Both incoming and outgoing link degrees are useful evidence in identifying topical relevance [4, 9]. We use the combined $indegree(d) + outdegree(d)$ as a document "prior" probability $P_{link}(d)$. This is easy to

incorporate in a standard language model. Of course, local link evidence is not query-independent, so $P_{link}(d)$ is not an actual *prior* probability. We note that for runs where we combine the article or section text score with a category distance score, we get a different score distribution. With these runs we use the link evidence more carefully by taking the log of the link degree as $P_{link}(d)$.

Focused Link degrees : We also constructed a focused local link graph based on the retrieved elements of the top 100 articles. Instead of using all links between the top 100 articles, we only use the outgoing links from sections that are retrieved for a given topic. The main idea behind this is that link anchors appearing closer to the query terms are more closely related to the search topic. Thus, if for an article a_i in the top 100 articles only section s_j is retrieved, we use only the links appearing in section s_j that point to other articles in the top 100. This local link graph is more focused on the search topic, and potentially suffers less from infiltration of important but off-topic articles. Once the focused local link graph is constructed, we count the number of incoming + outgoing links as the focused link prior $P_{foc_link}(d)$.

Article ranking : based on [3], we use the article ranking of an article index run and group the elements returned by a section index run as focused results.

Cut-off(n) : When we group returned elements per article for the Relevant in Context task, we can choose to group all returned elements of an article, or only the top ranked elements. Of course, further down the results list we find less relevant elements, so grouping them with higher ranked elements from the same article might actually hurt precision. We set a cut-off at rank n to group only the top returned elements by article.

3.2 Runs

Combining the methods described in the previous section with our baseline runs leads to the following official runs.

For the Thorough Task, we submitted two runs:

UamsTAdbi100 : an article index run with relevance feedback. The top 100 results are re-ranked using the link degree prior $P_{link}(d)$. This run was submitted to the Thorough task.

UamsTSdbi100 : a section index run with relevance feedback. We cut off the results list at rank and re-rank the focused results of the top 100 articles using the link prior $P_{link}(d)$. This run was submitted to the Thorough task.

For the Focused Task, we submitted two runs:

UamsFSdbi100CAS : a section index run combined with the Wikipedia category distance scores. The results of the top 100 articles are re-ranked using the link degree prior. Finally, the CAS filter is applied to boost results with target elements in the xpath. This run was submitted to the Focused task.

UamsFSs2dbi100CAS : a section index run combined with the Wikipedia category distance scores. The results of the top 100 articles are re-ranked using the focused link degree prior $P_{foc_link}(d)$. Finally, the CAS filter is applied to boost results with target elements in the xpath. This run was submitted to the Focused task.

For the Relevant in Context Task, we submitted two runs:

UamsRSCMACMdbi100 : For the article ranking we used the article text score combined with the manual category distance score as a baseline and re-ranked the top 100 articles with the log of the local link prior $P_{link}(d)$. The returned elements are the top results of a combination of the section text score and the manual category distance score, grouped per article. This run was submitted to the Relevant in Context task.

UamsRSCWACWdbi100 : For the article ranking we used the article text score combined with the WordNet category distance score as a baseline and re-ranked the top 100 with the log of the local link prior $P_{link}(d)$. The returned elements are the top results of a combination of the section text score and the wordnet category distance score, grouped per article. This run was submitted to the Relevant in Context task.

For the Best in Context Task, we submitted two runs:

UamsBAfbCMdbi100 : an article index run with relevance feedback combined with the Wikipedia category distance scores, using the local link prior $P_{link}(d)$ to re-rank the top 100 articles. The Best-Entry-Point is the start of the article. This run was submitted to the Best in Context task.

UamsBAfbCMdbi100 : a section index run with relevance feedback combined with the Wikipedia category distance scores, using the focused local link prior $P_{foc_link}(d)$ to re-rank the top 100 articles. Finally, the CAS filter is applied to boost results with target elements in the xpath. The Best-Entry-Point is the start of the article. This run was submitted to the Best in Context task.

3.3 Thorough Task

Results of the Thorough Task can be found in Table 1. We make the following observations:

– Standard relevance feedback improves upon the baseline. The Wikipedia category distances are even more effective. The WordNet category distances are somewhat less effective, but still lead to improvement for MAiP.
– Combining relevance feedback with the WordNet categories hurts performance, whereas combining feedback with the Wikipedia categories improves MAiP. However, for early precision, the Wikipedia categories without feedback perform better.
– The link prior has a negative impact on performance of article level runs. The official run *UamsTAdbi100* is based on the *Article RF* run, but with the top 100 articles re-ranked using the local link prior.

