Deriving a Domain Specific Test Collection from a Query Log

Avi Arampatzis[1] **Jaap Kamps**[1,2] **Marijn Koolen**[1] **Nir Nussbaum**[2]
[1] Archives and Information Science, University of Amsterdam
[2] ISLA, Informatics Institute, University of Amsterdam

Abstract

Cultural heritage, and other special domains, pose a particular problem for information retrieval: evaluation requires a dedicated test collection that takes the particular documents and information requests into account, but building such a test collection requires substantial human effort. This paper investigates methods of generating a document retrieval test collection from a search engine's transaction log, based on submitted queries and user-click data. We test our methods on a museum's search log file, and compare the quality of the generated test collections against a collection with manually generated and judged known-item topics. Our main findings are the following. First, the test collection derived from a transaction log corresponds well to the actual search experience of real users. Second, the ranking of systems based on the derived judgments corresponds well to the ranking based on the manual topics. Third, deriving pseudo-relevance judgments from a transaction log file is an attractive option in domains where dedicated test collections are not readily available.

1 Introduction

Cultural heritage, and other special domains, pose a particular problem for information retrieval. Progress in information retrieval depends heavily on the availability of suitable test collections consisting of a set of documents; a set of search topics; and (human) relevance judgments. Standard benchmarks, such as those developed at TREC (2007), have been developed using newspaper and newswire data. Whilst these test collections are immensely useful to evaluate generic properties of retrieval systems, such as fundamental ranking principles, they do not capture the specific context of particular domains (Ingwersen and Järvelin, 2005). To take cultural heritage as an example, the documents are cultural heritage descriptions which are different in character from newspaper articles, and also the search requests and relevance judgments about art are more subjective than factual queries about news (Koolen et al., 2007). As a result, special domains like cultural heritage require a dedicated test collection that takes the particular documents and information requests into account, but building such a test collection requires substantial human effort.

We opt for a different approach. Search engines commonly store the actions of users in transaction logs, which allow an unobtrusive way of studying user behaviour. Logs contain valuable information such as what searchers are looking for, what results they find interesting enough to click on, etc. In this paper, we investigate methods of extracting queries and user-clicks (on the search result items) from transaction logs in order to create a quality test collection for Document Retrieval.

A quality test collection for Document Retrieval is traditionally considered as a set of queries on a document collection with *complete* and *reliable* relevance judgements. Complete in the sense that all documents are judged for relevance against all queries, and reliable in the sense that judgements are sta-

Proceedings of the Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2007), pages 73–80,
Prague, 28 June 2007.

ble across a majority of human assessors. Nevertheless, considering the fact that a test collection is used *"as a mechanism for comparing system performance"* (Voorhees, 2002), the requirements for completeness and reliability may be relaxed somewhat.

The Text REtrial Conference (TREC) has traditionally used incomplete judgements for comparing system effectiveness via the "pooling" method (Jones and van Rijsbergen, 1975), and it is also well-known that human assessor agreement is relatively low (Voorhees and Harman, 2005). Consequently, test collections which *preserve* the effectiveness ranking of several systems can be considered of equivalent quality in the context of comparing system effectiveness. In order to evaluate the quality of test collections extracted in various ways from a transaction log, it would be sufficient to compare their ability to rank several retrieval systems against a reference system ranking produced by an already known good test collection not produced from the log.

One can think of several ways of extracting queries and clicks from a transaction log and turning them into a set of queries with relevance judgments. A simple (and naive) way would be to treat every query typed by a user as a topic, and every result that the user clicked on as a positive relevance judgment. However, such an approach may not lead to a good test set. Previous research on user click behaviour has shown that clicks on search engine results do not directly correspond to explicit, absolute relevance judgments, but can be considered as *relative* relevance judgments (Joachims et al., 2005), i.e., if a user skips result *a* and clicks on result *b*, than the user preference reflects $rank(b) > rank(a)$. Moreover, the occurrence frequencies of queries and the numbers of retrieved items vary significantly across queries which may lead to wide variation in effectiveness.

