universitätbonn

Rheinische
Friedrich-Wilhelms-
Universität Bonn

A Living Lab Evaluation Environment for Academic Document Repositories

by

Narges Tavakolpoursaleh

Supervisors:
Prof. Dr. Sören Auer,
Dr. Philipp Schaer

A thesis submitted in partial fulfillment for the
degree of Master of Science

in the
Department of Computer Science

June 2016

Declaration of Authorship

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Date and Place:

Signature:

Rheinische Friedrich-Wilhelms-Universitat Bonn

Abstract

Department of Computer Science

Master of Science

by Narges Tavakolpoursaleh

Information Retrieval (IR) is known as an empirical science which needs experimental research to complement the theoretical results. In this regard, Living Labs for evaluation are recently taken into consideration in the field of Information Retrieval evaluation.

The Living Lab for IR evaluation represents a user-centric research methodology that researchers can use to evaluate their ranking algorithm in a live setting with real users in their natural task environments. Schuth et al. provided a living lab platform for IR researchers in order to evaluate their ranking algorithms on the one hand, and on the other hand, for sites with their search engine to benefit from R&D and most likely obtain a much-improved retrieval system. In practice, however, preparing the sites to take part in a Living Lab environment raises numerous challenges regarding the design, development, and implementation of the required architecture that supports the connection, data transformation between the site and living lab environment. The extension should also perform tasks like replacing site's ranking with the experimental ranking, following and recording the users' interaction with the search results in the experimental mode, etc.

We have implemented such an extension for SSOAR (Social Science Open Access Repository), which is a DSpace-based repository platform for the academic documents. This extension adds a new functionality to the DSpace platform and provides a living laboratory environment for SSOAR Discovery Module for experimenting on the ranking algorithms using SSOAR's components involves SSOAR's queries, documents, and users. We have empirically evaluated our approach by participating in the TREC 2016 OpenSearch track.

Acknowledgements

I would first like to thank my advisor, Professor Dr. Sören Auer from the Enterprise Information Systems department (EIS) at the Institute for Applied Computer Science at University of Bonn, for accepting me to do my thesis in his group, for his constructive criticisms, insightful comments, and hard questions.

Particularly, I would like to express my sincere gratitude to my advisor Dr. Philipp Schaer, Knowledge Discovery team leader at the Leibniz Institute for the Social Science (GESIS), for the continuous support of my master study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I also would like to thank Dr. Christoph Lange for the excellent guidance, encouragement, and advice he has provided throughout my thesis period.

A very special thanks goes out to my colleagues in GESIS specially, Jan Steinberg and Gerrit Hübbers who supported me greatly and were always willing to help me.

Last but not the least, I would like to thank my family for supporting and continuous encouragement throughout my years of study.

Contents

Declaration of Authorship i

Abstract ii

Acknowledgements iii

List of Figures vi

List of Tables viii

1 Introduction 1

2 State of the Art 7

2.1 System-based IR evaluation . . . 7

2.2 User-based IR evaluation . . . 10

2.3 User-based vs. System-based IR Evaluation . . . 11

2.4 Interactive IR evaluation . . . 12

2.5 Living Labs . . . 12

2.5.1 LL4IR Open Search 2016 . . . 15

2.5.2 LL4IR API . . . 15

2.5.3 LL4IR Dataset . . . 16

2.5.4 LL4IR Evaluation Phases . . . 19

2.6 Academic Document Repositories . . . 20

2.6.1 DSpace . . . 20

2.6.2 SSOAR . . . 24

3 Living Labs for Information Retrieval in Practice 26

3.1 Product Search with REGIO JATEK . . . 27

3.2 Ranking Approach . . . 30

3.3 Results . . . 32

3.4 Limitations . . . 35

4 Transforming DSpace into a Living Lab 37

4.1 Data structure in DSpace Repositories . . . 37

4.2 Search in DSpace Repositories 39
4.3 Implementation of Living Labs Environment for DSpace IR Evaluation 40
4.3.1 Collection Provider 42
4.3.2 Logging For Living Lab 53
4.3.3 Implementation of An Interleaver 54
4.3.4 Feedback Extractor 56
4.3.5 Data Uploader 58
4.4 Activation of Living Labs for DSpace Repository 59
4.5 Participation in TREC OpenSearch as Site 61
4.6 Signing Up for Living Lab OpenSearch as Researcher to Implement Experimental Ranking Algorithms 62
4.6.1 Experimental Ranking for SSOAR 62
4.6.2 Experimental Ranking for CiteSeerx 64

5 Evaluation **67**

6 Conclusions **72**
6.1 Limitations and Future Work 73

Appendix **81**

List of Figures

1.1 Information Retrieval Research area . . . 3
1.2 Living Labs Communication Channel . . . 4
1.3 The Location of Site Extension in Living Labs Communication Channel . . . 6

2.1 A Possible System Architecture for a Living Lab. From Azzopardi, L., & Balog, K. (2011, September). Towards a living lab for information retrieval research and development. In International Conference of the Cross-Language Evaluation Forum for European Languages (pp. 26-37). Springer Berlin Heidelberg. . . . 14
2.2 A Screenshot of SSOAR Search Result Page . . . 23

3.1 Inventory Changes During the First Round of the LL4IR 2015 Challenge, Schuth et al. [31] . . . 29
3.2 Distribution of impressions per topic for the first and official CLEF round (blue) and the second unofficial test round (red) . . . 33

4.1 DSpace data model . . . 38
4.2 Search Use Case in SSOAR . . . 39
4.3 General schema of Dspace extension for living labs . . . 41
4.4 Sequential diagram for adding Living Labs to the DSpace Repository 43
4.5 Comparison of Number of daily Simple Queries Across Three Years 44
4.6 Comparison of Number of daily Simple Queries over the Months from August 2013 to Jun 2015 . . . 45
4.7 300 of the most frequently searched queries (term search) on SSOAR during 23 month started from August 2013 until Jun 2015 . . . 45
4.8 100 of the most frequently searched (browsed) communities and collections on SSOAR during 22 month started from August 2013 until Jun 2015 . . . 48
4.9 Class diagram for DSpace living labs extension . . . 53
4.10 Living Labs in discovery performance Flowchart . . . 60
4.11 A Screenshot of TREC Open Search Dashboard Page, Representing the General Statistic of SSOAR Data Set Including the Number of Uploaded Queries, Documents, and the Impressions, `http://trec-open-search.org/` . . . 62
4.12 Counter Cumulative Distribution Function of Downloads . . . 63
4.13 Counter Cumulative Distribution Function of Views . . . 63

4.14 A Screenshot of CiteSeerx search Page 64
4.15 A Screenshot of CiteSeerx's candidate document 65
4.16 Our Stats for Site CiteSeerX 66

5.1 A Screenshot of TREC Open Search Dashboard Page `http://trec-open-search.org/` 69
5.2 Our Participation Details in TREC Open Search Dashboard Page `http://trec-open-search.org/` 70

List of Tables

3.1 The LL4IR API's Endpoints . . . 29

3.2 Solr field configuration for the available product metadata. Since different Configurations for round #1 and #2 were used we also report on the usage of the fields for the two evaluation rounds. . . . 32

3.3 Some basic statistics on the results. . . . 34

3.4 Outcome, wins, losses and ties from round #1 and #2. . . . 35

4.1 8 most searched terms in SSOAR simple search between 2013 and 2015 . . . 46

4.2 10 most searched community and collection in SSOAR between 2013 and 2015 . . . 49

4.3 Descriptions . . . 52

4.4 Example of interleaved result of experimental and site ranking . . . 56

5.1 Gesis Statistics for Site SSOAR . . . 70

Listings

3.1 Sample Candidate Products for a Head Query of REGIO Responded Form Living Labs API . . . 28
3.2 Sample output of the outcome documentation . . . 36
4.1 JSON contains SSOAR Head Queries For Living Labs . . . 49
4.2 SSOAR Document List for Query Term broeskamp . . . 51
4.3 Document Metadata for docid ssoar-d2 . . . 52
4.4 Recorded (logged) Search Event . . . 54
4.5 Feedback for ranking S1 . . . 59
4.6 Living Labs Configuration File . . . 61
4.7 A Candidate Document of CiteSeerX's Query Term Semantic Wikipedia . . . 65

Chapter 1

Introduction

The research in the field of information retrieval is conducted in different aspects of IR, such as development, indexing, evaluation, etc. Among this research, there is an enormous interest in the study of the development and evaluation of information retrieval tools. Since each innovation in retrieval technique should be evaluated and assessed, these two lines of research are tied together. Therefore, IR evaluation is an important and essential area in the field of developing and designing information retrieval tools.

Kelly (2009) has categorized the study of information retrieval - including developing and evaluating - into two major types. System-focus studies are the first category; systems are the center of interest in such studies. Experimental Cranfield models, which are the basic IR evaluation models, are an example of a system-based evaluation that focuses on measuring the relevance of documents in relation to the queries using assessors. In this model, researchers share common resources known as test collection, which contains documents, queries, and relevance assessments that determine which document is relevant to which query. This collection is used to quantify the effectiveness of an IR system according to evaluation measures. However, no user is involved and no search takes place. Most of Text Retrieval Conference (TREC) studies are examples of this type. The second category is user-focus studies. Regardless of the systems, these studies are conducted with an intense focus on human information seeking behavior. However, the results of these studies are helpful in improving interactive IR systems. Research of this type uses experimental approaches and interviews used in psychology to verify human behaviors. Both the complexity of studying the user and the system simultaneously, and the relevance of these two in the development and evaluation of online interactive models caused the formation of an area of different types of user-system studies, which Kelly placed in between the two main categories (system and user) with different inclinations towards one of these classes. In user-system studies, the user may be a central focus or play an assistant role while studying the system.

Kelly (2009) placed the study of Interactive Information Retrieval (IIR) in the middle of these two research studies, expressing that IIR perceives both the user as well as the system (Figure 1.1). TREC Interactive studies are located in this group, with the goal of improving the search process. IIR benefits from other studies such as, traditional information retrieval studies on one hand, and Psychology and Human Computer Interaction on the other.

By increasing the number of Internet tools and the interest in interactive search systems, the field of IIR research has grown fast. IIR deals with various research aspects, and among these it is, evaluation that is considered the central aspect. In this process, the evaluation of IIR can be assessed based on various criteria i.e. the user interface design of an IIR system, usability, and how people interact with the system besides the traditional IR evaluation. Nevertheless, the basic evaluation purpose of an IR system is to learn the effectiveness and performance of the searching process, which determines how well the investigated system retrieves and ranks relevant results.

Test collection-based evaluation originating from the Cranfield model is the basic IR evaluation model for the specification of relevance in IR systems. Test collection facilitates a central and shared resource, which has the benefit of being reused in different experiments and allows the comparison of multiple retrieval systems [28]. However, the process of offline judgments by expert assessors is expensive and time-consuming. Other laboratory evaluation approaches have been proposed to reduce these costs by enlisting the potential source of human assessors. Among those we can mention are the policy of coerced judgments (INEX and the enterprise track of TREC), crowd sourcing experiments, which employ the power of the crowds [28], or using laboratories study participants as assessors. These, however, are still not realistic and need to be abstracted and controlled.

Starting in 1990, online search engines suddenly started to become everyone's favorite information discovery tools. This size of search activity is indeed a very good opportunity for researchers to study the huge number of users worldwide, to understand the users better, and as the result improve the search engines themselves [7]. As long as online search engines continue to make progress, their quality must be evaluated.

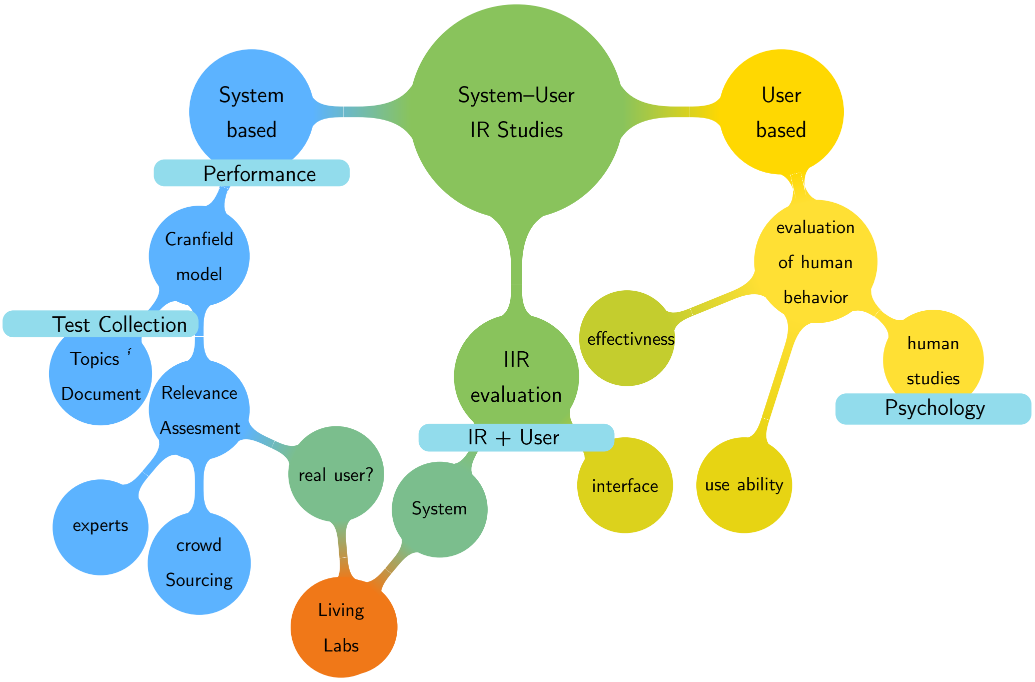


Figure 1.1: Information Retrieval Research area

This has made a new form of evaluation possible, in which the masses of real users can be considered as relevance assessors. Using a massive crowd of assessors would increase the reliability of relevance judgment. Considering the fact that the IIR systems attempt to fulfill the information needs of their users, this leads to the researchers' desire to make IR evaluation not only more achievable but also more realistic. It is not only the evaluation, but also the whole area of IR that can take advantage of studying real users in the real world. For researchers working at search engine companies, a huge amount of recorded user data is available in logs, which they have already started to navigate being evaluated in a real world environment (living labs) instead of the laboratory. A living lab transforms a real world application to an experiential environment to set up an experiential learning, where users of the application are implicitly involved in the experimentation when they are doing their natural task in their real world setting.

However, the benefit of using the real users is monopolized by industry, and due to some reasons such as competition and privacy, this knowledge is accessible exclusively to individual companies. Consequently, it has caused a knowledge gap to develop between academia and industry. This has the potential to be a threat to the academic, since research quality in this area and the search tools may not improve as fast as the growing IR system requires [7].

The evaluation and development of IR systems are becoming more complex and needs a wide scale of researchers from various fields in the form of a research community. A shared resource should be available and facilitate the research in this area with diverse focuses. “A living laboratory on the web that brings researchers together is needed to facilitate ISSS evaluation” (Kelly 2009). In this regard, the Living Lab for Information Retrieval (LL’13) workshop at CIKM brought academia and industry together for the first time. This workshop aimed to determine the possible steps required to create Living Labs for the IR evaluation platform. Such an evaluation platform covers search components, data collection tools, methods, and measures (Kelly 2009).

Living Labs for Information Retrieval evaluation (LL4IR) held at CLEF 2015 attempted to follow the progress of LL’13. This lab provided such an experimental platform to the IR research community for the first time [31]. It offers an evaluation environment and, a unique infrastructure, which is used for various types of sources.

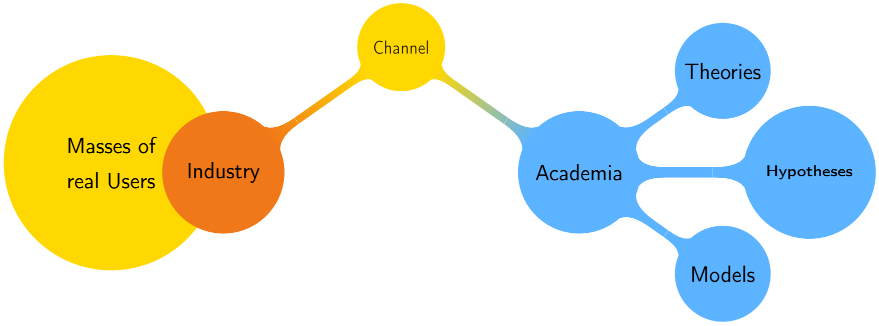


Figure 1.2: Living Labs Communication Channel

Problem and Specific Goal of this thesis

For IR researchers who participate in a living lab for IR, the use of search engines' components to verify their own rankings is a straightforward process, includes getting the set of queries and documents from the living lab's API , generating rankings for each query and sharing them with the search engines through the API. From the participating search engines' perspective to the contrary, it would not be a trivial work. It is required to implement an experimental environment in the existing application for replacing site's ranking with interleaved ranking, tracking the user interactions with experimental ranking and generating feedback. Consequently, the process of transforming an existing platform to an experimental environment for a search engine without hurting its principle raises some challenging issues.

The idea is therefore to provide an extension (See Figure 1.3) for an standard repository software to make it easier for search engine operators to participate in the challenge. DSpace or DuraSpace is one of those standard repository software, which operates many relevant sites. Social Science Open Access Repository (SSOAR) is one of the DSpace-based sites and will run Living lab for IR enhanced by this extension. SSOAR use the extension to provide the living lab environment in which researchers evaluate how SSOAR's end users use the system.

