Using Wikipedia Categories and Links in Entity Ranking

Anne-Marie Vercoustre, Jovan Pehcevski, James A. Thom

Using Wikipedia Categories and Links in Entity Ranking

Anne-Marie Vercoustre[1], Jovan Pehcevski[1], and James A. Thom[2]

[1] INRIA Rocquencourt, France
{anne-marie.vercoustre,jovan.pehcevski}@inria.fr
[2] RMIT University, Melbourne, Australia
james.thom@rmit.edu.au

Abstract. This paper describes the participation of the INRIA group in the INEX 2007 XML entity ranking and ad hoc tracks. We developed a system for ranking Wikipedia entities in answer to a query. Our approach utilises the known categories, the link structure of Wikipedia, as well as the link co-occurrences with the examples (when provided) to improve the effectiveness of entity ranking. Our experiments on the training data set demonstrate that the use of categories and the link structure of Wikipedia, together with entity examples, can significantly improve entity retrieval effectiveness. We also use our system for the ad hoc tasks by inferring target categories from the title of the query. The results were worse than when using a full-text search engine, which confirms our hypothesis that ad hoc retrieval and entity retrieval are two different tasks.

1 Introduction

Entity ranking has recently emerged as a research field that aims at retrieving entities as answers to a query [5, 8, 10, 11]. Here, unlike in the related field of entity extraction, the goal is not to tag the names of the entities in documents but rather to get back a list of the relevant entity names. It is a generalisation of the expert search task explored by the TREC Enterprise track [9], except that instead of ranking people who are experts in the given topic, other types of entities such as organizations, countries, or locations can also be retrieved and ranked.

The Initiative for the Evaluation of XML retrieval (INEX) is running a new track on entity ranking in 2007, using Wikipedia as its document collection [3]. There are two tasks in the INEX 2007 XML entity ranking (XER) track: *entity ranking*, which aims at retrieving entities of a given category that satisfy a topic described in natural language text; and *list completion*, where given a topic text and a small number of entity examples, the aim is to complete this partial list of answers. Two data sets were used by the participants of the INEX 2007 XER track: a *training* data set, comprising 28 XER topics which were adapted from the INEX 2006 ad hoc topics and proposed by our INRIA participating group; and a *testing* data set, comprising 73 XER topics most of which were proposed

```
<inex_topic>
<title>
European countries where I can pay with Euros
</title>
<description>
I want a list of European countries where I can pay with Euros.
</description>
<narrative>
Each answer should be the article about a specific European country
that uses the Euro as currency.
</narrative>
<entities>
   <entity ID="10581">France</entity>
   <entity ID="11867">Germany</entity>
   <entity ID="26667">Spain</entity>
</entities>
<categories>
<category ID="185">european countries<category>
</categories>
</inex_topic>
```

Fig. 1. Example INEX 2007 XML entity ranking topic

and assessed by the track participants. The main purpose of having two data sets is to allow participants to tune the parameters of their entity ranking systems on the training data set, and then use the optimal parameter values on the testing data set.

An example of an INEX 2007 XER topic is shown in Figure 1. Here, the `title` field contains the plain content only query, the `description` provides a natural language description of the information need, and the `narrative` provides a detailed explanation of what makes an entity answer relevant. In addition to these fields, the `entities` field provides a few of the expected entity answers for the topic (task 2), while the `categories` field provides the target category of the expected entity answers (task 1).

In this new track, the expected entities correspond to Wikipedia articles that are likely to be referred to by links in other articles. As an example, the query "European countries where I can pay with Euros" [3] should return a list of entities (or pages) representing relevant countries, and not a list of entities representing non-relevant (country or other) names found in pages about the Euro and similar currencies.

In this paper, we describe our approach to ranking entities from the Wikipedia XML document collection. Our approach is based on the following principles:

1. A good entity page is a page that answers the query (or a query extended with names of target categories or entity examples).

2. A good entity page is a page associated with a category close to the target category (task 1) or to the categories of the entity examples (task 2).
3. A good entity page is referred to by a page answering the query; this is an adaptation of the HITS [6] algorithm to the problem of entity ranking.
4. A good entity page is referred to by contexts with many occurrences of the entity examples (task 2). A broad context could be the full page that contains the entity examples, while smaller and more narrow contexts could be elements such as paragraphs, lists, or tables.

After a short presentation of the INEX Wikipedia XML collection used for entity ranking, we provide a detailed description of our entity ranking approach and the runs we submitted for evaluation to the INEX 2007 XER track. We also report on our run submissions to the INEX 2007 ad hoc track.