The challenge we take up has several dimensions which can be summarized in the following questions:

- How can we derive topics and pseudo-relevance judgments from a transaction log file, and how does this impact the quality of the generated test collection?
- How does system effectiveness on the automatically generated test collection compare to the effectiveness on a set of manually constructed known-item topics?

If automatic methods of building test collections are indeed feasible, this opens up a whole new dimension of possibilities for Information Retrieval evaluation: there is an enormous lengths of transaction logs generated daily at numerous web-sites and at on-line search engines.

The rest of this paper is organized as follows. Next, in Section 2 we discuss transaction logs in general, and the specific transaction log from a museum that we'll use in the case study of this paper. Section 3 details how we have extracted topics and pseudo-relevance judgments from a museum's log file, and their evaluation. Then, in Section 4, we evaluate the merits of the derived test collection in comparison to human generated and judged topics. We end with Section 5 in which we summarize our findings.

2 Transaction Logs

2.1 Previous Work

There has been substantial interest in using click-through data from transaction logs as a form of implicit feedback (Dumais et al., 2003). A range of implicit feedback techniques have been used for query expansion and user profiling in information retrieval tasks (Oard and Kim, 2001; Kelly and Teevan, 2003). Joachims et al. (2005, p.160) conclude that "the implicit feedback generated from clicks shows reasonable agreement with the explicit judgments of the pages".

Transaction logs have been analysed to study user search behaviour in Web search engines (Chau et al., 2005) and digital libraries (Jones et al., 2000), amongst others (Jansen, 2006). In Chau et al. (2005), user behaviour is studied using the transaction log of a website's search engine and is compared to that of general purpose search engines. They find that the number of query terms used for website search engines is comparable to queries submitted to general purpose search engines, but the search topics and terms are different.

In this paper, we go one step further and try to exploit the user behaviour implicit in the data to con-

