Exploiting the Category Structure of Wikipedia for Entity Ranking

Rianne Kaptein[a,1], Jaap Kamps[b]

[a] *Oxyme, Cronenburg 150, 1081GN AMSTERDAM, The Netherlands*
[b] *University of Amsterdam, Department of Media Studies, Turfdraagsterpad 9, 1012XT AMSTERDAM, The Netherlands*

Abstract

The Web has not only grown in size, but also changed its character, due to collaborative content creation and an increasing amount of structure. Current Search Engines find Web pages rather than information or knowledge, and leave it to the searchers to locate the sought information within the Web page. A considerable fraction of Web searches contains named entities. We focus on how the Wikipedia structure can help rank relevant entities directly in response to a search request, rather than retrieve an unorganized list of Web pages with relevant but also potentially redundant information about these entities. Our results demonstrate the benefits of using topical and link structure over the use of shallow statistics.

Our main findings are the following. First, we examine whether Wikipedia category and link structure can be used to retrieve entities inside Wikipedia as is the goal of the INEX (Initiative for the Evaluation of XML retrieval) Entity Ranking task. Category information proves to be a highly effective source of information, leading to large and significant improvements in retrieval performance on all data sets. Secondly, we study how we can use category information to retrieve documents for ad hoc retrieval topics in Wikipedia. We study the differences between entity ranking and ad hoc retrieval in Wikipedia by analyzing the relevance assessments. Considering retrieval performance, also on ad hoc retrieval topics we achieve significantly better results by exploiting the category information. Finally, we examine whether we can automatically assign target categories to ad hoc and entity ranking queries. Guessed categories lead to performance improvements that are not as large as when the categories are assigned manually, but they are still significant. We conclude that the category information in Wikipedia is a useful source of information that can be used for entity ranking as well as other retrieval tasks.

Keywords: Wikipedia, Entity Ranking, Category structure, Link structure

Email addresses: rianne@oxyme.com (Rianne Kaptein), kamps@uva.nl (Jaap Kamps)

[1] Work done while at the University of Amsterdam

Preprint submitted to Elsevier *April 15, 2012*

1. Introduction

The Web contains an unprecedented amount of information, and continues to grow, making shallow statistics so powerful that they can solve tasks that were assumed to require deep “intelligence” [14]. But apart from its size, the Web has also changed in character. First, there is an increasing amount of structure—topical structure, document structure, link structure, tag structure, et cetera—either imposed or evolving as a light-weight organization of information and knowledge. Second, Web users are no longer passive consumers, but are active contributors of information in Wikis, Blogs and online communities—in fact they can publish whatever they want to share with the rest of the world. This immediately prompts the question how we can unleash the power of the Web’s structure and collaborative content, and what benefits this will bring over the use of shallow statistics.

Researchers in artificial intelligence (AI) have realized for long that “semantics” is key to realizing advanced applications like machine translation, question answering or sophisticated information retrieval (IR) [25]. Early on AI took the path of knowledge-rich symbolic rules, and IR took the path of knowledge-poor statistics, and the two fields effectively separated. With the advent of the Web, and the availability of large-scale text corpora, both strands of research are united again, and statistical methods prevail in both fields. Despite their great effectiveness, there are some dissatisfying aspects to the effectiveness of relatively shallow statistics and machine learning on large data volumes. First, they work remarkably well some tasks, where the data holds all required cues explicitly or implicitly. Yet they have difficulty in tasks that rely on common-sense or background knowledge, or on purposeful dialogue with a user. Admittedly this is a vague distinction that seems to be shifting over the years. Second, related to the product and process schools in AI, it seems unlikely that human cognition is modelled well by the statistical models. There are many benefits to be had if we were able to fruitfully combine our insights in statistics and machine learning with our insights in human cognition, if only as a second step to take input and feedback, and to communicate results more effectively to the users of AI systems. Third, and perhaps the main dissatisfaction, is that of our scientific understanding of the problems we study. We know how to “solve” a hard task by using statistics and machine learning on a massive amount of data—which is of clear value in itself—but we fail yet to truly understand the problem, nor its solution, nor why its hard or easy, nor how we as human are capable of solving it. Hence, we have a particular interest in more informed methods that on the one hand exploit available statistics but on the other hand combine this with meaningful and interpretable semantic structure.

A resource that is large enough to generate meaningful statistics, and contains interpretable semantic structure is Wikipedia. In this paper we therefore investigate how we can exploit Wikipedia for an information retrieval task, that is to rank relevant entities in response to a search request. Current Search Engines find Web pages rather than information or knowledge directly, and leave it to the searchers to locate the sought information within the Web page. A considerable fraction of Web searches contains named entities [e.g., 30]. Searchers looking for entities are better served by presenting a ranked list of entities directly, rather than an unorganized list of Web pages with relevant but also potentially redundant information about these entities. The goal of the entity ranking task is to return entities instead of documents or text as are returned for most common search tasks. Entities can be for example persons, organizations, books, or movies. We focus on Wikipedia, an important structured information resource on the Web. Wikipedia is a free encyclopedia that anyone can edit, consisting of millions of articles that adhere to a certain structure. Wikipedia is a highly structured resource and includes an extensive collection of categories that are used to categorize Wikipedia articles. Wikipedia being a structured resource is a great asset that can be exploited for many AI tasks such as word sense disambiguation [24], question answering [1] and information retrieval.

The nature and structure of Wikipedia presents new opportunities to solve problems that were thought to require deep understanding capabilities and where bottlenecks such as high cost and scalability where applicable in the past. Combining the benefits of the structured information and the large scale of Wikipedia, creating the opportunity to use probabilistic methods, we can now efficiently process all of the information contained in Wikipedia. Furthermore, the scale of Wikipedia offers the opportunity to go far beyond toy problems, and experiment with real-world problems and applications. In addition, the information retrieval community has created test collections for different retrieval tasks using Wikipedia as a document collection. The advanced IR methodology helps us evaluate the quality of search results and compare different approaches in great detail.

The notion of "entity" is a complex one, yet Wikipedia provides a simple and effective solution to the problem of named entity extraction. Many Wikipedia pages are in fact entities, and by using the category information we can distinguish the entities from other types of documents. The titles of the Wiki-pages are the named entity identifiers. Using the redirects alternative entity identifiers can also be extracted. The Wikipedia categories can be associated with entity types, which makes it possible to extract entities where any Wiki-page belonging to a target entity type can be considered as an entity. This nicely illustrates the power of modern Web resources as Wikipedia, where human computing has already identified and classified entities in a way that is very similar to the intent of searchers.

An issue in all entity ranking tasks is how to represent entities, returning only the name of the entity is not enough. It has been shown for QA systems users generally prefer answers embedded in context, regardless of the perceived reliability of the source documents [22]. It can be assumed that for entity ranking the same holds, that is searchers like to see evidence, for example surrounding text, why an entity is relevant to their query. Since in this paper we rank entities in Wikipedia, we can exploit the structure of Wikipedia to represent entities, that is an entity is represented by its Wikipedia page and the page title is the entity identifier. Moreover, we can also take advantage of the encyclopedic structure of Wikipedia. The Web is highly redundant making it easy to find "some" relevant information but very hard to give complete and exact answers. Search results will be dominated by the most popular one or two entities on the Web, pushing down other relevant entities. Since Wikipedia has an encyclopedic structure, enforced by its millions of editors, each entity occurs in principle only once and we do not have to worry about redundant information. For ambiguous entity names there are special disambiguation pages, where the different meanings of the entity name are listed. These disambiguation pages are also an interesting source of information to return more diverse search results to a query [28].

One of the challenges in exploiting the category information is that Wikipedia categories are created and assigned by different human editors, and are therefore less rigorous, coherent and consistent than usual ontologies. With 150,000 categories to choose from it is a non-trivial task to assign the correct categories to a Wikipedia page. Some categories that should be assigned can be missing, and too general or too specific categories can be assigned to a page. A Wikipedia page is usually assigned to multiple categories. Wikipedia guidelines are to place articles only in the most specific categories they reasonably fit in (a modern version of Cutter's rule from library science [8]). Peer reviewing is employed to improve the quality of pages and categorizations. Given the noisy nature of any Web data, it is very well possible that relevant pages are not assigned to the designated target category. The category can either be a few steps away in the category graph, or similar categories can be relevant. Another issue is that some of the target categories provided in the entity ranking topics are redirected, e.g. "Category:Movies" is redirected to "Category:Films". These categories in principle should not contain any pages, and are not included in the category graph. The entity ranking techniques that will be described in this paper, are robust enough to be able to deal with these issues.

In the typical Web search scenario the shallowness on the user side is a main bottleneck for delivering more accurate retrieval results. Users provide only two to three keywords on average to search in the complete Web. In an ideal situation the user only has to submit a short keyword query, and inspect the first few top results to fulfill his information need. We want to deliver a satisfactory answer to the user's information need requiring the least effort possible from the user. We propose to overcome the shallowness of query and results by using context, either implicitly elicited from data or explicitly through user interaction. In this paper we study how we can use Wikipedia's structure to add context in the form of category information to overcome the shallowness on the user side and retrieve information from Wikipedia. Entity ranking in Wikipedia can be an important stepping stone to ranking entities on the complete Web. Wikipedia can be used as a pivot to seach web entities by following the external links on the Wikipedia pages, or by searching the homepages of the entities identified in Wikipedia [21].

In this paper we address the following main research question:

- How can we exploit the structure of Wikipedia to retrieve entities?

We start by looking at how we can retrieve entities inside Wikipedia, which is also the task in the INEX entity ranking track. INEX[2] is an information retrieval evaluation forum that provides an IR test collection to evaluate the task of entity ranking using Wikipedia as its document collection. Our first research question is:

1. How can we exploit category and link information for entity ranking in Wikipedia?

Since a requirement for a relevant result in entity ranking is to retrieve the correct entity type, category information is of great importance for entity ranking. Category information can also be regarded in a more general fashion, as extra context for your query, which could be exploited for ad hoc retrieval. Our second research question is therefore:

2. How can we use entity ranking techniques that use category information for ad hoc retrieval?

Since usually ad hoc queries do not have target categories assigned to them, and providing target categories for entity ranking is an extra burden for users, we also examine ways to assign target categories to queries. Our third research question is:

3. How can we automatically assign target categories to ad hoc and entity ranking queries?

This paper is organized as follows. Next, in Section 2, we describe the Wikipedia test collection and topics we are using. In Section 3 we describe the models used to exploit category and link information, how information is combined and how categories are assigned automatically to topics. In Section 4 we describe our experiments. In Section 5 we describe related work and compare our performance to the state-of-the-art. Finally, in Section 6 we draw our conclusion.