Table 1: Results for the Ad Hoc Track Thorough Task (runs labeled "UAms" are official submissions)

Run id	**MAiP**	**iP[0.00]**	**iP[0.01]**	**iP[0.05]**	**iP[0.10]**
UamsTAdbi100	0.2676	0.5350	0.5239	0.4968	0.4712
UamsTSdbi100	0.2139	0.5022	0.4915	0.4639	0.4400
Article	0.2814	0.5938	0.5880	0.5385	0.4981
Article RF	0.2967	0.6082	0.5948	0.5552	0.5033
Article + Cat(Wiki)	0.2991	**0.6156**	**0.6150**	**0.5804**	**0.5218**
Article + Cat(WordNet)	0.2841	0.5600	0.5499	0.5203	0.4950
Article RF + Cat(Wiki)	**0.3011**	0.6006	0.5932	0.5607	0.5177
Article RF + Cat(WordNet)	0.2777	0.5490	0.5421	0.5167	0.4908
$(Article + CAT(Wiki)) \cdot P_{link}(d)$	0.2637	0.5568	0.5563	0.4934	0.4662
$(Article + CAT(WordNet)) \cdot P_{link}(d)$	0.2573	0.5345	0.5302	0.4924	0.4567
Section	0.1403	0.5525	0.4948	0.4155	0.3594
Section RF	0.1493	0.5761	0.5092	0.4296	0.3623
Section + Cat(Wiki)	0.1760	0.6147	0.5667	0.5012	0.4334
Section + Cat(WordNet)	0.1533	0.5474	0.4982	0.4506	0.3831
Section RF + Cat(Wiki)	0.1813	0.5819	0.5415	0.4752	0.4186
Section RF + Cat(WordNet)	0.1533	0.5356	0.4794	0.4201	0.3737

Table 2: Results for the Ad Hoc Track Focused Task (runs labeled "UAms" are official submissions)

Run id	**MAiP**	**iP[0.00]**	**iP[0.01]**	**iP[0.05]**	**iP[0.10]**
UamsFSdbi100CAS	0.1726	0.5567	0.5296	0.4703	0.4235
UamsFSs2dbi100CAS	0.1928	**0.6328**	0.5997	0.5140	0.4647
Section	0.1403	0.5525	0.4948	0.4155	0.3594
Section RF	0.1493	0.5761	0.5092	0.4296	0.3623
Section + Cat(Wiki)	0.1760	0.6147	0.5667	0.5012	0.4334
Section RF + Cat(Wiki)	0.1813	0.5819	0.5415	0.4752	0.4186
Article + Cat(Wiki)	0.2991	0.6156	0.6150	**0.5804**	**0.5218**
Article RF + Cat(Wiki)	**0.3011**	0.6006	0.5932	0.5607	0.5177
UamsRSCMACMdbi100	0.2096	0.6284	**0.6250**	0.5363	0.4733
UamsRSCWACWdbi100	0.2132	0.6122	0.5980	0.5317	0.4782

– On the section level run it leads to improvement. The official run *UamsTSdbi100* is based on the *Section RF* run, but with the results of the top 100 articles re-ranked using the local link prior. Here, the link prior increases MAiP from 0.1493 to 0.2139.
– Section index runs miss too much relevant information. They perform much worse than the article index runs.

3.4 Focused Task

We have no overlapping elements in our indexes, so no overlap filtering is done. Table 2 shows the results for the Focused Task. We make the following observations:

Table 3: Results for the Ad Hoc Track Relevant in Context Task (runs labeled "UAms" are official submissions)

Run id	**MAgP**	**gP[5]**	**gP[10]**	**gP[25]**	**gP[50]**
UamsRSCMACMdbi100	0.1771	0.3192	0.2794	0.2073	0.1658
UamsRSCWACWdbi100	0.1678	0.3010	0.2537	0.2009	0.1591
Article	0.1775	0.3150	0.2773	0.2109	0.1621
Article RF	0.1880	0.3498	0.2956	0.2230	0.1666
Article + Cat(Wiki)	0.1888	0.3393	0.2869	**0.2271**	0.1724
Article + Cat(WordNet)	0.1799	0.2984	0.2702	0.2199	0.1680
Article RF + Cat(Wiki)	**0.1950**	**0.3528**	**0.2979**	0.2257	**0.1730**
Article RF + Cat(WordNet)	0.1792	0.3200	0.2702	0.2180	0.1638

