An Exploration of Retrieval-Enhancing Methods for Integrated Search in a Digital Library

Diana Ransgaard Sørensen, Toine Bogers, Birger Larsen
Royal School of Library and Information Science, Birketinget 6, 2300, Copenhagen, Denmark
{drs,tb,blar}@iva.dk

ABSTRACT

Integrated search is defined as searching across different document types and representations simultaneously, with the goal of presenting the user with a single ranked result list containing the optimal mix of document types. In this paper, we compare various approaches to integrating three different types of documents (bibliographic records for articles and books as well as full-text articles) using the iSearch collection: combining all document types in a single index, weighting the different document types using priors, and using collection fusion techniques to merge the retrieval results on three separate indexes corresponding to each of the document types. We find that a properly optimized retrieval model on a single combined index containing all documents without any special treatment performs no worse than our weighting and fusion methods, suggesting that more work is needed on alternative approaches to integrated search.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.3 Information Search and Retrieval

General Terms

Algorithms, Experimentation, Performance

Keywords

Integrated search, prior weighting, collection fusion

1 Introduction

The need for effective *integrated search* is becoming increasingly important as more and more textual sources are being digitized and made available through search engines. Integrated search is defined as searching across many different document types and representations simultaneously, with the end goal of presenting the user with a single ranked result list containing the optimal mix of document types. Digital libraries are a prototypical example of a domain where different types of document representations need to be integrated into a single ranking, e.g. bibliographic records for books, articles, conference papers, and multimedia as well as (increasingly) full-text versions of these information objects. Integrating these different types of document representations can be complicated due to differences in formats, metadata quality, and imbalances in the amount of text between different representation types.

A naive approach to integrating different document types and representations in digital libraries would be to add everything to the same index and use an out-of-the-box, unoptimized search engine for retrieval. However, this does not take into account the inherent differences between different document types: certain types could come to dominate the document rankings unfairly using such a one-size-fits-all approach. For instance, full-text documents could be expected to be returned more frequently than records containing only metadata, because there are more ways of matching user queries. How can we address these imbalances so that the most relevant documents are returned to the user instead of just the documents with the most text?

This paper focuses on how to best rank and combine different types of document representations to produce the most relevant results. For our experiments, we use iSearch, a test collection based on the digital physics library arXiv.org[1], which contains three different types of document representations of varying information density [6]. We compare retrieval over a single index with all document types combined with using priors as well as techniques from collection fusion to weight the different document types. We find that while properly optimizing the retrieval parameters can have a significant positive influence on retrieval performance, our weighting methods are ineffective in improving performance. Our main contribution is therefore an overview of what *does* and does *not* work on the iSearch collection.

We discuss work related to integrated search in the next section. Section 3 describes our methodology. We describe our baseline runs on a combined index of all document types in Section 4. In Sections 5 and 6 we respectively describe our experiments with weighting different document types and collection fusion methods to improve retrieval performance. We discuss our findings and conclude in Section 7.

2 Background

As more information becomes available in digital form there is an increasing need to access information across types and genres. Examples of services that rise to this challenge include web search engines diversifying over different media

[1] http://arxiv.org/

types and mixing these in universal search result lists, as well as academic library integrated search systems that search across different collections, document types, genres and with different levels of representation. Also recent efforts in The European Library[2] and Europeana[3] to provide access to the collective contents of national libraries, archives, galleries and museums of Europe emphasize the need for solutions that can integrate search across very diverse collections.