2. Experimental Data

In this section, we will discuss the Wikipedia, and the various sets of search requests and relevance judgments we will use to evaluate the effectiveness of our novel approach to entity ranking.

In this paper we make use of the 2006 and 2009 Wikipedia test collections created by INEX. Both document collections are a snapshot of the English Wikipedia. For the INEX tracks from 2006 to 2008 the Wikipedia collection of 2006 is used, which consists of a snapshot from Wikipedia from early 2006 containing 659,338 articles [13]. Since then Wikipedia has significantly grown, and for the 2009 INEX tracks a new snapshot of the collection is used. It is

[2] http://www.inex.otago.ac.nz/

extracted in October 2008 and consists of 2.7 million articles [34]. An example of a Wikipedia page can be found in Figure 1.

Wikipedia distinguishes between the following types of categories[3]:

- **Content categories** are intended as part of the encyclopedia, to help readers find articles, based on features of the subjects of those articles. Content categories can again be divided into two types of categories:
  - **Topic categories** are named after a topic, usually sharing the name with the main article on that topic, e.g. "Category:Netherlands" contains articles relating to the topic Netherlands.
  - **Set categories** are named after a class, usually in the plural, e.g. "Category:Cities in the Netherlands" contains articles whose subjects are cities in the Netherlands.
- **Project categories** are intended for use by editors or automated tools, based on features of the current state of articles, or used to categorize non-article pages, e.g. stubs, articles needing cleanup or lacking sources.

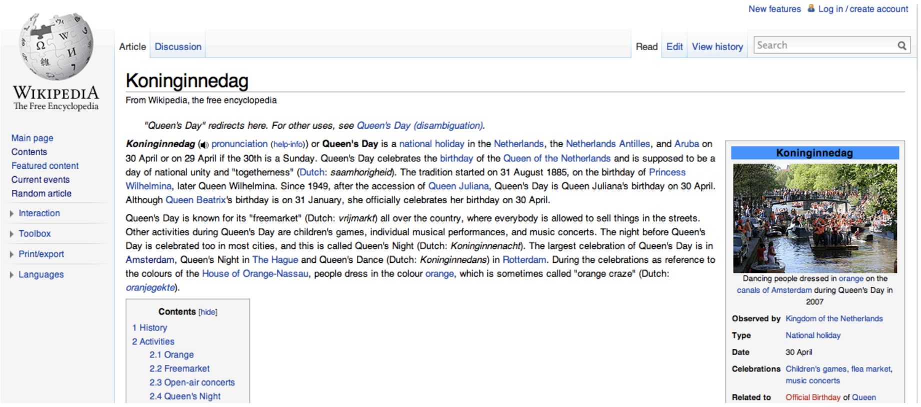
The content categories can not only help readers to find articles, also retrieval systems can use the content categories to retrieve articles. The set categories correspond with the entity types or target categories that are essential for the entity ranking task. Both the topic and the set categories can be used in the ad hoc retrieval task as sources of query context.

Wikipedia categories are organized in a loose hierarchy. Some cycles of linked categories exist, but the guideline is to avoid them. In Figure 2 a small part of the category hierarchy is shown. The category "Dutch styles in music" has one subcategory: "Dutch hiphop", which in turn has again some subcategories. Three pages are assigned to the category, and it has two parent categories, listed at the bottom of the page: "Dutch music" and "European music genres".

Wikipedia takes some measures to prevent that similar categories coexist. If two similar categories are discovered, one category is chosen and whenever people try to use the other category, they are redirected to the chosen category. For example if someone tries to assign or find "Category:Authors", he is redirected to "Category:Writers". Also if some different spelled versions of the same category exists, category redirects are used, i.e., "Ageing" redirects to "Aging", and "Living People" redirects to "Living people". This system is in use not only for categories, but also for pages. For example, the Wikipedia page in Figure 1 is reached by going to the "Queen's Day", where you immediately get redirected to the "Koninginnedag" page (the Dutch translation). The redirect pages can thus also provide synonym and cross lingual information. Wikipedia's category information can provide valuable information when searching for entities or information, but we have to take into account that the data is noisy.

The difference between entity ranking and ad hoc retrieval in general is that instead of searching for relevant text, you are searching for 'conceptual' results

[3] `http://en.wikipedia.org/wiki/Wikipedia:Categorization`



Koninginnedag

From Wikipedia, the free encyclopedia

"Queen's Day" redirects here. For other uses, see Queen's Day (disambiguation).

Koninginnedag (pronunciation (help·info)) or **Queen's Day** is a national holiday in the Netherlands, the Netherlands Antilles, and Aruba on 30 April or on 29 April if the 30th is a Sunday. Queen's Day celebrates the birthday of the Queen of the Netherlands and is supposed to be a day of national unity and "togetherness" (Dutch: *saamhorigheid*). The tradition started on 31 August 1885, on the birthday of Princess Wilhelmina, later Queen Wilhelmina. Since 1949, after the accession of Queen Juliana, Queen's Day is Queen Juliana's birthday on 30 April. Although Queen Beatrix's birthday is on 31 January, she officially celebrates her birthday on 30 April.

Queen's Day is known for its "freemarket" (Dutch: *vrijmarkt*) all over the country, where everybody is allowed to sell things in the streets. Other activities during Queen's Day are children's games, individual musical performances, and music concerts. The night before Queen's Day is celebrated too in most cities, and this is called Queen's Night (Dutch: *Koninginnenacht*). The largest celebration of Queen's Day is in Amsterdam, Queen's Night in The Hague and Queen's Dance (Dutch: *Koninginnedans*) in Rotterdam. During the celebrations as reference to the colours of the House of Orange-Nassau, people dress in the colour orange, which is sometimes called "orange craze" (Dutch: *oranjegekte*).

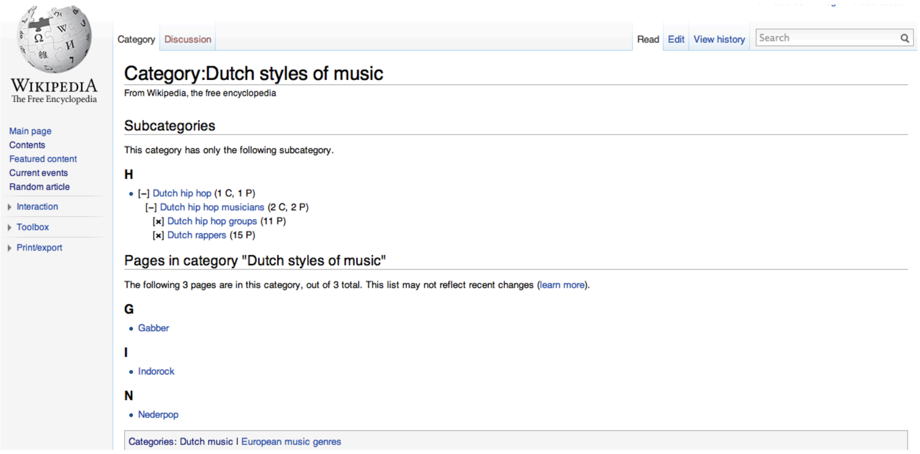
Contents [hide]
1 History
2 Activities
2.1 Orange
2.2 Freemarket
2.3 Open-air concerts
2.4 Queen's Night

Koninginnedag

Dancing people dressed in orange on the canals of Amsterdam during Queen's Day in 2007

Observed by	Kingdom of the Netherlands
Type	National holiday
Date	30 April
Celebrations	Children's games, flea market, music concerts
Related to	Official Birthday of Queen

Figure 1: Wikipedia page for "Koninginnedag" (Queen's day).



Category:Dutch styles of music

From Wikipedia, the free encyclopedia

Subcategories

This category has only the following subcategory.

H

- [–] Dutch hip hop (1 C, 1 P)
  - [–] Dutch hip hop musicians (2 C, 2 P)
    - [×] Dutch hip hop groups (11 P)
    - [×] Dutch rappers (15 P)

Pages in category "Dutch styles of music"

The following 3 pages are in this category, out of 3 total. This list may not reflect recent changes (learn more).

G

- Gabber

I

- Indorock

N

- Nederpop

Categories: Dutch music | European music genres

Figure 2: Wikipedia page for the category "Dutch styles of music".

that represent the relevant entities. Entities can be of different types. A popular type of entity ranking is people search, other entity types can be movies, books, cities, etc. One of the difficulties in entity ranking is how to represent entities. Some supporting evidence in addition to the entity id or name is needed to confirm that an entity is relevant. When we rank entities in Wikipedia, we simply use Wikipedia pages to represent entities and to provide the supportive evidence [42].

An entity ranking query topic consists of a keyword query and one or a few target categories which are the desired entity types. A description and narrative are added to clarify the query intent. A few relevant example entities are included in the topics for the list completion task, which we do not consider in this paper. For retrieval we only use the topic titles and the target categories of the entity ranking topics. An example entity ranking topic is given in Figure 3.

```
<inex_topic topic_id="79">
  <title>Works by Charles Rennie Mackintosh</title>
  <description>I am interested in works by Charles Rennie
    Mackintosh, especially buildings.
  </description>
  <narrative>I have seen some of the Mackintosh’s pieces of
    furnitures, but not his buildings.  I am interested in
    locating them as I plan a trip to Scotland.
  </narrative>
  <categories>
    <category>Buildings and structures</category>
  </categories>
  <entities>
    <entity id="433083">Glasgow School of Art</entity>
    <entity id="2981244">Queen"s Cross Church</entity>
  </entities>
</inex_topic>
```

Figure 3: INEX entity ranking topic 79

The structure of ad hoc topics is similar to the entity ranking topics, but they do not include target categories. An example ad hoc topic is given in Figure 4.

A main difference between the INEX entity ranking and ad hoc retrieval tasks lies in the assessments. In ad hoc retrieval, a document is judged relevant if any piece of the document is relevant. In the entity ranking track, a document can only be relevant if the document is of the correct entity type, resulting in far less relevant documents. The correct entity type is specified during topic creation as a target category.