2 INEX Wikipedia XML collection

Wikipedia is a well known web-based, multilingual, free content encyclopedia written collaboratively by contributors from around the world. As it is fast growing and evolving it is not possible to use the actual online Wikipedia for experiments, and so we need a stable collection to do evaluation experiments that can be compared over time. Denoyer and Gallinari [4] have developed an XML-based corpus based on a snapshot of the Wikipedia, which has been used by various INEX tracks in 2006 and 2007. It differs from the real Wikipedia in some respects (size, document format, category tables), but it is a very realistic approximation.

2.1 Entities in Wikipedia

The entities have a name (the name of the corresponding page) and a unique ID in the collection. When mentioning such an entity in a new Wikipedia article, authors are encouraged to link every occurrence of the entity name to the page describing this entity. This is an important feature as it allows to easily locate potential entities, which is a major issue in entity extraction from plain text.

However in this collection, not all potential entities have been associated with corresponding pages. For example, if we look for Picasso's artworks, only three paintings ("Les Demoiselles d'Avignon", "Guernica", and "Le garçon à la pipe") get associated pages. If the query was "paintings by Picasso", we would not expect to get more than three entity pages for Picasso's paintings, while for the online Wikipedia there are about thirty entities, yet not that many compared to the actual number of his listed paintings.

The INEX XER topics have been carefully designed to make sure there is a sufficient number of answer entities. For example, in the Euro page (see Fig. 2), all the underlined hypertext links can be seen as occurrences of entities that are each linked to their corresponding pages. In this figure, there are 18 entity references of which 15 are country names; specifically, these countries are all "European Union member states", which brings us to the notion of category in Wikipedia.

"The **euro** ... is the official currency of the Eurozone (also known as the Euro Area), which consists of the European states of Austria, Belgium, Finland, France, Germany, Greece, Ireland, Italy, Luxembourg, the Netherlands, Portugal, Slovenia and Spain, and will extend to include Cyprus and Malta from 1 January 2008."

Fig. 2. Extract from the Euro Wikipedia page

2.2 Categories in Wikipedia

Wikipedia also offers categories that authors can associate with Wikipedia pages. There are 113,483 categories in the INEX Wikipedia XML collection, which are organised in a graph of categories. Each page can be associated with many categories (2.28 as an average).

Wikipedia categories have unique names (e.g. "France", "European Countries", "Countries"). New categories can also be created by authors, although they have to follow Wikipedia recommendations in both creating new categories and associating them with pages. For example, the Spain page is associated with the following categories: "Spain", "European Union member states", "Spanish-speaking countries", "Constitutional monarchies" (and some other Wikipedia administrative categories).

When searching for entities it is natural to take advantage of the Wikipedia categories since they would give a hint on whether the retrieved entities are of the expected type. For example, when looking for entities "authors", pages associated with the category "Novelist" may be more relevant than pages associated with the category "Book".

3 Our entity ranking approach

Our approach to identifying and ranking entities combines: (1) the full-text similarity of the answer entity page with the query; (2) the similarity of the page's categories with the target categories (task 1) or the categories attached to the entity examples (task 2); and (3) the contexts around entity examples (task 2) found in the top ranked pages returned by a search engine for the query.

We have built a system based on the above ideas, and a framework to tune and evaluate a set of different entity ranking algorithms.

3.1 Architecture

The system involves several modules and functions that are used for processing a query, submitting it to the search engine, applying our entity ranking algorithms, and finally returning a ranked list of entities. We use Zettair[3] as our choice for a full-text search engine. Zettair is a full-text information retrieval (IR) system

[3] http://www.seg.rmit.edu.au/zettair/

developed by RMIT University, which returns pages ranked by their similarity score to the query. In a recent comparison of open source search engines, Zettair was found to be "one of the most complete engines" [7]. We used the Okapi BM25 similarity measure that has proved to work well on the INEX 2006 Wikipedia test collection [1].

Our system involves the following modules and functions:

- the topic module takes an INEX topic as input (as the topic example shown in Fig. 1) and generates the corresponding Zettair query and the list of target categories and entity examples (as an option, the names of target categories or example entities may be added to the query);
- the search module sends the query to Zettair and returns a list of ranked Wikipedia pages (typically 1500);
- the link extraction module extracts the links from a selected number of highly ranked pages,[4] together with the information concerning the paths of the links (using an XPath notation);
- the category similarity module calculates a weight for a page based on the similarity of the page categories with target categories or those of the entity examples (see 3.2);
- the linkrank module calculates a weight for a page based (among other things) on the number of links to this page (see 3.4); and
- the full-text IR module calculates a weight for a page based on its initial Zettair score (see 3.4).