Case studies reveal that users generally find it challenging to search across various heterogeneous bibliographic types for relevant information, in comparison to the web search engines and full-text systems they are familiar with [3]. The complexity of accessing the various 'vertical' sources and systems offered by digital libraries may threaten to potentially turn users away from libraries altogether [10]. Attempts have been made to address this issue, by federating different but simultaneous retrieval results into one result list [12]. However, studies of student and researcher information acquisition patterns in digital libraries indicate that users prefer simple systems that contain all sources in one and are easy to search [13]. This type of integrated search [2, 6] is similar to the notion of aggregated search [9], and is currently being developed and implemented in large scale systems in research libraries and as commercial products. However, academic research on how to best integrate results of different types has been limited because of the lack of suitable testbeds for such studies. The recently released iSearch test collection[4] is one of the few examples of an IR test collection designed for integrated search experiments (see Section below). As the collection has not yet been widely used we carry out a number of initial experiments testing how to best integrate different document types.

3 Methodology

Our goal is to study optimization of search engine performance through document weighting and data fusion in an integrated search scenario.

3.1 The iSearch collection

A suitable test collection for our experiments is iSearch [6], which contains scientific documents from physics, collected from arXiv.org and from the union catalogue for all Danish libraries[5]. iSearch comprises three types of content: (i) 143,571 full length articles (PF), (ii) 291,246 article metadata records (PN), and (iii) 18,443 book metadata records (BK). The iSearch test collection is specifically designed for IR evaluation in an integrated search scenario representing a hybrid digital library with a large amount of articles represented by metadata (ii), a sizable amount of articles in full text (i) and a smaller set of books represented by metadata records (iii). For the full length articles we extracted the full text from the PDF version of the articles. The article metadata records contain the article title, subject(s), and typically a description of several lines. The book records contain much less information, typically the title, subject(s), and in some cases a description.

iSearch comes with a set of 65 queries and their relevance assessments, which have been created by 23 lecturers and experienced postgraduate and graduate students from three different university departments of physics. The queries represent real information seeking tasks of their authors. Each query contains five different fields, each of which corresponds to a different aspect of the user's information need and context. These five query fields are: description of information sought, user background, work task, ideal answer, keywords.

The iSearch relevance assessment of each query was made by the same user who formulated that query, by examining a pool of up to 200 documents retrieved for that query. Those 200 documents offered to the users for assessment were retrieved by manual search, making use of any clues in the five query fields and by exploiting available search operators and facilities, such as fielded search, proximity operators, classification codes etc. The goal of this assessment approach has been to carry out an exhaustive yet precise search much as a university librarian would do. The users assessed the relevance of those documents on a 4-point scale: highly, fairly, marginally and non-relevant.

3.2 Experimental setup

Indexing & retrieval We used the Indri 5.0 toolkit[6] with the retrieval model described in [7] which allows for the evaluation of structured queries using an IR algorithm based on language modeling. Indri offers three variants of this IR model, based on one of three different smoothing methods:

- **Jelinek-Mercer (JM) smoothing** is used as a mixture model of the document and collection language models [15]. The λ parameter ($0 \leq \lambda \leq 1$) controls the influence of the collection language model; higher values boost the collection language model, and lower values boost the document language model.
- **Bayesian smoothing using Dirichlet priors (DIR)** uses the Dirichlet distribution as the conjugate prior for Bayesian analysis [15]. The μ parameter controls the smoothing based on the document length and ranges from 0 to 5000 (as a practical upper value).
- **Two-stage smoothing (TWO)** combines Jelinek-Mercer and Dirichlet smoothing, and as such has both the λ and μ parameters that influence the smoothing.

In addition to optimizing the different parameters for the retrieval models mentioned above, we also examine the value of stop word filtering and stemming. We use the SMART stop word list and Krovetz stemming. We construct two types of indexes: (1) a combined index of all the documents and document types of iSearch, and (2) three separate indexes, one for each document type BK, PF and PN.

