Exploiting Locality of Wikipedia Links in Entity Ranking

Jovan Pehcevski, Anne-Marie Vercoustre, James Thom

To cite this version:

Jovan Pehcevski, Anne-Marie Vercoustre, James Thom. Exploiting Locality of Wikipedia Links in Entity Ranking. The 30th annual European Conference on Information Retrieval (ECIR), Apr 2008, Glasgow, Scotland, United Kingdom. inria-00232794

HAL Id: inria-00232794

https://hal.inria.fr/inria-00232794

Submitted on 1 Feb 2008

Exploiting Locality of Wikipedia Links in Entity Ranking

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

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

Abstract. Information retrieval from web and XML document collections is ever more focused on returning entities instead of web pages or XML elements. There are many research fields involving named entities; one such field is known as entity ranking, where one goal is to rank entities in response to a query supported with a short list of entity examples. In this paper, we describe our approach to ranking entities from the Wikipedia XML document collection. Our approach utilises the known categories and the link structure of Wikipedia, and more importantly, exploits link co-occurrences to improve the effectiveness of entity ranking. Using the broad context of a full Wikipedia page as a baseline, we evaluate two different algorithms for identifying narrow contexts around the entity examples: one that uses predefined types of elements such as paragraphs, lists and tables; and another that dynamically identifies the contexts by utilising the underlying XML document structure. Our experiments demonstrate that the locality of Wikipedia links can be exploited to significantly improve the effectiveness of entity ranking.

1 Introduction

The traditional entity extraction problem is to extract named entities from plain text using natural language processing techniques or statistical methods and intensive training from large collections. The primary goal is to tag those entities and use the tag names to support future information retrieval. *Entity ranking* has recently emerged as a research field that aims at retrieving entities as answers to a query. Here 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 [14], except that instead of ranking people who are experts in the given topic, other types of entities such as organisations, countries, or locations can also be retrieved and ranked. For example, the query "European countries where I can pay with Euros" should return a list of entities representing relevant countries, and not a list of entities about the Euro and similar currencies.

The Initiative for the Evaluation of XML retrieval (INEX) has a new track on entity ranking, using Wikipedia as its XML document collection [7]. Two tasks are explored by the INEX 2007 entity ranking 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. The inclusion of the target category (in the first task) and entity examples (in the second task) makes these quite different tasks from the task of full-text retrieval, and the combination of the query and entity examples (in the second task) makes it quite different from the task addressed by an application such as Google Sets[3] where only entity examples are provided.

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 entity examples).
2. A good entity page is a page associated with categories close to the categories of the entity examples.
3. A good entity page is pointed to by a page answering the query; this is an adaptation of the HITS [10] algorithm to the problem of entity ranking.
4. A good entity page is pointed to by contexts with many occurrences of the entity examples. 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.

Specifically, we focus on whether the locality of Wikipedia links around entity examples can be exploited to improve the effectiveness of entity ranking.

2 Related work

In this section, we review some related work on link analysis and entity disambiguation and extraction.

Link analysis To calculate the similarity between a document and a query, most information retrieval (IR) systems use statistical information concerning the distribution of the query terms, both within the document and the collection as a whole. However, in hyperlinked environments, such as the World Wide Web and Wikipedia, link analysis is important. PageRank [3] and HITS [10] are two of the most popular algorithms that use link analysis to improve web search.

We use the idea behind PageRank and HITS in our approach; however, instead of counting every possible link referring to an entity page in the collection (as with PageRank), or building a neighbourhood graph (as with HITS), we only consider pages that are pointed to by a selected number of top-ranked pages for the query. This also makes our link ranking algorithm to be query-dependent (just like HITS), which allows for it to be dynamically calculated at query time.

[3] http://labs.google.com/sets

Cai et al. [4] recognise that most popular linkrank algorithms treat a web page as a single node, despite the fact that the page may contain multiple semantic blocks. Using the visual presentation of a page to extract the semantic structure, they adapted PageRank and HITS to deal with block nodes rather than full page nodes. Nie et al. [12] propose a topical link analysis model that formalises the idea of splitting the credit (the authority score) of a source page into different topics based on topical distribution. Our entity ranking approach is based on a similar idea, except that instead of using topics for discrimination we use list-like contexts around the entity examples.