The global score for a page is calculated as a linear combination of three normalised scores coming out of the last three modules (see 3.4).

The architecture provides a general framework for evaluating entity ranking which allows for some modules to be replaced by more advanced modules, or by providing a more efficient implementation of a module. It also uses an evaluation module to assist in tuning the system by varying the parameters and to globally evaluate our entity ranking approach.

The current system was not designed for online entity ranking in Wikipedia. First, because we are not dealing with the online Wikipedia, and second because of performance issues. The major cost in running our system is in extracting the links from the selected number of pages retrieved by the search engine. Although we only extract links once by topic and store them in a database for reuse in later runs, for an online system it would be more efficient to extract and store all the links at indexing time.

3.2 Using Wikipedia categories

To make use of the Wikipedia categories in entity ranking, we define similarity functions between:

[4] We discarded external links and some internal collection links that do not refer to existing pages in the INEX Wikipedia collection.

– the categories of answer entities and the target categories (task 1), or
– the categories of answer entities and a set of categories attached to the entity examples (task2).

Similarity measures between concepts of the same ontology, such as tree-based similarities [2], cannot be applied directly to Wikipedia categories, mostly because the notion of sub-categories in Wikipedia is not a subsumption relationship. Another reason is that categories in Wikipedia do not form a hierarchy (or a set of hierarchies) but a graph with potential cycles [10, 12].

Task 1 We first define a similarity function that computes the ratio of common categories between the set of categories $\mathsf{cat}(t)$, associated to an answer entity page t, and the set $\mathsf{cat}(C)$ which is the union of the provided target categories C:

$$S_C(t) = \frac{|\mathsf{cat}(t) \cap \mathsf{cat}(C)|}{|\mathsf{cat}(C)|} \tag{1}$$

The target categories will be generally very broad, so it is to be expected that the answer entities would not be directly attached to these broad categories. Accordingly, we experimented with several extensions of the set of categories, both for the target categories and the categories attached to answer entities.

We first experimented with extensions based on using sub-categories and parent categories in the graph of Wikipedia categories. However, on the training data set, we found that these category extensions overall do not result in an improved performance [10], and so they were not used in our INEX 2007 runs.

Another approach is to use lexical similarity between categories. For example, "european countries" is lexically similar to "countries" since they both contain the word "countries" in their names. We use an information retrieval approach to retrieve similar categories, by indexing with Zettair all the categories, using their names as corresponding documents. By sending both the title of the topic T and the category names C as a query to Zettair, we then retrieve all the categories that are lexically similar to C. We keep the top M ranked categories and add them to C to form the set $\mathsf{TCcat}(C)$. On the training data set, we found that the value M=5 is the optimal parameter value used to retrieve the likely relevant categories for this task [10]. We then use the same similarity function as before, where $\mathsf{cat}(C)$ is replaced with $\mathsf{TCcat}(C)$.

We also experimented with two alternative approaches: by sending the category names C as a query to Zettair (denoted as $\mathsf{Ccat}(C)$); and by sending the title of the topic T as a query to Zettair (denoted as $\mathsf{Tcat}(C)$). On the training data set we found that these two approaches were less effective than the $\mathsf{TCcat}(C)$ approach [10]. However, we used the $\mathsf{Tcat}(C)$ category set in the ad-hoc runs where the target category is not provided.

Task 2 Here, the categories attached to entity examples are likely to correspond to very specific categories, just like those attached to the answer entities. We define a similarity function that computes the ratio of common categories between

the set of categories attached to an answer entity page $\mathsf{cat}(t)$ and the set of the union of the categories attached to entity examples $\mathsf{cat}(E)$:

$$S_C(t) = \frac{|\mathsf{cat}(t) \cap \mathsf{cat}(E)|}{|\mathsf{cat}(E)|} \tag{2}$$

3.3 Exploiting locality of links

For task 2, exploiting locality of links around entity examples can significantly improve the effectiveness of entity ranking [8]. The idea is that entity references (links) that are located in close proximity to the entity examples, especially in list-like elements, are likely to refer to more relevant entities than those referred to by links in other parts of the page. Here, the very notion of *list* involves grouping together objects of the same (or similar) nature. We are therefore looking for links that co-occur with links to entity examples in such list-like elements.