Researchers will be provided with prominent (head) queries from the log files. For each query, researchers calculate their own result ranking from the documents in the document collection of SSOAR. For each query, end users see a list of search results that interleaves the system's results, which in the case of SSOAR are provided by the Solr search server with the researchers' results. Then, the end users' actions (whether or not they click on a search result) are recorded, and serve as feedback to the developers of the system (here: the search/ranking algorithm and UI of DSpace) The feedback expected from the challenge is a comparison of how well the rankings of different researcher teams perform against the system's ranking (it does this by returning rankings that the end users find better/worse/equal than the system's ranking.)

The goal of the study is to address the following questions:

RQ1 What is the current state of the art in the living labs for IR evaluation

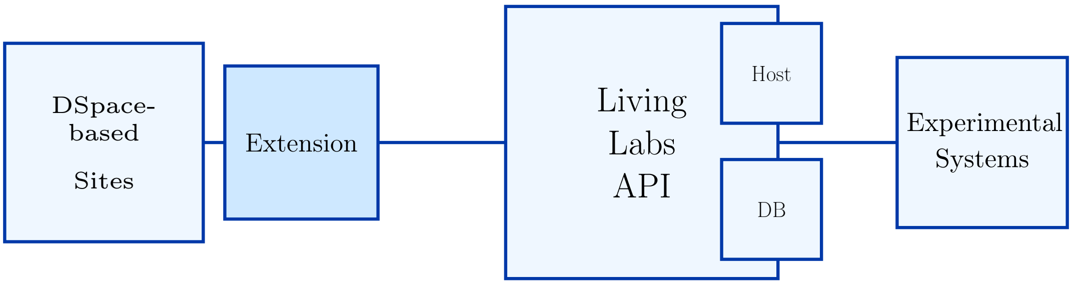


FIGURE 1.3: The Location of Site Extension in Living Labs Communication Channel

methodology and what an evaluation experiment using this methodology is like?

RQ2 What can we learn by participating in a Living Lab?

RQ3 What kind of infrastructure is required to support a living lab for a DSpace based repository?

RQ4 How can such an infrastructure be built to reach qualities such as adaptability, efficiency, maintainability, reliability, and reusability?

The next chapter (2) gives a brief overview of different information retrieval evaluation methods and relevant work, and introduces the idea and structure of living labs for information retrieval as well (RQ1). We address the RQ2 in Chapter 3 based on the publication [30] of the participation in LL4IR challenge where we investigated a living lab for IR environment in practice and reported on our experiences with the evaluation of an experimental IR system in a live setting. In Chapter 4 we describe our approach for transforming a use case (SSOAR) to an IR test environment (RQ3). Chapter 5 evaluates our transforming approach and answers RQ4 by considering the strengths and limitations of our method. Finally, in Chapter 6 we conclude the thesis and discuss the limitations and the future works.

Chapter 2

State of the Art

2.1 System-based IR evaluation

The focus in the system-based evaluation is on the pure system. There is no real user to search and evaluate the system in these kinds of evaluations. The Cranfield experiment (Cleverdon) series conducted since 1960 are known to be the first kinds of system-based IR evaluation. The aim of these experiment series was the improvement of the effectiveness of IR systems (Cleverdon, 1970). The components that were used in the Cranfield experiments include a test collection of documents, a set of test queries, and a set of relevance judgments (query relevance document).

A series of experiments known as TREC Tracks that started in 1992 are the next examples of system-based evaluation. Most of System-based IR evaluation studies also used the Cranfield components for the evaluation of an IR system [28].

Cranfield Model Components

Test Collection

Test collection together with evaluation measures as a quantifier is mainly used to predict the effectiveness score of retrieval systems. The researchers deploy this score to simulate a user's search process and compare the different IR systems or different configurations of a single system. A test collection contains a test set of queries, a set of documents, and a set of the query-relevant documents (Qrel or set of relevance judgment). The runs retrieved by the IR system are assessed according to this relevance judgment and then specifies the effectiveness of the system. The initial development of test collections and measures refers to the time period from 1950 to 1990. In 1953, Cleverdon and Throne innovated for the first time the method of test collection for IR evaluation. This method is also known as Cranfield collection. The objective of this innovation was to evaluate the effectiveness of a librarian's retrieval methods to find the documents requested by library patrons [28]. The system retrieval approach was to partition the document into two subsets: query relevant and non-relevant documents (Boolean Retrieval). Consequently, in the time of development, the evaluation measures were mostly based on the number of relevant items that are retrieved by the system which is the measure of recall. The concepts recall and precision in studying the performance of information retrieval systems are considered as in Equation 2.1.

$$
\begin{aligned}
recall &= \frac{\text{number of relevant documents retrieved}}{\text{number of relevant documents}} \\
precision &= \frac{\text{number of relevant documents retrieved}}{\text{number of retrieved documents}}
\end{aligned}
\tag{2.1}
$$

After the innovation of test collections for evaluation of a search system, researchers started to investigate, modify, and improve test collections and their related evaluation measures. The size of test collections available for academia was small, whereas commercial search companies had access to a large number of test collections. At this time the needs for larger text collections became apparent. The National Institute of Standards and Technology (NIST) together with U.S. Department of Defense sponsored in 1991 the Text Retrieval Conference (TREC) [1]. The TREC program is intended to provide the information retrieval community with the requirements for large-scale evaluation of text retrieval. The TREC also organizes a workshop each year. These workshop series are based on a set of goals, aiming to support researchers in these areas with large test collections and to gather the researchers at conferences and facilitate the publishing of research works.

By the development of web search engines in the late 1990s, researchers became

[1] http://trec.nist.gov/

interested in a ranked retrieval approach and development of ranking functions and weighting factors of relevancy. The ranked retrieval approach proposed by Luhn [22] in 1958 sorts the relevant items to a search query based on their relevance weights (term-frequency factor). The development of this approach created the need for new evaluation measures that balance the relevant rate of items to a given query.

Relevance Judgment

The Cranfield component that determines and assesses the system using the information needs and the documents is Relevance Judgment. This contains a set of documents for each test query that have been judged (by some experts) to be relevant to each query. This mapping process is specified by a number of the human resources and through the assimilation of reality that are known as domain experts. The number of documents from Cranfield collections are small, so the relevance judgment by domain expert is applicable. For large collections, a standard approach, which is called pooling (Sparck Jones and Van Rijsbergen 1975), is used to select a subset of documents (top-ranked documents) retrieved by an information retrieval system for evaluation [34]. This subset is being assessed by the relevant domain experts.

In 1956, Gull modified a form of test collection containing a set of queries and documents. He asked two independent groups to collect all relevant documents for each query. The result was two different sets of relevance judgment, which indicates that assessors of different groups may have completely different interpretations of queries (inconsistency in judgment by human). Saracevic [29] has studied the effects of such inconsistency on results of IR evaluations. In his research, Saracevic drew some general conclusions: more judges result in less overlap (small intersection) and the higher the expertise and the laboratory condition is, the higher the overlap between different judgments will be (up to 80%).

Measure Metrics

The most used (typically by TREC) metrics to measure how well the system performs, having the relevance judgment (mostly binary relevance), are derived from precision and recall parameters and are derivatives of the "relevance" concept (see Equation 2.1). Among these measurements are F-measure 2.2, Mean Average Precision (MAP), Mean Reciprocal Rank (MRR). The F-measure is used often for evaluating the performance of the IR and is a combination of the precision and recall. A high F-measure reveals that both precision and recall are high.

$$F - measure = \frac{2.precision.recall}{precision + recall} \tag{2.2}$$

MAP is the most standard measure in the TREC community. It is used for comparing search algorithms and has been shown to have a good stability [23]. Normalized Discounted Cumulative Gain (NDCG) [15] can be used to state the quality of the ranking system by grading each retrieved document. Contrary to other measures, NDCG considers not only the degree of relevance for each document but also a discount to weight the document's positions in a ranking. The measure of DCG for a ranking at the position is defined recursively. It has shown in Equation 2.3.

$$DCG[i] = \begin{cases} G[1] \quad if \quad i = 1, \\ \frac{DCG[i-1]+G[i]}{log_b i} \quad otherwise \end{cases} \tag{2.3}$$

2.2 User-based IR evaluation

User-based IR evaluation approaches were initiated in the 1970's and have grown rapidly since 1980. The users (information seekers) are the important parts of IR evaluation, and the usefulness of the modern IR system is measured based on information seekers' points of view. Contrary to the system based studies, user-based studies concentrate more on the users themselves. It focuses on the human and deals with psychology and cognitive science concerning human information needs, behaviors (HIB) and preferences as well as its influence on the information search behavior.

McKechnie et al. [24] analyzed 247 human information behavior articles published between 1993 to 2000. He observed that these articles are mainly centered on surveys, interviews, and experiments methods.

Study of Human Searching Behaviors

Information searching behavior is the study of searchers' interactions toward information systems. It comprises Human Computer Interaction (HCI), e.g. usability and usefulness studies as well as human judgment in assessing the relevance level of information retrieved by an IR system such as studies of the system effectiveness in retrieving relevant results [39].

System Usefulness is a study in the field of digital library evaluation and connects to the studies of information behavior and HCI. It determines the measure of user satisfaction with the system by observing the user's interaction with the system and analyzing the user's information searching behaviors - including user need formulation and expression (querying) and relevance assessment. This measure shows how successful the systems are in fulfilling the tasks that are issued by the system's user [36].

System Effectiveness refers to the measure of a system's ability to distinguish the items that are relevant to the user's information needs, and to separate them from irrelevant items. It deals with the studies of measuring the effectiveness of the system in performing its expected task as well as the user's information needs. Information Science (IS) in the field of HCI.

2.3 User-based vs. System-based IR Evaluation

Using system-based IR evaluation methods are popular and used widely since they are independent of the users, while they use domain experts' relevance judgment with some measures like precision and recall. However, such an evaluation lacks a variety of different conceptions of the system, which is expressed by users [8]. The user-centered evaluation, which concentrates on the user's behavior has eliminated this leakage, which nevertheless poses the problem of accessing users' behavior/-preference data. In other words, such an evaluation cannot be conducted without the users.

2.4 Interactive IR evaluation

Interactive information retrieval evaluation studies include both system and user aspects. The research in these areas has different evaluation axes, ranging from system focus such as the evaluation of information retrieval algorithms, which deal more with system performance to user focus, which deals with interactive retrieval and human-computer interaction issues such as system design and interface, usability, usefulness, ease of use, user satisfaction, etc. This research zone is comprised of of both HCI and IR studies for the evaluation of Interactive IR. Interactive IR evaluation research in its initial concept covers especially interaction and the user. However, the research in these areas began getting into either system or user-focus directions in the 1970's. Starting in the 1980's by focusing on human-centered perspectives in IR, these studies have become more oriented towards one of these dimensions [18]. Kelly (2013) reported a systematic review of 40 years Interactive IR evaluation studies from 1967 to 2006. This historical review analyses the works that have been done during these 40 years in the area of interactive IR and concludes the necessity of further works in this field.

2.5 Living Labs

The Living Labs' user-centered research concept was created by Jarmo Suominen and encompass research methodologies that involve real users of real life contexts for establishing experiments, developing, and evaluation. Living Labs for IR evaluation are proposed by the IR community as an alternative to the traditional offline evaluation environment with its standard test collections, which has been used by TREC, CLEF, and INEX.

Living lab for IR aims to provide a new IR evaluation environment with real users performing tasks in real applications. Such an evaluation paradigm provides an environment with a common data repository for researchers, which enables them to not only observe the real users' interactions but also make a realistic experiment and test their IR model on a live-real system.

HARD [2] and TREC Interactive tracks [10], amongst others, brought out the importance of involving real users in the IR evaluation.

The researcher's delight in having a living lab for tracking user information searching behavior raised and invigorated the IR evaluation workshop in the future of IR Evaluation on the SIGIR 2009 [17]. This was attended by CLEF, INEX, NTCIR and TREC and aimed to investigate and discuss the current state of IR evaluation, its weaknesses, and the needs of realistic approaches for the evaluation. Moreover, they attempted to address the different issues that appear relevant to realistic IR evaluation. For instance, the limited scale of observation studies with real users [27], the possibility of user model deployment for prediction of user behavior or using the living lab to record user interaction and explicit feedback (Human in loop report) [14] and the possible way to gather user data.

The deployment of online controlled experiments like A/B testing on different parts of live applications of big companies who have a lot of users like Google, Bing, Yahoo, Amazon, and Facebook have taught many lessons in the development and improvement of the different parts of their systems from user interface design and interactive techniques to recommendation system and the relevance or ranking algorithm. These lessons have shown that "shipping is not the goal, shipping something useful to the customer is the goal" [19].

Nevertheless, the benefit of using real data for the evaluation of IR is available only to major industrial research labs. "The basic idea of living labs for IR is that rather than individual research groups independently developing experimental search Infrastructures and gathering their own groups of experimental searchers for IR evaluations, a central and shared experimental environment is developed to facilitate the sharing of resources" [3].

An access to a shared resource of real data is desirable for IR researchers. This needs an experimental environment that connects researchers to data resources and consequently raises the need of an infrastructure design for such an environment. Azzopardi and Balog proposed the first architecture for such a living lab evaluation environment (see Figure 2.1).

The proposed system architecture connects four independent services consisting of an evaluation forum (A), which act as a bridge between available applications (B) and researchers (C) that are registered to use the real data of live applications. A living lab application provided by the live application provides the necessary usage data and users' interaction with the non-commercial variant of the live application (D). These are made available to researchers through an API in the living lab forum (A).

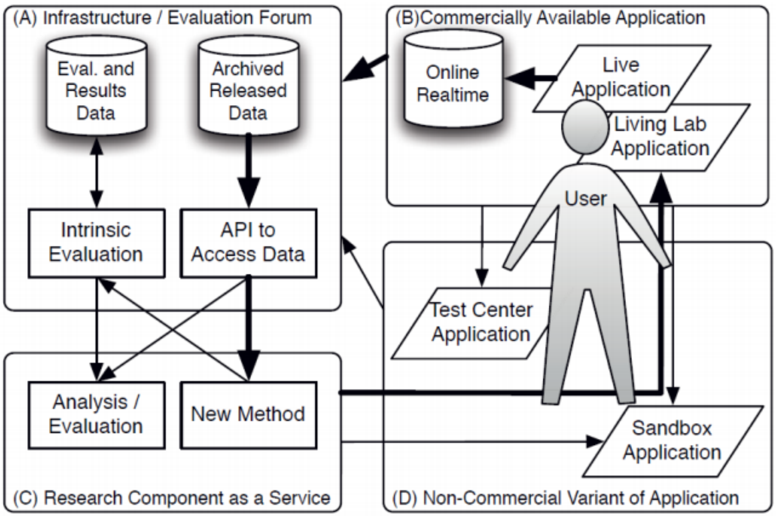


FIGURE 2.1: A Possible System Architecture for a Living Lab. From Azzopardi, L., & Balog, K. (2011, September). Towards a living lab for information retrieval research and development. In International Conference of the Cross-Language Evaluation Forum for European Languages (pp. 26-37). Springer Berlin Heidelberg.

The living lab for IR focused primarily on the commercial sites and was proposed to open up this kind of sites to academic research so that researchers can take advantage of site's users and evaluate their system in the real applications. However, there are also non-commercial sites like academic institutional repositories or digital libraries which have numerous users. Accordingly, these sites are also started to be considered by living labs as potential sources.

Recent years have seen several numbers of workshops on Living Labs for Information Retrieval Evaluation. Among the last workshops are Balog et al. [4, 5] that aimed to provide a living labs evaluation paradigm accessible to the wider IR search community.

The Living Labs for Information Retrieval Evaluation (LL'13) workshop at CIKM in 2013 was held in the USA as the first attempt to bring people together to discuss challenges and to propose practical next steps. It tried to provide a central experimental environment for sharing resources. To continue the directions of LL'13, CLEF 2015 held the first Living Labs for Information Retrieval Evaluation Challenge Lab (LL4IR). The LL4IR lab contributes to both the understanding of online evaluation as well as an understanding of the generalization of retrieval techniques across different use-cases.

The last workshop [2] is an attempt to investigate the difference of rankings when using historical clicks, online experiments, and manual relevance assessments. They gave lab participants a limited access to data consisting of head queries, documents, and a list of candidate documents for each query. The participants then got an opportunity to test their IR systems on real live systems with real live users and improve their IR model using users' feedback. However, participants do not have full control over the results shown to the user. The participant's rankings are interleaved with the ranking of the production system. In addition, the feedback is not given immediately and cannot be used in the given search session.

2.5.1 LL4IR Open Search 2016

The main objective of this lab is to provide a platform for participants to experience a realistic evaluation of their ranking systems in real applications with real users. The Lab facilitates a connection between commercial sites and researchers. All data exchange is done through the lab. Researchers who participants in the lab can access the dataset of the commercial site and use this data as the input of their experimental system. The lab allows the conduction of the experimental system outcome to the commercial site where the experimental systems are tested in a live environment with real users. The assessment then is done through user feedback. The lab compares the feedback of different experimental systems and announces the result.

2.5.2 LL4IR API

A Living Lab for IR requires a means of transferring information (queries, ranking and feedback) between end users of the site and experimental search systems. In LL4IR, all communications between sites and experimental systems are established via an API (Figure 1.3).