We run our experiments on the following topic sets:

- Ad hoc topics
  - AH07: Ad hoc topics 414-543, consisting of 99 assessed ad hoc topics.
    - AH07a: 19 Ad hoc topics that have been used to create the entity ranking topics 30-59.
    - AH07b: The remaining 80 ad hoc topics.
- Entity ranking topics
  - ER07a: Entity ranking topics 30-59, consisting of 19 assessed entity ranking topics derived from ad hoc topics of the 2007 track.
  - ER07b: Entity ranking topics 60-100, consisting of 25 assessed genuine entity ranking topics of the 2007 track.
  - ER08: Entity ranking topics 101-149, consisting of 35 assessed genuine entity ranking topics of the 2008 track.

```
<inex_topic topic_id="524">
  <title>home heating solar panels</title>
  <description>Find information about solar panels in relation
     to home heating.
  </description>
  <narrative>Friends are planning to build a new house and have
    heard that using solar energy panels for heating can save a
    lot of money.  Since they do not know anything about home
    heating and the issues involved, they have asked for your
    help.  You are uncertain as well, and do some research to
    identify some issues that need to be considered in deciding
    between more conventional methods of home heating and solar
    panels.
  </narrative>
</inex_topic>
```

Figure 4: INEX ad hoc topic 524

  - ER09: Entity ranking topics 60-149, a selection of 55 entity ranking topics from 2007 and 2008 to be used with the 2009 Wikipedia collection.

Set ER07b consists of genuine entity ranking topics, set AH07b consists of genuine ad hoc topics. Set ER07a and set AH07a consist of the same topics, but with different relevance assessments, i.e., entity ranking assessments for set ER07a and ad hoc assessments for set AH07a. These different topic sets allow us to explore the relations between ad hoc retrieval and entity ranking, which is the topic of the next section.

Now that we have found some indications that category information is indeed useful for entity ranking topics, and could also be useful for ad hoc topics, in the next section we describe how we can make use of the category information.

3. Retrieval Model

In this section we describe our baseline retrieval model, how we use category information and link information for entity ranking, how we combine these sources of information, and how we assign categories to query topics automatically.

3.1. Baseline Retrieval Model

Our baseline retrieval model is a standard language model. For retrieval we make use of Indri [37], an open source search engine, which incorporates the language modeling approach. The baseline model uses Jelinek-Mercer smoothing to smooth the probability of a query term occurring in a document with the probability of the query term occurring in the background corpus as follows:

$$P(Q|D) = \prod_{t \in Q} \lambda P(t|D) + (1-\lambda)P(t|B)$$

where Q is the query, D the document, and B the background collection.

The optimal value of λ depends on the type of query as well as the document collection. An optimal value of $\lambda = 0.15$ is found by [15] in an experiment using the small Cranfield document collection consisting of 1,398 abstracts on aerodynamics. Zhai and Lafferty [47] find optimal values around 0.7 for long queries, but also show that for short keyword queries the smoothing parameter has less impact and little smoothing is needed, leading to an optimal value of λ is large, around 0.9.[4] From the TREC Terabyte tracks, it is also known that the .GOV2 collection requires little smoothing i.e., a value of 0.9 for λ gives the best results [18]. In this paper we therefore use the value $\lambda = 0.9$.

3.2. Exploiting Category Information

Although for each entity ranking topic one or a few target categories are provided, relevant entities are not necessarily associated with these provided target categories. Relevant entities can also be associated with descendants of the target category or other similar categories. Therefore, simply filtering on the target categories is not sufficient. Also, since Wikipedia pages are usually assigned to multiple categories, not all categories of an answer entity will be similar to the target category. We calculate for each target category the distances to the categories assigned to the answer entity. To calculate the distance between two categories, we tried three options. The first option (*binary distance*) is a very simple method: the distance is 0 if two categories are the same, and 1 otherwise. The second option (*contents distance*) calculates distances according to the contents of each category, and the third option (*title distance*) calculates a distance according to the category titles. For the title and contents distance, we need to estimate the probability of a term occurring in a category. To avoid a division by zero, we smooth the probabilities of a term occurring in a category with the background collection:

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

where C, the category, consists either of the category title to calculate title distance, or of the concatenated text of all pages belonging to that category to calculate contents distance. B is the entire Wikipedia document collection, which is used to estimate background probabilities.

Instead of using maximum likelihood estimation to estimate the probability $P(t|C)$, we estimate $P(t|C)$ with a parsimonious model. The parsimonious

[4] In [47] λ determines the weight of the term probability in the background collection $P(t|B)$, instead of the weight of the term probability in the document $P(t|D)$. Therefore they report on an optimal value of 0.1.

language model overcomes some of the weaknesses of the standard language modeling approach. Instead of blindly modeling language use in a (relevant) document, we should model what language use distinguishes a document from other documents. The exclusion of words that are common in general English, and words that occur only occasionally in documents, can improve the performance of language models and decrease the size of the models. This so-called parsimonious model was introduced by [36] and practically implemented by [16].

The model is estimated using *Expectation-Maximization*:

$$\begin{aligned} \text{E-step}: \quad & e_t = tf_{t,C} \cdot \frac{\alpha P(t|C)}{\alpha P(t|C) + (1-\alpha) P(t|B)} \\ \text{M-step}: \quad & P(t|C) = \frac{e_t}{\sum_t e_t}, \text{ i.e., normalize the model} \end{aligned}$$

In the initial E-step, the maximum likelihood estimates are used to estimate $P(t|C)$. The E-step benefits terms that occur relatively more frequent in the document as in the whole collection. The M-step normalizes the probabilities. After the M-step terms that receive a probability below a certain threshold are removed from the model. In the next iteration the probabilities of the remaining terms are again normalized. The iteration process stops after a fixed number of iterations.

We use KL-divergence to calculate distances between categories, and calculate a category score that is high when the distance is small, and the categories are similar as follows:

$$S_{cat}(C_d|C_t) = -D_{KL}(C_d|C_t) = -\sum\nolimits_{t \in D} \left(P(t|C_t) * \log \left(\frac{P(t|C_t)}{P(t|C_d)} \right) \right) \tag{2}$$

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.

In contrast to [40], where a ratio of common categories between the categories associated with an answer entity and the provided target categories is calculated, we take for each target category 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:

$$S_{cat}(d|QT) = \sum\nolimits_{C_t \in QT} \underset{C_d \in d}{\operatorname{argmax}} S(C_d|C_t) \tag{3}$$

3.3. Exploiting Link Information

We implement two options to use the link information: *relevance propagation* and *document link degree prior*. For the document link degree prior we use the same approach as in [20]. The prior for a document d is:

$$P_{Link}(d) = 1 + \frac{Indegree_{Local}(d)}{1 + Indegree_{Global}(d)} \tag{4}$$

The local indegree is equal to the number of incoming links from within the top ranked documents retrieved for one topic. The global indegree is equal to the number of incoming links from the entire collection.

The second use of link information is through relevance propagation from initially retrieved entities, as was done in the 2007 entity ranking track by [39].

$$P_0(d) = P(q|d)P_i(d) = P(q|d)P_{i-1}(d) + \sum\nolimits_{d' \to d} (1 - P(q|d'))P(d|d')P_{i-1}(d') \tag{5}$$

Probabilities $P(d|d')$ are uniformly distributed among all outgoing links from the document. Documents are ranked using a weighted sum of probabilities at different steps:

$$P_{Link}(d) = \mu_0 P_0(d) + (1 - \mu_0) \sum\nolimits_{i=1}^{K} \mu_i P_i(d) \tag{6}$$

For K we take a value of 3, which was found to be the optimal value by [39]. We try different values of μ_0 and distribute $\mu_1 ... \mu_K$ uniformly, i.e., $\mu_1 ... \mu_K = 1/3$.

3.4. Combining information

Finally, we have to combine our different sources of information. We start with our baseline model which is a standard language model. We explore two possibilities to combine information. First, we make a linear combination of the document, link and category score. All scores and probabilities are calculated in the log space, and then a weighted addition is made.

Besides the category score, we also need a query score for each document. This score is calculated using a standard language model with Jelinek-Mercer smoothing without length prior:

$$P(q_1, ..., q_n|d) = \sum_{i=1}^{n} \lambda P(q_i|d) + (1 - \lambda)P(q_i|B) \tag{7}$$

Finally, to combine the query score and the category score, both scores are calculated in the log space, and then a weighted addition is made.

$$S(d|QT) = (1 - \mu)P(q|d) + \mu S_{cat}(d|QT) \tag{8}$$

Link information is accounted for in a similar fashion:

$$S(d|QT) = (1 - \beta)P(q|d) + \beta P_{Link}(d) \tag{9}$$

We also combine both link category and link information with the query score as follows:

$$S(d|QT) = (1 - \mu - \beta)P(q|d) + \mu S_{cat}(d|QT) + \beta P_{Link}(d) \tag{10}$$

Alternatively, we can use a two step model. Relevance propagation takes as input initial probabilities as calculated by the baseline document model score. Instead of the baseline probability, we can use the scores of the run that combines the baseline score with the category information. Similarly, for the link degree prior we can use the top results of the baseline combined with the category information instead of the baseline ranking.

3.5. Target Category Assignment

Besides using the target categories provided with the entity ranking query topics, we also look at the possibility of automatically assigning target categories to entity ranking and ad hoc topics. Since the entity ranking topic assessments heavily depend on the target categories used during assessment, the automatically assigned categories will have to be suitably similar to the provided target categories in order to perform well. The advantage of automatically assigning target categories is that no effort from a user is required.

Furthermore, in the 2008 runs 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 always the case. In addition to the manually assigned target categories, we have therefore automatically created sets of target categories.

There are many ways to do automatic topic categorization, for example by using text categorization techniques. For now we keep it simple here and exploit the existing Wikipedia categorization of documents. From our baseline run we take the top N results, and look at the T most frequently occurring categories belonging to these documents, while requiring categories to occur at least twice. These categories are assigned as target categories to the query topic.

As stated in the introduction, a distinction between *topic categories* (named after a topic) and *set categories* (named after a class or entity type) can be made. Entity ranking topics look for a collection of pages belonging to the same set category or entity type, instead of just any type of document. Ad hoc topics look for any type of document as long as it belongs to the correct topic category.