- The runs shown are the same as those for the Thorough task. Since the measures used are also the same, the results are also the same. The Wikipedia categories are very effective in improving performance of both the article and section index runs.
- The official Focused runs *UamsFSdbi100CAS* and *UamsFSs2dbi100CAS* are the link prior and focused link prior versions of the *Section + Cat(Wiki)* run. Both runs are also CAS filtered. The document level link degrees hurt performance, while the focused link degrees improve performance.
- The *Article + Cat(Wiki)* run has a slightly lower iP[0.00] than the official *UamsFSs2dbi100CAS*, but a somewhat higher iP[0.01]. The section index is less effective for the Focused task than the article index.
- For comparison, we also show the official Relevant in Context run *UamsRSCMACMdbi100*, which uses the same result elements as the *Section + Cat(Wiki)* run, but groups them per article and uses the $(Article + Cat(Wiki)) \cdot P_{link}(d)$ run for the article ranking. This improves the precision at iP[0.01]. The combination of the section run and the article run gives the best performance.

3.5 Relevant in Context Task

For the Relevant in Context Task, we group result per article. Table 3 shows the results for the Relevant in Context Task. We make the following observations:

- A simple article level run is just as effective for the Relevant in Context task as the much more complex official runs, which uses the $Article + Cat(Wiki) \cdot log(P_{link}(d))$ run for the article ranking, and the *Section + Cat(Wiki)* run for the top 1500 sections.
- Both relevance feedback and category distance improve upon the baseline article run. Combining relevance feedback with the Wikipedia category distance gives the best results.
- The WordNet categories again hurt performance of the relevance feedback run.

Table 4: Results for the Ad Hoc Track Best in Context Task (runs labeled "UAms" are official submissions)

Run id	MAgP	gP[5]	gP[10]	gP[25]	gP[50]
UamsBAfbCMdbi100	0.1543	0.2604	0.2298	0.1676	0.1478
UamsBSfbCMs2dbi100CASart1	0.1175	0.2193	0.1838	0.1492	0.1278
UamsTAdbi100	0.1601	0.2946	0.2374	0.1817	0.1444
Article	0.1620	0.2853	0.2550	0.1913	0.1515
Article RF	0.1685	**0.3203**	**0.2645**	0.2004	0.1506
Article + Cat(Wiki)	0.1740	0.2994	0.2537	**0.2069**	**0.1601**
Article + Cat(WordNet)	0.1670	0.2713	0.2438	0.2020	0.1592
Article RF + Cat(Wiki)	**0.1753**	0.3091	0.2625	0.2001	0.1564
Article RF + Cat(WordNet)	0.1646	0.2857	0.2506	0.1995	0.1542

3.6 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. We make the following observations:

– Same patterns. Relevance feedback helps, so do Wikipedia and WordNet categories. Wikipedia categories are more effective than relevance feedback, WordNet categories are less effective. Wikipedia categories combined with relevance feedback gives further improvements, WordNet combined with feedback gives worse performance than feedback alone. Links hurt performance. Finally, the section index is much less effective than the article index.

4 Book Track

In the INEX 2009 Book Track we participated in the Book Retrieval and Focused Book Search tasks. Continuing our efforts of last year, we aim to find the appropriate level of granularity for Focused Book Search. The BookML markup has XML elements on the page level. In the assessments of last year, relevant passages often cover multiple pages [8]. With larger relevant passages, query terms might be spread over multiple pages, making it hard for a page level retrieval model to assess the relevance of individual pages.

Can we better locate relevant passages by considering larger book parts as retrievable units? One simple option is to divide the whole book in sequences of n pages. Another approach would be to use the logical structure of a book to determine the retrievable units. The INEX Book corpus has no explicit XML elements for the various logical units of the books, so as a first approach we divide each book in sequences of pages.

Book index : each whole book is indexed as a retrievable unit.
Page index : each individual page is indexed as a retrievable unit.
5-Page index : each sequence of 5 pages is indexed as a retrievable unit. That is, pages 1-5, 6-10, etc., are treated as text units.

We submitted six runs in total: two for the Book Retrieval (BR) task and four for the Focused Book Search (FBS) task. The 2009 topics consist an overall topic statement and one or multiple sub-topics. In total, there are 16 topics and 37 sub-topics. The BR runs are based on the 16 overall topics. The FBS runs are based on the 37 sub-topics.