[2] http://living-labs.net/clef-lab/

The experimental system produces runs by taking the data sources (queries and documents) from the API and applying its own ranking algorithm. The produced experimental runs are then uploaded via the API by participants who experiment their algorithms. The site uploads the runs and interleaves each run with its production system run. The items in each interleaved run are then being clicked by the site's users, the clicks are uploaded via the API, and finally each run (ranking systems) being evaluated by a pairwise comparison approach.

2.5.3 LL4IR Dataset

Two commercial sites participated in this lab (product and web search). These two applications provided a set of frequent queries (head queries) with a set of their system response to each query. The product search data set contains a set of (product) entities. Each of these entities is characterized by set of metadata-presenting product attributes.

While product search submitted actual terms for query and data sets, web search sites stated those with feature vectors. The data set in LL4IR consisted of head queries, a set of documents for each query, and metadata for each document. Click-through rates are also added as metadata for some of the documents.

Head queries for a search engine are the queries that are frequently issued. These queries are used in the lab through their large scale of their click-through data. This means that the ranking algorithm has no chance to learn tail queries (queries which are made rarely). The lab organizer listed three reasons why they decided to use only head queries. Firstly, new ranking is guaranteed to be evaluated via frequent query occurrence. Secondly, the volume level of these queries stay fixed, and thirdly, an appropriate amount of usage data and historical clicks for these queries are available [31].

Ranking is a list of the documents that correspond to a query (user information need). This list is sorted by some criteria known as relevancy. The measure of a document's relevancy to a query determines the position of the document in the ranking list. The most relevant documents to a query are positioned on the top of the ranking list. Ranking functions use different methods to measure the relevancy of a document.

Feedback is a ranking that has been shown to a user and whose item may have been clicked on by the user in such a way that a set of attributes is assigned to each item. The “click” attribute has the value of “true” if a user has clicked on it, or “false” if he has not. The “team” attribute gets the value “participant” if it belongs to the experimental system’s run and“site” if it belongs to the site’s run.

Evaluation Metric

Users submit clicks and evaluate the rankings while they are not aware of the study. This leads the evaluation to be more realistic. The experimental system is evaluated online in such a way that users are shown experimental system rankings (treatment) interleaved with a baseline system (control) ranking. Each item in ranking is labeled by its owner (treatment or control). Users’ clicks on an item in the ranked list are submitted as positive scores for its rankings. A CTR indicates a fraction of clicks on a specific entity out of the total number of clicks on any entity and time to click for an entity in order to determine the time difference between showing the query result to the user and when they clicked on the item. A metric of the click-through rate and the time to click is used to compare the two systems and to confirm which system performs better.

Click-through rate (CTR) is computed using initiative user clicks on the products of live applications. The system can predict user behaviors by ranking existing products based on their CTR. However, new products do not have CTR. For a product, CTR is counted normally as the percentage of the number of clicks on the product divided by the number of times it has shown to the user.

$$\frac{\#clicks}{\#impressions} * 100$$

CTR for ranking can be calculated under other definitions as well, e.g., as the average of clicks per SERP (Search Result Page) or as the fraction of pages which receive any click [32].

A/B Testing vs. Interleaving

In online evaluation, users' implicit feedback (clicks on items) demonstrates the user preference of items retrieved from the search engine. Controlled experiments are a methodology that uses the web and users' implicit feedback to evaluate ideas. In controlled experiments, logs are analyzed for the identification of behavior patterns. Kohavi has shown in his studies that the employment of controlled experiments has a significant effect on return-on-investigate (ROI) [20].Researchers use controlled experiment such as A/B testing and interleaving to evaluate different methods and theories such as ranking algorithms and recommendation systems. A/B tests and interleaving are both well known methods for the online evaluation of information retrieval systems.

A/B testing is known as the gold standard of online evaluation methods for information retrieval systems. The Idea of A/B testing or split testing is to redirect random samples of users to a variant system (two or more systems) where metrics of interest are collected. Then the metrics on experimental groups are analyzed and the performance of the systems is compared to discover which idea is better than the other. This experiment must run statistical tests to confirm that the differences are not causality. Techniques of machine learning or data mining are often used to discover significant differences in experimental groups. This approach is the simplest controlled experiment and is widely used because it is easy to deploy and understand [21].

Interleaving or combined ranking is another popular online evaluation method proposed by Joachims et al. [16], which mixes two different ranking systems so that the first n items of combined ranking contains top K_a and K_b items of the ranking system A and B, and so that the numbers of items selected from A and B are either equal or have a difference of one; $|K_a - K_b| \leq 1$. This means that the combined ranking has almost equal numbers of items of both systems. This experiment has the risk of system usability reduction. However, it has the advantage that compared to A/B testing, a user's experience with the system is not hurt by just being presented with the worse ranking. It requires less traffic in comparison to A/B testing to get significant metrics and consequently is faster than A/B testing. However, it is mainly useful for evaluations such as ranking algorithms.

Team-Draft Interleaving (TDI) [26] and Balanced Interleaving (BI) [16] are two popular interleaving algorithms, which are most commonly used. In TDI, a combined list of ranking A and B is shown to the users. Each item in the result list belongs to ranking system A or B or none of them. If the amount of user clicks on system, items of A are more than the items of system B, system A is considered to be more efficient. The living lab uses TDI to evaluate experimental ranking systems against the search engine system. The lab organizers prefer the Interleaving method by because of the high sensitivity (two orders of magnitude more) of the method compared to the AB testing [32]. Another reason that makes this method preferable is that the number of times that two rankings should be shown to the users to make two systems comparable is one out of two; with just one impression two systems can be compared. Finally, using this method reduces the risk of showing bad rankings to the users.

2.5.4 LL4IR Evaluation Phases

The evaluation campaign is performed in two phases: training and testing. During the training phase, participants can update their ranking. The feedback is made available as soon as the sites releases it through the API. Participants are able to upload their ranking before - or only ones during - the test phase. During the test phase, participants cannot update their rankings. The outcomes - without individual feedback - are announced at the end of the testing phase. The outcome of the testing phase determines the final performance of the experimental rankings against the baseline. Outcomes are used as the primary metrics for comparing rankings.

Outcomes are used as the primary metrics for comparison of rankings [31]. Outcomes are calculated as follows:

$$\frac{\#Wins}{\#Wins + \#Losses}$$

Wins belong to the participant (experimental system) when the items that are assigned to him are clicked more than items that are assigned to site. The participant loses when he gets fewer clicks, and ties occurs when the site and the participant receive the same amount of clicks on their items. Outcomes are announced at the end of the testing phase and compare each experimental system with the site baseline.

2.6 Academic Document Repositories

2.6.1 DSpace

DSpace as an open source platform is developed by Hewlett-Packard Company (HP) and MIT Libraries for the preservation of digital library for the long term and is released since November 2002.

DSpace[3] is community-based and has been made available free to be used for digital repositories.

DSpace has been intended to offer a repository that facilitates the deposit and discovery. It preserve a professional repository to collect, preserve, index, distribute and in general manage research materials and scholarly publications. DSpace uses the Handle System (where resource address is identified by a unique handle assigned by a common registration service) from the Corporation for National Research Initiatives (CNRI) for long-term preservation.

Organizations in education, research, health-care, government and business uses DSpace to easily share and preserve their digital collections that may include.

It employs qualified standard metadata, such as title, abstract, language, keywords to characterize its items. This metadata is located in the item record in DSpace and is indexed for browsing and searching the system within a collection or communities which may include scientific publications, representations of books, theses, 3-D digital scans of objects, photographs, film, video, research data sets and many other forms of content [9].

DSpace is written in Java and uses relational database PostgreSQL. As web server and Java server engine DSpace uses Apache Tomcat. It uses also another open source software and libraries like RDF toolkit and is running on UNIX.

[3] http://www.dspace.org

DSpace architecture has three-layer: storage, business, and application. The storage layer is handled by PostgreSQL. The business layer copes with the whole workflow, content management, search and browse modules. Last layer, the application, includes the DSpace interfaces.

DSpace has several interfaces. The end user interface, XMLUI, supports search and retrieval of archived items by browsing or searching the metadata[33].

More than 1000 commercial and academic institution and organizations, national libraries and research center worldwide are using DSpace for institutional repositories. Among these organizations is Society for Critical Aesthetics, University of Kassel, Lufthansa Aviation Group, and Leibniz Institute for Social Sciences (GESIS) where this work has been done.

Solr

Solr [4] is an open source enterprise search platform and builds on Apache Lucene search technology. Lucene is a Java-based full-text information retrieval library and is managed by the Apache Software Foundation [5]. An IndexWriter and IndexSearcher are two components in Lucene library that are used to index and search the documents. It implements also spellchecking, hit highlighting, and advanced analysis/tokenization. Lucene scores the documents according to their Term Frequency–Inverse Document Frequency (TF–IDF) and lengthNorm:

$$Scoring : TF * IDF * LenghtNorm$$

where:

- **TF** Term Frequency in the document: A document's score is higher if it contains more instances of the search term.
- **IDF** Inverse Document Frequency: rarer terms and words in the corpus are generally more important and score higher.
- **lengthNorm** Matches in shorter fields score higher.

[4] http://lucene.apache.org/solr/

[5] http://www.apache.org

Solr, the fastest growing Lucene subproject, is optimized to search complex queries in large volumes of natural-language text and ranks the results based on their relevance to the query [13]. Solr's relevancy scores are based on the Solr Similarity classes, which define how to determine and measure the relevancy score. The Solr's default Similarity class use two-pass model for the calculation of the similarity, including a Boolean model to filter out the none-relevance documents, and a Vector Space model for scoring the relevance documents, based on the cosine Similarities between the query and the documents vectors. The Equation 2.4 demonstrates the calculation of relevancy.

$$Score(q,d) = \sum_{t\,in\,q} \Big(TF(t\,in\,d)\ \cdot IDF(t)^2\ \cdot\ t.getboost().\,norm(t,d)\Big).\,coord(q,d).\,queryNorm(q) \tag{2.4}$$

where:

$$TF(t\,in\,d) =\ \text{the number of term occurrences in document}^{1/2}$$

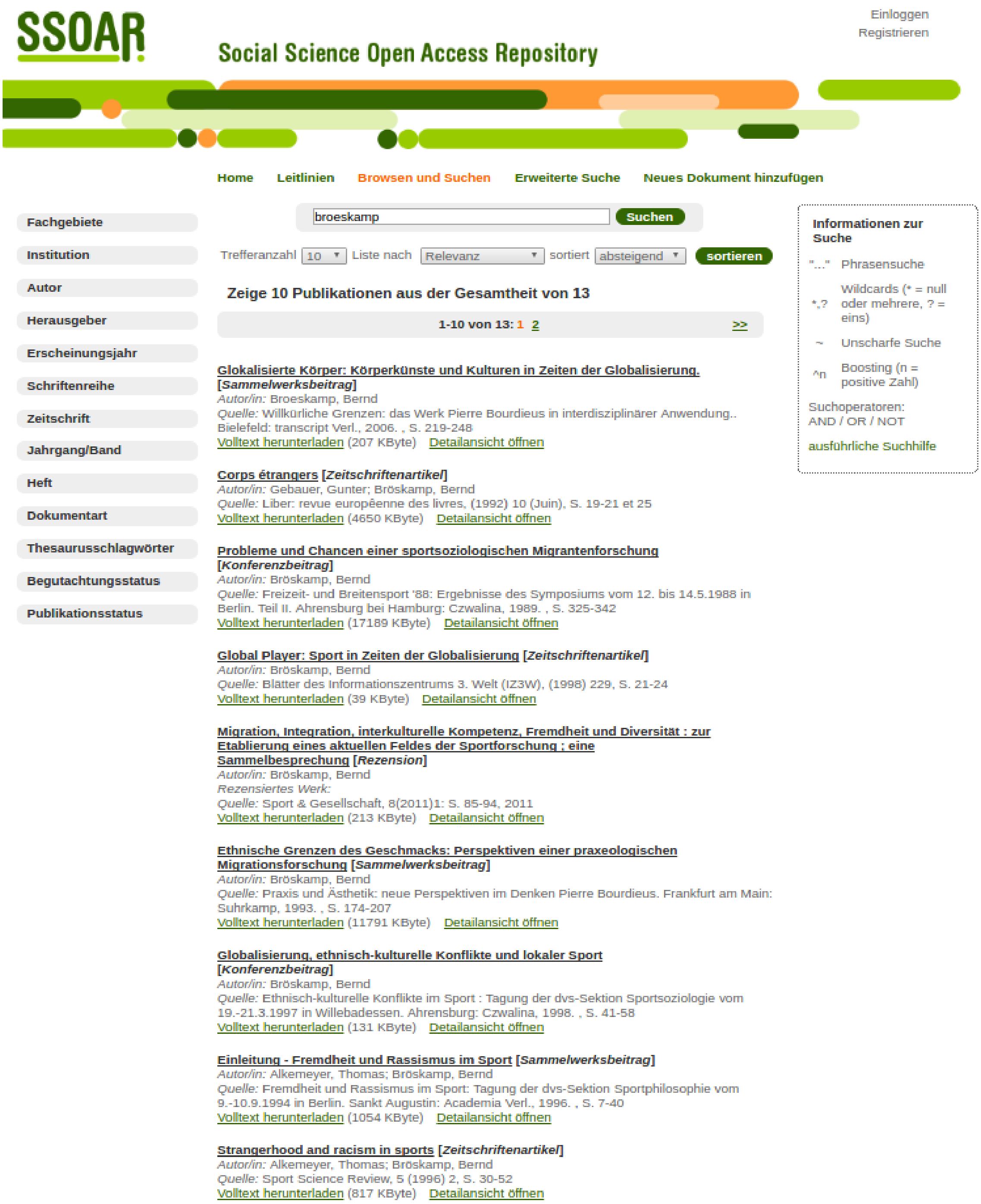
$$IDF(t\,in\,d) =\ 1 + log(\frac{numDocs}{docFreq+1})$$

$$norm(t,d) =\ d.getBoost() \cdot lengthNorm(f) \cdot f.getBoost()$$

$$coord(q,d) =\ \frac{\text{the numer of terms in document from query}}{\text{the number of terms in query}}$$

$$queryNorm(q) =\ 1/\Big(q.getBoost()^2 \cdot \sum_{t\,in\,q}\big(IDF(t)\cdot t.getBoost()\big)^2\Big)^{1/2}$$

Solr is used as the search and browses infrastructure in the DSpace -based institutional repositories. DSpace implements the querying of Solr through its Discovery module whose classes are located in the "org.dspace.discovery" package of DSpace source code. In DSpace queries are passed to Solr and the returned results are displayed within the DSpace UI.



Zeige 10 Publikationen aus der Gesamtheit von 13

1-10 von 13: 1 2 >>

Glokalisierte Körper: Körperkünste und Kulturen in Zeiten der Globalisierung. [*Sammelwerksbeitrag*]
Autor/in: Broeskamp, Bernd
Quelle: Willkürliche Grenzen: das Werk Pierre Bourdieus in interdisziplinärer Anwendung.. Bielefeld: transcript Verl., 2006. , S. 219-248
Volltext herunterladen (207 KByte) Detailansicht öffnen

Corps étrangers [*Zeitschriftenartikel*]
Autor/in: Gebauer, Gunter; Bröskamp, Bernd
Quelle: Liber: revue européenne des livres, (1992) 10 (Juin), S. 19-21 et 25
Volltext herunterladen (4650 KByte) Detailansicht öffnen

Probleme und Chancen einer sportsoziologischen Migrantenforschung [*Konferenzbeitrag*]
Autor/in: Bröskamp, Bernd
Quelle: Freizeit- und Breitensport '88: Ergebnisse des Symposiums vom 12. bis 14.5.1988 in Berlin. Teil II. Ahrensburg bei Hamburg: Czwalina, 1989. , S. 325-342
Volltext herunterladen (17189 KByte) Detailansicht öffnen

Global Player: Sport in Zeiten der Globalisierung [*Zeitschriftenartikel*]
Autor/in: Bröskamp, Bernd
Quelle: Blätter des Informationszentrums 3. Welt (IZ3W), (1998) 229, S. 21-24
Volltext herunterladen (39 KByte) Detailansicht öffnen

Migration, Integration, interkulturelle Kompetenz, Fremdheit und Diversität : zur Etablierung eines aktuellen Feldes der Sportforschung ; eine Sammelbesprechung [*Rezension*]
Autor/in: Bröskamp, Bernd
Rezensiertes Werk:
Quelle: Sport & Gesellschaft, 8(2011)1: S. 85-94, 2011
Volltext herunterladen (213 KByte) Detailansicht öffnen

Ethnische Grenzen des Geschmacks: Perspektiven einer praxeologischen Migrationsforschung [*Sammelwerksbeitrag*]
Autor/in: Bröskamp, Bernd
Quelle: Praxis und Ästhetik: neue Perspektiven im Denken Pierre Bourdieus. Frankfurt am Main: Suhrkamp, 1993. , S. 174-207
Volltext herunterladen (11791 KByte) Detailansicht öffnen

Globalisierung, ethnisch-kulturelle Konflikte und lokaler Sport [*Konferenzbeitrag*]
Autor/in: Bröskamp, Bernd
Quelle: Ethnisch-kulturelle Konflikte im Sport : Tagung der dvs-Sektion Sportsoziologie vom 19.-21.3.1997 in Willebadessen. Ahrensburg: Czwalina, 1998. , S. 41-58
Volltext herunterladen (131 KByte) Detailansicht öffnen

Einleitung - Fremdheit und Rassismus im Sport [*Sammelwerksbeitrag*]
Autor/in: Alkemeyer, Thomas; Bröskamp, Bernd
Quelle: Fremdheit und Rassismus im Sport: Tagung der dvs-Sektion Sportphilosophie vom 9.-10.9.1994 in Berlin. Sankt Augustin: Academia Verl., 1996. , S. 7-40
Volltext herunterladen (1054 KByte) Detailansicht öffnen

Strangerhood and racism in sports [*Zeitschriftenartikel*]
Autor/in: Alkemeyer, Thomas; Bröskamp, Bernd
Quelle: Sport Science Review, 5 (1996) 2, S. 30-52
Volltext herunterladen (817 KByte) Detailansicht öffnen

Figure 2.2: A Screenshot of SSOAR Search Result Page

2.6.2 SSOAR

GESIS as one of the information infrastructure institutes of the Leibniz-Association [6] has launched "Social Science Open Access Repository" (SSOAR) [7], which is one of the four big disciplinary Open Access Repositories in Germany. The SSOAR is a DSpace based platform and a freely accessible full-text server with the Open Access strategy. It collects around 36,000 electronic full texts from the domain of the Social, Political, and Management Sciences and Humanities and makes these texts available according to the Berlin Declaration on Open Access to Scientific Knowledge. SSOAR offers the social scientists, scientific associations, and publishers the opportunity to self-archive their publications, to publish their works on the web.