Consider the example of the Euro page shown in Fig. 2, where France, Germany and Spain are the three entity examples (as shown in Fig. 1). We see that the 15 countries that are members of the Eurozone are all listed in the same paragraph with the three entity examples. In fact, there are other contexts in this page where those 15 countries also co-occur together. By contrast, although there are a few references to the United Kingdom in the Euro page, it does not occur in the same context as the three examples (except for the page itself).

We have identified in the Wikipedia collections three types of elements that correspond to the notion of lists: paragraphs (tag `p`); lists (tags `normallist`, `numberlist`, and `definitionlist`); and tables (tag `table`). We use an algorithm for identifying the (static) element contexts on the basis of the leftmost occurrence of any of the pre-defined tags in the absolute XPaths of entity examples. The resulting list of element contexts is sorted in a descending order according to the number of distinct entity examples contained by the element. If two elements contain the same number of distinct entity examples, the one that has a longer XPath length is ranked higher. Finally, starting from the highest ranked element, we filter all the elements in the list that either contain or are contained by that element. We end up with a final list of (one or more) non-overlapping elements that represent the statically defined contexts for the page.[5]

Consider Table 1, where the links to entity examples are identified by their absolute XPath notations. The three static contexts that will be identified by the above algorithm are the elements `p[1]`, `normallist[1]` and `p[3]`. The first two element contexts contain the three (distinct) examples, while the last one contains only one entity example.

The drawback of this approach is that it requires a predefined list of static elements that is completely dependent on the collection. The advantage is that

[5] In the case when there are no occurrences of the pre-defined tags in the XPath of an entity example, the document element (`article[1]`) is chosen to represent the element context.

Table 1. List of links referring to entity examples (France, Germany, and Spain), extracted from the page 9272.html, for the INEX 2007 XER topic shown in Fig. 1.

Page		Links		
ID	Name	XPath	ID	Name
9472	Euro	/article[1]/body[1]/p[1]/collectionlink[7]	10581	France
9472	Euro	/article[1]/body[1]/p[1]/collectionlink[8]	11867	Germany
9472	Euro	/article[1]/body[1]/p[1]/collectionlink[15]	26667	Spain
9472	Euro	/article[1]/body[1]/p[3]/p[5]/collectionlink[6]	11867	Germany
9472	Euro	/article[1]/body[1]/normallist[1]/item[4]/collectionlink[1]	10581	France
9472	Euro	/article[1]/body[1]/normallist[1]/item[5]/collectionlink[2]	11867	Germany
9472	Euro	/article[1]/body[1]/normallist[1]/item[7]/collectionlink[1]	26667	Spain
9472	Euro	/article[1]/body[1]/normallist[1]/item[8]/collectionlink[1]	26667	Spain

the contexts are fast to identify. We have also experimented with an alternative algorithm that dynamically identifies the link contexts by utilising the underlying XML document structure. On the training data set, we found that this algorithm does not significantly improve the entity ranking performance compared to the algorithm that uses the static contexts [8].

3.4 Score Functions and parameters

The core of our entity ranking approach is based on combining different scoring functions for an answer entity page, which we now describe in more detail.

LinkRank score The linkrank function calculates a score for a page, based on the number of links to this page, from the first N pages returned by the search engine in response to the query. The number N has been kept to a relatively small value mainly for performance issues, since Wikipedia pages contain many links that would need to be extracted. We carried out some experiments with different values of N and found that N=20 was a good compromise between performance and discovering more potentially good entities.

The linkrank function can be implemented in a variety of ways. We have implemented a linkrank function that, for an answer entity page t, takes into account the Zettair score of the referring page $z(p)$, the number of distinct entity examples in the referring page $\#ent(p)$, and the locality of links around the entity examples:

$$S_L(t) = \sum_{r=1}^{N} \left(z(p_r) \cdot g(\#ent(p_r)) \cdot \sum_{l_t \in L(p_r,t)} f(l_t, c_r | c_r \in C(p_r)) \right) \qquad (3)$$

where $g(x) = x + 0.5$ (we use 0.5 to allow for cases where there are no entity examples in the referring page); l_t is a link that belongs to the set of links

$L(p_r, t)$ that point from the page p_r to the answer entity t; c_r belongs to the set of contexts $C(p_r)$ around entity examples found for the page p_r; and $f(l_t, c_r)$ represents the weight associated to the link l_t that belongs to the context c_r.