The automatic assignment of categories is applied in the same way to entity ranking and ad hoc topics, but when we look at the automatically assigned categories for the entity ranking topics in almost all cases the category can be considered as a (usually low level) entity type. For the ad hoc topics still a considerable number of set categories are assigned, but topical categories occur regularly here as well. In order to compare manual and automatic assignment of categories on the ad hoc topics as well, we have manually assigned target categories to the ad hoc topics. These categories can be either topic or set categories, the category that seems closest to the query topic is selected, e.g for the query "Steganography and its techniques" the category "Steganography" is assigned as target category.

4. Experiments

In this section we describe our experiments with entity ranking and ad hoc retrieval in Wikipedia.

4.1. Experimental Set-up

In this paper we experiment with two different tasks. First of all we experiment with the entity ranking task as defined by INEX. We will make runs on the topic sets from 2007 to 2009. The 2007 topic set is used to experiment with settings of different parameters, and these parameter settings are tested on the 2008 and 2009 topics. Secondly, we experiment with ad hoc retrieval using category information on the ad hoc topic sets from 2007, and provide an analysis of the relevance assessment sets of the ad hoc and the entity ranking topics. We compare automatic and manual category assignment for ad hoc and entity ranking topics.

For our experiments we use query topics from the ad hoc and entity ranking tracks. The goal of the INEX ad hoc track is to investigate the effect of structure in the query and the documents. Results consist of XML elements or document passages rather than Wikipedia pages. The ad hoc assessments are based on highlighted passages. Since we only do document retrieval and do not return document elements or passages, we have to modify the ad hoc assessments. In our experiments, a document is regarded as relevant if some part of the article is regarded as relevant, i.e., highlighted by the assessor [19], which is similar to the TREC guidelines for relevance in ad hoc retrieval. This way we can reuse the relevance assessments of the so-called "Relevant in Context Task" to calculate MAP and precision evaluation measures. Ad hoc topics consist of a title (short keyword query), an optional structured query, a one line description of the search request and a narrative with more details on the requested topic and the task context.

To create our baseline runs incorporating only the content score, we use Indri [37]. Our baseline is a language model using Jelinek-Mercer smoothing with $\lambda = 0.9$. We apply pseudo-relevance feedback, using the top 50 terms from the top 10 documents. The category score is usually calculated for the top 500 documents of the baseline run. These documents are reranked to produce the run that combines content and category score. In one experiment we increase the number of documents to rerank to 2500. Only the top 500 results are taken into account when MAP is calculated. Since relevant pages could be found outside the initial top 500, by reranking 2500 pages more pages with relevant categories will be included in the top 500 results.

In addition to the manually assigned topic categories during the topic creation, we automatically assign topic categories. For the automatically assigned categories, we have two parameters, N the number of top results to use, and T the number of target categories that is assigned for each topic. For the parameter μ, which determines the weight of the category score, we tried values from 0 to 1, with steps of 0.1. The best values of μ turned out to be on the low end of this spectrum, therefore we added two additional values of μ: 0.05 and 0.02.

To evaluate our approach we use the following measures which are all standard measures in the Information Retrieval community. *Mean Average Precision* (MAP) provides a measure of the quality of the ranking across all recall levels. For a single information need, average precision is the average of the precision value obtained for the set of top k documents in the ranking after each relevant document is retrieved. MAP is the average of the average precision for a set of information needs. MAP is calculated as follows [23]:

$$MAP = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j}\text{Precision}(R_{jk}) \tag{11}$$

where the set of relevant documents for an information need $q_j \in Q$ is $\{d_1, \ldots, d_{m_j}\}$ and R_{jk} is the set of ranked retrieval results from the top result until you get to document d_k.

For search tasks it is important to measure how many good results there are on the first result page, since this is all most users look at [17]. Precision (the fraction of retrieved documents that are relevant) is therefore measured at fixed low levels of retrieved results, i.e., precision at 10 documents (P10).

A relatively novel performance measure that handles graded relevance judgements to give more credit to highly relevant documents is *Discounted Cumulative Gain* (DCG) [7]. The gain or usefulness of examining a document is accumulated starting at the top of the ranking and may be reduced or discounted at lower ranks. The DCG is the total gain accumulated at a particular rank k and is calculated as:

$$DCG_k = rel_1 + \sum_{i=2}^{k}\frac{rel_i}{log_2 i} \tag{12}$$

where rel_i is the graded relevance level of the document retrieved at rank i. To facilitate averaging across queries with different numbers of relevant documents, DCG values can be normalised by comparing the DCG at each rank with the DCG value for the perfect or ideal ranking for that query. The *Normalised Discounted Cumulative Gain* (NDCG) is defined as:

$$NDCG_k = \frac{DCG_k}{IDCG_k} \tag{13}$$

where IDCG is the ideal DCG value for that query. NDCG can be calculated at fixed cut-off values for k such as $NDCG_5$, or at the total number of R relevant documents for the query ($NDCG_R$).

Table 1: ER07b Results Using Link Information

# docs for local indegree	Weight link prior	MAP	P10
Baseline		0.1840	0.1920
50	0.6	0.1898$^-$	**0.2040**$^-$
50	0.5	0.1876$^-$	0.2000$^-$
100	0.7	0.1747$^-$	0.2000$^-$
100	0.3	0.1909$^-$	0.1920$^-$
500	0.5	**0.1982**$^{\circ}$	0.2000$^-$
500	0.3	0.1915$^-$	**0.2040**$^{\circ}$
1,000	0.5	0.1965$^-$	0.1960$^-$
1,000	0.4	0.1965°	0.2000$^-$

Significance of increase or decrease over baseline according to t-test, one-tailed, at significance levels 0.05($^{\circ}$), 0.01(◍), and 0.001($^{\bullet}$).

4.2. Entity Ranking Results

We apply our entity ranking methods to the entity ranking tasks over the years to answer our first research question: *How can we exploit category and link information for entity ranking in Wikipedia?*

4.2.1. Entity Ranking 2007 topics

For our training data we use topic set ER07b which consists of the 25 genuine entity ranking test topics that were developed for the 2007 entity ranking track. For our baseline run and to get initial probabilities we use the language modeling approach with Jelinek-Mercer smoothing, Porter stemming and pseudo relevance feedback. We tried different values for the smoothing λ. We found $\lambda = 0.9$ gives the best results, with a MAP of 0.1840 and a P10 of 0.1920. Applying pseudo relevance feedback has a positive effect on MAP. When no pseudo-relevance feedback is applied, results are not as good with a MAP of 0.1638. Early precision is slightly better though when no pseudo-relevance feedback is applied, with a P10 of 0.1929.

Now that we have a baseline run, we experiment with the document link degree prior, the category information, and their combination. For the document link degree prior we have to set two parameters: the number of top documents to use, and the weight of the document prior. For the number of top documents to use, we try 50, 100, 500 and 1,000 documents. For the weight of the prior we try all values from 0 to 1 with steps of 0.1. Only weights that give the best MAP and P10 are shown in Table 1. Unfortunately, applying a link degree prior does not lead to much improvement in the results. Most improvements are small and not significant. The best number of top documents to use is 500, here we find a significant improvement in MAP (from 0.1840 to 0.1982) for a weight of the document prior of 0.5, and a significant improvement in P10 (from 0.1920 to 0.2040) for a weight of 0.3 for the document prior.

Table 2: ER07b Results Using Category Information

Category representation	Weight	MAP	P10
Baseline		0.1840	0.1920
Binary	0.1	0.2145^{-}	0.1880^{-}
Contents	0.1	0.2481$^{\color{gray}\bullet}$	0.2320°
Title	0.1	0.2509°	0.2360°
Contents	0.05	**0.2618**$^{\color{gray}\bullet}$	**0.2480**$^{\color{gray}\bullet}$
Title	0.05		

Table 3: ER07b Results Combining Category and Link Information

Link Info	Weight	MAP	P10
Linear Combination			
Prior	0.3	0.2682$^{\color{gray}\bullet}$	0.2640$^{\color{gray}\bullet}$
Prop.	0.1	**0.2777**$^{\color{gray}\bullet}$	**0.2720**$^{\color{gray}\bullet}$
Two Step Model			
Prior	0.5	0.2526$^{\color{gray}\bullet}$	0.2600$^{\color{gray}\bullet}$
Prop.	0.2	0.2588$^{\color{gray}\bullet}$	**0.2960**$^{\bullet}$
Prop.	0.1	**0.2767**$^{\color{gray}\bullet}$	0.2720$^{\color{gray}\bullet}$

The results of using category information are summarized in Table 2. The weight of the baseline score is 1.0 minus the weight of the category information. For all three distances, a weight of 0.1 gives the best results. In addition to these combinations, we also made a run that combines the original score, the contents distance and the title distance. When a single distance is used, the title distance gives the best results. The combination of contents and title distance gives the best results overall.

In our next experiment we combine all information we have, the baseline score, the category and the link information. Firstly, we combine all scores by making a linear combination of the scores and probabilities. Secondly, we combine the different sources of information by using the two step model (see Table 3). Link information is mostly useful to improve early precision, depending on the desired results we can tune the parameters to get optimal P10, or optimal MAP. Relevance propagation performs better than the document link degree prior in both combinations.

4.2.2. Entity Ranking 2008 topics

Next, we test our approach on the 35 entity ranking topics from 2008. We use the parameters that gave the best results on the ER07b topics, i.e., baseline with pseudo-relevance feedback and $\lambda = 0.9$, weights of contents and title category information is 0.1, or 0.05 and 0.05 in the combination. For the link prior we use the top 100 results, and the two-step model is used to combine the information. In Table 4 our results on the 2008 topics are shown. Results are reported using

Table 4: ER08 Results Using Category and Link Information

# Results	Category repr.				Link info		xinfAP	P10
	Baseline						0.1586	0.2257
500	Title	0.1			No		0.3059$^\bullet$	0.4171$^\bullet$
	Title	0.2			No		0.3164$^\bullet$	0.4400$^\bullet$
			Cont.	0.1	No		0.3031$^\bullet$	0.4086$^\bullet$
			Cont.	0.2	No		0.3088$^\bullet$	0.4200$^\bullet$
	Title	0.05	Cont.	0.05	No		0.3167$^\bullet$	0.4343$^\bullet$
	Title	0.1	Cont.	0.1	No		0.3189$^\bullet$	0.4400$^\bullet$
	Title	0.05	Cont.	0.05	Prior	0.5	0.3196$^\bullet$	0.4371$^\bullet$
	Title	0.05	Cont.	0.05	Prop.	0.1	0.3324$^\bullet$	0.4543$^\bullet$
2500	Title	0.1			No		0.3368$^\bullet$	0.4343$^\bullet$
	Title	0.2			No		0.3504$^\bullet$	0.4514$^\bullet$
	Title	0.2			Prop.	0.1	**0.3519**$^\bullet$	**0.4629**$^\bullet$

an inferred AP (xinfAP), the official measure of the track, where the assessment pool is created by a stratified random sampling strategy [46], and P10. The behaviour of the xinfAP measure is similar to the MAP measure. Using the category information leads to an improvement of 100% over the baseline, the score is doubled. Even when we rerank the top 500 results retrieved by the baseline using only the category information, the results are significantly better than the baseline, with a xinfAP of 0.2405. Since the category information is so important, it is likely that relevant pages can be found outside the top 500. Indeed, when we rerank the top 2500, but still evaluating the top 500, our results improve up to a xinfAP of 0.3519. Furthermore, we found that on the 2008 topics doubling the weights of the category information to 0.2 leads to slightly better results. Similar to the 2007 results, relevance propagation performs better than the link prior, and leads to small additional improvements over the runs using category information.