Authors and publishers enable free access to scientific information of social science research by archiving their works in SSOAR. It aims to be a central archive for the secondary publisher of quality-controlled literature for social science research in Germany and strives to be on"Green Road to Open Access" (OA).

SSOAR publishes pre-prints, post-prints, and publishers' versions of published works as well as works for the first time. SSOAR archives and makes different types of documents available in open access as journal articles and contributions to edited volumes, contributions to working papers, monographs, theses, and dissertations.

According to Alexa [8] 41% of SSOAR visitors comes through google.de, 6.8% through google.com, 3.6% through gesis.org and 3.2% through bing. The average number of SSOAR users is accounted as 14,000 users with 25,000 downloads per month in 2013 [11]. The E-Tracker displayed 647,467 number of downloads in 2015. In the same year the number of domain uses is 422,078 and the total number of visitors is recorded to be 461,011 by the E-Tracker.

To enhance the publications' visibility, find-ability, and getting higher ranks in search engines like Google, the publications in SSOAR are being cataloged and indexed using controlled social science vocabulary. By self-archiving the users are able to add controlled keywords from the thesaurus of the social sciences, assign keywords, and add an abstract. The SSOAR team classifies the applications using

[6] http://www.gesis.org/
[7] http://www.ssoar.info
[8] http://www.alexa.com/

the Social Science Classification [9]. To increase the visibility of the full texts on the Internet, documents' rich metadata are made available to Open Archives Initiative (OAI) service providers. All metadata is made available under a Creative Commons Zero (CC0) license and can be harvested via the standardized web interface and used freely without any limitation [1].

SSOAR is based on DSpace and uses DSpace (Solr / Discovery) Retrieval Systems.

Participating in a Living Lab provides a research and development (R&D) environment for study, evaluation, and improvement of SSOAR search engine. It also let other researchers in the field of information retrieval to use the component of SSOAR for training and evaluation of their retrieval algorithms in a live environment using SSOAR users.

It can give SSOAR team the opportunity to experiment the retrieval system of SSOAR in a live environment with experimental ranking systems and to gain much-improved ranking approaches for SSOAR.

Figure 2.2 shows the schema of SSOAR search result page.

[9] http://www.gesis.org/fileadmin/upload/dienstleistung/tools_standards/klass.pdf

Chapter 3

Living Labs for Information Retrieval in Practice

This chapter is based on the publication "Historical Clicks for Product Search:GESIS at CLEF LL4IR 2015" [30]. The explanation and the implementation of the ranking algorithm were my contributions in this paper.

In 2015, the Living Labs for Information Retrieval initiative (LL4IR) for the first time organized a lab in the form of a challenge at the CLEF conference series [31]. This lab can be seen as a pilot evaluation lab or as stated by the organizers: "the first round". Our participation goal was to get a first-hand experience with the lab's API and living lab evaluation methodology. Furthermore, to obtain the feedback of our ranking approach impressed by real users and eventually to perceive and realize if we can improve our approach by observing the user feedback and whether we can conclude an assertion to reason the performance of our approach having the living labs evaluation outcome. We investigate the impact of living labs on IR Evaluation and prerequisites for using SSOAR as our use-case (RQ2).

The first edition of LL4IR started with two specific use-cases; A product search of REGIO Jatek [1] which is an e-commerce site and web search of Seznam which is Czech Republic commercial web search engine [2] cooperated in the LL4IR challenge by sharing a data collection and a small part of their traffic with the participating researchers of the Living Labs to evaluating their runs.

[1] http://www.regiojatek.hu

[2] https://www.seznam.cz/

We took part in the product search task with the aim of learning about the procedures and the systems used. We designed a simple system based on the Solr search engine and historical click data.

3.1 Product Search with REGIO JATEK

One of the two use-cases were the product search On the eCommerce site REGIO JATEK (toy retailer) which is placed in first top five online shops in Hungary [31]. This use-case introduced a collection of 100 queries, each consists of approximately 1.18 terms with a permanent search volume. For each query, a set of product (candidate products) consisting about 50 products is given. Candidate products are described as all the products of REGIO that contain at least a query terms in any of its description fields. Each product is introduced by some meta information, among these were: (1) Historical click information for queries, (2) collection statistics, (3) product taxonomy, (4) product photos, (5) date/time when the product first became available, (6) historical click information for products, and (7) sales margins. Each product was also assigned to a set of semantic annotation like the brand, gender and age recommendation, toy characters and queries that return the given product. Table 3.2 shows most of the products' attributes represented for product search use-case. Given this information collection, the product search task is to rank the determined products set for each query. This task had some challenges regarding the instability of products' prices and availability status. This made the ranking task in testing phase difficult since the participants are not able to obtain the updated data, including possible products' actual prices or inventory status. Such a problem caused the status in which an experimental ranking (run) contained products which had not been available during the testing phase, and as a consequence, they had to be eliminated from the ranking before offering to the users. It was also being stated that newly arrived products were included in ranking returned for a query in the production system, whereas they weren't available on candidate products and the participants were not able to include them in their ranking. Figure 3.1 shows the REGIO online-shop inventory changes, regarding the products which were attended in the challenge, during the first round of the challenge. In the graph, the number of newly arrived products, the products that become available after being unavailable for a while and the products that become unavailable, over a period of two weeks are illustrated sequentially in green, blue and red. As the figure demonstrates, the majority of inventory changes caused by the daily new arrival and become unavailable products. It is shown that the number of the products which become again available is significantly smaller than the number of the products which are eliminated from an inventory list daily. Declared by REGIO, the average of the unavailability ratio of all submitted rankings was 44%. These inventory changes by REGIO are mentioned as a phenomenon of a the product search system which is an undesirable (in the sense of living lab) but the inevitable natural behavior of such a use-case. This condition reduced the participant's probability of winning from 0.5 to 0.28.

```
{
    "doclist": [
        {
            "docid": "R-d1244",
            "site_id": "R",
            "title": "M\u0171anyag sportpoh\u00e1r Monster High "
        },
        {
            "docid": "R-d1245",
            "site_id": "R",
            "title": "Monster High Scaris Parav\u00e1ros Vadmacska
   baba kieg\u00e9sz\u00edt\u0151kkel"
        },
        {
            "docid": "R-d1242",
            "site_id": "R",
            "title": "TOLL MONSTER HIGH 4 AZ 1-BEN PTS"
        },
        ....

    "qid": "R-q1"
}
```

LISTING 3.1: Sample Candidate Products for a Head Query of REGIO Responded Form Living Labs API

To generate our runs we decided to re-use the available historical click data for products and a keyword-based relevance score (TF-IDF weighting and the vector space modeling) derived from a Solr indexation of the available product metadata. Living Labs API provided The indexation configuration of the given fields is listed in table 3.2. Living Labs organizer implements a REST-full service API As the participant, we could connect to the API using an individual key which we obtained

Table 3.1: The LL4IR API's Endpoints

URL	Description
/api/participant/query/KEY	Get the query set, including query ids and query string
/api/participant/doclist/KEY/(qid)	Get the candidate products ids (docid) for query (qid)
/api/participant/doc/KEY/(docid)	Get the product's details of product (docid)
/api/participant/run/KEY/(qid)	Commit a run for query (qid)
/api/participant/feedback/KEY/(qid)/(runid)	Get feedback for run (runid) of query (qid)
/api/participant/outcome/KEY/(qid)	Get all the users interactions the runs of query (qid)

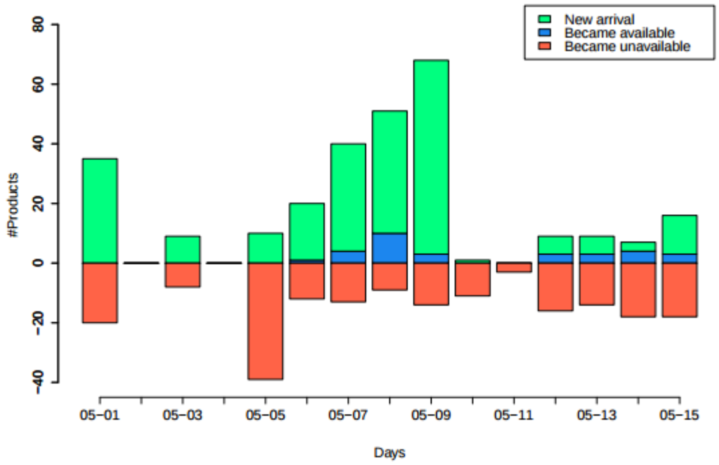


Figure 3.1: Inventory Changes During the First Round of the LL4IR 2015 Challenge, Schuth et al. [31]

from the registration. Through our key, we were able to communicate with the API and obtain and retrieve the data, including the head queries, candidate products for each query with their metadata as well as committed the runs. The format of all transforming data is JSON. The URL to derive and deliver the living lab data are listed in the table 3.1. To make the communication to the endpoints URL easy we could implement a client to talk to the API. However the lab organizer provided sample clients for both site and participants to talk to the living labs API [31]. The clients are open source written in Python and made available through Bitbucket [3]. The Listing 3.1 shows the response of our request(/api/participant/doclist/KEY/R-q1) through the API for obtaining the product candidate for the query with the ID of R-q1. The returned JSON files from the API may contain some encoded character as we can see in some titles of 3.1 that the encoded character "\u0171" is replaced for the Unicode Hungarian character Ű.

[3] https://bitbucket.org/living-labs/ll-api

3.2 Ranking Approach

We used a Solr search server (version 5.0.0) and indexed all available metadata provided by the API for every document related to the REGIO given queries. For each product, we additionally stored and indexed the corresponding query number in order to retrieve all available candidate products. The retrieved results were made up of the candidate products that were ranked according to the Solr score based on the query string. Furthermore, the available historical click rates were added as a weighting factor in the final ranking. Since not all the products were adorned with the rate of click through (As may the products are newly arrived in the system), we couldn't take advantage of these for ranking of all the products. We observed that some query candidate products contained no term (neither in its product name nor in its description fields) which can be matched with The query string and therefore we were not able to retrieve every query related document on a mere query string-based search. To fix this issue we added the query number to the query itself as a Boolean query part and used Solr query Syntax and the default Solr QParserPlugin. This syntax allows the use of Boolean operators and of different boosting factors which were both used in the query formulation:

$$qid : query[id]^\wedge 0.0001 \;\; OR \;\; (qid : query[id]^\wedge 0.0001 \;\; AND \;\; query[str])$$

Using this query string we got a Solr-ranked list of candidate products to the corresponding query which was then re-ranked using the historical click rates as outlined in algorithm 1. Basically, it's a linear combination of a boosted search on the document id (field name docid) and the vector space-based relevance score of the query string. As a result, a product in the produced ranking got a higher ranking score if the query terms appear more often (compare to the products in lower ranking) in the product metadata, rare in the whole collection and was clicked on a lot in the past. This is typical "the rich are getting richer" approach where formally successful products are more likely to be once again ranked high in the result list. The approach was inspired by a presentation by Bialecki [6].

Solr Configuration

As stated before, we used a Solr search server to obtain our runs. We tried to construct a simple system using the original Solr configuration and imported the

REGIO products using the originally provided Solr schema and the fields from the table 3.2. We did not include any language specific configurations for stemmers or stop word lists since the Hungarian Solr stemmer returned the same results as the generic stemmer. We used the following standard components for “text general” fields like description:

- StandardTokenizerFactory: A general purpose tokenizer, which divides a string into tokens with various types.
- StopFilterFactory: Words from the Solr included stopword lists are discarded.
- LowerCaseFilterFactory: All letters are indexed and queried as lowercase.

Algorithm 1 Re-ranking algorithm merging the Solr ranking score and the Historical click rates.

```
Data: Runs of production system correspond to the queries to products of REGIO
      JATEK site
Result: Runs of our experimental system according to the document's fields and
        click-through rate
for query in queries do
    run = get_doclist(query)
    ctr = get_ctr(query)
    for doc in run do
        doc_detail = get_docDetail(doc)
        BuildSolrIndex(doc_detail,qid)
    end
    myQuery = (docid$_1$ ^ctr$_1$ OR docid$_2$ ^ctr$_2$ OR ...OR docid$_n$ ^ctr$_n$) OR
    (qid^0.0001 AND query['str'])
    myRun = solr.search(myQuery)
    update_runs(key, myRun , feedback)
end
```

The indexation configuration of the given fields is listed in table 3.2. To generate our runs in the first round, we utilized a very limited set of fields (only the title and description field) which configures Solr to search for query terms in only titles and descriptions. We changed Solr search configuration to index the rest of the fields in the second round and accordingly included all the available metadata in the search.

Table 3.2: Solr field configuration for the available product metadata. Since different Configurations for round #1 and #2 were used we also report on the usage of the fields for the two evaluation rounds.

field	Description	Multi_Val	Round #1	Round #2
product_name	title of product		✓	✓
product_ID	ID of product	✓		✓
category	the category of product			✓
main_category	the main Category of the product			✓
brand	the name of product's brand			✓
photos	list of product's photo	✓		✓
short description	a short description of the product			✓
description	a full description of product		✓	✓
category_id	the ID of category			✓
main_category_id	the ID of the product main category			✓
bonus_price	the sale price if the product is on sale			✓
price	the normal price of product			✓
available	the products' availability status			✓
age_ min	recommended minimum age			✓
age_max	recommended maximum age			✓
characters	the character of toy e.g. Spiderman	✓		✓
gender	gender recommendation	✓		✓
arrived	first became available date			✓

3.3 Results

The final resolutions of the challenge were determined by comparing the performance of the participated systems toward the site.

These outcomes were documented by giving numbers on the (1) impressions per query, (2) wins, losses, and ties calculated against the production system, and (3) the calculated outcome:

$$\frac{\#wins}{\#wins + \#losses}$$

As noted in the documentation a win "is defined as the experimental system having more clicks on results assigned to it by Team Draft Interleaving than clicks on results assigned to the production system". This means that any value below 0.5 can be realized as a performance worse than the production system. Due to the problem of unavailable items in the online shop (44% of items in the query candidates products were unavailable during the first round) the expected outcome had to be corrected to 0.28 $P(participant > site) = (10.44) * 0.5 = 0.28$. as unavailable items were not filtered out for participating systems. Indeed, the REGIO site filtered the unavailable items after interleaving. Our system received 523 impressions in the two-week test period. This makes roughly 37.4 impressions

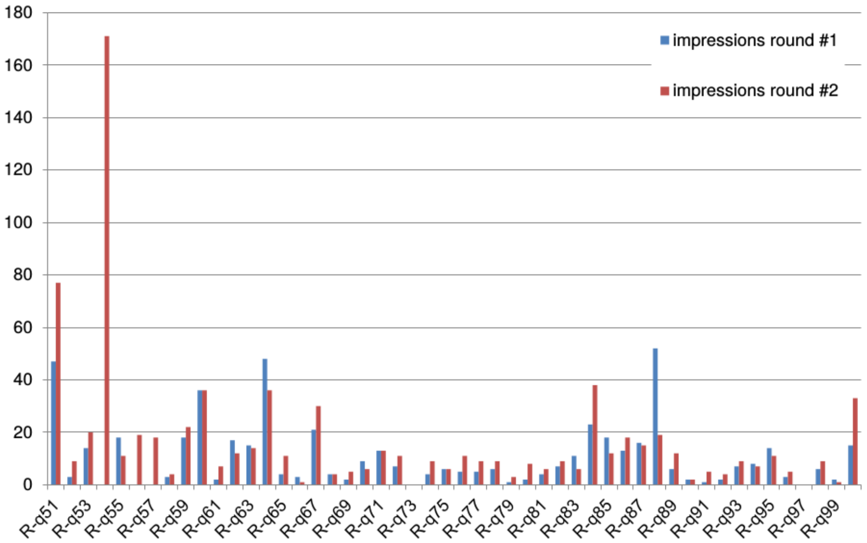


Figure 3.2: Distribution of impressions per topic for the first and official CLEF round (blue) and the second unofficial test round (red)

per day and 1.6 impressions per hour. Although we don't have any comparable numbers we interpret these impression rates to be quite low. If we compare to the other teams we received the lowest number of impressions while for example system UiS-Mira received 202 more impressions in the same time period (725 impressions which are 38% more impressions than we got). This is not quite in line with the principle of giving fair impression rates between the different teams. Another thing regarding the impressions is the fact that different queries had very different impression rates. Figure 3.2 illustrates this phenomenon clearly. It shows that whereas some of the runs got more than 50 impressions some were not shown to the users at all.