Evaluation The topics in the iSearch collection come with graded relevance judgments, so we used Normalized Discounted Cumulated Gain (NDCG) [4] as our evaluation metric. NDCG credits retrieval methods for their ability to retrieve highly relevant results at top ranks. We use `trec_eval` 8.1[7]. Wherever appropriate, we determine the significance of differences between two runs using a two-tailed paired Student's t-test. We will denote significant differences against the baseline run using $^{\triangle}$ (and $^{\triangledown}$) for $\alpha = .05$ and $^{\blacktriangle}$ (and $^{\blacktriangledown}$)

[2] http://www.theeuropeanlibrary.org/
[3] http://www.europeana.eu/
[4] http://itlab.dbit.dk/~isearch
[5] http://www.danbib.dk/index.php?doc=english

[6] Available at http://www.lemurproject.org/indri/
[7] The `trec_eval` program computes NDCG with the modification that the discount is always $\log_2(rank + 1)$ so that rank 1 is not a special case.

Table 1: Out-of-the-box and optimized baseline runs using NDCG. $^{\triangle}$ $^{\blacktriangle}$ mark degrees of stat. significance.

	Model	λ	μ	Stop	Stem	NDCG
Default	Jelinek-Mercer	0.4	-	No	No	0.2779
	Dirichlet	-	2500	No	No	**0.2856**
	Two-stage	0.4	2500	No	No	0.2783
Optimized	Jelinek-Mercer	0.5	-	Yes	Yes	**0.3263** $^{\blacktriangle}$
	Dirichlet	-	1500	No	Yes	0.3136 $^{\triangle}$
	Two-stage	0.5	0	Yes	Yes	0.3263 $^{\blacktriangle}$

for $\alpha = .01$. E.g. $^{\triangle}$ signals a significant improvement of, for instance, a fusion run over the baseline run at $\alpha = .05$. For each topic we retrieve up to 1000 documents.

4 Baseline runs

A naive approach to integrating different types of document types and representations in digital libraries would be to add everything to the same index and use an out-of-the-box, unoptimized search engine to retrieve the document from it. We believe this to be a sub-optimal strategy for integrated search, because it does not take into account the inherent differences between different document types: certain types could come to dominate the document rankings unfairly using such a one-size-fits-all approach.

One of our working hypotheses in this paper is that, at the very least, we need to optimize our search engine on such a combined index: using the default, out-of-the-box settings does not provide the best retrieval performance, because the default parameter settings are necessarily a generalization over many different collections. To examine the value of optimization on a combined index of all three document types, we first generated retrieval runs using the default settings. The results of these default runs, as well as the default settings, are shown in the top half of Table 1. They show that Dirichlet smoothing outperforms the other two models with an NDCG score of 0.2856, although the difference with the default JM and TWO runs is not statistically significant.

Using the default, out-of-the-box settings does not necessarily provide the best retrieval performance, because the default parameter settings are a generalization over many different collections. We therefore optimized the three retrieval models by performing an exhaustive sweep of the possible parameter values: (i) **Stop word filtering**: Yes or no; (ii) **Krovetz stemming**: Yes or no; (iii) **smoothing parameters**: $\lambda \in [0 - 1]$ in steps of 0.1, $\mu \in \{0 - 5000\}$ in steps of 500.

The results of our optimization experiments can be found in the bottom half of Table 1. They show that optimization significantly improves the performance of the retrieval models on our combined index: optimization increases the NDCG scores by 17.4%, 9.8% and 17.2% for the JM, DIR, and TWO models respectively. The best performing model is JM with an NDCG score of 0.3263, which will serve as our baseline in the rest of the paper[8].

These results clearly support our expectations that optimization of the retrieval model produces beneficial results on a combined index of document types. However, we believe that simply treating all document representation types the same is not necessarily the best approach to integrated search. We believe that different methods are needed to weight the different document types differently to get the best performance in our integrated search scenario. We test some of our ideas for this in the next two sections.

[8]While the optimized TWO run achieves the same performance as JM, this is due to the value for μ being equal to 0, which means this is conceptually equal to JM smoothing with identical values of λ.