4.2.3. Entity Ranking 2009 topics

A second testing round has been done using the 2009 entity ranking topics, which use the new Wikipedia '09 collection. We use the same parameters as for the ER08 topics, and rerank the top 2,500 results using the category titles to compute the distances between categories. Since the link information only led to minor improvements, it is not considered. Also we only use the category titles and not the category contents to calculate distances between categories, which is faster and we do not have to go through the complete collection to create the language models of the contents of each category. The results of the runs can be found in Table 5. Results are reported here using the official measures of the track, i.e., an inferred AP (xinfAP) and NDCG. Only the best runs with the according weights are shown in the table. We see that using the category information still leads to significant improvements over the baseline, but the improvements are not as large as before. Besides testing our approach with the parameter settings from ER08, we created a new type of run where we apply score normalization. Scores are normalized using the min-max normalization method before they are combined. The normalization of scores does lead to additional improvement.

Table 5: ER09 Results Using Category Information

Category repr.	Weight	xinfAP	NDCG
Baseline		0.171	0.441
Title	0.1	0.201•	0.456°
Title, normalized	0.2	**0.234•**	**0.501•**

4.3. Ad Hoc Retrieval Results

Besides using category information for entity ranking, we also experiment with using category information for ad hoc retrieval to answer our second research question: *How can we use entity ranking techniques that use category information for ad hoc retrieval?*

In these experiments we have manually assigned target categories to the ad hoc retrieval topics. For the entity ranking topics we use the target categories assigned during topic creation. Our results expressed in MAP are summarized in Table 6. This table gives the query score, which we use as our baseline, the category score, the combined score using $\mu = 0.9$ and the best score of their combination with the corresponding value of μ, which is the weight of the category score.

The baseline score on the entity ranking topics is quite low as expected. Using only the keyword query for article retrieval, and disregarding all category information, can not lead to good results since the relevance assessments are based on the category information. For the ad hoc topics on the other hand, the baseline scores are much better.

The best value for μ differs per topic set, but for all sets μ is quite close to 0. This does not mean however that the category scores are not important, which is also clear from the improvements achieved. The reason for the high μ values is that the category scores are in a larger order of magnitude, because instead of scoring a few query terms, all the terms occurring in the language model of the category are scored. So even with small weights, the category score contributes significantly to the total score. Normalizing the scores, like we have done in the ER09 track using min-max normalization, can give a more realistic estimation of the value of the category information. From the four topic sets, the baseline scores best on the two ad hoc topic sets AH07a and AH07b. There is quite a big difference between the two entity ranking topic sets, where the topics derived from the ad hoc topics are easier than the genuine entity ranking topics. The topics derived from the ad hoc topics are a selection of the complete ad hoc topic set, and mostly easy topics with a lot of relevant pages are selected. The genuine entity ranking topics are developed by the participants in the INEX entity ranking track who have less insight into topic difficulty.

Table 6: Ad Hoc vs. Entity Ranking results in MAP

		Query	Category	Comb.	Best Score
Set	$\mu = 0.0$	$\mu = 1.0$	$\mu = 0.1$	μ	
ER07a	0.2804	0.2547^{-}	0.3848$^{\bullet}$	0.2	0.4039$^{\bullet}$
ER07b	0.1840	0.1231^{-}	0.2481$^{\circledcirc}$	0.1	0.2481$^{\circledcirc}$
AH07a	0.3653	0.2067°	0.4308$^{\circledcirc}$	0.1	0.4308$^{\circledcirc}$
AH07b	0.3031	0.1761$^{\bullet}$	0.3297$^{\circledcirc}$	0.05	0.3327$^{\bullet}$

The entity ranking topics benefit greatly from using the category information with significant MAP increases of 44% and 35% for topic sets ER07a and ER07b respectively. When only the category score is used to rerank the top 1000 results, the scores are surprisingly good, for set ER07a MAP only drops a little with no significant difference from 0.2804 to 0.2547. Apparently the category score really moves up relevant documents in the ranking. When we use the category information for the ad hoc topics with manually assigned categories improvements are smaller than the improvements on the entity ranking topics, but still significant with MAP increases of 18% and 10% for set AH07a and AH07b respectively. So, we have successfully applied entity ranking techniques to improve retrieval on ad hoc topics. The improvements are bigger on the ad hoc topics that are later converted into entity ranking topics, indicating that queries that can be labeled as entity ranking topics benefit the most from using category information.

We see that entity ranking topics profit more from the use of category information than ad hoc topics. In order to gain information on category distributions within the retrieval results, we analyze the relevance assessment sets. We show some statistics in Table 7. The ad hoc topics contain more relevant pages. This was to be expected, since documents in the entity ranking task, do not only have to contain relevant information on entities, but in addition the documents have to belong to a relevant category type. The relevance assessment set of topic set ER07a, contains all relevant pages from topic set AH07a. Of these pages 41.4% are relevant for the entity ranking task.

For each topic we determine the most frequently occurring category in either all pages or only the relevant pages, we call this the majority category. The target category is the category that is manually assigned during the topic creation, e.g., the target category for the example topic in Figure 3 is 'Buildings and structures'. We calculate what percentages of pages are assigned to the majority category and the target category. For the ad hoc topic sets the categories are the most diverse, only around 6-7% of the pages belong to the same category. The categories in the entity ranking topic sets are more focused, with percentages ranging from 16.3% of pages in set ER07b, to 31.6% of the pages in set ER07a belonging to the majority category.

The majority categories in the relevant pages are quite large within these relevant pages, around 60% for the entity ranking topics, and still around 32% for the ad hoc topics. What is interesting for the entity ranking topics, is that this percentage is much higher than the percentage of relevant pages belonging to the target category. This means that there are categories other than the target category, which are good indicators of relevance. In many cases the majority category is more specific than the target category, e.g. to our example topic "Works by Charles Rennie Mackintosh' target category "Buildings and structures" is assigned. The majority category in the relevant pages is "Charles Rennie Mackintosh buildings'. This category is far more specific, and using it probably leads to better results. For all topic sets we see that from the relevant pages a far higher percentage belongs to the majority category than non-relevant pages. This is in line with our findings, that category information is not only beneficial for entity ranking topics, but also ad hoc topic results can be improved if the right target categories can be found.

For the entity ranking topics we can also determine how many of the pages belong to one of the specified target categories. In fact, only 11.3% of set ER07b pages and 16.7% of set ER07a pages belong to a target category. The runs used to create the pool for topic set ER07a are ad hoc runs, so the target categories have not been taken into consideration here. In topic set ER07b however the target categories were available, but here less pages belong to the target category indicating that target categories themselves are not treated as an important feature in the submitted runs. Considering that 11.1% of the non-relevant pages also belong to the target category, this is a good decision.

Over all kinds of pages, set ER07a has more focused categories than set ER07b, the genuine entity ranking set. This can be explained by the fact that the pages in set ER07a were already assessed as relevant for the ad hoc topic, so at least topically they are more related. Comparing the ER07b results to the ER08 results, we see that the assessment statistics are quite similar, but that the ER08 results are a bit more focused on pages belonging to the target and majority categories and that a considerable higher percentage of the relevant pages belongs to the target category.

Comparing the ER08 results on the Wikipedia'06 collection to the ER09 results on the Wikipedia'09 collection, we see that a higher percentage of relevant pages is found. The number of pages belonging to the majority category stays roughly the same, but the percentage of pages belonging to the target category has gone down significantly. Not only have the systems returned less pages belonging to the target category, also a smaller part of the relevant pages belongs to the target category. This is probably caused by the fact that the categorization of Wikipedia pages has become more fine grained, while the target categories of the queries remained the same. Also less pages belong to the majority category of the relevant pages, which is another sign that the categories assigned to pages have become more diverse.The systems also evolved, and return less pages belonging to the target categories.

4.4. Manual vs. Automatic Category Assignment

Our final set of experiments in this paper compares the performance of manually and automatically assigned target categories to answer our third research

Table 7: Relevancy in judged pages for ad hoc and entity ranking topics

Set	AH07a	AH07b	ER07a	ER07b	ER08	ER09
Avg. # of pages in pool	611	612	83	485	394	314
Avg. % relevant pages	0.13	0.09	0.41	0.04	0.07	0.20
Pages with majority category of all pages:						
all pages	0.066	0.059	0.316	0.163	0.252	0.254
relevant pages	0.200	0.200	0.426	0.313	0.363	0.344
non-relevant pages	0.045	0.048	0.167	0.154	0.241	0.225
Pages with majority category of relevant pages:						
all pages	0.047	0.047	0.281	0.084	0.189	0.191
relevant pages	0.318	0.316	0.630	0.590	0.668	0.489
non-relevant pages	0.016	0.028	0.074	0.064	0.155	0.122
Pages with target category:						
all pages			0.167	0.113	0.208	0.077
relevant pages			0.387	0.277	0.484	0.139
non-relevant pages			0.048	0.111	0.187	0.064

question: *How can we automatically assign target categories to ad hoc and entity ranking topics?*

We will first discuss the ad hoc results, and then study the entity ranking topics in more detail. Before we look at at the results, we take a look at the categories assigned by the different methods. In Table 8 we show a few example topics from the ER07 track together with the categories as assigned by each method. As expected the pseudo-relevant target categories (PRF) are more specific than the manually assigned target categories. The number of common Wikipedia categories in the example entities (Examples) 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. Almost all categories extracted from the pages are ’set categories’, what is coherent with the entity ranking topics where the target entity types correspond to one of more set categories.