The Outcome in the First Round

Our ranking algorithm in the first round did not perform very well due to some obvious misconfiguration and open issues of our implementation. In fact, we provided the least efficient ranking compared to the other three participants [31]. We could achieve an outcome of 0.2685 by getting 40 wins vs. 109 losses and 374 ties. On the other hand, no other participant was able to beat the baseline with an outcome rate of 0.4691. The best performing system received an outcome rate of 0.3413 (system UiS-Mira) and was able to be better than the expected outcome of 0.28 but yet below the baseline provided by the organizers.

Table 3.3: Some basic statistics on the results.

	round #1	round #2
test queries	50	50
impressions total	523	816
impressions per day	37.4	58.1
queries with no impressions	5	2
queries with outcome > 0.5	4	13
queries with outcome > baseline	10	13

Listing 3.2 shows an example of outcomes of the first round represented by LL4IR. The outcomes declare that our run the query with the ID of "R-q60" are shown 36 times during the two weeks of the test period to the REGIO users. From these 36 impressions, 11 times we (our ranking) have got fewer clicks than the site ranking (losses), 23 times we got the same number of clicks (ties) and just two times we got more clicks than the site (win). The general score as the outcome of our run for this query is $2/(2 + 11) \approx 0.16$.

The Outcome in the Second Round

We also took part in the 2nd evaluation round and adapted some parameters of the system and tried to compensate the deficiencies of our approach from the first round. As there was a misconfiguration in the Solr system of round #1 we only searched the titles and description of products. We fixed this bug so that for round #2 we correctly indexed all available metadata fields. Another issue from round #1 was that not all 50 test queries were correctly calculated. We only used the historical click data for 1 test query and 13 training queries. For other queries, we used just the standard Solr ranking without any click history boosting. We fixed this issue for round #2 and calculated all the 50 queries according to the described boosting approach. After we corrected these two points we observe a clear increase in the outcomes. The outcome increased to 0.4520 by getting 80 wins, 97 losses, and 639 ties. Although the performance increase might be due to the fixes introduced by the organizers regarding unavailable items we could still see some positive effects: The performance of the other teams increased too, but while we were the weakest team in round #1 we were now able to provide the second best system performance. We also outperformed the winning system from round #1. Nevertheless, we (and no other system) were able to compete with

Table 3.4: Outcome, wins, losses and ties from round #1 and #2.

	outcome	wins	losses	ties
round #1	0.2685	40	109	374
round #2	0.4520	80	97	639

the production system. Comparing the number of impressions we observed a clear increase in queries that are above the 0.5 thresholds and the baseline (13 queries each) and the impressions in total and per day are also increased. The issue of unbalanced impression rates stays the same for round #2 (see figure 3.2).

3.4 Limitations

We were provided with a data collection with Hungarian content. This was the first challenge in this task since we don't know Hungarian and we couldn't take advantage of the semantic of contents. Besides the language problem was the name of the brand. Although some well-known name were also familiar to us like Spiderman (we knew that it is a toy character) but many of them were undefined for us. It was the same problem we had also for the name of the brands. Because of these limitations, we decided to use a language and content-agnostic approach which is independent of the meaning of the content.

It could have taken advantage of additionally provided metadata like the classes of the two-level deep topical categorization system to which the products were assigned to. Since we are unfamiliar with their categorization system, we could only add the category names to the index like other fields.

A typical problem with real world systems was also present in the available queries: Real world users tend to use rather short queries. For the 100 available query strings, only 15 had more than one word and only 2 had more than 2 words. (R-q22: "Bogyo es Baboca" and R-q50: "my little pony"). The average word length per query was 1.17 and the average string length was 7.16 characters. Another factor that we did not think of, although it was clearly stated in the documentation and in the key concepts 6, was the fact that no feedback data was available during the test phase. As this came to our mind way too late, we were only able to include historical click data for some queries. Therefore, the validity of our results from round #1 is weak as there are too few queries to really judge on the influence of the historical click data vs. live click data. We were not able to consider new feedback data for our rankings after the official upload and the beginning of the test phase. All the uploaded rankings were "final" and only depend on historical clicks. While this is of course due to the experimental setup, not truly a "living" component in the living lab environment. On top of that, not every document received clicks and therefore some documents are missing any hint of being relevant at all. Last but not least we had to struggle with speed issues of the LL4IR platform itself. As mentioned in the workshop report of 2014 there are known issues on "scaling up with the number of participants and sites talking to the API simultaneously"[5]. Although they state that these bottlenecks have been identified and that they have started addressing these it still takes some time to correspond with the API.

```
{
" outcomes ": [
        {
            " impressions ": 36 ,
            " losses ": 11 ,
            " outcome ": "0.15384615384615385" ,
            " qid ": "R − q60 " ,
            " site_id ": " R " ,
            " test_period ": {
                " end ": " Sat , 16 May 2015 00:00:00 −0000" ,
                " name ": " CLEF LL4IR Round #1" ,
                " start ": " Fri , 01 May 2015 00:00:00 −0000"
            },
            " ties ": 23 ,
            " type ": " test " ,
            " wins ": 2
        }
    ]
}
```

LISTING 3.2: Sample output of the outcome documentation

Chapter 4

Transforming DSpace into a Living Lab

4.1 Data structure in DSpace Repositories

Data in DSpace is formed with a set of communities, which are groups that contribute content to DSpace. Departments, research institutes, and schools are examples of communities in a university environment. Each community can be either a rooted community or a branch of another community and may include several sub–communities and collections. A collection is a group of related content that contains the content items or files. An item consists of three parts: (1) metadata, (2) bundles (e.g. LICENCE bundle, ORIGINAL bundle) and (3) bitstreams such as files (stream of bits 0s and 1s). Each item is described by a set of metadata. Communities and Collections can be described by a small number of metadata fields, commonly just title (name), identifier, logo, and description. However, the number of metadata fields for items is adaptable. Consequently, items in DSpace can be set up with a large amount of metadata. DSpace has considered an abstract metadata record –, known as Dublin Core – for each item besides its other metadata. Dublin Core facilitates the items' discovery. DSpace supports different metadata schemes to describe the items. Bundles are used to classify items and are collections of files (bitstreams). A common bundle is the ORIGINAL bundle, which contains the originally deposited files. Each digital object, i.e. Collection, community and an item in DSpace has a handle, which is a persistent identifier

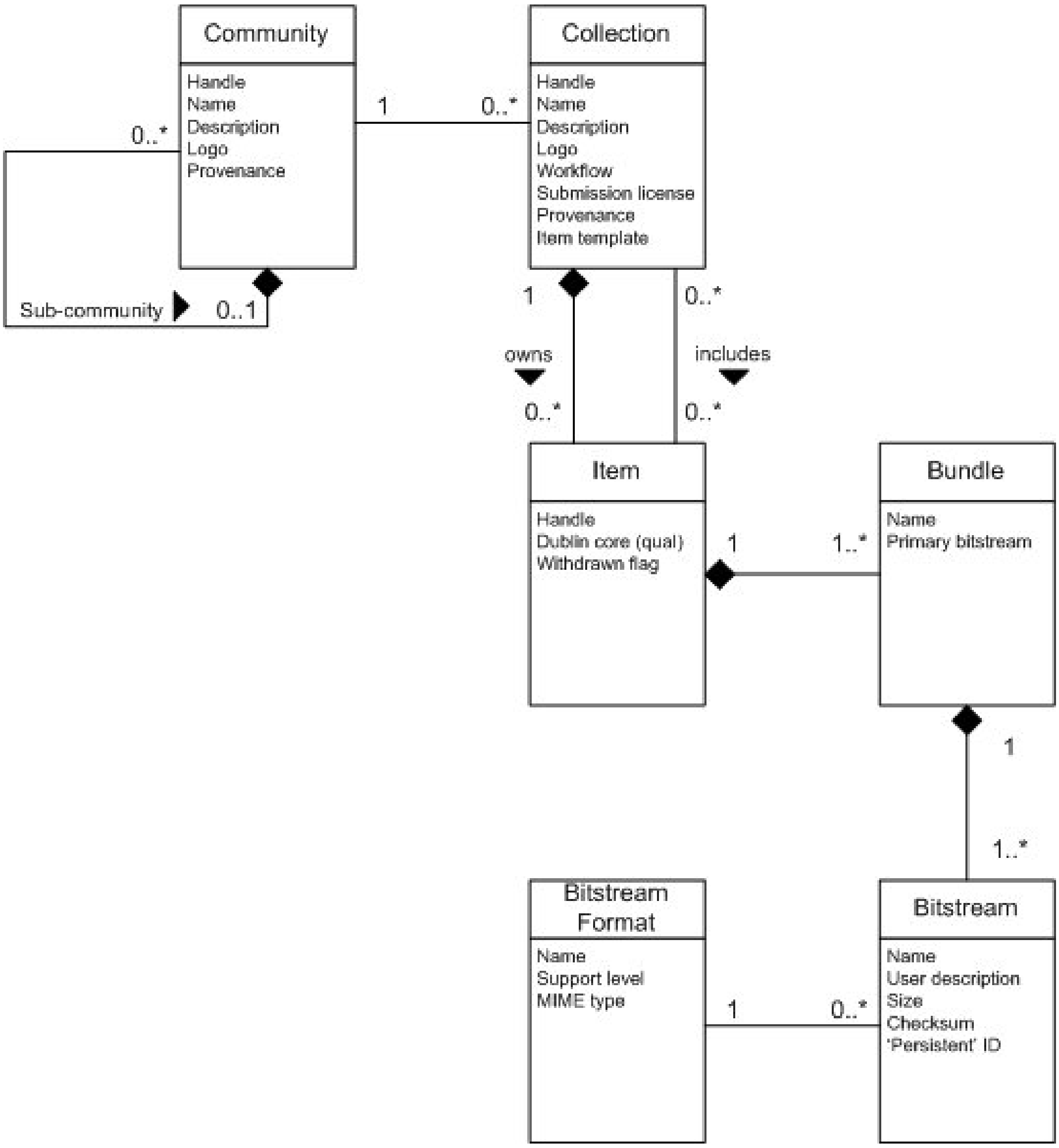


FIGURE 4.1: DSpace data model
http://DSpace.org/sites/DSpace.org/files/archive/1_5_2Documentation/ch02.html

for the digital object if they are registered in a handle server. Multiple relationships exist between collection and items. Collections are the owner of the items. Therefore, each item belongs to at least one collection. This data model is aimed to reflect the structure of the organizations that use the DSpace system [9]. Figure 4.1 depicts the structure in the space data model.

4.2 Search in DSpace Repositories

Search and indexing modules in DSpace create an easy way to index new content and perform searches on DSpace communities and collections. It supports an extensive and configurable search and browses framework [9] using the discovery module based on Apache Solr for all Search and Browse. Solr enables qualified searching, faceting, search results filtering, and accessing of the usage data (for statistics).

There are several ways for users to find their items or documents. First, the Simple Search is where a user enters a query term and DSpace shows a Solr ranked result to the user. Secondly, there is Browse and Search, which browses the available collections and communities selected by the user and DSpace return results sorted by relevance. A third approach is Advance Search, where a user can perform a combination search. In the simple and browse search a user can continue the discovering process and filter the returned results to reach his information needs. See Figure 4.2.

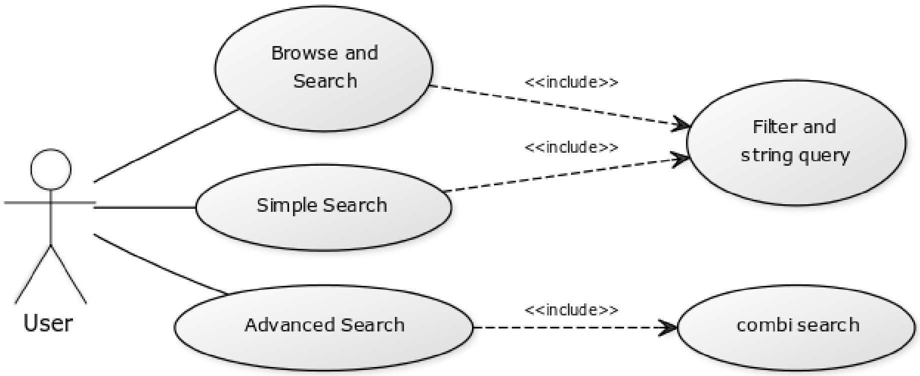


Figure 4.2: Search Use Case in SSOAR

The Discovery Module in DSpace is responsible for simple, faceted, and browse searching. The general process of searching in a DSpace repository is shown in following the use-case.

User	System
Search a query String	The system, shows a list of ranked items sorted by relevance(default)
Select an item Y	The System shows the item Y in details
View the Bitstream	The System shows the document to the user

4.3 Implementation of Living Labs Environment for DSpace IR Evaluation

SSOAR based on DSpace has active users and a search engine, which are the prerequisites for participating in a Living lab for information retrieval. Besides, participating in the OpenSearch evaluation campaign and transforming DSpace discovery module to a living lab environment can be achieved within five steps. Figure 4.3 illustrates the infrastructure of a living lab's environment, connections, and elements in DSpace.

- First, implementing a data collector (4.3.1) to collect Living Labs collection contains head queries, a document list for each head query, and document detail for each document on the list.
- Second, it implements a logger (4.3.2), which logs all the interested events (user interaction by searching).
- Third, it improvises an interleaver (4.3.3) which interleaves original ranking with site ranking.
- Fourth, a feedback extractor (4.3.4) to provide feedback from the log file for the living lab and
- Fifth, a site client to upload all this data (collections and feedback) to the living lab's API (4.3.5).

DSpace supports different discovery mechanisms, among them are search and browsing through the user interface. Browsing enabled the end users to search in

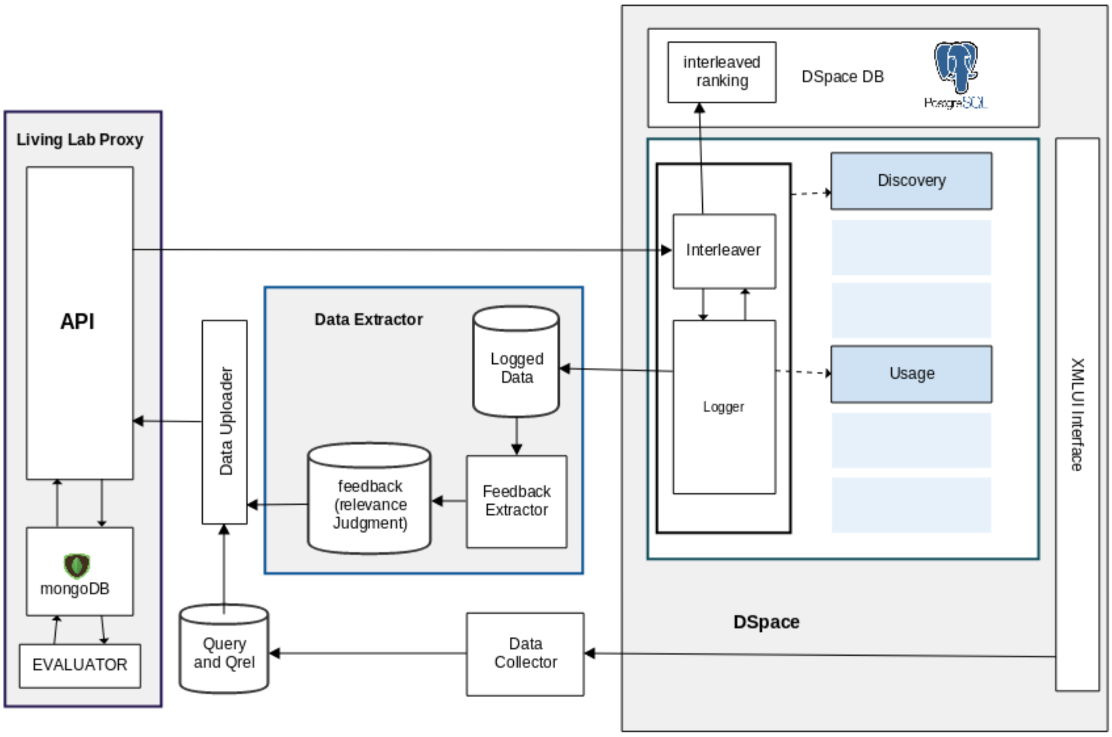


Figure 4.3: General schema of Dspace extension for living labs

a particular community, collection, author, publisher, etc. A standard DSpace discovery process (I) is depicted in Diagram 4.4. In such a DSpace natural discovery process, an end user performs a search in the repository by creating simple term query, browsing a collection/community, faceted browsing, etc. Correspondingly, the Solr service which supports Search and Browse in DSpace discovery returns a ranking of relevant documents for the query. Eventually, DSpace shows this ranking to the user. The user may proceed to click on the retrieved result's items to view or download matching items, or he may generate another query. by filtering the retrieval result. This discovery process with living lab (II) includes an event logger, which logs the queries created by a user. Each time a user performs a search, the living lab's event logger logs the query. At this point, the standard process of the search is repurposed if the generated query is on the list of experimental queries for the lab. After proofing some lab parameters and checking if certain conditions are met for the experiment - as will be explained in detail in Section 4.4- the site sends a request for an experimental ranking of that particular query to the living lab API. The API then may return an experimental ranking with an identifier (if there is any). The received experimental ranking will be sent together with the site ranking to an Interleaver (Team Draft Interleaver in this lab). The interleaver interleaves two rankings and labels each item with the name of its ranking session ID, i.e. site and experimental ranking identifier. The interleaved ranking is then returned and shown to the user. This labeled ranking, together with possible user clicks on the items, are recorded by the event logger. Figure 4.9 shows the corresponding modules that are added in the discovery component of DSpace: (1) LiLaLogger for logging the lab events including query calls, interleaved results, and clicks on retrieving items, (2) LiLaInterleaver for interleaving the experimental ranking with site ranking, (3) LiLaDB for the communication with living lab's API and for catching the interleaved results, and (4) LiLaConnector for checking living lab's parameters and activating the lab. Finally, a feedback extractor extracts feedback such as clicks and stores them in JSON format. The feedback should be uploaded through site client to the API for the evaluation and comparison of experimental ranking systems.