Book : a standard Book index run. Up to 1000 results are returned per topic.
Book RF : a Book index run with Relevance Feedback (RF). The initial queries are expanded with 50 terms from the top 10 results.
Page : a standard Page index run.
Page RF : a Page index run with Relevance Feedback (RF). The initial queries are expanded with 50 terms from the top 10 results.
5-page : a standard 5-Page index run.
5-Page RF : a 5-Page index run with Relevance Feedback (RF). The initial queries are expanded with 50 terms from the top 10 results.

At the time of writing, no relevance assessments have been made. Therefore we cannot yet provide any evaluation results.

5 Entity Ranking

In this section, we describe our approach to the Entity Ranking Track. Our goals for participation in the entity ranking track are to refine last year's entity ranking method, which proved to be quite effective, and to explore the opportunities of the new Wikipedia collection. The most effective part of our entity ranking approach last year was combining the documents score with a category score, where the category score represents the distance between the document categories and the target categories. We do not use any link information, since last year this only lead to minor improvements [6].

5.1 Category information

For each target category we estimate the distances to the categories assigned to the answer entity. The distance between two categories is estimated according to the category titles. Last year we also experimented with a binary distance, and a distance between category contents, but we found the distance estimated using category titles the most efficient and at the same time effective method.

To estimate title distance, we need to calculate the probability of a term occurring in a category title. To avoid a division by zero, we smooth the probabilities of a term occurring in a category title with the background collection:

$$P(t_1, ..., t_n|C) = \sum\nolimits_{i=1}^{n} \lambda P(t_i|C) + (1 - \lambda) P(t_i|D)$$

where C is the category title and D is the entire wikipedia document collection, which is used to estimate background probabilities. We estimate $P(t|C)$ with a parsimonious model [2] that uses an iterative EM algorithm as follows:

$$\text{E-step:} \qquad e_t = tf_{t,C} \cdot \frac{\alpha P(t|C)}{\alpha P(t|C) + (1-\alpha)P(t|D)}$$

$$\text{M-step:} \qquad P(t|C) = \frac{e_t}{\sum_t e_t}, \text{i.e. normalize the model}$$

The initial probability $P(t|C)$ is estimated using maximum likelihood estimation. We use KL-divergence to calculate distances, and calculate a category score that is high when the distance is small as follows:

$$S_{cat}(C_d|C_t) = -D_{KL}(C_d|C_t) = -\sum\nolimits_{t \in D} \left(P(t|C_t) * \log \left(\frac{P(t|C_t)}{P(t|C_d)} \right) \right)$$

where d is a document, i.e. an answer entity, C_t is a target category and C_d a category assigned to a document. The score for an answer entity in relation to a target category $S(d|C_t)$ is the highest score, or shortest distance from any of the document categories to the target category.

For each target category we take only the shortest distance from any answer entity category to a target category. So if one of the categories of the document is exactly the target category, the distance and also the category score for that target category is 0, no matter what other categories are assigned to the document. Finally, the score for an answer entity in relation to a query topic $S(d|QT)$ is the sum of the scores of all target categories:

$$S_{cat}(d|QT) = \sum\nolimits_{C_t \in QT} \operatorname*{argmax}_{C_d \in d} S(C_d|C_t)$$

A new feature in the new Wikipedia collection is the assignment of YAGO/-WordNet categories to documents as described in Section 2.2. These WordNet categories have some interesting properties for entity ranking. The WordNet categories are designed to be conceptual, and by exploiting list information, pages should be more consistently annotated. In our official runs we have made several combinations of Wikipedia and WordNet categories.

5.2 Pseudo-Relevant Target Categories

Last year we found a discrepancy between the target categories assigned manually to the topics, and the categories assigned to the answer entities. The target categories are often more general, and can be found higher in the Wikipedia category hierarchy. For example, topic 102 with title ‘Existential films and novels’ has as target categories ‘films’ and ‘novels,’ but none of the example entities belong directly to one of these categories. Instead, they belong to lower level categories such as ‘1938 novels,’ ‘Philosophical novels,’ ‘Novels by Jean-Paul Sartre’ and ‘Existentialist works’ for the example entity ‘Nausea (Book).’ In this case the estimated category distance to the target category ‘novels’ will be small, because the term ‘novels’ occurs in the document category titles, but this is not