Entity disambiguation and extraction Kazama and Torisawa [9] explore the use of Wikipedia as external knowledge to improve named entity recognition, by using the first sentence of a Wikipedia page to infer the category of the entity attached to that page. These categories are then used as features in their named entity tagger. We do not use inferred categories in our approach; instead, we use categories that were explicitly associated with the entity page by Wikipedia authors. Cucerzan [6] also uses Wikipedia for entity disambiguation by exploiting (amongst other features) co-references in static contexts such as titles, links, paragraphs and lists. Callan and Mitamura [5] investigate if the entity extraction rules can be dynamically generated. Their rules are based on heuristics exploiting a few pre-defined HTML contexts such as lists and tables. The contexts are weighted according to the number of contained examples; the best contexts are then used to dynamically extract new data. We use pre-defined contexts in our entity ranking approach; however, we also develop a new algorithm that dynamically determines the contexts around entity examples.

ESTER [2] was recently proposed as a system for searching text, entities and relations. ESTER relies on the Wikipedia links to identify the entities and on the context of the links for disambiguation (using 20 words around the anchor text instead of just the anchor text). This approach primarily focuses on improving the efficiency of the proposed system, while we are more interested in improving the effectiveness of entity ranking.

3 The Wikipedia XML document 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 [8] have developed an XML-based corpus based on a snapshot of the Wikipedia, which has been used by various INEX tracks in 2006. It differs from the real Wikipedia in some respects (size, document format, category tables), but it is a very realistic approximation.

Entities in Wikipedia In Wikipedia, an entity is generally associated with an article (a Wikipedia page) describing this entity. For example, there is a page for

> “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. 1. Extract from the Euro Wikipedia page

every country, most famous people or organisations, places to visit, and so forth. In Wikipedia nearly everything can be seen as an entity with an associated page.

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. For example, in the Euro page (see Fig. 1), 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; more specifically, these countries are all “European Union member states”, which brings us to the notion of category in Wikipedia.

Categories in Wikipedia Wikipedia offers categories that authors can associate with Wikipedia pages. 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. 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”.

4 Our entity ranking approach

We are addressing the task of ranking entities in answer to a query supplied with a few examples (task 2). However, our approach can also be used for entity ranking tasks where the category of the target entities is given and no examples are provided (task1).

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

4.1 Architecture

Our entity ranking approach 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 used Zettair[4] as our choice for a full-text search engine. Zettair is a full-text IR system developed by RMIT University, which returns pages ranked by their similarity scores to the query. Zettair is "one of the most complete engines" according to a recent comparison of open source search engines [11]. We used the Okapi BM25 similarity measure which was shown to be very effective on the INEX 2006 Wikipedia test collection [1].

The architecture of our approach is described as follows. The *topic* module takes an INEX topic as input and generates the corresponding Zettair query and the list of entity examples (as one option, the names of the entity examples may be added to the query). The *search* module sends the query to Zettair and returns a list of scored Wikipedia pages. The *link extraction* module extracts the links to target entities from a selected number of highly ranked pages, together with the information about the paths of the links (using an XPath notation). The *linkrank* module calculates a weight for a target entity based on (amongst other factors) the number of links to this entity and the number of entity examples that appear in the context of the link. The *category similarity* module calculates a weight for a target entity based on the similarity of its categories with that of the entity examples. The *full-text IR* module calculates a weight for a target entity based on its initial Zettair score. Finally, the global score for a target entity is calculated as a linear combination of three normalised scores coming out of the last three modules.

The above architecture provides a general framework for entity ranking which allows for replacing some modules 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 modules by varying the parameters and to globally evaluate the entity ranking approach.

4.2 Score functions and parameters

The global score of an entity page is derived by combining three separate scores: a linkrank score, a category score, and a full-text similarity score.

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 parameter N has been kept to a relatively small value mainly for performance purposes, since Wikipedia pages contain many links that would need to be extracted. We carried out experiments with different values of the parameter N, by varying it between 5 and 100 with a step of 5, 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 a target entity page t, takes into account the Zettair score of the referring page $z(p)$, the number of distinct entity

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

examples in the referring page $\#ent(p)$, and the locality of links around the entity examples:

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

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 to the target entity t from the page p_r; c_r is a context around entity examples that belongs to a set of contexts $C(p_r)$ 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 contexts are explained in full detail in sub-section 4.3.