5 Prior weighting

The three iSearch document types have quite different characteristics, such as a notable difference in document length (cf. Section 3.1). We therefore investigate whether assigning different weights to each document type might smooth the inherent differences between document types and hence improve overall performance in an integrated search scenario. We implement this smoothing as prior probabilities in the language modeling framework [1]. Instead of assuming that the prior probability of a document is uniform for all documents, we vary this prior probability according to document types. We assign priors in the range [0.0001, 0.2, 0.4, 0.6, 0.8, 1.0], and add the log of these as priors to the combined index. These values are not a quantification of a certain feature found in the documents; we set them to assign different weights to different document types. We thus take a data-driven approach where we experiment with all prior values on all documents types and their combinations.

Table 2: Prior weights (columns 1-3) of the top-10 best performing weighted runs per document type.

BK weight	PN weight	PF weight	NDCG
1.0	0.2	0.2	0.3176
0.8	0.2	0.2	0.3155
1.0	0.4	0.2	0.3143
0.6	0.2	0.2	0.3141
1.0	0.2	0.4	0.3140
1.0	0.4	0.4	0.3136
0.8	0.4	0.2	0.3125
0.8	0.2	0.4	0.3123
1.0	0.4	0.6	0.3120
0.4	0.2	0.2	0.3116

With the range described above, there are 216 unique prior combinations of the three document types. Table 2 shows the top 10 best performing runs measured with NDCG. It can be observed that none of the runs outperform the optimized baseline (NDCG = 0.3263), although the difference is not statistically significant ($t(65) = -1.1939$, $p = 0.2368$). The best performing weighted runs overwhelmingly display the same weighting trend of higher weights for BK (1.0, 0.8, 0.6) and lower weights for PN and PF (0.4, 0.2). The lack of performance improvements means, however, that using uniform priors for document types is not an effective way of improving integrated search performance.

6 Fusion

An alternative to weighting the different document types using priors and generating a single, integrated list of results is *fusing* multiple retrieval runs into a single run. Despite the inevitable increase in computational overhead when fusing multiple runs, it also allows us to combine runs that have been optimized for a particular collection or document type, thereby improving potential retrieval performance of the fused run.

An important distinction to make when fusing retrieval runs in IR is the one between *collection fusion*, where the results of one or more algorithms on *different* document collections are integrated into a single results list, and *results fusion*, where the results of *different* retrieval algorithms on the *same* collection are combined [14]. In our integrated search scenario, we are interested in investigating whether fusing retrieval runs optimized over *different document types* can provide us with better performance than using a combined index as we described in Section 4. This corresponds to the original notion of collection fusion. First, we index the different document types in separate indexes and then merge the optimal retrieval runs over the different indexes. This way, each document type can be retrieved using the optimal retrieval model and parameter settings for that document type, without having to resort to a single, parameter setting that could be sub-optimal for the combination of all document types in a single index. We created separate indexes for each of the three document types BK, PF, and PN, and optimized performance on each index using an exhaustive parameter sweep as described in Section 4.

Different retrieval runs can generate wildly different ranges of similarity values, so we apply normalization to each retrieval result to map the score into the range $[0, 1]$. We normalize the original retrieval scores $score_{original}$ using the maximum and minimum retrieval scores $score_{max}$ and $score_{min}$ according to the formula proposed by Lee [5]:

$$score_{norm} = \frac{score_{original} - score_{min}}{score_{max} - score_{min}}. \quad (1)$$

When fusing different retrieval runs, there is an additional choice of combining the runs based on the retrieval scores or the ranks of the retrieved documents. These two options are commonly referred to as *score-based fusion* and *rank-based fusion* in the related work. The decision between score-based and rank-based fusion can also be seen as a decision of what should be normalized: the item ranks or the item scores. Early studies suggested that using retrieval scores over document ranks for data fusion results in superior performance [5], but later studies have found few significant differences between the two [11]. In the experiments described in this section we explore both options.