For the automatic assignment of target categories, we have to set two parameters: the number of top ranked documents N and the number of categories T. The retrieval results of our experiments on the AH07 set, with different values of N and T, expressed in MAP are summarized in Table 9. This table gives the query score, which we use as our baseline, the category score, the combined score using a weight of $\mu = 0.1$ for the category score and the best score of their combination with the corresponding value of μ.

When we use the category information for the ad hoc topics with manually assigned categories MAP improves significantly with an increase of 11.3%. Using the automatically assigned topics, almost the same results are achieved. The best automatic run uses the top 50 documents and takes the top 3 categories, reaching a MAP of 0.3502, a significant improvement of 11.1%. Assigning one

Table 8: Example Target Categories

Topic	**olympic classes dinghie sailing**	**Neil Gaiman novels**	**chess world champions**
Manual	dinghies	novels	chess grandmasters world chess champions
PRF	dinghies sailing	comics by Neil Gaiman fantasy novels	chess grandmasters world chess champions
Examples	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

target category leads to the worst results. It is better to assign multiple categories to spread the risk of assigning a wrong category. Similarly, using more than the top 10 ranked documents leads to better results. Differences between using the top 20 and the top 50 ranked documents are small.

We continue with experiments on the entity ranking topics. We use $N = 10$ and $T = 2$ for the remaining experiments in this section. Results of manual and automatic category assignment on the ER07 data sets can be found in Table 10. When we look at the category scores only, the automatically assigned topics perform even better than the manually assigned categories. Looking at the combined scores, the manually assigned target categories perform somewhat better than the automatically assigned categories. However, for both topic sets using the automatically assigned categories leads to significant improvements over the baseline.

During the automatic assignment we use the top 10 results of the baseline run as surrogates to represent relevant documents. So we would expect that if the precision at 10 is high, this would lead to good target categories. However, precision at 10 of the baseline for topic set ER07b, is only 0.2640, but the category score is almost as good as the query score (0.1840 and 0.1779 respectively).

The question remains why the combined scores of the automatically assigned categories are worse than the combined scores of the manually assigned categories while their category scores are higher. The automatically assigned categories may find documents that are already high in the original ranking of the baseline run, since the categories are derived from the top 10 results. The manually assigned categories do not necessarily appear frequently in the top results of the baseline, so the category scores can move up relevant documents that were ranked low in the baseline run.

Finally, we take a look at the entity ranking results of 2009. Again we have manually and automatically assigned categories, but this time the scores are normalized before combining the query and the category score. The results of

Table 9: AH07 Results in MAP for Manual and Automatic Cat. Assignment

Cats		Category	Comb.	Best Score	
N	T	$\mu = 1.0$	$\mu = 0.1$	μ	
Baseline					0.3151
Manual		0.1821$^{\bullet}$	0.3508$^{\bullet}$	0.1	0.3508$^{\bullet}$
Top 10	1	0.1640$^{\bullet}$	0.3334°	0.05	0.3368$^{\bullet}$
Top 20	1	0.1793$^{\bullet}$	0.3306^{-}	0.05	0.3390$^{\bullet}$
Top 50	1	0.1798$^{\bullet}$	0.3364°	0.05	0.3457$^{\bullet}$
Top 10	2	0.1815$^{\bullet}$	0.3380$^{\odot}$	0.05	0.3436$^{\bullet}$
Top 20	2	0.1919$^{\bullet}$	0.3326°	0.05	0.3471$^{\bullet}$
Top 50	2	0.1912$^{\bullet}$	0.3323^{-}	0.05	0.3502$^{\bullet}$
Top 10	3	0.1872$^{\bullet}$	0.3379$^{\odot}$	0.05	0.3445$^{\bullet}$
Top 20	3	0.1950$^{\bullet}$	0.3265^{-}	0.05	0.3457$^{\bullet}$
Top 50	3	0.1959$^{\bullet}$	0.3241^{-}	0.05	0.3459$^{\bullet}$
Top 10	4	0.1873$^{\bullet}$	0.3370$^{\odot}$	0.05	0.3439$^{\bullet}$
Top 20	4	0.1970$^{\bullet}$	0.3275^{-}	0.05	0.3477$^{\bullet}$
Top 50	4	0.1932$^{\bullet}$	0.3172^{-}	0.02	0.3442$^{\bullet}$

Table 10: ER07 Results in MAP for Manual and Automatic Cat. Assignment

		Query	Category	Comb.	Best Score	
Assignment	Set	$\mu = 0.0$	$\mu = 1.0$	$\mu = 0.1$	μ	
Manual	ER07a	0.2804	0.2547^{-}	0.3848$^{\bullet}$	0.2	0.4039$^{\bullet}$
Manual	ER07b	0.1840	0.1231^{-}	0.2481$^{\odot}$	0.1	0.2481$^{\odot}$
Auto	ER07a	0.2804	0.2671^{-}	0.3607$^{\odot}$	0.1	0.3607$^{\odot}$
Auto	ER07b	0.1840	0.1779^{-}	0.2308^{-}	0.2	0.2221°

Table 11: ER09 Results for Manual and Automatic Cat. Assignment

Cats	μ	$\#Rel$	P10	MAP
Baseline	0	1042	0.2164	0.1674
Auto.	0.1	982^{-}	0.2509^{-}	0.2014$^{\odot}$
Auto.	0.2	911°	0.2382^{-}	0.1993°
Man.	0.1	**1180**$^{\bullet}$	0.2982$^{\bullet}$	0.2350$^{\bullet}$
Man.	0.3	1178$^{\odot}$	0.3127$^{\bullet}$	**0.2396**$^{\bullet}$
Man.	0.4	1171$^{\odot}$	**0.3145**$^{\bullet}$	0.2376$^{\bullet}$

the runs can be found in Table 11. The run that uses the official categories assigned during topic creation performs best, and significantly better than the baseline. Because we normalize the scores the weights of the category information go up, a weight of 0.4 even leads to the best P10. Here the category information proves to be almost as important as the query itself. The runs with automatically assigned entity types reach a performance close to the manually assigned topics. Although $P10$ is low in the baseline run, the 10 top ranked documents do provide helpful information on entity types. Most of the automatic assigned categories are very specific, for example 'College athletics conferences' and 'American mystery writers'. For one topic the category exactly fits the query topic, the category 'Jefferson Airplane members' covers exactly query topic 'Members of the band Jefferson Airplane'. Unsurprisingly, using this category boosts performance significantly. The category 'Living people' is assigned to several of the query topics that originally also were assigned entity type 'Persons'. This category is one of the most frequently occurring categories in Wikipedia, and is assigned very consistently to pages about persons. In the collection there are more than 400,000 pages that belong to this category. This large number of occurrences however does not seem to make it a less useful category.

5. Related Work

In this section, we will discuss related work on entity ranking, list questions in Question Answering (QA), exploiting the structure of Wikipedia and other knowledge sources, and the impact of topical structure on information access. We finish with comparing our work to other approaches in the INEX evaluation forum.

Entity ranking in Wikipedia is quite different from entity ranking on the general Web. By considering each page in Wikipedia as an entity, the problem of named entity recognition is avoided, and the entity ranking task becomes more similar to the document retrieval task on Wikipedia. Furthermore, we return the complete Wikipedia page as evidence for the relevance of the page. We do not consider the extraction of specific features of information about the entity, which is the topic of much related work and also the start of work on entity ranking approaches. Early named entity recognition systems were making use of handcrafted rule-based algorithms and supervised learning using extensive sets of manually labeled entities. More recent work used unsupervised entity extraction and resort to machine learning techniques. See [27] for a survey of named entity recognition. Wikipedia and IMDB are used as a seed list of named entity-type pairs in [44]. Subsequently, the Web is searched for occurrences of the names of entities. Recurring patterns or templates in the text around the names are extracted and filtered, and then used to extract more entity mentions of the target type.

An interesting language modeling approach to entity ranking on the Web is presented in [29]. In this case, entities are scientific papers extracted from different Web sources such as Citeseer and DBLP. Instead of aggregating all information on an entity into a large bag of words, records from each data source have their own language model, and the information from the different datasources is weighted according to the accuracy of the extraction of the data from the Web source. Also they try to incorporate structural information in their model to weigh fields, corresponding to features of the entity, differently.

Their methods outperform a bag-of-words representation of entities, and adding the structural information leads to additional improvements.

Related work can also be found in the Question Answering field. TREC (Text Retrieval Conference) ran a Question Answering track until 2007 in which list questions were included, where list questions are requests for a set of instances of a specified type (person, organization, thing or event) [9]. This task is quite similar to our entity ranking task, but even more similar to the TREC related entity finding task [5]. Topics in both of these tasks include a target entity to which the answers or retrieved entities should be related.

Many QA systems answer questions by first extracting a large list of possible candidate answers, and then filtering or reranking these answers based on some criteria such as type information, which is similar to our approach where we also rerank initially retrieved documents according to their categories. Expected answer types of a question restrict the admissible answers to specific semantic classes such as river, country, or tourist attractions. Expected answer types are assigned using supervised machine learning techniques, while the types of candidate answers are extracted making use of Wordnet and domain information contained in geographical name information systems. Different scoring methods are used to capture the relation between a candidate answer and an answer type [35]. State-of-the-art question answering systems exploit lexico-semantic information throughout the process, which leads to significant enhancements of information retrieval techniques. Bottlenecks in QA systems are the derivation of the expected answer type and keyword expansion to include morphological, lexical, or semantic alternations [26].

The task we are dealing with here is also related to other tasks which use a source of query context such as a category directory like DMOZ. Wei and Croft [43] manually assign topic categories from the DMOZ directory to queries according to some basic rules. A topical model is built from the documents in the selected topic category, and queries are smoothed with the topical model to build a modified query. A query likelihood model using this modified query does not outperform a relevance model using pseudo-relevance feedback. A combination of applying the relevance model for queries with low clarity scores meaning clear queries and the topical model smoothing otherwise, leads to minor improvements over the relevance model.