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

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

Category similarity score To calculate the category similarity score, we use a very basic similarity function that computes the ratio of common categories between the set of categories associated with the target page $\mathsf{cat}(t)$ and the set of the union of the categories associated with the entity examples $\mathsf{cat}(E)$:

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

Z score The full-text (Z) score assigns the initial Zettair score to a target entity page. If the target entity does not appear among the initial ranked list of pages returned by Zettair, then its Z score is zero:

$$S_Z(t) = \begin{cases} z(t) & \text{if page } t \text{ was returned by Zettair} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

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

$$S(t) = \alpha S_L(t) + \beta S_C(t) + (1 - \alpha - \beta) S_Z(t) \tag{4}$$

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: $\alpha = 1, \beta = 0$, 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.[5] More combinations of the two parameters are explored in the tuning phase of our approach (section 5).

4.3 Exploiting locality of links

The main assumption behind the idea of exploiting locality of links in entity ranking is that references to entities (links) located in close proximity to the entity examples, which typically appear in list-like contexts, are more likely to represent relevant entities than links that appear in other parts of the page. Here, the notion of *list* refers to grouping together objects of the same (or similar) nature. The aim is therefore to assign a bigger weight to links that co-occur with links to entity examples in such list-like contexts.

Consider the example of the Euro page shown in Fig. 1. Let us assume that the topic is “European countries where I can pay with Euros”, and France, Germany and Spain are three entity examples. 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).

Statically defined contexts We have identified three types of elements that correspond to list-like contexts in the Wikipedia XML document collection: paragraphs (tag `p`); lists (tags `normallist`, `numberlist`, and `definitionlist`); and tables (tag `table`). We design two algorithms for identifying the static contexts: one that identifies the context on the basis of the leftmost occurrence of the pre-defined tags (`StatL`), and another that uses the rightmost occurrence of the pre-defined tags to identify the context (`StatR`). We do this to investigate whether the recursive occurrences of the same tag, as often found in many XML documents in the INEX Wikipedia collection, has an impact on the ability to better identify relevant entities.

Consider Table 1, where the links to entity examples are identified by their absolute XPath notations. The three non-overlapping elements that will be identified by the `StatL` algorithm are the elements `p[1]`, `p[3]`, and `normallist[1]`, while with the `StatR` algorithm `p[5]` will be identified instead of `p[3]` in addition to also identifying the other two elements.

The main drawback of the static approach is that it requires a pre-defined list of element contexts which is totally dependent on the document collection. The advantage is that, once defined, the list-like contexts are easy to identify.

[5] This is not the same as the plain Zettair score, as apart from target entities corresponding to the highest N pages returned by Zettair, the remaining entities are all generated by extracting links from these pages, which may or may not correspond to the ranked pages returned by Zettair.

Table 1. List of links referring to entity examples (France, Germany, and Spain), extracted for the Euro topic.

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

Dynamically defined contexts To determine the contexts dynamically, we adapted the concept of *coherent retrieval elements* [13] initially used to identify the appropriate granularity of elements to return as answers in XML retrieval.

For the list of extracted entities corresponding to entity examples, a *Coherent Retrieval Element* (CRE) is defined as an element that represents the *lowest common ancestor* (LCA) of at least two entity examples. To identify the CREs, we sequentially process the list of extracted entity examples by considering every pair of elements, starting from the first element down to the element preceding the last element in the list. For each pair of elements, their LCA is chosen to represent a dynamic context (a CRE). Starting from the first identified CRE, we filter the overlapping elements and end up with a final list of (one or more) non-overlapping CREs that represent the dynamically defined contexts for the page.[6] We refer to this algorithm as `DynCRE`.

For example, the two dynamic contexts that will be identified for the list of extracted entity examples shown in Table 1 are `p[1]` and `normallist[1]`. Although `body[1]` was also initially identified as a CRE, it was subsequently filtered from the final list since it overlaps with `p[1]` (the first identified CRE).

The main advantage of the dynamic approach is that it is independent of the document collection, and it does not require a pre-defined list of contexts. The possible drawback is that narrow contexts containing only one entity example (such as `p[5]` in Table 1) are never identified.

5 Experimental results

We now present results that investigate the effectiveness of our entity ranking approach when using different types of contexts around the entity examples.

[6] When the page contains *exactly one* entity example, the document element (`article[1]`) is chosen to represent a CRE.