4.3.1 Collection Provider

Living labs require a test collection that contains head queries and candidate's documents with their details. To collect this information from the site, a collection provider (Data Collector in Figure 4.3) for living labs is implemented. This collection provider gets a list of queries for the lab as input, and records query relevant documents with their metadata (documents detailed information) as candidates' documents in special JSON formats, which are required by the living lab's API. It gets as input a set of head queries and stores document lists and documents' metadata in JSON files.

DSpace Log Analyses for Obtaining the Head Queries

For the living lab, we are required to prepare head queries to represent a set of the most frequent queries in our system. Living labs expect to get a static sample of 100 high frequent queries for the lab during the experimental period. For that we analyzed the contents of the previous log files of the SSOAR portal between August 2013 and June 2015 (712 days or almost 23 months) and extracted the queries that are called more frequently.

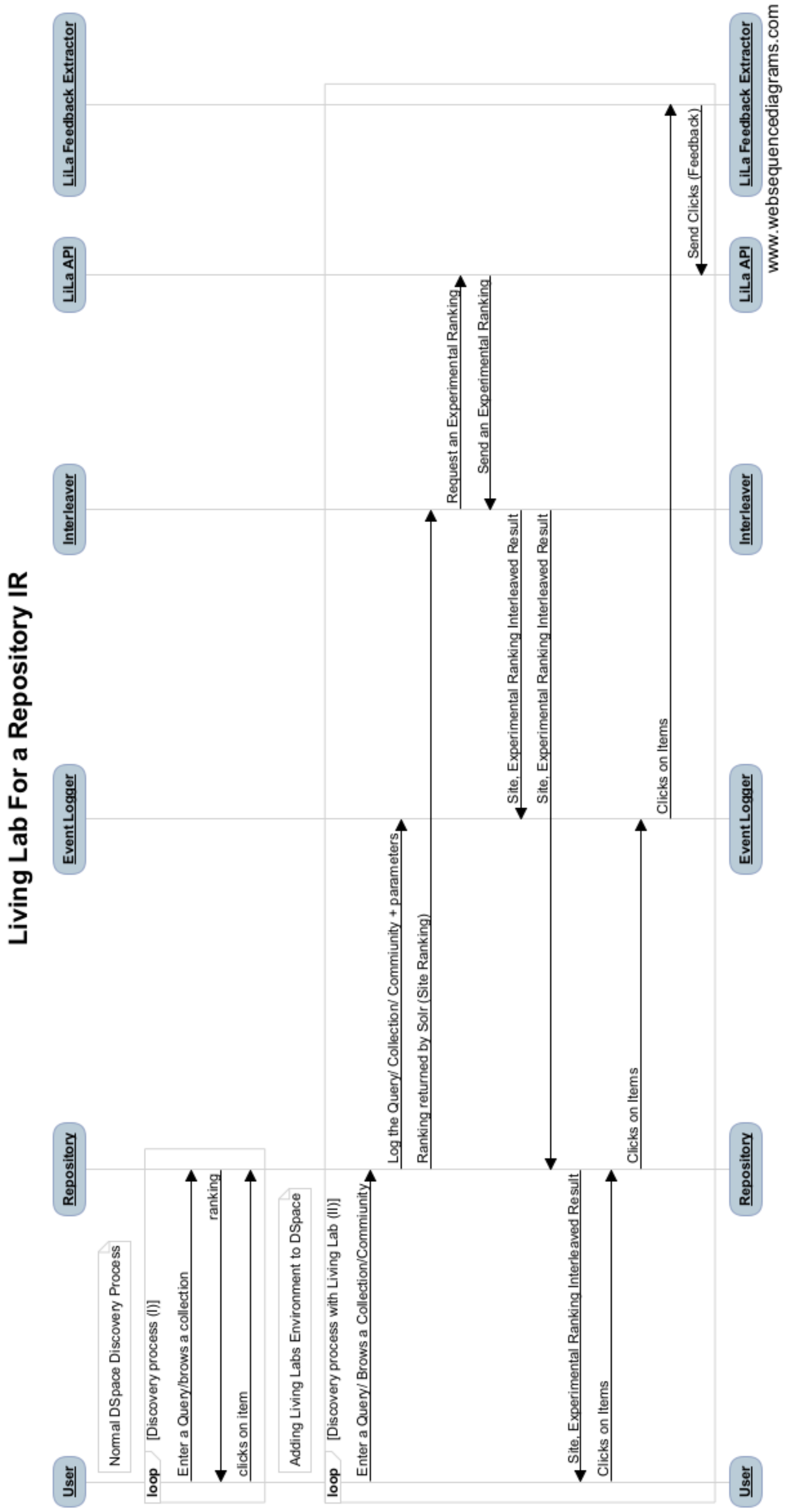

Living Lab For a Repository IR
User
Repository
Event Logger
Interleaver
LiLa API
LiLa Feedback Extractor
Normal DSpace Discovery Process
loop
[Discovery process (I)]
Enter a Query/brows a collection
ranking
clicks on item
Adding Living Labs Environment to DSpace
loop
[Discovery process with Living Lab (II)]
Enter a Query/ Brows a Collection/Commiunity
Log the Query/ Collection/ Commiunity + parameters
Ranking returned by Solr (Site Ranking)
Request an Experimental Ranking
Send an Experimental Ranking
Site, Experimental Ranking Interleaved Result
Site, Experimental Ranking Interleaved Result
Site, Experimental Ranking Interleaved Result
Clicks on Items
Clicks on Items
Clicks on Items
Send Clicks (Feedback)
User
Repository
Event Logger
Interleaver
LiLa API
LiLa Feedback Extractor
www.websequencediagrams.com

FIGURE 4.4: Sequential diagram for adding Living Labs to the DSpace Repository

Default DSpace logger logs contain a simple log of events that arise within the DSpace code, such as search process interactions between the user and the site, as query calls and clicks on items. With this data we were able to extract simple query strings as well as browse collections and communities with their handle and calculate query frequencies over time. Therefore we could extract the most frequent queries of SSOAR that have default DSpace logs.

We were able to collect information from logs using two different regular expression patterns in order to obtain (1) the simple searched queries and (2) browsed collections and communities queries. The extracted information includes user data and interactions' events as user IDs and IP addresses, query terms, and session ids with event modification times.

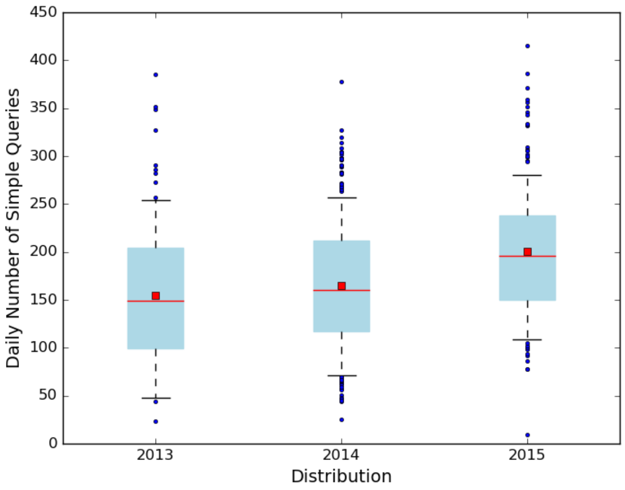


Figure 4.5: Comparison of Number of daily Simple Queries Across Three Years

Simple search

The distribution of the number of query impressions per day between August 2013 to June 2015 is demonstrated in Figure 4.5. The graph maps the daily number of queries in years, from the beginning of August 2013 until the 11th of July 2015. The average number of query calls in 2014 on SSOAR was 165 queries per day - almost the same average was obtained for the number of queries in the last five months of 2013. In the first six months of the year 2015, this average increased to 202 queries per day.

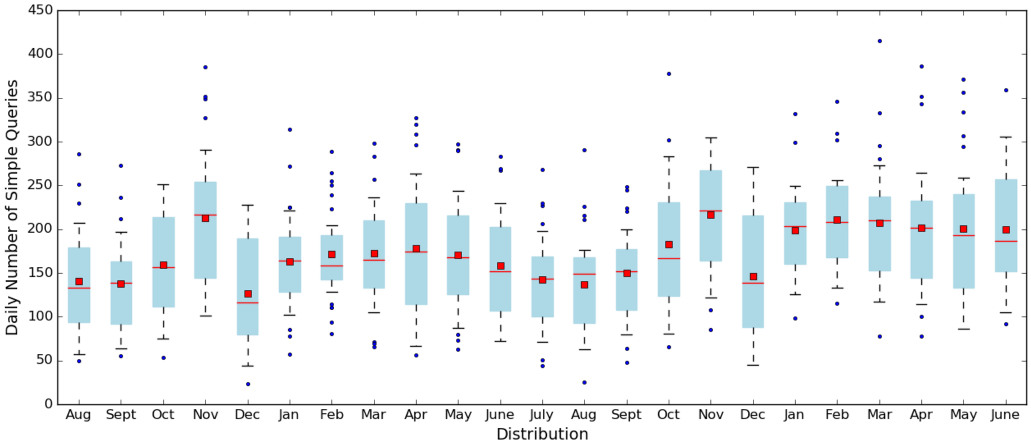


FIGURE 4.6: Comparison of Number of daily Simple Queries over the Months from August 2013 to Jun 2015

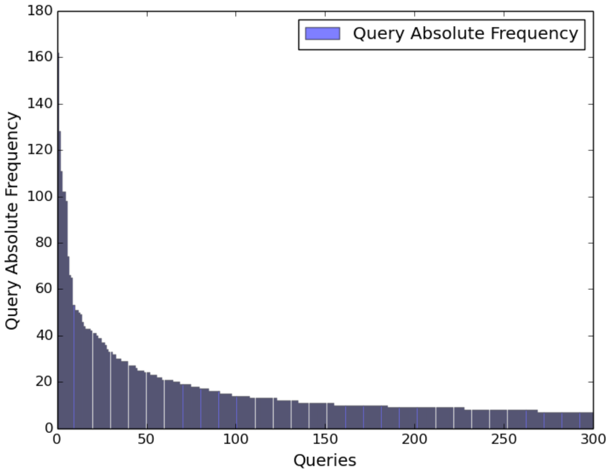


FIGURE 4.7: 300 of the most frequently searched queries (term search) on SSOAR during 23 month started from August 2013 until Jun 2015

Figure 4.6 illustrates and compares the distribution of monthly searched simple term queries on SSOAR in the mentioned period. The impressions' average reaches a peak in November and falls sharply in December to its lowest level. Tracking SSOAR user simple searches in this period, we collected 20503 distinct query terms.

Figure 4.7shows the frequency of the first 300 most-searched simple queries in the above mentioned period. One can observe that the 70 most common query terms have the frequency of between 20 and 162 impressions during this period.

Table 4.1 shows the first eight most frequently asked query terms. The most

Query	Frequency of 23-Months	Per Week
kommunikation@gesellschaft	162	1.829
broeskamp	128	1.431
fritz schuetze	116	1.284
computerspiel	102	1.159
weiss, felix	98	1.113
schuetze	85	0.886
europa regional	67	0.75
avo	66	0.75

Table 4.1: 8 most searched terms in SSOAR simple search between 2013 and 2015

frequency belongs to the query term “kommunication@gesellschaft” with 162 query calls in almost 23 months, with the average of fewer than two impressions in a week. When the frequency of a query call in these intervals remains the same, it means that there are almost eight query calls in each month, or four calls in two weeks.

Although the minimum amount of frequency is not intended by the TREC Living labs organizer, it is still mentioned that the head queries should have an appropriate number of frequencies for training and evaluating (comparing) the experimental ranking systems in the lab. The long series of tail queries exposed on the right side of the graph 4.7, in comparison to the small number of high-frequency queries on the left side reveals the limited data (from simple search resources) to share with the living labs, as the shared queries with living labs are supposed to have enough frequency.

Browse Search

In simple search we observed a very long tail of queries (least frequent) that are very uncommon. The tail queries are not useful for the living labs, as they will not be queried enough. Since we could not find appropriate queries with a high enough frequency from a simple search scenario, we decided to observe SSOAR’s other user search interfaces. Browse and Search interface is one of the search interfaces in SSOAR with a faceted search concept and facilitates browsing the documents that belong to a particular research area (collection and communities) as well as documents that belong to the special author, publisher, and so on.

After browsing, a user is able to (1) select the documents to view or download or (2) continue to search in the result list by entering query terms and/or filtering more items to make the query more specific. We decided to investigate whether we can consider the first scenario as an alternative to obtaining the head queries. We extracted the browsing calls from log files during the mentioned period.

Extracting the patterns of browse searches from log files from the mentioned period, we were able to collect 129 distinct browsing categories. Figure 4.8 shows the absolute number of frequencies of browsing 100 communities and collections. It is observable that a third of these have the impressions of more than 5000.

Table 4.2 shows these numbers precisely for the first 10 most frequently browsed communities and collections. The main category "Sociology" was chosen 25193 times during this period. We were convinced that browsing is more popular for SSOAR users than the simple keyword based search, and that the impressions are promising for the living labs. Accordingly, we generated the majority of head queries from the browsing categories besides a few numbers of the keyword searches. We have 134 defined categories (Fachgebiete in German) in SSOAR, which are communities and collections in DSpace. The provided head queries for training and testing the living lab's ranking systems are recorded by living labs in specified JSON format that is shown in Listing 4.1. Each query provides the information depicting its (1) creation_time, (2) qid, which is a unique query identification for living lab's participants (it is generated by living lab's application once the query is uploaded and stored in living lab's database), (3) qstr, the query terms, which is a string for simple term search and the title of collection or community for browse search, (4) site_qid, which is our unique identification for queries that is generated by the site (our collection provider) and is used in our system to recognise the queries that are under experimentation for the living labs, and (5) a type that determines for which phases (training or test) the query is intended.

The Identification of Query Types

Each query in the list of experimental queries for the living lab (head queries) has an identifier in its site. As we have chosen a list of mixed of simple term queries (very few numbers) and browsed queries, we need to use an approach to the system in order to identify between these two types of queries and determine according to the query type the corresponding steps.

To make the system differentiate between a simple search and browsing queries, we added a mark at the beginning of the queries' identities. "SSS" (SSOAR Simple search term) should be appended at the beginning of the identity for the simple term queries and "SSB" (SSOAR Browse search) for the queries that are browsed. The rest of a query identifier has consisted of the Base64 encoding of the handle ID for the browsing queries, and the Base64 encoding of terms of simple search. For instance, the browsing queries "collection/50200" belong to the research area of "Naturwissenschaften" and is identified by the ID :SSBY29sbGVjdGlvbi81MDIwMA. The first three characters of the identifier (SSB) determine that the query is a browsing query and the rest "Y29sbGVjdGlvbi81MDIwMA" is a Base64 encoding of the search area identifier or handle ID (collection/50200) in SSOAR.

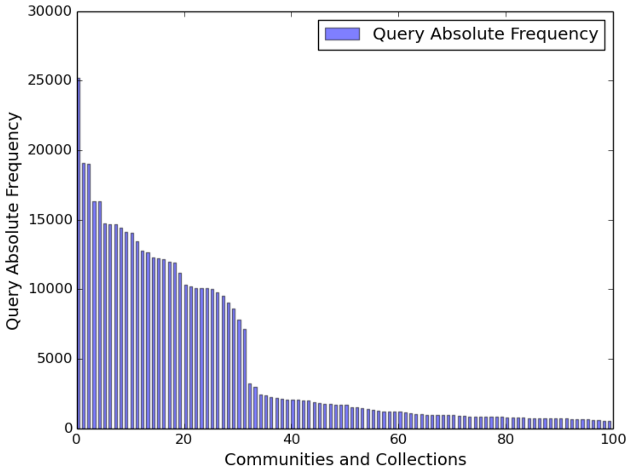


FIGURE 4.8: 100 of the most frequently searched (browsed) communities and collections on SSOAR during 22 month started from August 2013 until Jun 2015

Obtaining the Candidates Documents for each Head Query

For each head query specified for the living labs, we provided a list of candidate documents that is known in Living Labs as "doclist" . To prepare a document list for each query, we implemented a module in our data collector (Figure 4.3, which read the provided JSON file containing the head queries (Listing 4.1) and

Query	Frequency of 23-Months	Type	Per Week
Sociology	25193	community	282.79
Education and Pedagogics	19112	community	213.84
Basic Research in the Social Sciences	19009	community	213.69
Social Sciences	16322	community	183.34
Economics	16320	community	183.590
Social Problems	14719	collection	165.67
Psychology	14695	community	164.25
Science of Communication	14653	community	165.05
Humanities	14453	community	162.90
Jurisprudence Science	14115	community	159.30

TABLE 4.2: 10 most searched community and collection in SSOAR between 2013 and 2015

```
{
   "queries": [
        {
            "creation_time": "Thu, 17 Mar 2016 11:13:55 -0000",
            "qid": "ssoar-q1",
            "qstr": "broeskamp",
            "site_qid": "SSSYnJvZXNrYW1w",
            "type": "train"
        },
        {
            "creation_time": "Mon, 08 Feb 2016 16:23:01 -0000",
            "qid": "ssoar-q2",
            "qstr": "Naturwissenschaften, Technik(wissenschaften),
  angewandte Wissenschaften",
            "site_qid": "SSBY29sbGVjdGlvbi81MDIwMA",
            "type": "train"
        },
        {
            "creation_time": "Thu, 17 Mar 2016 11:13:55 -0000",
            "qid": "ssoar-q3",
            "qstr": "Social Problems",
            "site_qid": "SSBY29sbGVjdGlvbi8yMDUwMA==",
            "type": "train"
        },
       ...
              ]
}
```

LISTING 4.1: JSON contains SSOAR Head Queries For Living Labs

generated the correspond query using the qstr, site_qid, and some parameters listed in Table 4.3.1. The query should return at most 100 documents, ordered according to their relevance in decremental order. Our module sends the generated HTTP request for simple and browsing queries to SSOAR in order to get the result from the site:

simple search

```
discover?query=QUERY&submit=Suchen&rpp=100&sort_by=score&order=DESC
```

brows search

```
handle/HANDLE/discover?query=&submit=Suchen&rpp=100&sort_by=score&order=DESC
```

The retrieved list of each query is recorded by its document IDs, which contains unique IDs for each document in our data collection in the determined JSON format as "doclist" in Listing 4.2.