Bai et al. [2] compares the automatic and the manual assignment of topical domains. Here the topic domains do not come from an existing topic hierarchy, but the users can define their own domains. Domain models are created by either using the relevant documents for the in-domain queries, or by using the top 100 documents retrieved with the in-domain queries. Additionally, automatic query classification is done by calculating KL-divergence scores. Although the accuracy of the automatic query classification is low, the effectiveness of retrieval is only slightly lower than when the query domain is assigned manually. Both lead to significant improvements over a baseline that does not incorporate topical context.

Ravindran and Gauch [33] designed a conceptual search engine where users can input DMOZ topic categories as context for their search. Document scores for retrieval are a combination of the keyword match and the topic category match. Additionally, search results are pruned, i.e documents that do not match any of the topic categories provided with the query are removed.

Topical categories as a source of query context have also been used in TREC for ad hoc retrieval. The topics in TREC 1 and 2 include a topical domain in the query topic descriptions, which can be used as topical context. It has been shown that these topical domains can successfully be used as query context for ad hoc retrieval [2]. In this paper the automatic and the manual assignment of topical categories is compared. Category models are created by using the relevant documents or the top 100 documents retrieved for the in-category queries. The top terms in the category models are used to expand the query. Automatic query classification is done by calculating KL-divergence scores. Although the accuracy of the automatic query classification is low, the effectiveness of retrieval is only slightly lower than when the query topic category is assigned manually.

Besides topical categories, also tags can be used a source of query context. The social network site Delicious[5] is annotated by users and provides category information in the form of informal tags. Much of the early work on social annotations uses this resource, we will discuss two of these papers here. Wu et al. [45] present a semantic model that is statistically derived from the frequencies of co-occurrences among users, resources and tags. The semantic model helps to disambiguate tags and groups synonymous tags together in concepts. The derived semantic model can be used to search and discover semantically related Web resources, even if the resource is not tagged by the query tags and does not contain any query keywords.

Two aspects of social annotations that can benefit Web search are explored in [6]. These aspects are: the annotations are usually good summaries of corresponding Web pages and the count of annotations indicates the popularity of Web pages. Their approach is able to find the latent semantic association between queries and annotations, and successfully measures the quality (popularity) of a Web page from the Web users perspective.

The INEX evaluation forum has generated many entity ranking papers. INEX has run an entity ranking track from 2007 to 2009 using Wikipedia as the test collection [42, 12, 11]. Using category information is essential in this track, and almost all participants use the category information in some form. Another source of information that is exploited is link information. We will discuss some of the best performing approaches related to our approach. Vercoustre et al. [40] use Wikipedia categories to define similarity functions between the categories of retrieved entities and the target categories. The similarity scores are estimated based on the ratio of common categories between the set of categories associated with the target categories and the union of the categories associated with the candidate entities [41] or by using lexical similarity of category names [40]. Besides the entity ranking task, they also try to tackle the ad hoc retrieval task using the same approach. To categorize the ad hoc topics, the query title

[5] `http://delicious.com/`

Table 12: Comparison of our best runs to official INEX Entity Ranking Results

Year	Measure	Off. Run	Unoff. Run	INEX Run
2007	MAP	N.A.	**0.313**	0.306
2008	xinfAP	0.317	**0.352**	0.341
2009	xinfAP	0.201	0.234	**0.517**

is sent to an index of categories that has been created by using the names of the categories, and the names of all their attached entities. Their model works well for entity ranking, but when applied to ad hoc topics the entity ranking approach performs significantly worse than the basic full-text retrieval run. Another extension to their entity ranking approach is to integrate topic difficulty prediction. A topic is classified into one of four classes representing the difficulty of the topic. According to the topic classification a number of retrieval parameters is set. Although a small increase in performance can be achieved when two classes of difficulty are used, the improvements are not significant [31].

Random walks to model multi-step relevance propagation from the articles describing entities to all related entities and further are used in [39]. After relevance propagation, the entities that do not belong to a set of allowed categories are filtered out the result list. The allowed category set leading to the best results included the target categories with their child categories up to the third level.

A probabilistic framework to rank entities based on the language modelling approach is presented in [3]. Their model takes into account for example the probability of a category occurrence and allows for category-based feedback. Finally, in addition to exploiting Wikipedia structure i.e., page links and categories, Demartini et al. [10] apply natural language processing techniques to improve entity retrieval. Lexical expressions, key concepts, and named entities are extracted from the query, and terms are expanded by means of synonyms or related words to entities corresponding to spelling variants of their attributes.

A comparison of our best official and unofficial runs to the best runs officially submitted to INEX can be found in Table 12. Our entity ranking results compare favourably to other approaches on the INEX data sets. We have to note here that for our unofficial runs we have optimized some parameters using the test set, e.g. for the 2008 runs we reranked the top 500 results in the official runs, but after the evaluation results were released, we discovered it is better to rerank a larger number of results, 2500 in this case. Topic sets ER07a and ER07b together form the test data of the 2007 INEX entity ranking track. Our best score on this test data is achieved with $\mu = 0.2$ which leads to a MAP of 0.313. This score is better than any of the official submitted runs, of which the best run of Tsikrika et al. [39] achieves a MAP of 0.306 [42].

For the 2008 entity ranking track we submitted official runs. Of our submitted runs, the run using category information based on the category titles reranking 500 results performed best, with a MAP of 0.317 and ranking third among all runs. Reranking the top 2500 results leads to additional improvements, increasing MAP to 0.352, and these unofficial runs outperform the best official run of Pehcevski et al. [31], which achieves a MAP of 0.341 [12].

Considering the 2009 entity ranking track, we again ranked among the top participants in this track [11]. The topics for the 2009 track consisted of a selection of topics from the previous tracks. Only the document collection changed: a new version of Wikipedia was used. We were outperformed by two approaches. One approach used the relevance assessments available from prior years, promoting documents previously assessed as relevant, achieving xinfAP scores up to 0.517 [4]. Ramanathan et al. [32] combine a number of expansion and matching techniques based on the page titles, categories and extracted entities and n-grams. An initial set of relevant documents is recursively expanded using the document titles, category information, proximity information and the prominent n-grams. Next, documents not representing entities are filtered out using category and WordNet information. Finally, the entities are ranked using WordNet tags, category terms and the locality of query terms in the paragraphs. Using many elements beside the category information used in our approach, a xinfAP of 0.270 is achieved, which is better than our best official run with a xinfAP of 0.201, as well as our best unofficial run with a xinfAP of 0.234.

Unfortunately, we cannot compare our ad hoc retrieval runs to official INEX ad hoc runs. The original INEX ad hoc task is not a document retrieval task, but a focused retrieval task, and participants return XML elements as results, making the comparison unfair. Vercoustre et al. [40] have done experiments similar to ours, testing their entity ranking approach on the INEX 2007 ad hoc topics, the combination of topic sets AH07a and AH07b. Their entity ranking approach does not outperform their standard document retrieval run. The standard run is generated by Zettair[6], an information retrieval system developed by RMIT University, using the Okapi BM25 similarity measure, which proved to work well on earlier INEX test collections, and was ranked among the top participants in the official INEX 2007 ad hoc track results. Zettair achieves a MAP of 0.292. Calculated over all 99 topics, our baseline run achieves a MAP of 0.315, so we can say we have a strong baseline. In contrast to the approach of Vercoustre et al. [40], using the category information in our approach leads to further significant improvements over this strong baseline.

6. Conclusion

In this paper we have experimented with retrieving entities from Wikipedia exploiting its category structure. We presented our entity ranking approach where we use category and link information to answer our first research question: *How can we exploit category and link information for entity ranking in Wikipedia?* Category information is the factor that proves to be most useful and we can do more than simply filtering on the target categories. Category information can both be extracted from the category titles and from the contents

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

of the category. Link information can also be used to improve results, especially early precision, but these improvements are smaller. Our second research question was : *How can we use entity ranking techniques that use category information for ad hoc retrieval?* Our experiments have shown that using category information indeed leads to significant improvements over the baseline for ad hoc topics. Considering our third research question: *How can we automatically assign target categories to ad hoc and entity ranking topics?*, automatically assigned categories prove to be good substitutions for manually assigned target categories. Similar to the runs using manually assigned categories, using the automatically assigned categories leads to significant improvements over the baseline for all topic sets.

Our work can be extended in a number of ways. First of all, to calculate the similarity of categories, we calculate KL-divergence between the names of the categories, or the contents of the categories. Another option to calculate similarity between categories is to exploit the existing category hierarchy in Wikipedia and use a path-based measure to estimate similarity, which has been proven to be effective for computing semantic relatedness of concepts [38]. Besides using path-based measures to estimate similarity of categories, these measures could also be useful in the list completion task to find entities similar to the example entities. Another line of future work is the automatic assignment of categories. We have experimented with an approach that uses pseudo-relevance feedback to extract the most frequently occurring categories of the top N results, but it might be possible to obtain better categories with more sophisticated AI approaches, such as text categorization techniques. Finally, other sources of topical information can be extracted using the Wikipedia structure besides the category information. Many Wikipedia pages for example contain a so-called 'infobox,' a consistently-formatted table which is present in articles with a common subject. Also we could exploit structured information that extends the information in Wikipedia, which is available for example in the collaborative knowledge base Freebase. Furthermore, the INEX 2009 Wikipedia test collection also includes semantic tags [34] which can be exploited in a similar way as the category information.

In response to our main research question: *How can we exploit the structure of Wikipedia to retrieve entities?*, we found that Wikipedia is an excellent knowledge resource, which is still growing and improving every day, and we have shown that we can effectively exploit its category structure to retrieve entities. Effectively retrieving documents and entities from Wikipedia can also benefit other Web search tasks. For example, Wikipedia can be used as a pivot to rank entities on the Web [21]. Our main conclusion is that the category structure of Wikipedia can be effectively exploited, in fact not only for entity ranking, but also for ad hoc retrieval, and with manually assigned as well as automatically assigned target categories.

The general conclusion is that both the topical and the link structure of Wikipedia can be used to generate knowledge-rich answers—the entities themselves represented by their entry-pages—opposed to just a long list of relevant and redundant information that needs substantial further processing by our searcher.

This may seem just a small step, but it is an important step in exploring how the implicit or explicit structure of the modern Web can benefit many AI tasks.

Acknowledgments

We thank the reviewers for the constructive comments that helped shaped this paper, and express our gratitude to the editors of the special issue for applying the deadlines with consideration. This research was supported by the Netherlands Organization for Scientific Research (NWO, under project # 612.-066.513).