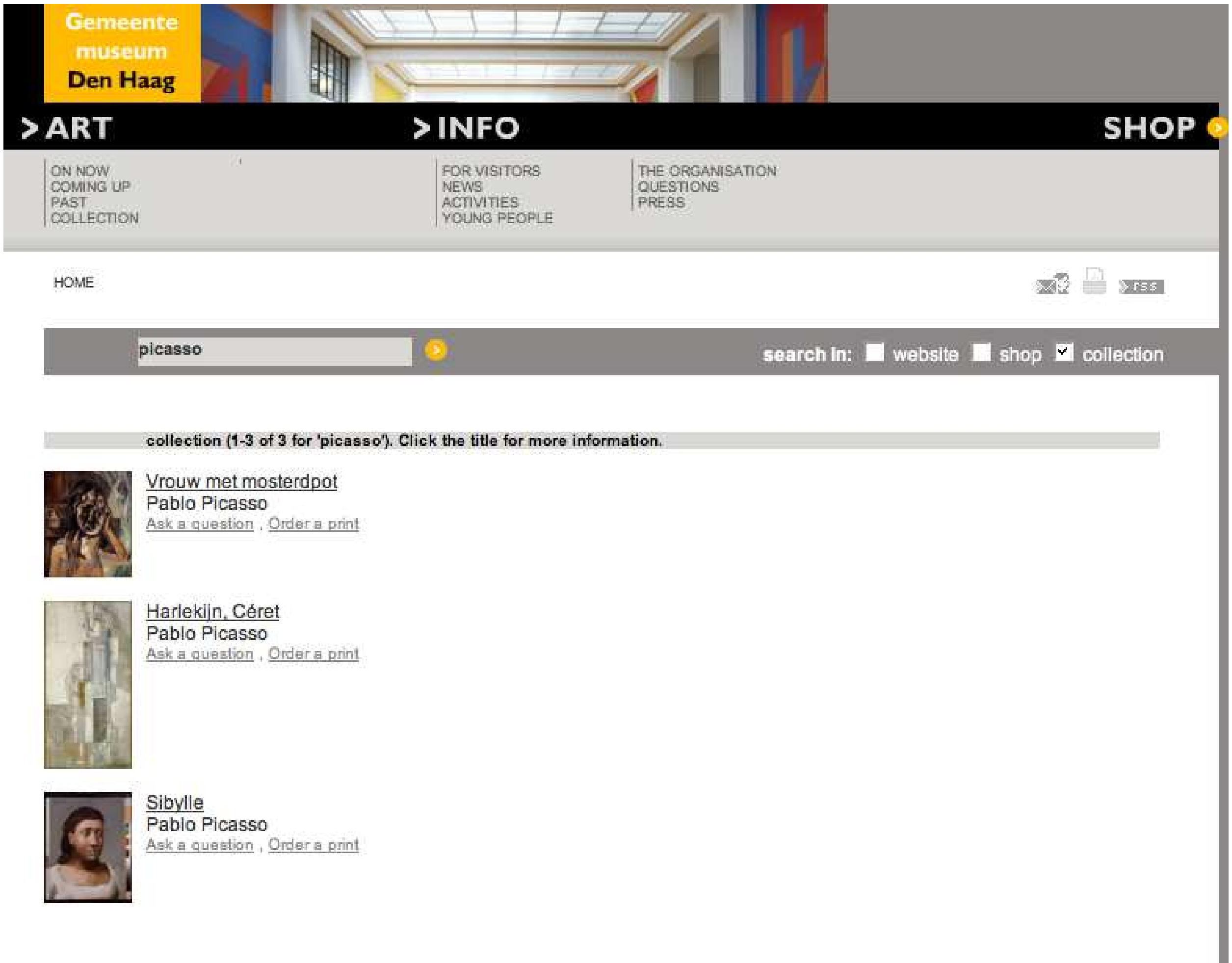

Figure 1: The search engine of the Gemeentemuseum's website.

struct a test set with real user needs, queries and judgments.

2.2 A Website's Search Engine

The website of the *Haags Gemeentemuseum*[1] in the Hague, the Netherlands, offers a search engine for three different parts of the *Gemeentemuseum*, the website content, the on-line shop, and the highlights of the museum's object collection (see Figure 2.1). The searchable on-line collection consists of 1,127 objects, the highlights of the museum, from a total database of 116,493 museum objects. The metadata of these objects are stored in a legacy system, and queries are matched against the title and creator fields (Koolen et al., 2007). The descriptions contain many more fields, however. The objects database treats the query as a Boolean AND query, and returns a warning if there is no object description containing all terms in one field. Although the database allows a drop-back to the individual terms, the website search engine retains a strict Boolean AND query and returns an empty result list.

The transaction log contains the transactions from the server side. The website uses a Java script to interact with the search engine. The query itself is not stored in the transaction log. If a user clicks on a result that leads to another web page in the domain, or to an item in the shop, this click is registered in the transaction, but the actual query is not. If a user clicks on a result from the object collection however, the database query is stored in the transaction log, from which we can extract the actual user query, and the object that user wants to see.

[1] http://www.gemeentemuseum.nl

This has an effect on the queries found in the log file. Queries containing both title and creator names often lead to an empty result list, as there is no single field containing both creator and title terms. The database looks for all the terms in one field at a time, and will not match with any object. With an empty result list, users cannot click on an object and hence, the query is not logged. Another effect is that all the results that users can click on have all the query terms in either the title or creator field. Although end users sometimes express their information needs in terms different from the terms chosen by indexers, i.e. the curators in the museum (Markkula and Sormunen, 2000), this discrepancy cannot be observed in the log-file data.

This may lead to the concern that the topics that can be extracted from the transaction log are "easy" topics, since the relevant descriptions necessarily contain all the query terms. It is unclear whether this affects the extracted topic set significantly, since we will look only at the relative ranking of systems over a set of queries. We will compare the ability to rank systems of our automatically generated topic sets with the system ranking ability of a manual topic set. If the extracted topic sets preserve the system ranking of the manual topic set, the bias in the topic sets towards "easy" topics has no negative influence on the quality of the topic sets.

3 Experiments and setup

We have obtained the log files covering a period of one and a half years, between September 14, 2005 and February 26, 2007.

From the transaction log, we extracted the queries and the object identifiers from the database query, and turned them into Qrels, i.e., the object is relevant for the query.