For each document in a document list, we need to provide an entity in a JSON format containing (1) site_docid the identities which match to the identities in the document list, (2) a title for the document, and (3) content that can have different optional metadata describing the document. After all the entities in a document list are uploaded, we can upload the document list. Each document uploaded into the API is assigned a new ID generated by Living Labs. For each uploaded document, a docid is generated, which is the document identities considered for participants.

Query Parameters	
QUERY	Simple Search Term (qstr f.e. broeskamp)
HANDLE	Brows Search Id (collection/community id f.e. collection/30301)
max Results	100 (The specified maximum number of candidates documents for each query)
Sort Order	DESC (Most Relevant First)
Sort Field	score (in Solr)

```
{
    "doclist": [
        {
            "docid": "ssoar-d2",
            "site_docid": "document6675",
            "title": "Glocalized bodies: body arts and cultures in
   time of globalization"
        },
        {
            "docid": "ssoar-d3",
            "site_docid": "document31134",
            "title": "Corps \u00e9trangers"
        },
        {
            "docid": "ssoar-d4",
            "site_docid": "document13462",
            "title": "Fremdheit und Rassismus im Sport"
        },
        {
            "docid": "ssoar-d5",
            "site_docid": "document13480",
            "title": "Integration/ Strangerhood"
        },
        ...
            ],

    "site_qid": "SSSYnJvZXNrYW1w"
}
```

LISTING 4.2: SSOAR Document List for Query Term broeskamp

Obtaining the Metadata for each Head Query's Candidates Document

Each document in the candidates' documents (doclist) should have an entity, including a site_docid, a title, and content. DSpace supports documents with rich metadata. We added some of this metadata as abstract, author, publisher, and the year of publication to describe the documents. Data Collector recorded the metadata of each document in document lists in a JSON format that is shown in Listing 4.3.

Field	Description
docid	Document ID presented to the participants
site_docid	Document ID for the site
title	Title of the Document (in English if available, otherwise in original language)
author	the name of author with format Last name, first-name
abstract	An abstract of document if available
description	A description about the document
identifier	The document URN/DOI identifier
issued	The publication date
language	The language of the document
publisher	The publisher of the document
subject	The subject of the document
type	The type of the article
creation_time	the date in which the our document description from the site is recorded

TABLE 4.3: Descriptions

```
{
   "content": {
       "abstract": "Der Text nimmt seinen Ausgangspunkt in Bourdieus
    \"Programm    f\u00fcr eine Soziologie des Sports\", das um das
    Konzept der transnationalen sozialen R\u00e4ume erweitert wird. \
   u00dcber die  Beschreibung transnationaler Sportr\u00e4ume hinaus
   werden diese dann in umfassendere Zusammenh\u00e4nge gestellt ,..."
   ,
       "author": "Broeskamp, Bernd",
       "description": "Kein Verlagsvertrag abgeschlossen",
       "identifier": "urn:nbn:de:0168-ssoar-66756",
       "issued": "2006",
       "language": "de",
       "publisher": "Bielefeld",
       "subject": "Theorieanwendung",
       "type": "Sammelwerksbeitrag"
   },
   "creation_time": "Wed, 16 Mar 2016 15:04:45 -0000",
   "docid": "ssoar-d2",
   "site_docid": "document6675",
   "title": "Glocalized bodies: body arts and cultures in time of
   globalization"
}
```

LISTING 4.3: Document Metadata for docid ssoar-d2

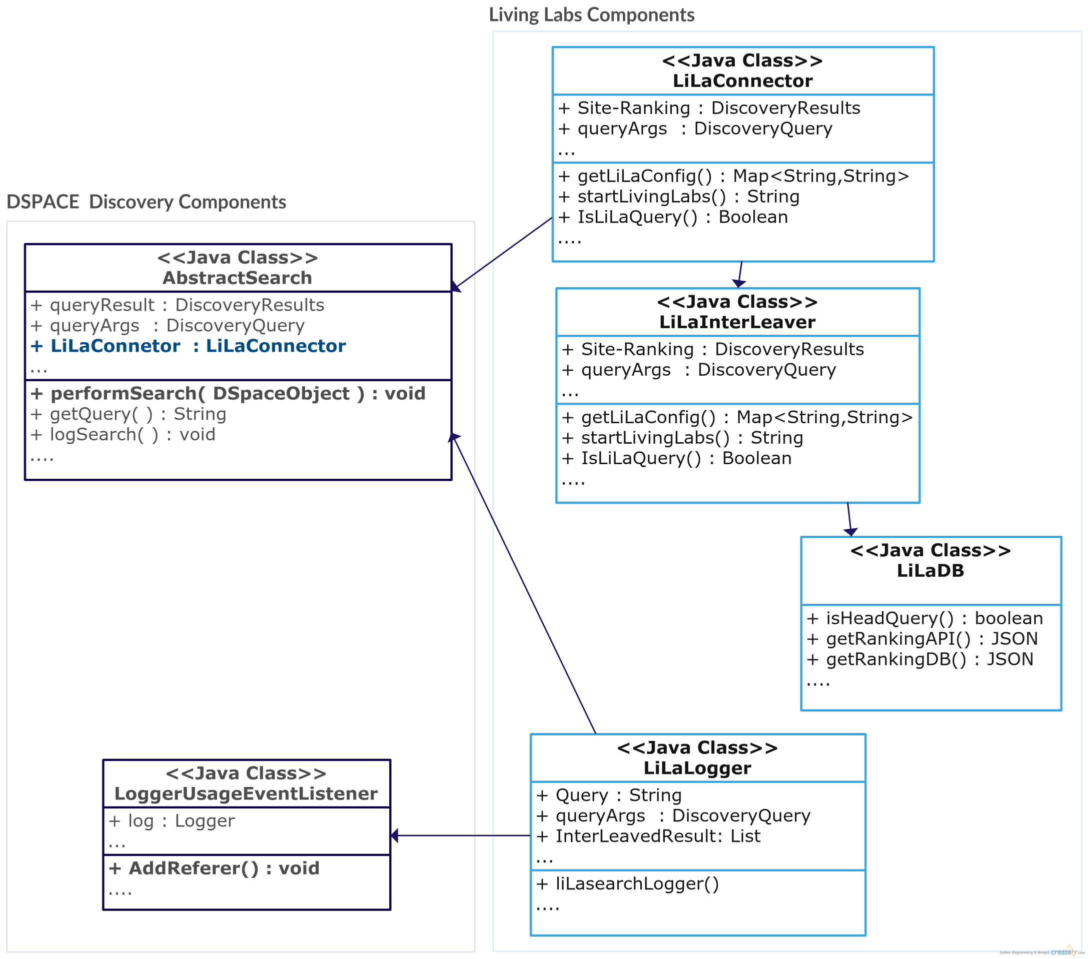

Living Labs Components
DSPACE Discovery Components
<<Java Class>>
LiLaConnector
+ Site-Ranking : DiscoveryResults
+ queryArgs : DiscoveryQuery
...
+ getLiLaConfig() : Map<String,String>
+ startLivingLabs() : String
+ IsLiLaQuery() : Boolean
....
<<Java Class>>
AbstractSearch
+ queryResult : DiscoveryResults
+ queryArgs : DiscoveryQuery
+ LiLaConnetor : LiLaConnector
...
+ performSearch(DSpaceObject) : void
+ getQuery( ) : String
+ logSearch( ) : void
....
<<Java Class>>
LiLaInterLeaver
+ Site-Ranking : DiscoveryResults
+ queryArgs : DiscoveryQuery
...
+ getLiLaConfig() : Map<String,String>
+ startLivingLabs() : String
+ IsLiLaQuery() : Boolean
....
<<Java Class>>
LiLaDB
+ isHeadQuery() : boolean
+ getRankingAPI() : JSON
+ getRankingDB() : JSON
....
<<Java Class>>
LoggerUsageEventListener
+ log : Logger
...
+ AddReferer() : void
....
<<Java Class>>
LiLaLogger
+ Query : String
+ queryArgs : DiscoveryQuery
+ InterLeavedResult: List
...
+ liLasearchLogger()
....

FIGURE 4.9: Class diagram for DSpace living labs extension

4.3.2 Logging For Living Lab

Standard DSpace logs the user interactions with the system as calling in queries, clicks on items (view), and downloads. However, the lists of returning documents i.e. the result of the keywords or queries entered by users are not recorded in the DSpace logs. Consequently, if an item in a list of query-relevant documents (Qrel) is clicked, one cannot establish to which query result and with which ranking the document belongs. Therefore, there is a need for a special logger for the living lab environment that logs all the required information including a ranked retrieval result of interested queries.

We have implemented a new logger (LiLaLogger as shown in Figure 4.9) that, in addition to the queries and user and query parameters, also logs the ranking results retrieved from Solr at the moment the queries are issued (see Listing 4.4).

For each viewed item we added a referrer to determine and record the query (URL) that leads to the item. As a result, the location of viewing items in the result list is clarified in the referrer. Having this recorded log we are able to track the user interactions and determine which item in which position is clicked after a query is created.

```
2016-03-08 12:15:07,086 [LiLaSearchLogger.java:http-bio-8084-exec]
-anonymous:
session_id=E9D3494B9E8EF8A78F97C9276BD7579F:
ip_addr=0:0:0:0:0:0:0:1:search: ,
query( broeskamp)
SessionID(E9D3494B9E8EF8A78F97C9276BD7579F),
UserID(null),
HostIP(0:0:0:0:0:0:0:1),
Handle(null) ,
Sort(score),
Currentpage(1),
Ppage(10),
Order(DESC),
result(
     {'clicked': 'false','docid':'document/6675','team':'none'},
     {'clicked': 'false','docid':'document/31134','team':'none'},
     {'clicked': 'false','docid':'document/6679','team':'s1'},
     {'clicked': 'false','docid':'document/6716','team':'site'},
     {'clicked': 'false','docid':'document/6715','team':'site'},
     {'clicked': 'false','docid':'document/13482','team':'s1'},
     {'clicked': 'false','docid':'document/31170','team':'s1'},
     {'clicked': 'false','docid':'document/13462','team':'site'},
     {'clicked': 'false','docid':'document/13480','team':'s1'},
     {'clicked': 'false','docid':'document/13479','team':'site'},
     {'clicked': 'false','docid':'document/6673','team':'s1'},
     {'clicked': 'false','docid':'document/6672','team':'site'},
     {'clicked': 'false','docid':'document/6685','team':'s1'})
```

LISTING 4.4: Recorded (logged) Search Event

4.3.3 Implementation of An Interleaver

The interleaving process can be switched on and off through related parameter of the living lab's config file. A user in a session enters a query to browse all

100 documents sorted by relevance to the query. Dspace checks if the query is an experimental query. If it is the case, Dspace sends a request to living labs API (GET ranking) and asks for an experimental ranking to the query. Each time the query is issued, the site should retrieve a new ranking from the API. When a ranking through API is called, an experimental ranking suggested by a participant of living labs challenge is returned according to least-served basis [31]. Therefore, participants can update their runs each time (during the training phase).

The documents listed in experimental rankings should be the same as the documents in the site ranking. The living lab API ensures that in the returned rankings are called by the site. We should on the other hand update the doclist if it has changed (when new documents are added to the doclist or if some documents are removed) .

The rankings submitted by participants may include documents that are not available at the time the query is issued or documents which are removed from the database. These documents need to be removed from the participant ranking list before the interleaving is performed, this process is shown in Figure 4.10 as Filtering.

The experimental ranking returned from API contains at most 100 documents. This ranking will be interleaved with the first 100 documents in the ranking returned from the site.

In order to show the user stable rankings for each query, we have to place them in a cache. The result will be saved as an interleaved result and will be shown to the user of the corresponding session. As it required in the OpenSearch, we used “Team Draft” Interleaving in our implementation for the living lab.

Team Draft Interleaving

Radlinski et al. [26] proposed a Team Draft Interleaving (TDI) approach for almost identical rankings. This method is compared to the procedure of selecting players (items) from a list of players by two captains for their teams. One of the captain begins the team selection process by random. The captains select their most preferred players from among the remaining players for their team in turn. Before starting to assign teams, TDI adds any common prefix to the list. For

TABLE 4.4: Example of interleaved result of experimental and site ranking

experimental ranking	site ranking	interleaved result
doc-13482	doc-6675	doc-13482
doc-31170	doc-31134	doc-6675
doc-6679	doc-6716	doc-31134
doc-6675	doc-6715	doc-31170
doc-31134	doc-13462	doc-6716
doc-13462	doc-13479	doc-6679
doc-13480	doc-13480	doc-6715
doc-6673	doc-13482	doc-13462
doc-6716	doc-6673	doc-13479
doc-6672	doc-31170	doc-13480
doc-6685	doc-6672	doc-6673
doc-6715	doc-6685	doc-6672
doc-13479	doc-6679	doc-6685

instance, if documents were letters in lists $A = \{a, b, c, d, e\}$ and $B = \{a, b, c, f, g\}$, the interleaving list will be $I_{team} = \{a, b, c, d_A, f_B, e_A, g_B\}$, i.e. $Team_A = \{d, e\}$ and $Team_B = \{f, g\}$. The first three items are assigned to "no team" . This means that users clicking on these three items are not considered as credits for team A or B. The next step after adding the common prefix includes picking the next item from the randomly selected ranking, which is not still on the interleaved list and adding it to the interleaved list. These steps continue until all of the rankings in one of the lists exist in the interleaved list (Algorithm 2). Accordingly, the maximum difference between the number teams' members is one ($|size(team_A) - size(team_B)| < 2$).

The evaluation of two rankings is assessed by comparing the rankings' performances. When a user clicks on an item which is assigned to a team in interleaved ranking, it is considered as a plus point for that team. The total number of clicks for each team determines the out-performance of one team over another.If the number of clicks on items that belong to team A, $h_A = |\{click_{I_n} : I_n \in TeamA\}|$ is greater than $h_B = |\{click_{I_n} : I_n \in TeamB\}|$, $(h_A > h_B)$, a preference for ranking system A is inferred. Likewise, $h_A = h_B$, declares no preference over two teams.

4.3.4 Feedback Extractor

The living labs expected the sites to upload obtained feedback of the site's users for the transferred experimental ranking when they are generated. Each experimental ranking received from the API has an identity known as session IDs. The feedback returned to the API should have this identity with it. The feedback can be stored

Algorithm 2 Team Draft Interleaver

```
Input  : Rankings A = (a1,a2,...) and A = (a1,a2,...)
Init   : L ← (); TeamA ← 0 ; TeamB ← 0; i ← 1
while $A[i] = B[i]$ do
    L ← L + A[i]
     i ← i + 1
end
while $(s\ i : A[i] \notin L) \wedge (\exists\ j : B[j] \notin L)$ do
    L ← $L + A[i]$
     i ← $i + 1$
     if $(|TeamA| < |TeamB|) \vee ((|TeamA| = |TeamB|) \wedge (RandBit()=1))$ then
        k ← $min_i$ {i: A[i] ∉ L }
         L ← L + A[k]
         TeamA ← TeamA ∪ { A[k]}

    else
        k ← $min_i$ {i: Bı ∉ L }
         L ← L + B[k]
         TeamB ← TeamB ∪ { B[k]}
    end
end
Output: Interleaved ranking L, Team A, Team B
```

several times for the same ranking with its session ID if the ranking gets further clicks afterward in the same session.

A Feedback Extractor (Figure 4.3) is needed to extract the feedback from the logged data, format them to a particular JSON required by the living labs' API, and transfer (upload) them to the API (see Data Uploader in Figure 4.3).

To obtain feedback (clicks), the observed items (in the view events) that reference a query result page (is followed by a search event) using the same session ID are recovered and collected, and the "clicked" attributes of these items (4.4) in the search result list of the query are set to "true" . Living labs required the sites to implement a caching mechanism so that the same user would be presented with the same interleaved ranking for a given query in a particular session [31].