We investigated two types of document type fusion: (1) round-robin merging and (2) linearly combining the normalized retrieval scores. *Round-robin merging* is arguably one of the simplest collection fusion techniques, where we merge the three retrieval runs r_1, r_2, and r_3 (representing the BK, PF, and PF indexes) in the following way. We start by taking the documents returned at rank 1 in runs r_1, r_2, and r_3 and inserting those at the top of our merged result list. Then we take the documents returned at rank 2 in each of the three runs and append them to the merged list, followed by the third-highest ranked documents from each run, and so on. We continue until we exhaust the individual retrieval runs or until our merged run contains 1000 results. The order in which the individual runs r_1, r_2, and r_3 are consulted for new documents is randomly selected at the start, but kept the same for the duration of the merging process.

In our *linear combination* method, we produce a weighted combination of runs r_1, r_2, and r_3 by multiplying the normalized retrieval scores $score_{norm}(i, r_n)$ for each document i retrieved in run r_n by a weight w_n for that retrieval run according to:

$$score_{merged}(i) = \sum_{n=1}^{3} w_n \cdot score_{norm}(i, r_n) \quad (2)$$

Our three indexes contain disjoint document sets, so the final score $score_{merged}(i)$ will never be the sum of two or more retrieval scores for the same document i. The reason for weighting the runs representing the different indexes separately is that certain document types might come to dominate the rankings if we do not re-weight them. To avoid the exhaustive parameter sweep that comes with weighting more than two runs, we used a random-restart hill climbing algorithm to approximate the optimal weights for our individual document types. We randomly initialized the weights for each run, then varied each weight between 0 and 1 with increments of 0.1. We selected the value for which the NDCG score is maximized and then continued with the next weight. The order in which run weights were optimized was randomized, and we repeated the optimization process until the settings converged. We repeated this process 100 times, as the simple hill climbing algorithm is susceptible to local maxima. We then selected the weights that result in the best performance and generated the merged results list using these optimal weights.

Table 3 contains the results of our experiments with document type fusion for round-robin merging and linear combination with both score- and rank-normalization[9]. The results show that round-robin merging is not an effective strategy for integrating different document types in a search engine: it performs significantly worse than the baseline established in Section 4. Given the simple nature of the algorithm, which does not take performance differences between the runs into account, this is not surprising.

Table 3: Round-robin merging and linear combination (LC) with score- and rank-normalization. LC optimal weights are shown in columns 2-4. Baseline scores of the combined index included for reference. Bold mark best overall score.

Fusion method	Run weights			NDCG
	w_{BK}	w_{PN}	w_{PN}	
Round-robin merging	-	-	-	0.2144▼
LC (score norm.)	0.9	0.6	0.6	0.2896▼
LC (rank norm.)	1.0	1.0	1.0	**0.3286**
Baseline	-	-	-	0.3263

Linearly combining the three score-normalized runs is similarly ineffective, leading to an 11% decrease in performance. The combination of rank-normalization and linear combination does lead to a small improvement of 0.7%, which is not statistically significant. It is interesting to observe that this small improvement is obtained by weighting all document types equally, which means we cannot confirm our hypothesis that certain document types should be weighted more strongly than others to achieve better performance. We are therefore forced to conclude that document type fusion on individual indexes for the document types does not seem to be an effective strategy, based on this first investigation.

[9]Round-robin merging only looks at the document ranks while ignoring the normalized retrieval scores, so the distinction between score- and rank-normalization is meaningless here.

7 Discussion & conclusions

In this paper we examined two main approaches to the problem of integrated search: one where we treat all document types equally in a combined index containing all documents of all types, and one where we attempt to use various forms of weighting to promote certain types of documents over others. In our experiments with the combined, unweighted index we found that it pays to properly optimize the retrieval model used. As could be expected, we found significant performance gains for the optimized IR models compared to using an out-of-the-box, unoptimized retrieval algorithm.

We expected that weighting the document types in our combined index differently could boost performance even further, but this was not the case. The best weighted runs showed lower performance than the optimised baseline. The trend of the best weighted runs was that book records tended to have higher weights and article metadata and full text lower weights. This is an interesting finding with potential implications to Digital Library search functions.