We use the following terminology:

- **User**: the client side of the transaction, identified by ip-address.
- **Transaction**: any exchange between client (user) and server (system), corresponding to a line in the transaction log.
- **Session**: A sequence of transactions by the same user, where the maximum interval between transaction n and $n + 1$ is 1 hour. More than 1 hour of inactivity signals a session boundary.
- **Query**: the string typed by the user as it appears in the transaction log.
- **Result**: the identifier of the museum object, used to retrieve the object data from the object database.

Topic set	# Topics	Query length average	Query length median	Avg. # rel. docs
Raw	7,531	1.18	1	2.38
Union	1,183	1.38	1	3.86
Intersection	974	1.42	1	1.41
Manual	150	2.38	2	1.00

Table 1: Statistics on the extracted topic sets.

3.1 Extraction methods

We used 3 extraction methods to construct a test set:

1. **Raw queries**: each query appearing in the log is used, i.e. the bag of queries. Here, a topic consist of a query and the corresponding clicked results from one session. If the same user types the same query in another session, this is treated as a new topic.
2. **Unique union**: All unique queries are used, i.e. the set of queries. All the results clicked by all users typing the same query are considered relevant documents.
3. **Unique intersection**: All unique queries are used, i.e. the set of queries. The intersection of the results clicked by all users typing the same query are considered relevant documents. Thus, a result is relevant only if all users who typed the query, clicked on that result.

Table 1 shows statistics on the resulting topic sets. In calculating these numbers, stop words were removed from the queries. As most queries are in Dutch, we used the standard Snowball stopword list for Dutch (Snowball, 2007). The queries are very short on average. For the Raw, Union and Intersection topic sets, the queries with 1 term form 84%, 70% and 68% of the query sets respectively. There are 1,183 unique queries, and on average, 3.86 results are clicked by at least one user. Understandably, the Intersection set has less topics than the Union set, as there are queries with no single result clicked on by all users. Also, the average number of relevant documents per topic is lower for the intersection set.

We created 150 Known-Item topics by hand and used this test set, referred to as KI-topics, on the same collection and include the results as a comparison with the new test sets. Table 1 shows the statistics of these human generated topics in the last row. These search request have more verbose topic statements with a median length of 2, compared to a median length of 1 for the query log topics. Also the number of relevant documents differs considerably, with a unique relevant page for the human known-item topics, and several “clicked” pages per query for the transaction log.

3.2 Retrieval system

To see if our test sets lead to a stable system ranking, we need a number of retrieval systems to compare their ranking on the different test collections. To get a number of different systems, we simply use a standard retrieval model with different parameter settings to create different runs.

We use a standard language model (Hiemstra, 2001). Our system is an extension to Lucene (ILPS, 2005) and uses Jelinek-Mercer smoothing, controlled by the parameter λ, and a length prior, controlled by the parameter β, i.e., for a collection D, document d and query q:

$$P(d|q) = P(d) \cdot \prod_{t \in q} \left((1-\lambda) \cdot P(t|D) + \lambda \cdot P(t|d) \right), \tag{1}$$

where

$$P(t|d) = \frac{tf_{t,d}}{|d|} \tag{2}$$

$$P(t|D) = \frac{\text{doc\_freq}(t, D)}{\sum_{t' \in D} \text{doc\_freq}(t', D)} \tag{3}$$

$$P(d) = \frac{|d|}{\sum_{d' \in D} |d|} \tag{4}$$

We assign a prior probability to an document d relative to its length in the following manner:

$$P(d) = \frac{|d|^{\beta}}{\sum_d |d|^{\beta}}, \tag{5}$$

System	λ	β
A	0.10	0
B	0.50	0
C	0.90	0
D	0.10	1
E	0.50	1
F	0.90	1
G	0.10	2
H	0.50	2
I	0.90	2

Table 2: Parameter settings for the different systems.

where $|d|$ is the length of a document d. The β parameter introduces a length bias which is proportional to the document length with $\beta = 1$ (the default setting). For more details on language models and smoothing, see (Hiemstra, 2001). For details on the effect of the length parameter, see (Kamps et al., 2004).