References

[1] D. Ahn, V. Jijkoun, G. Mishne, K. Müller, M. de Rijke, S. Schlobach, Using Wikipedia in the TREC QA Track, in: TREC '04: The Thirteenth Text Retrieval Conference, National Institute of Standards and Technology (NIST).

[2] J. Bai, J.Y. Nie, H. Bouchard, G. Cao, Using query contexts in information retrieval, in: SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, New York, NY, USA, 2007, pp. 15–22.

[3] K. Balog, M. Bron, M. de Rijke, Category-based query modeling for entity search, in: ECIR '10: Advances in Information Retrieval: 32nd European Conference on IR Research, volume 5993 of *LNCS*, Springer, 2010, pp. 319–331.

[4] K. Balog, M. Bron, M. de Rijke, W. Weerkamp, Combining term-based and category-based representations for entity search, in: INEX '09: Focused Retrieval and Evaluation: 8th International Workshop of the Initiative for the Evaluation of XML Retrieval, volume 6203 of *LNCS*, Springer Verlag, Berlin / Heidelberg, 2010, pp. 265–272.

[5] K. Balog, A.P. de Vries, P. Serdyukov, P. Thomas, T. Westerveld, Overview of the TREC 2009 entity track, in: TREC '09: The Eighteenth Text REtrieval Conference Notebook, National Institute for Standards and Technology (NIST), 2009.

[6] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, Z. Su, Optimizing web search using social annotations, in: WWW '07: Proceedings of the 16th international conference on World Wide Web, ACM, New York, NY, USA, 2007, pp. 501–510.

[7] B. Croft, D. Metzler, T. Strohman, Search Engines: Information Retrieval in Practice, Addison Wesley, 2009.

[8] C.A. Cutter, Rules for a dictionary catalog, Govt. Print. Off, 2nd edition, 1889.

[9] H.T. Dang, D. Kelly, J.J. Lin, Overview of the TREC 2007 question answering track., in: TREC '07: The Sixteenth Text REtrieval Conference, National Institute of Standards and Technology (NIST).

[10] G. Demartini, C.S. Firan, T. Iofciu, R. Krestel, W. Nejdl, Why finding entities in Wikipedia is difficult, sometimes, Information Retrieval, Special Issue on Focused Retrieval and Result Aggregation 13 (2010) 534–567.

[11] G. Demartini, T. Iofciu, A.P. de Vries, Overview of the INEX 2009 entity ranking track, in: INEX '09: Focused Retrieval and Evaluation: 8th International Workshop of the Initiative for the Evaluation of XML Retrieval, volume 6203 of *LNCS*, Springer Verlag, Berlin / Heidelberg, 2010, pp. 254–264.

[12] G. Demartini, A.P. de Vries, T. Iofciu, J. Zhu, Overview of the INEX 2008 entity ranking track, in: Advances in Focused Retrieval: 7th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX '08), volume 5631 of *LNCS*, Springer Verlag, Berlin / Heidelberg, 2009, pp. 243–252.

[13] L. Denoyer, P. Gallinari, The Wikipedia XML Corpus, SIGIR Forum 40 (2006) 64–69.

[14] A. Halevy, P. Norvig, F. Pereira, The unreasonable effectiveness of data, IEEE Intelligent Systems 24 (2009) 8–12.

[15] D. Hiemstra, Using Language Models for Information Retrieval, Ph.D. thesis, Center for Telematics and Information Technology, University of Twente, 2001.

[16] D. Hiemstra, S. Robertson, H. Zaragoza, Parsimonious language models for information retrieval, in: SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, New York, NY, USA, 2004, pp. 178–185.

[17] B.J. Jansen, A. Spink, How are we searching the world wide web?: A comparison of nine search engine transaction logs, Information Processing and Management 42 (2006) 248–263.

[18] J. Kamps, Effective smoothing for a terabyte of text, in: TREC '05: The Fourteenth Text REtrieval Conference, National Institute of Standards and Technology (NIST), 2006.

[19] J. Kamps, S. Geva, A. Trotman, A. Woodley, M. Koolen, Overview of the INEX 2008 ad hoc track, in: INEX '08: Advances in Focused Retrieval: 7th International Workshop of the Initiative for the Evaluation of XML Retrieval, volume 5631 of *LNCS*, Springer Verlag, Berlin / Heidelberg, 2009, pp. 1–28.

[20] J. Kamps, M. Koolen, The importance of link evidence in Wikipedia, in: ECIR 2008: Advances in Information Retrieval: 30th European Conference on IR Research, volume 4956, Springer Verlag, Berlin / Heidelberg, 2008, pp. 270–282.

[21] R. Kaptein, P. Serdyukov, A.P. de Vries, J. Kamps, Entity ranking using Wikipedia as a pivot, in: CIKM '10: Proceedings of the 19th ACM Conference on Information and Knowledge Management, ACM, New York, NY, USA, 2010, pp. 69–78.

[22] J.J. Lin, D. Quan, V. Sinha, K. Bakshi, D. Huynh, B. Katz, D.R. Karger, What makes a good answer? The role of context in Question Answering, in: INTERACT'03: IFIP TC13 International Conference on Human-Computer Interaction, IOS Press, 2003, pp. 25–32.

[23] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[24] R. Mihalcea, Using Wikipedia for automatic word sense disambiguation, in: NAACL-HLT '07: Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2007, pp. 196–203.

[25] M. Minsky (Ed.), Semantic Information Processing, The MIT Press, Cambridge MA, 1968.

[26] D. Moldovan, M. Paşca, S. Harabagiu, M. Surdeanu, Performance issues and error analysis in an open-domain question answering system, ACM Transactions on Information Systems (TOIS) 21 (2003) 133–154.

[27] D. Nadeau, S. Sekine, A survey of named entity recognition and classification, Lingvisticae Investigationes 30 (2007) 3–26.

[28] R. Navigli, G. Crisafulli, Inducing word senses to improve web search result clustering, in: EMNLP '10: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 116–126.

[29] Z. Nie, Y. Ma, S. Shi, J.R. Wen, W.Y. Ma, Web object retrieval, in: WWW '07: Proceedings of the 16th international conference on World Wide Web, ACM, New York, NY, USA, 2007, pp. 81–90.

[30] M. Paşca, Weakly-supervised discovery of named entities using web search queries, in: CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, ACM, New York, NY, USA, 2007, pp. 683–690.

[31] J. Pehcevski, J. Thom, A.M. Vercoustre, V. Naumovski, Entity ranking in Wikipedia: utilising categories, links and topic difficulty prediction, Information Retrieval, Special Issue on Focused Retrieval and Result Aggregation 13 (2010) 568–600.

[32] M. Ramanathan, S. Rajagopal, V. Karthik, M. Murugeshan, S. Mukherjee, A recursive approach to entity ranking and list completion using entity determining terms, qualifiers and prominent n-grams, in: INEX '09: Focused Retrieval and Evaluation: 8th International Workshop of the Initiative for the Evaluation of XML Retrieval, volume 6203 of *LNCS*, Springer, Berlin / Heidelberg, 2010, pp. 292–302.

[33] D. Ravindran, S. Gauch, Exploiting hierarchical relationships in conceptual search, in: CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management, ACM, New York, NY, USA, 2004, pp. 238–239.

[34] R. Schenkel, F.M. Suchanek, G. Kasneci, YAWN: A semantically annotated Wikipedia XML corpus, in: BTW '07: Proceedings of GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW2007), pp. 277–291.

[35] S. Schlobach, D. Ahn, M. de Rijke, V. Jijkoun, Data-driven type checking in open domain question answering, Journal of Applied Logic 5 (2007) 121–143.

[36] K. Spärck-Jones, S. Robertson, D. Hiemstra, H. Zaragoza, Language modelling and relevance, in: Language Modeling for Information Retrieval, Kluwer Academic Publishers, 2003, pp. 57–71.

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

[38] M. Strube, S. Ponzetto, WikiRelate! Computing semantic relatedness using Wikipedia, in: AAAI '06: Proceedings of the 21st national conference on Artificial intelligence, AAAI Press, 2006, pp. 1419–1420.

[39] T. Tsikrika, P. Serdyukov, H. Rode, T. Westerveld, R. Aly, D. Hiemstra, A.P. de Vries, Structured document retrieval, multimedia retrieval, and entity ranking using PF/Tijah, in: INEX 2006: Comparative Evaluation of XML Information Retrieval Systems: Fifth Workshop of the INitiative for the Evaluation of XML Retrieval, volume 4518 of *LNCS*, Springer Verlag, Berlin / Heidelberg, 2007, pp. 306–320.

[40] A.M. Vercoustre, J. Pehcevski, J.A. Thom, Using Wikipedia categories and links in entity ranking, in: INEX 2007: Focused access to XML documents: 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, volume 4862 of *LNCS*, Springer Verlag, Berlin / Heidelberg, 2008, pp. 321–335.

[41] A.M. Vercoustre, J.A. Thom, J. Pehcevski, Entity ranking in Wkipedia, in: SAC '08: Proceedings of the 2008 ACM symposium on Applied computing, ACM, New York, NY, USA, 2008, pp. 1101–1106.

[42] A.P. de Vries, A.M. Vercoustre, J.A. Thom, N. Craswell, M. Lalmas, Overview of the INEX 2007 entity ranking track, in: INEX 2007: Focused access to XML documents: 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, volume 4862 of *LNCS*, Springer Verlag, Berlin / Heidelberg, 2008, pp. 245–251.

[43] X. Wei, W.B. Croft, Investigating retrieval performance with manually-built topic models, in: RIAO '07: Large Scale Semantic Access to Content (Text, Image, Video, and Sound), pp. 333–349.

[44] C. Whitelaw, A. Kehlenbeck, N. Petrovic, L. Ungar, Web-scale named entity recognition, in: CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management, ACM, New York, NY, USA, 2008, pp. 123–132.

[45] X. Wu, L. Zhang, Y. Yu, Exploring social annotations for the semantic web, in: WWW '06: Proceedings of the 15th international conference on World Wide Web, ACM, New York, NY, USA, 2006, pp. 417–426.

[46] E. Yilmaz, E. Kanoulas, J.A. Aslam, A simple and efficient sampling method for estimating AP and NDCG, in: SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, ACM, New York, NY, USA, 2008, pp. 603–610.

[47] C. Zhai, J. Lafferty, A study of smoothing methods for language models applied to ad hoc information retrieval, in: SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, New York, NY, USA, 2001, pp. 49–56.