The weighting function $f(l_r, c_r)$ is represented as follows:

$$f(l_r, c_r) = \begin{cases} 1 & \text{if } c_r = p_r \text{ (the context is the full page)} \\ \\ 1 + \#ent(c_r) & \text{if } c_r = e_r \text{ (the context is an XML element)} \end{cases}$$

A simple way of defining the context of a link is to use its full embedding page [11]. In this work we use smaller contexts using predefined types of elements such as paragraphs, lists and tables (as described in sub-section 3.3).

Category similarity score As described in sub-section 3.2, the category score $S_C(t)$ for the two tasks is calculated as follows:

task 1

$$S_C(t) = \frac{|\mathsf{cat}(t) \cap \mathsf{cat}(C)|}{|\mathsf{cat}(C)|} \tag{4}$$

For task 1, we consider variations on the category score $S_C(t)$ based on lexical similarities of category names (see sub-section 3.2), by replacing $\mathsf{cat}(C)$ with $\mathsf{TCcat}(C)$.

task 2

$$S_C(t) = \frac{|\mathsf{cat}(t) \cap \mathsf{cat}(E)|}{|\mathsf{cat}(E)|} \tag{5}$$

On the training data set, we found that extending the set of categories attached to both entity examples and answer entities did not increase the entity ranking performance [10], and so for task 2 we do not use any category extensions.

Z score The Z score assigns the initial Zettair score to an answer entity page. If the answer page does not appear among the initial ranked list of pages returned by Zettair, then its Z score is zero:

$$S_Z(t) = \begin{cases} z(t) & \text{if page } t \text{ was returned by Zettair} \\ \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Global score The global score $S(t)$ for an answer entity page is calculated as a linear combination of three normalised scores, the linkrank score $S_L(t)$, the category similarity score $S_C(t)$, and the Z score $S_Z(t)$:

$$S(t) = \alpha S_L(t) + \beta S_C(t) + (1 - \alpha - \beta) S_Z(t) \quad (7)$$

where α and β are two parameters that can be tuned differently depending on the entity retrieval task.

We consider some special cases that allow us to evaluate the effectiveness of each module in our system: $\alpha = 1, \beta = 0$, which uses only the linkrank score; $\alpha = 0, \beta = 1$, which uses only the category score; and $\alpha = 0, \beta = 0$, which uses only the Z score.[6] More combinations for the two parameters are explored in the training phase of our system. The optimal combination is then used on the testing data set.

4 Experimental results

In this section, we present results that investigate the effectiveness of our entity ranking approach when applied to both the INEX 2007 XER and ad hoc tracks.

We first tune the system parameters using the training collection, and then we apply the optimal values on the test collection. We submitted three runs for task 1 and three runs for task 2. For this track, we aim at investigating the impact of using various category and linkrank similarity techniques on the entity ranking performance. We also compare the performances of our entity ranking runs to that achieved by a full-text retrieval run. For the ad hoc track, we submitted three entity ranking runs that correspond to the three individual modules of our system and compare it with the full text Zettair run submitted by RMIT. For this track, we aim at investigating the impact of using our entity ranking approach on the ad hoc retrieval performance.

4.1 XER training data set (28 topics)

The XER training data set was developed by our participating group. It is based on a selection of topics from the INEX 2006 ad hoc track. We chose 27 topics that we considered were of an "entity ranking" nature, where for each page that had been assessed as containing relevant information, we reassessed whether or not it was an entity answer, and whether it *loosely* belonged to a category of entity we had *loosely* identified as being the target of the topic. If there were entity examples mentioned in the original topic these were used as entity examples in the entity topic. Otherwise, a selected number (typically 2 or 3) of entity examples were chosen somewhat arbitrarily from the relevance assessments. We also added the Euro topic example (shown in Fig. 1) from the original INEX description of the XER track [3], resulting in total of 28 entity ranking topics.

[6] This is not the same as the plain Zettair score, as apart from answer entities corresponding to the highest N pages returned by Zettair, the remaining entity answers are all generated by extracting links from these pages, which may or may not correspond to the initial 1500 pages retrieved by Zettair.

Table 2. Performance scores for Zettair and our three XER submitted runs on the training data set (28 topics), obtained for task 1 with different evaluation measures. For each measure, the best performing score is shown in bold.