3.3 Experimental Set-up

In our experiments we will emulate a set of different retrieval systems by using arbitrary parameter settings for smoothing (λ) and length prior (β). This will result in a range of different rankings of documents, and we can compare their retrieval effectiveness on our various topic sets. In this way, we can compare the system ranking of the automatically generated topic sets with the system ranking of a manually crafted topic set.

We made 9 different runs with each topic set, using 3 different values (0.10, 0.50 and 0.90) for the smoothing parameter λ, corresponding to heavy, average and little smoothing respectively, and 3 different values (0, 1 and 2) for the length prior β corresponding to no length normalization and length normalization proportional to the document length.

To measure the correlation of the system rankings resulting from the different topic sets, we look at Kendall's tau coefficient.

4 Results

Table 3 shows the detailed results for all runs over all topics sets. As noted above, we will focus on the relative system rankings over topic sets. We limit our analysis to the performance in terms of mean-

Topics	# Topics	MRR	Success@10
Raw topics $\beta = 0, \lambda = 0.10$	7,527	0.5974	0.8023
Raw topics $\beta = 0, \lambda = 0.50$	7,527	0.5970	0.8030
Raw topics $\beta = 0, \lambda = 0.90$	7,527	0.5970	0.8031
Raw topics $\beta = 1, \lambda = 0.10$	7,527	0.5673	0.7506
Raw topics $\beta = 1, \lambda = 0.50$	7,527	0.5765	0.7574
Raw topics $\beta = 1, \lambda = 0.90$	7,527	0.5767	0.7574
Raw topics $\beta = 2, \lambda = 0.10$	7,527	0.5531	0.7427
Raw topics $\beta = 2, \lambda = 0.50$	7,527	0.5618	0.7468
Raw topics $\beta = 2, \lambda = 0.90$	7,527	0.5644	0.7474
Union $\beta = 0, \lambda = 0.10$	1,183	0.6908	0.8191
Union $\beta = 0, \lambda = 0.50$	1,183	0.6925	0.8233
Union $\beta = 0, \lambda = 0.90$	1,183	0.6927	0.8233
Union $\beta = 1, \lambda = 0.10$	1,183	0.6622	0.7887
Union $\beta = 1, \lambda = 0.50$	1,183	0.6772	0.8005
Union $\beta = 1, \lambda = 0.90$	1,183	0.6782	0.8005
Union $\beta = 2, \lambda = 0.10$	1,183	0.6216	0.7566
Union $\beta = 2, \lambda = 0.50$	1,183	0.6477	0.7828
Union $\beta = 2, \lambda = 0.90$	1,183	0.6515	0.7870
Intersection $\beta = 0, \lambda = 0.10$	974	0.6481	0.8008
Intersection $\beta = 0, \lambda = 0.50$	974	0.6505	0.8049
Intersection $\beta = 0, \lambda = 0.90$	974	0.6506	0.8049
Intersection $\beta = 1, \lambda = 0.10$	974	0.6187	0.7690
Intersection $\beta = 1, \lambda = 0.50$	974	0.6329	0.7793
Intersection $\beta = 1, \lambda = 0.90$	974	0.6341	0.7793
Intersection $\beta = 2, \lambda = 0.10$	974	0.5783	0.7310
Intersection $\beta = 2, \lambda = 0.50$	974	0.6053	0.7618
Intersection $\beta = 2, \lambda = 0.90$	974	0.6093	0.7659
KI-topics $\beta = 0.0\lambda = 0.10$	150	0.5446	0.7067
KI-topics $\beta = 0.0\lambda = 0.50$	150	0.5590	0.7267
KI-topics $\beta = 0.0\lambda = 0.90$	150	0.5608	0.7200
KI-topics $\beta = 1.0\lambda = 0.10$	150	0.5253	0.7067
KI-topics $\beta = 1.0\lambda = 0.50$	150	0.5465	0.7200
KI-topics $\beta = 1.0\lambda = 0.90$	150	0.5516	0.7200
KI-topics $\beta = 2.0\lambda = 0.10$	150	0.4602	0.6667
KI-topics $\beta = 2.0\lambda = 0.50$	150	0.5196	0.7133
KI-topics $\beta = 2.0\lambda = 0.90$	150	0.5292	0.7133

Table 3: Mean Reciprocal Rank and Success@10 for all topic sets on the web site objects.