5.1 Test collection

Since there was no existing set of topics with relevance assessments for entity ranking, we developed our own test collection, which we made available as a training set for other participants in the INEX 2007 entity ranking track. So for these experiments we used our own test collection based on a selection of topics from the INEX 2006 ad hoc track, since most of these topics reflect real-life tasks represented by queries very similar to the short Web queries. 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 usually 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. To this set of 27 topics we also added the Euro topic example that we had created by hand from the original INEX description of the entity ranking track [7], resulting in total of 28 entity ranking topics.

We use mean average precision (MAP) as our primary method of evaluation, but also report results using several alternative IR measures: mean of P[5] and P[10] (mean precision at top 5 or 10 entities returned), and mean R-precision. 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 those provided.

5.2 Full page context

We used the context of the full page to determine suitable values for the parameters α and β, and also to try out some minor variations to our entity ranking approach (such as whether or not to include the names of the entity examples in the query sent to Zettair).

We calculated MAP over the 28 topics in our test collection, 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 (0.3570) on this data set is achieved for $\alpha = 0.1$ and $\beta = 0.8$. We also trained using mean R-precision instead of MAP as our evaluation measure, but we also observed the same optimal values for the two parameters.

We used a selected number of runs to carry out a more detailed investigation of the performance achieved by each independent module and by the optimal module combination. We also investigated whether adding names of the entity examples to the query sent to Zettair would have a positive performance impact. The results of these investigations are shown in Tables 2(Q) and 2(QE).

Several observations can be drawn from these results. First, adding names of the entity examples to the query sent to Zettair generally performs worse for all but the linkrank module, for which we see a consistent performance improvement. Second, different optimal values are observed for the two parameters in the two tables, which suggest that adding the entity examples to the query can

Table 2. Performance scores for runs using the context of the full page, obtained by different evaluation measures. Queries sent to Zettair include only terms from the topic title (Q), or terms from the topic title and the names of entity examples (QE). For each measure, the best performing score is shown in bold.

	P[r]			
Run	5	10	R-prec	MAP
Zettair	0.2286	0.2321	0.2078	0.1718
α0.0–β0.0	0.2286	0.2321	0.2135	0.1780
α0.0–β1.0	0.3643	0.3071	0.3151	0.3089
α1.0–β0.0	0.1571	0.1571	0.1385	0.1314
α0.1–β0.8	**0.4714**	**0.3857**	**0.3902**	**0.3570**
α0.2–β0.6	0.4357	0.3786	0.3751	0.3453

(Q) Topic title

	P[r]			
Run	5	10	R-prec	MAP
Zettair	0.2000	0.1714	0.1574	0.1427
α0.0–β0.0	0.2000	0.1714	0.1775	0.1533
α0.0–β1.0	0.3357	0.2821	0.2749	0.2674
α1.0–β0.0	0.1857	0.1750	0.1587	0.1520
α0.1–β0.8	0.3357	0.3286	0.3109	0.3140
α0.2–β0.6	**0.3429**	**0.3357**	**0.3362**	**0.3242**

(QE) Topic title and entity examples

dramatically influence the retrieval performance. Third, we observe that the best entity ranking approaches are those that combine the ranking evidence from the three modules (runs α0.1–β0.8 for Q and α0.2–β0.6 for QE). With MAP, these two runs perform significantly better ($p < 0.05$) than the plain Zettair full-text retrieval run, and they are also significantly better than any of the three runs representing each individual module in our entity ranking approach.

These results therefore show that the global score (the combination of the three individual scores), optimised in a way to give more weight on the category score, brings the best value in retrieving the relevant entities for the INEX Wikipedia document collection. However, the results also show that using only the linkrank module and the context of the full page results in a very poor entity ranking strategy, which is why below we also experiment with narrow contexts.

5.3 Static and dynamic contexts

We now investigate whether using smaller and more narrow contexts has a positive impact on the effectiveness of entity ranking. Tables 3(Q) and 3(QE) show the results of this investigation. These results reflect the case when only the linkrank module (α1.0–β0.0) is used by our entity ranking approach.

As in the case with using the full page context, for all the four runs we observe a consistent performance improvement when names of the entity examples are added to the query sent to Zettair. Importantly, when compared to the baseline (the full page context), we observe a substantial increase in performance for the three runs that use smaller and more narrow contexts, irrespective of the type of query used. These increases in performance are all statistically significant ($p < 0.05$). However, the type of query sent to Zettair (Q or QE) seems to have an impact on the best performance that could be achieved by these three runs. Specifically, with MAP the `StatL` run performs best among the three runs when only the topic title is used as an input query (Q), while the `StatR` run is best when using terms from the topic title and the names of entity examples (QE). In

Table 3. Performance scores for runs using different types of contexts in the linkrank module (α1.0–β0.0), obtained by different evaluation measures. Queries sent to Zettair include only terms from the topic title (Q), or terms from the topic title and the names of entity examples (QE). For each measure, the best performing score is shown in bold.