				P[r]			
Run	cat-sim	α	β	5	10	R-prec	MAP
Zettair		–	–	0.229	0.232	0.208	0.172
run 1	$\mathsf{cat}(C)$-$\mathsf{cat}(t)$	0.0	1.0	0.229	0.250	0.215	0.196
run 2	$\mathsf{TCcat}(C)$-$\mathsf{cat}(t)$	0.0	1.0	0.307	0.318	0.263	0.242
run 3	$\mathsf{TCcat}(C)$-$\mathsf{cat}(t)$	0.1	0.8	**0.379**	**0.361**	**0.338**	**0.287**

We use mean average precision (MAP) as our primary method of evaluation, but also report results using several alternative measures that are typically used to evaluate the retrieval performance: mean of P[5] and P[10] (mean precision at top 5 or 10 entities returned), and mean R-precision (R-precision for a topic is the P[R], where R is the number of entities that have been judged relevant for the topic). For task 1 all the relevant entities in the relevance assessments are used to generate the scores, while for task 2 we remove the entity examples both from the list of returned answers and from the relevance assessments, as the task is to find entities other than the provided examples.

Task 1 Table 2 shows the performance scores on the training data set for task 1, obtained for Zettair and our three submitted XER runs. Runs 1 and 2 use only the category module ($\alpha = 0.0$, $\beta = 1.0$) while run 3 uses a combination of linkrank, category, and Z scores ($\alpha = 0.1$, $\beta = 0.8$). Runs 2 and 3 use lexical similarity for extending the target categories.

We observe that the three entity ranking runs outperform the plain Zettair run, which suggests that using full-text retrieval alone is not an effective entity ranking strategy. The differences in performance between each of the three runs and Zettair are statistically significant ($p < 0.05$) only for the two entity ranking runs that use lexical similarity between categories (runs 2 and run 3 in Table 2).

When comparing the performances of the runs that use only the category module, we observe that run 2 that uses lexical similarity between category names ($\mathsf{TCcat}(C)$) is more effective than the run that uses the target categories only ($\mathsf{cat}(C)$). With MAP, the difference in performance between the two runs is statistically significant ($p < 0.05$). We also observe that the third run, which uses combined scores coming out from the three modules, performs the best among the three. To find the optimal values for the two combining parameters for this run, we calculated MAP over the 28 topics in the training data set as we varied α from 0 to 1 in steps of 0.1. For each value of α, we also varied β from 0 to $(1 - \alpha)$ in steps of 0.1. We found that the highest MAP score (0.287) is achieved for $\alpha = 0.1$ and $\beta = 0.8$ [10]. This is a 19% relative performance improvement over the best score achieved by using only the category module ($\alpha 0.0$–$\beta 1.0$). This performance improvement is statistically significant ($p < 0.05$).

Table 3. Performance scores for Zettair and our three XER submitted runs on the training data set (28 topics), obtained for task 2 with different evaluation measures. For each measure, the best performing score is shown in bold.

				P[r]			
Run	cat-sim	α	β	5	10	R-prec	MAP
Zettair	–	–	–	0.229	0.232	0.208	0.172
run 1	cat(E)-cat(t)	1.0	0.0	0.214	0.225	0.229	0.190
run 2	cat(E)-cat(t)	0.0	1.0	0.371	0.325	0.319	0.318
run 3	cat(E)-cat(t)	0.2	0.6	**0.500**	**0.404**	**0.397**	**0.377**

Task2 Table 3 shows the performance scores on the training data set for task 2, obtained for Zettair and our three submitted XER runs. As with task 1, we again observe that the three entity ranking runs outperform the plain Zettair run. With the first two runs, we want to compare two entity ranking approaches: the first that uses scores coming out from the linkrank module (run 1), and the second that uses scores coming out from the category module (run 2). We observe that using categories is substantially more effective than using the linkrank scores. With MAP, the difference in performance between the two runs is statistically significant ($p < 0.05$).

Run 3 combines the scores coming out from the three modules. To find the optimal values for the two combining parameters for this run, we again varied the values for parameters α and β and we found that the highest MAP score (0.377) was achieved for $\alpha = 0.2$ and $\beta = 0.6$ [8]. This is a 19% relative performance improvement over the best score achieved by using only the category module. This performance improvement is statistically significant ($p < 0.05$).