Topic set	System ranking
Raw	$A \succ B \succeq C \succ F \succ E \succ D \succ I \succ H \succ G$
Union	$C \succ B \succ A \succ F \succ E \succ D \succ I \succ H \succ G$
Intersection	$C \succ B \succ A \succ F \succ E \succ D \succ I \succ H \succ G$
KI-topics	$C \succ B \succ F \succ E \succ A \succ I \succ D \succ H \succ G$

Table 4: Systems rankings of the 4 topic sets.

	KI-topics	Raw	Union	Intersect.
KI-topics	1.00			
Raw	0.67	1.00		
Union	0.83	0.83	1.00	
Intersection	0.83	0.83	1.00	1.00

Table 5: Rank correlation coefficients between the topic sets.

reciprocal rank (i.e., 1 over the rank at which the first relevant document is found). The rankings over the four different topic sets are given in Table 4 (based on the labeling introduced in Table 2).

The results show that ranking based on the Raw Topic set deviates slightly from ranking based on the Union and Intersection topic sets. The Union and Intersection topic sets result in exactly the same ranking. There is a clear grouping of systems with the same length prior. The systems without a length prior (A,B and C) outrank the systems with a length prior $\beta = 1$ (D, E and F), which in turn outrank the systems with length prior $\beta = 2$ (systems G, H and I). Within these groups, the system ranks correspond to the smoothing parameter settings. A higher λ value corresponds to a higher rank. The only deviation is observed in the ranking based on the Raw Topic set. Here, the lowest value for λ leads to the best performance for the systems with no length prior.

If we compare the three automatically generated topic sets to the manual known-item topic set, we see some more differences. For the manual topics, systems E and F, which have a unit length prior, outrank system A, which has no length prior. A possible explanation for this is that the higher λ of systems E and F help the longer queries of the manual topic set. In the other topic sets, most of the queries have only one term, so smoothing has very little influence. This same effect might explain why system I outranks system D.

If we look at the correlation coefficient (Table 5), we see a positive correlation between all topic sets. As the Union and Intersection topic sets lead to the same system ranking, they have a correlation of 1. The system ranking of the Raw topic set shows the lowest correlation with the other topic sets, but the correlation with the manual topic set is still high, indicating that all the extraction methods lead to topic sets that have an ability to rank system similar to that of a manually constructed topic set. Of course, the number of known-item topics is much smaller than the other topic sets, but these initial results point out that the automatic generation of test collections from transaction logs makes sense.

5 Discussion and Conclusions

Cultural heritage, and other special domains, pose a particular problem for information retrieval: evaluation requires a dedicated test collection that takes the particular documents and information requests into account, but building such a test collection requires substantial human effort. We have investigated methods of generating a document retrieval test collection from a search engine's transaction log, based on submitted queries and user-click data. We tested our methods on a museum's search log file, and compared the quality of the generated test collections against a collection with manually generated and judged known-item topics.

Our main findings are the following. First, the test collection derived from a transaction log corresponds well to the actual search experience of real users. An important criterion of bench-marks is that they correspond well to the real-world phenomenon that they are supposed to measure. By basing the test collection directly on a large sample of real end-user interaction, with real information needs, we can ensure that the test collection reflects the information seeking behaviors of users well. This is of particular importance for domain-specific test collections, where results may be impacted by the particular type of information available, and the particular sorts of search requests that are likely to be issued.

Second, the ranking of systems based on the derived judgments corresponds well to the ranking based on the manual topics. We extracted three different sets of topics and corresponding pseudo-relevance judgments from the transaction log. All three sets result in very similar system rankings, indicating that the results are robust against particular choices in the extraction phase. The system rankings are corresponding well to a ranking based on human generated known-item topics. Given the promising initial results, we are currently working on a more rigorous comparative evaluation, with more human topics, and more diverse systems to be ranked, aiming to understand better the exact conditions under which the extracted test collections behave similar to human generated test collections—and when they behave differently.