In addition, we experimented with two standard collection fusion techniques to merge the retrieval results from three separate indexes, one for each document type. The lack of significant improvements (and in most cases significant decreases) in performance suggests that using relatively simple fusion techniques is not enough to determine the optimal way of integrating the different document types.

Overall, our results either suggest (1) that our methods of weighting the different document types were too simplistic to really affect performance, or (2) that in the iSearch collection there is no problem with imbalanced integration of different document types. At any rate, our experiments should be seen a case study on a specific collection, making it difficult to draw more general conclusions.

7.1 Future work

There are several promising avenues of research that could be pursued with the iSearch collection. A more extensive analysis of the performance of the individual document types could help us identify more fruitful techniques for weighting them properly. Another possibility could be to use the citation information from the documents available in the iSearch collection as an additional source of information. This way, more influential papers and books could be returned earlier, possibly reducing the dependence on the text present in the metadata records in the ranking process. In terms of prior weighting, we could calculate document-specific priors based on analysis of different document features, instead of painting the three document types with too broad a brush, and only assigning a prior based on the document type. Finally, we could explore different ranking models that use more information than the language modeling algorithms provided by Indri, such as Metzler et al.'s Markov random field model, which takes term dependencies into account [8].

Acknowledgements Funded by Denmark's Electronic Research Library (2007-003292) & the Danish Ministry of Culture Research Council (TAKT2008-040). We thank Christina Lioma, Haakon Lund, Marianne Lykke and Peter Ingwersen.

References

[1] W. B. Croft and J. Lafferty, editors. *Language Modeling for Information Retrieval*. Kluwer Academic Publishers, Dordrecht, 2003.

[2] S. Deb. TERI Integrated Digital Library Initiative. *The Electronic Library*, 24(3):366–379, 2006.

[3] C. Duddy. A Student Perspective on Accessing Academic Information in the Google Era. In *UKSG Annual Conference and Exhibition*, 2009.

[4] K. Järvelin and J. Kekäläinen. Cumulated Gain-based Evaluation of IR Techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[5] J. H. Lee. Analyses of Multiple Evidence Combination. *SIGIR Forum*, 31(SI):267–276, 1997.

[6] M. Lykke, B. Larsen, H. Lund, and P. Ingwersen. Developing a Test Collection for the Evaluation of Integrated Search. In *Proceedings of ECIR 2010*, number 5993 in LNCS, pages 627–630. Springer Verlag, 2010.

[7] D. Metzler and W. B. Croft. Combining the Language Model and Inference Network Approaches to Retrieval. *Information Processing & Management*, 40(5):735–750, 1997.

[8] D. Metzler and W. B. Croft. A Markov Random Field Model for Term Dependencies. In *Proceedings of SIGIR '05*, pages 472–479, New York, NY, 2005. ACM.

[9] V. Murdock and M. Lalmas. Workshop on Aggregated Search. *SIGIR Forum*, 42(2):80–83, 2008.

[10] N. O. Pors. Rationality and Educational Requirements: Exploring Students' Information Behaviour. In *Proceedings of IIiX '06*, pages 169–175, 2006.

[11] M. E. Renda and U. Straccia. Web Metasearch: Rank vs. Score-based Rank Aggregation Methods. In *Proceedings of SAC '03*, pages 841–846, New York, NY, USA, 2003. ACM.

[12] B. Schatz. Information Retrieval in Digital Libraries: Bringing Search to the Net. *Science*, 275(5298):327–334, 1997.

[13] G. Stone. Searching Life, the Universe and Everything? The Implementation of Summon at the University of Huddersfield. *Library Quarterly*, 20(1):24–52, 2010.

[14] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird. Learning Collection Fusion Strategies. In *Proceedings of SIGIR '95*, pages 172–179, New York, NY, 1995. ACM.

[15] C. Zhai and J. Lafferty. A Study of Smoothing Methods for Language Models Applied to Information Retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, 2004.