XER testing data set (73 topics)

Runs description Table 4 lists the six XER and four ad hoc runs that we submitted for evaluation in the INEX 2007 XER and ad hoc tracks, respectively. With the exception of the plain Zettair run, all the runs were created by using our entity ranking system. However, as seen in the table the runs use various parameters whose values are mainly dependent on the task. Specifically, runs differ depending on whether (or which) Zettair category index is used, which of the two types of link contexts is used, whether categories or example entities are used from the topic, and which combination of values is assigned to the α and β parameters.

For example, the run "run 3", which was submitted for evaluation in task 1 of the INEX 2007 XER track, can be interpreted as follows. The Wikipedia full-text Zettair index is used to extract the top 20 ranked Wikipedia pages, using the title from the INEX topic as a query. After extracting all links to potential answer entities from these 20 pages, the Zettair index of category names is used

Table 4. List of six XER and four ad hoc runs submitted for evaluation in the INEX 2007 XER and ad hoc tracks, respectively. "Cat-sim" stands for category similarity, "Ctx" for context, "Cat" for categories, "Ent" for entities, "T" for title, "TC" for title and categories, "C" for category names, "CE" for category and entity names, "FC" for full page context, and "EC" for element context.

				Category index				**Topic**	
Run ID	cat-sim	α	β	Query	Type	M	Ctx	Cat	Ent
Zettair		–	–	–	–	–	–	–	–
XER task 1									
run 1	$\mathsf{cat}(C)$-$\mathsf{cat}(t)$	0.0	1.0	–	–	–	FC	Yes	No
run 2	$\mathsf{TCcat}(C)$-$\mathsf{cat}(t)$	0.0	1.0	TC	C	5	FC	Yes	No
run 3	$\mathsf{TCcat}(C)$-$\mathsf{cat}(t)$	0.1	0.8	TC	C	5	FC	Yes	No
XER task 2									
run 1	$\mathsf{cat}(E)$-$\mathsf{cat}(t)$	1.0	0.0	–	–	–	EC	No	Yes
run 2	$\mathsf{cat}(E)$-$\mathsf{cat}(t)$	0.0	1.0	–	–	–	EC	No	Yes
run 3	$\mathsf{cat}(E)$-$\mathsf{cat}(t)$	0.2	0.6	–	–	–	EC	No	Yes
Ad hoc retrieval task									
run 1	$\mathsf{Tcat}(C)$-$\mathsf{cat}(t)$	0.0	0.0	T	CE	10	FC	No	No
run 2	$\mathsf{Tcat}(C)$-$\mathsf{cat}(t)$	1.0	0.0	T	CE	10	FC	No	No
run 3	$\mathsf{Tcat}(C)$-$\mathsf{cat}(t)$	0.0	1.0	T	CE	10	FC	No	No

to extract the top five ranked categories, using both the title and the category names (TC) from the INEX topic as a query. This set of five categories is used as an input set of target categories by the category module. The full page context (FC) is used to calculate the scores in the linkrank module. The final scores for answer entities are calculated by combining the scores coming out of the three modules ($\alpha = 0.1$, $\beta = 0.8$).

Results Results for XER task 1 and task 2 on the testing data set will be reported when they become available. The results obtained for our runs will also be compared with the results obtained for runs submitted by other track participants.

4.2 Ad hoc data set (99 topics)

There are no target categories and example entities provided for the ad hoc task. However, we wanted to apply our algorithm to test 1) whether some indication of the page categories would improve the retrieval performance, and 2) whether extracting new entities from the pages returned by Zettair would be beneficial for ad hoc retrieval.

We submitted four runs for the INEX 2007 ad hoc track: Zettair, representing a full-text retrieval run, and three entity ranking runs. As shown in Table 4, run 1 uses only the Z module for ranking the answer entities, run 2 uses only the linkrank module, while run3 uses only the category module. For each INEX 2007

Table 5. Performance scores for Zettair and our three XER submitted runs on the ad hoc data set (99 topics), obtained with different evaluation measures. For each measure, the best performing score is shown in bold.

			P[r]				Foc	RiC	BiC
Run	α	β	5	10	R-prec	MAP	iP[0.01R]	MAgP	MAgP
Zettair	–	–	**0.513**	**0.469**	**0.326**	**0.292**	**0.379**	**0.088**	**0.195**
run 1	0.0	0.0	**0.513**	**0.469**	0.303	0.247	0.379	0.075	0.165
run 2	1.0	0.0	0.339	0.289	0.170	0.121	0.235	0.031	0.070
run 3	0.0	1.0	0.406	0.368	0.208	0.157	0.287	0.050	0.115

ad hoc topic, we create the set of target categories by sending the title T of the query to the Zettair index of categories that has been created by using the names of the categories and the names of all their attached entities as corresponding documents.