Third, deriving pseudo-relevance judgments from a transaction log file is an attractive option in domains where dedicated test collections are not readily available. The results in the paper should not be interpreted as a claim to replace human relevance judgments with extracted topics and pseudo-relevance judgments. There are however many domains and tasks where no suitable test collection is available, and creating a new human test collection might be either impractical or even impossible. Recall that creating human judged test collections requires considerable effort: it is usually a community effort where a number of participating teams provide a diverse set of runs needed for pooling, or even engage in peer-assessments. Hence, deriving a test collection from a transaction log—if available—can be an attractive alternative.

Acknowledgments

This research is part of the MUSEUM (MUltiple-collection SEarching Using Metadata; `http://www.nwo.nl/catch/museum/`) project of the CATCH (Continuous Access To Cultural Heritage) research program in the Netherlands.

The authors were supported by the Netherlands Organization for Scientific Research (NWO, grants # 612.066.513, 639.072.601, and 640.001.501), and by the E.U.'s 6th FP for RTD (project MultiMatch contract IST-033104).

References

Michael Chau, Xiao Fang, and Olivia R. Liu Sheng. 2005. Analysis of the query logs of a web site search engine. *J. Am. Soc. Inf. Sci. Technol.*, 56(13):1363–1376.

Susan Dumais, Thorsten Joachims, Krishna Bharat, and Andreas Weigend. 2003. SIGIR 2003 workshop report: implicit measures of user interests and preferences. *SIGIR Forum*, 37:50–54.

Djoerd Hiemstra. 2001. *Using Language Models for Information Retrieval*. Thesis, University of Twente.

ILPS. 2005. The *ilps* extension of the *lucene* search engine. `http://ilps.science.uva.nl/Resources/`.

Peter Ingwersen and Kalervo Järvelin. 2005. *The Turn: Integration of Information Seeking and Retrieval in Context*. The Kluwer International Series on Information Retrieval. Springer Verlag, Heidelberg.

Bernard J. Jansen. 2006. Search log analysis: What is it; what's been done; how to do it. *Library and Information Science Research*, 28(3):407–432.

Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161. ACM Press, New York, NY, USA.

Karen Sparck Jones and C. van Rijsbergen. 1975. Report on the need for and provision of an "ideal" information retrieval test collection. British Library Research and Development report 5266, Computer Laboratory, University of Cambridge.

Steve Jones, Sally Jo Cunningham, Rodger J. McNab, and Stefan J. Boddie. 2000. A transaction log analysis of a digital library. *Int. j. on Digital Libraries*, 3(2):152–169. URL `citeseer.ist.psu.edu/jones00transaction.html`.

Jaap Kamps, Maarten de Rijke, and Börkur Sigurbjörnsson. 2004. Length normalization in xml retrieval. In Mark Sanderson, Kalervo Järvelin, James Allan, and Peter Bruza, editors, *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 80–87. ACM Press, New York, NY, USA.

Diane Kelly and Jaime Teevan. 2003. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37:18–28.

Marijn Koolen, Avi Arampatzis, Jaap Kamps, Nir Nussbaum, and Vincent de Keijzer. 2007. Unified access to heterogeneous data in cultural heritage. To appear.

Marjo Markkula and Eero Sormunen. 2000. End-user searching challenges indexing practices in the digital newspaper photo archive. *Information Retrieval*, 1:259–285.

Douglas W. Oard and Jinmook Kim. 2001. Modeling information content using observable behavior. In *Proceedings of the 64th Annual Meeting of the American Society for Information Science and Technology*, pages 38–45.

Snowball. 2007. Stemming algorithms for use in information retrieval. `http://www.snowball.tartarus.org/`.

TREC. 2007. Text REtrieval Conference. `http://trec.nist.gov/`.

Ellen M. Voorhees. 2002. The philosophy of information retrieval evaluation. In Carol Peters, Martin Braschler, Julio Gonzalo, and Michael Kluck, editors, *Evaluation of Cross-Language Information Retrieval Systems, CLEF 2001*, volume 2406 of *Lecture Notes in Computer Science*, pages 355–370. Springer.

Ellen M. Voorhees and Donna K. Harman, editors. 2005. *TREC: Experimentation and Evaluation in Information Retrieval*. MIT Press.