Table 5: Target Categories

	olympic classes dinghy sailing	**Neil Gaiman novels**	**chess world champions**
Assigned	dinghies	novels	chess grandmasters world chess champions
PR	dinghies sailing	comics by Neil Gaiman fantasy novels	chess grandmasters world chess champions
Wikipedia	dinghies sailing at the olympics boat types	fantasy novels novels by Neil Gaiman	chess grandmasters chess writers living people world chess champion russian writers russian chess players russian chess writers 1975 births soviet chess players people from Saint Petersburg
Wordnet	specification types	writing literary composition novel written communication fiction	entity player champion grandmaster writer chess player person soviet writers

always the case. In addition to the manually assigned target categories, we have therefore created a set of pseudo-relevant target categories. From our baseline run we take the top n results, and assign k pseudo-relevant target categories if they occur at least 2 times as a document category in the top n results. Since we had no training data available we did a manual inspection of the results to determine the parameter settings, which are $n = 20$ and $k = 2$ in our official runs. For the entity ranking task we submitted different combinations of the baseline document score, the category score based on the assigned target categories, and the category score based on the pseudo-relevant target categories. For the list completion task, we follow a similar procedure to assign target categories, but instead of using pseudo-relevant results, we use the categories of the example entities. All categories that occur at least twice in the example entities are assigned as target categories.

5.3 Results

Since the runs are not officially evaluated yet, in this section we will only look at the categories assigned by the different methods. In Table 5 we show a few example topics together with the categories as assigned ("Assigned") by each method. As expected the pseudo-relevant target categories ("PR") are more specific than the manually assigned target categories. The number of common Wikipedia categories in the example entities ("Wikipedia") can in fact be quite long. More categories is in itself not a problem, but also non relevant categories such as '1975 births' and 'russian writers' and very general categories such as 'living people' are added as target categories. Finally, the WordNet categories ("WordNet") contain less detail than the Wikipedia categories. Some general concepts such as 'entity' are included. With these kind of categories, a higher recall but smaller precision is expected.

6 Conclusion

In this paper we discussed our participation in the INEX 2009 Ad Hoc, Book, and the Entity Ranking Tracks. For the Ad Hoc Track we can conclude focused link evidence outperforms local link evidence on the article level for the Focused Task. Focused link evidence leads to high early precision. Using category information in the form of Wikipedia categories turns out to be very effective, and more valuable than WordNet category information. Since there are no results yet for the Entity Ranking Track and the Book Track, we cannot draw any conclusions about them here.

Acknowledgments Jaap Kamps was supported by the Netherlands Organization for Scientific Research (NWO, grants # 612.066.513, 639.072.601, and 640.001.-501). Rianne Kaptein was supported by NWO under grant # 612.066.513 Marijn Koolen was supported by NWO under grant # 640.001.501.

Bibliography

[1] K. N. Fachry, J. Kamps, M. Koolen, and J. Zhang. Using and detecting links in Wikipedia. In *Focused access to XML documents: 6th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2007)*, volume 4862 of *LNCS*, pages 388–403. Springer Verlag, Heidelberg, 2008.

[2] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 178–185. ACM Press, New York NY, 2004.

[3] J. Kamps and M. Koolen. The impact of document level ranking on focused retrieval. In *Advances in Focused Retrieval: 7th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2008)*, volume 5631 of *LNCS*. Springer Verlag, Berlin, Heidelberg, 2009.

[4] J. Kamps and M. Koolen. Is wikipedia link structure different? In *Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM 2009)*. ACM Press, New York NY, USA, 2009.

[5] 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 *LNCS*, pages 121–136. Springer Verlag, Heidelberg, 2007.

[6] R. Kaptein and J. Kamps. Finding entities in Wikipedia using links and categories. In *Advances in Focused Retrieval: 7th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2008)*, volume 5631 of *LNCS*. Springer Verlag, Berlin, Heidelberg, 2009.

[7] R. Kaptein, M. Koolen, and J. Kamps. Using Wikipedia categories for ad hoc search. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, New York NY, USA, 2009.

[8] G. Kazai, N. Milic-Frayling, and J. Costello. Towards methods for the collective gathering and quality control of relevance assessments. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 452–459, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6. doi: http://doi.acm.org/10.1145/1571941.1572019.

[9] M. Koolen and J. Kamps. What's in a link? from document importance to topical relevance. In *Proceedings of the 2nd International Conferences on the Theory of Information Retrieval (ICTIR 2009)*, volume 5766 of *LNCS*, pages 313–321. Springer Verlag, Berlin, Heidelberg, 2009.

[10] R. Schenkel, F. Suchanek, and G. Kasneci. YAWN: A semantically annotated wikipedia xml corpus. In *12th GI Conference on Databases in Business, Technology and Web (BTW 2007)*, March 2007.

[11] 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.

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

[13] 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.

[14] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: a language-model based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, 2005.