Table 5 shows the performance scores on INEX 2007 the ad hoc data set, obtained for Zettair and our three submitted entity ranking runs. Two retrieval scenarios are distinguished in the table: a *document retrieval* scenario (the first four result columns in Table 5), where we compare how well the runs retrieve relevant documents; and a *focused retrieval* scenario (the last three result columns in Table 5), where we compare how well the runs retrieve relevant information within documents.

For the document retrieval scenario, we observe that Zettair outperforms the other three XER runs. The differences in performance between Zettair and any of these three runs are statistically significant ($p < 0.05$). Among the three XER runs, the run that only uses the Z scores performs significantly better than the other two, followed by the run that only uses the category scores which in turn performs significantly better than the worst performing run that only uses the linkrank scores.

The same trend among the four runs is observed across the three sub-tasks of the focused retrieval scenario, where again Zettair is able to better identify and retrieve the relevant information compared to the other three XER runs.

The obvious conclusion of our our ad hoc experiments is that Zettair, which is especially designed for ad hoc retrieval, performs better than our entity ranking system specifically designed for entity retrieval.

5 Conclusion and future work

We have presented our entity ranking system for the INEX Wikipedia XML document collection which is based on exploiting the interesting structural and semantic properties of the collection. On the training data, we have shown that our system outperforms the full text search engine in the task of ranking entities.

On the other hand, using our entity ranking system for ad-hoc retrieval did not result in any improvement over the full-text search engine. This confirms

our hypothesis that that tasks of ad hoc retrieval and entity ranking are very different. Once the official results for the INEX 2007 XML entity ranking track are available, we will make further analysis and compare the effectiveness of our entity ranking system to those achieved by other participating systems.

Acknowledgements

Part of this work was completed while James Thom was visiting INRIA in 2007.

References

1. D. Awang Iskandar, J. Pehcevski, J. A. Thom, and S. M. M. Tahaghoghi. Social media retrieval using image features and structured text. 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 358–372, 2007.
2. E. Blanchard, P. Kuntz, M. Harzallah, and H. Briand. A tree-based similarity for evaluating concept proximities in an ontology. In *Proceedings of 10th conference of the International Fedederation of Classification Societies*, pages 3–11, Ljubljana, Slovenia, 2006.
3. A. P. de Vries, J. A. Thom, A.-M. Vercoustre, N. Craswell, and M. Lalmas. INEX 2007 Entity ranking track guidelines. In *INEX 2007 Workshop Pre-Proceedings*, 2007 (to appear).
4. L. Denoyer and P. Gallinari. The Wikipedia XML corpus. *SIGIR Forum*, 40(1):64–69, 2006.
5. S. Fissaha Adafre, M. de Rijke, and E. T. K. Sang. Entity retrieval. In *Proceedings of International Conference on Recent Advances in Natural Language Processing (RANLP - 2007), September 27-29, Borovets, Bulgaria*, 2007.
6. J. M. Kleinberg. Authoritative sources in hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
7. C. Middleton and R. Baeza-Yates. A comparison of open source search engines. Technical report, Universitat Pompeu Fabra, Barcelona, Spain, 2007. http://wrg.upf.edu/WRG/dctos/Middleton-Baeza.pdf.
8. J. Pehcevski, A.-M. Vercoustre, and J. A. Thom. Exploiting locality of Wikipedia links in entity ranking. Submitted for publication, 2007.
9. I. Soboroff, A. P. de Vries, and N. Craswell. Overview of the TREC 2006 Enterprise track. In *Proceedings of the Fifteenth Text REtrieval Conference (TREC 2006)*, pages 32–51, 2006.
10. J. A. Thom, J. Pehcevski, and A.-M. Vercoustre. Use of Wikipedia categories in entity ranking. In *Proceedings of the 12th Australasian Document Computing Symposium*, Melbourne, Australia, 2007 (to appear).
11. A.-M. Vercoustre, J. A. Thom, and J. Pehcevski. Entity ranking in Wikipedia. In *Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC08)*, Fortaleza, Brazil, 2008 (to appear).
12. J. Yu, J. A. Thom, and A. Tam. Ontology evaluation using Wikipedia categories for browsing. In *Proceedings of Sixteenth ACM Conference on Information and Knowledge Management (CIKM '07)*, Lisboa, Portugal, 2007.