	P[r]			
Run	5	10	R-prec	MAP
`FullPage`	0.1571	0.1571	0.1385	0.1314
`StatL`	0.2143	**0.2250**	**0.2285**	**0.1902**
`StatR`	**0.2214**	0.2143	0.2191	0.1853
`DynCRE`	**0.2214**	0.2107	0.2152	0.1828

(Q) Topic title

	P[r]			
Run	5	10	R-prec	MAP
`FullPage`	0.1857	0.1750	0.1587	0.1520
`StatL`	0.2429	0.2179	**0.2256**	0.2033
`StatR`	0.2429	**0.2214**	0.2248	**0.2042**
`DynCRE`	**0.2571**	0.2107	0.2207	0.1938

(QE) Topic title and entity examples

both cases the `DynCRE` run achieves the best early precision but overall performs worst among the three runs, although the differences in performance between each of the three run pairs are not statistically significant.

Implementing narrow contexts in our linkrank module allows for the locality of links to be exploited in entity ranking. By changing the context around entity examples, we would also expect the optimal values for the two combining parameters to change. We therefore varied the values for α and β and re-calculated MAP over the 28 topics in our test collection. For the three runs using narrow contexts we found that the optimal value for α has shifted from 0.1 to 0.2 (in the case of Q), while for the two static runs the optimal α value was 0.3 (in the case of QE). In both cases, the optimal value for β was found to be 0.6. The performances of the three optimal runs were very similar, and all of them substantially outperformed the optimal run using the full page context.

6 Conclusion and future work

We have presented our entity ranking approach for the INEX Wikipedia XML document collection which is based on exploiting the interesting structural and semantic properties of the collection. We have shown in our evaluations that the use of the categories and the locality of Wikipedia links around entity examples has a positive impact on the performance of entity ranking.

In the future, we plan to further improve our linkrank algorithm by varying the number of entity examples and incorporating relevance feedback that we expect would reveal other useful entities that could be used to identify better contexts. We also plan to carry out a detailed per-topic error analysis, which should allow us to determine the effect of the topic type on entity ranking. Finally, our active participation in the INEX entity ranking track will enable us to compare the performance of our entity ranking approach to those achieved by other state-of-the-art approaches.

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. H. Bast, A. Chitea, F. Suchanek, and I. Weber. ESTER: efficient search on text, entities, and relations. In *Proceedings of the 30th ACM International Conference on Research and Development in Information Retrieval*, pages 671–678, Amsterdam, The Netherlands, 2007.
3. S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the 7th International Conference on World Wide Web*, pages 107–117, Brisbane, Australia, 1998.
4. D. Cai, X. He, J.-R. Wen, and W.-Y. Ma. Block-level link analysis. In *Proceedings of the 27th ACM International Conference on Research and Development in Information Retrieval*, pages 440–447, Sheffield, UK, 2004.
5. J. Callan and T. Mitamura. Knowledge-based extraction of named entities. In *Proceedings of the 11th ACM Conference on Information and Knowledge Management*, pages 532–537, McLean, Virginia, 2002.
6. S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on EMNLP and CoNLL*, pages 708–716, Prague, The Czech Republic, 2007.
7. 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*, pages 481–486, 2007.
8. L. Denoyer and P. Gallinari. The Wikipedia XML corpus. *SIGIR Forum*, 40(1):64–69, 2006.
9. J. Kazama and K. Torisawa. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on EMNLP and CoNLL*, pages 698–707, Prague, The Czech Republic, 2007.
10. J. M. Kleinberg. Authoritative sources in hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
11. 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.
12. L. Nie, B. D. Davison, and X. Qi. Topical link analysis for web search. In *Proceedings of the 29th ACM International Conference on Research and Development in Information Retrieval*, pages 91–98, Seattle, Washington, 2006.
13. J. Pehcevski, J. A. Thom, and A.-M. Vercoustre. Hybrid XML retrieval: Combining information retrieval and a native XML database. *Information Retrieval*, 8(4):571–600, 2005.
14. 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.