For that, each experimental ranking (interleaved with site ranking) shown to the user is recorded in the database in order to be served afterward in the same session when the same user in the same session issues the same query multiple times, or if the user clicks to see more search result pages. The Listing shown in 4.5 is an example of the feedback for the living labs. Each document in the doclist has three attributes: (1) the team to which the item belongs, (2) the clicked condition (whether the user counted the document as relevant to his query), and (3) the document ID for the site.

4.3.5 Data Uploader

Living Labs API [1] acts as a bridge that enables the communication and data transformation between participants such as researchers and sites. The transformations are done via HTTP request (GET, Head, and PUT).

We need to implement a client which communicates with the living labs' API to (1) upload and update the test collections, (2) get the experimental rankings, and (3) upload and update the feedback. Living lab's organizers provided sample code that implements clients that talk to living labs, an an API for both participating sites and researchers. This code is made available in an open source form by living lab organizers. It is accessible in a GIT repository [2].

[1] http://doc.trec-open-search.org/en/latest/api-site.html

[2] https://bitbucket.org/living-labs/ll-api/

```
{
  "doclist": [
    {"team": "none",
      "clicked": true,
      "site_docid": "document6675"},

    {"team": "site",
      "clicked": false,
      "site_docid": "document6716"},

    {"team": "participant",
      "clicked": true,
      "site_docid": "document6679"},

    {"team": "site",
      "clicked": false,
      "site_docid": "document6715"},

    { "team": "participant",
      "clicked": false,
      "site_docid": "document13482"},
    ....
    ],
  "type": "tdi",
  "site_qid": "SSSYnJvZXNrYW1w",
  "modified_time": " 2016-03-03 19:39:07,863",
  "sid": "s1"
}
```

LISTING 4.5: Feedback for ranking S1

4.4 Activation of Living Labs for DSpace Repository

The Living lab is activated by setting the related variable in living lab's config file shown in Listing 4.6.

When the living lab is active and a query that is an experimental head query is issued, the standard search process in SSOAR is redirected to the lab process, which is illustrated in Figure 4.10. In the redirected search process, the lab

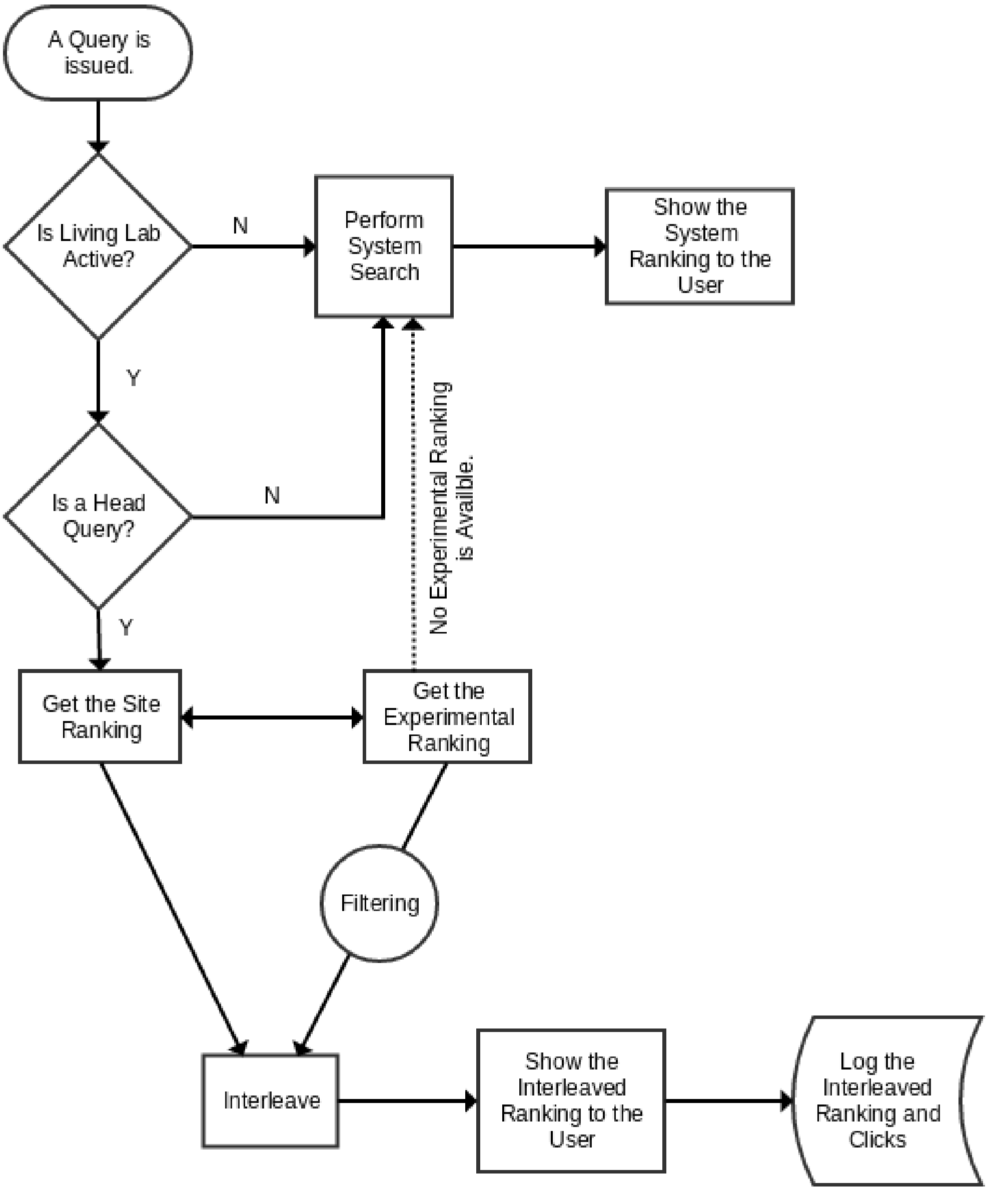

A Query is issued.
Is Living Lab Active?
N
Perform System Search
Show the System Ranking to the User
Y
Is a Head Query?
N
No Experimental Ranking is Availble.
Y
Get the Site Ranking
Get the Experimental Ranking
Filtering
Interleave
Show the Interleaved Ranking to the User
Log the Interleaved Ranking and Clicks

FIGURE 4.10: Living Labs in discovery performance Flowchart

component in DSpace sends a request to the living lab's API for obtaining an experimental ranking. When the API returns a ranking successfully, the lab component sends it - together with the site ranking - to the interleaver, where the two rankings combined together, are recorded in the living labs logger, and presented to the user that issued the query.

```
lila.livinglabs.active=true
lila.interleave=true
lila.headQuery.source=site
lila.qrel.max=100

lila.api =http://api.trec-open-search.org/api
lila.api.key =3DXXXXXXXXX-BJWKXGXXXXXXXXXXX
```

LISTING 4.6: Living Labs Configuration File

4.5 Participation in TREC OpenSearch as Site

TREC OpenSearch Track [3] runs living lab for the first time for re-ranking with the same aim as CLEF LL4IR. It uses the same structure of CLEF LL4IR and the tracking task is ad-hoc Academic Search.

The TREC OpenSearch API is a REST-full API. JSON is the unique format of all the data transformed. The search engines that share their data are called sites.

We participate as a site in this lab as SSOAR besides CiteSeerx [4] and Microsoft Academic Search [5] for experimenting the ranking methods during the lab. TREC OpenSearch evaluation will be perform in three rounds (Jun 1-July 15, Aug 1-Sep 15, and Oct 1-Nov 15).

CiteSeerx is a search engine that provides free access to millions of scientific literature in computer and information science. Together with SSOAR, CiteSeerx has started to share its components with OpenSearch from the first evaluation round.

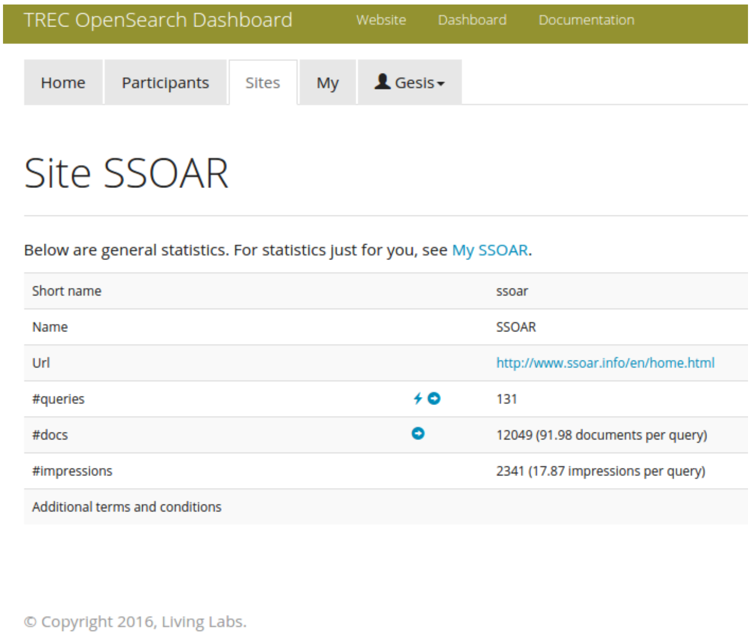
Microsoft Academic Search is developed by Microsoft Research. It serves as an experimental research service to explore how scholars, scientists, students, and practitioners find academic content, researchers, institutions, and activities. Besides indexing the academic publications it represents the key relationships between and among subjects, content, and authors, highlighting the critical links that help define scientific research. Microsoft Academic Search will participate in OpenSearch in the next evaluation rounds.

[3] http://trec-open-search.org/

[4] https://citeseerx.ist.psu.edu

[5] https://academic.microsoft.com/

We have generated 130 head queries (65 for training and 65 for test) in TREC ad-hoc style (queries + documents). We uploaded the data (See Figure 4.11), and used our implemented extension, which is adjusted to our site (SSOAR), for the interleaving task. We are following the clicks and sharing user interactions (clicks) with living lab's API.



Site SSOAR

Below are general statistics. For statistics just for you, see My SSOAR.

Short name		ssoar
Name		SSOAR
Url		http://www.ssoar.info/en/home.html
#queries	⚡ ➜	131
#docs	➜	12049 (91.98 documents per query)
#impressions		2341 (17.87 impressions per query)
Additional terms and conditions		

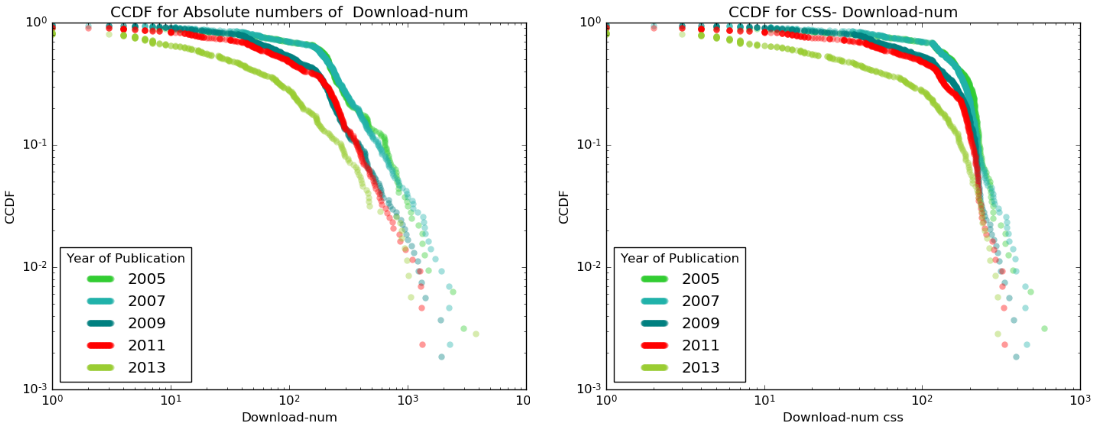
© Copyright 2016, Living Labs.

FIGURE 4.11: A Screenshot of TREC Open Search Dashboard Page, Representing the General Statistic of SSOAR Data Set Including the Number of Uploaded Queries, Documents, and the Impressions, http://trec-open-search.org/

4.6 Signing Up for Living Lab OpenSearch as Researcher to Implement Experimental Ranking Algorithms

4.6.1 Experimental Ranking for SSOAR

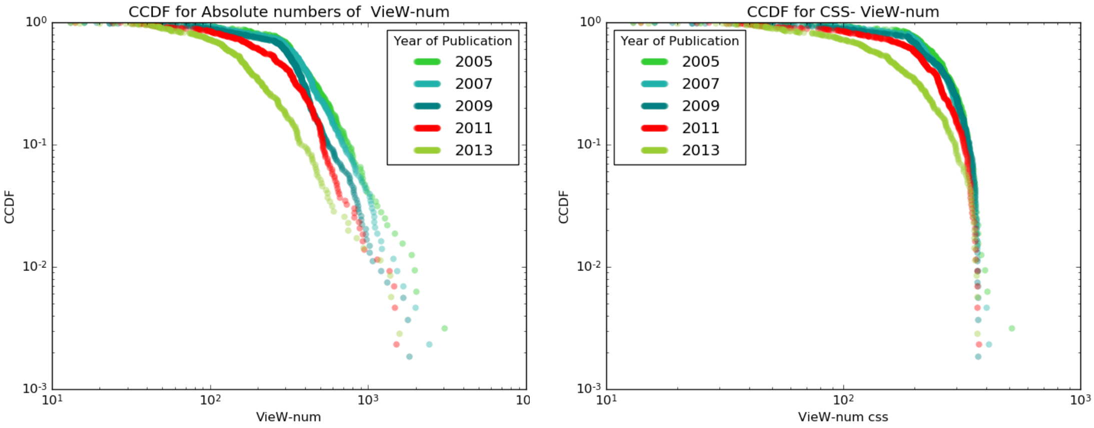
Currently twelve participants are registered for TREC Open Search. To show that our approach works, we participated also as a researcher who wants to evaluate the performance of his ranking algorithm and signed up for SSOAR. Accordingly, we generated our runs based on our ranking approach, which is based on Characteristic Scale and Score (CSS) introduced by Glänzel [12] combined with Solr score (TF-IDF). Inspired by the work of Plassmeier et al. [25], we applied a CSS score



(a) Number of Downloads

(b) The Application of CSS Score to Number of Downloads

FIGURE 4.12: Counter Cumulative Distribution Function of Downloads



(a) Number of Views

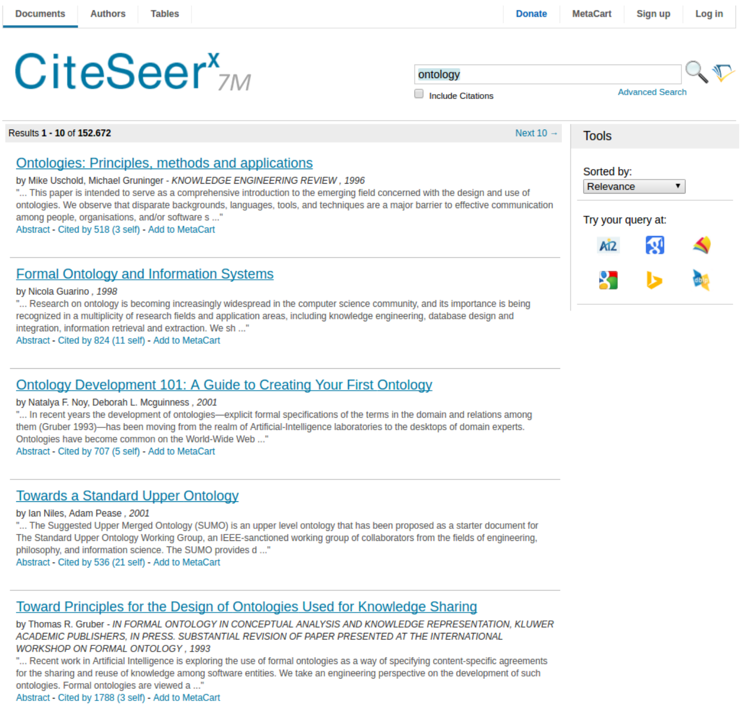
(b) The Application of CSS Score to Number of Views

FIGURE 4.13: Counter Cumulative Distribution Function of Views

to SSOAR usage data (the number of downloads and views for each document), which is extracted from Solr. The values of CSS are used to re-scale the usage distributions. The sum of CSS scores (for both the number of views and downloads) is multiplied by the document's Solr scores. Figures 4.12(a) and 4.13(a) plot the absolute number of downloads and views for documents published in the years 2005, 2007, 2009, 2011, and 2013. The application of a CSS score on the usage data is depicted in Figures 4.12(b) and 4.13(b). We also signed up for the other sites: CiteSeerx and Microsoft Academic Search (see Figure 5.2).

4.6.2 Experimental Ranking for CiteSeerx

CiteSeerx is a scientific digital library for computer and information science literature. Besides a digital library, CiteSeerx aims to contribute resources such as algorithms, data, metadata, services, techniques, and software that can help other digital libraries to improve. CiteSeerx uses its own methods and algorithms to index PostScript and PDF research articles on the Web. Figure 4.14 shows the screenshot of a CiteSeerx search result page.



Results **1 - 10** of **152.672**

Ontologies: Principles, methods and applications
by Mike Uschold, Michael Gruninger - *KNOWLEDGE ENGINEERING REVIEW , 1996*
"... This paper is intended to serve as a comprehensive introduction to the emerging field concerned with the design and use of ontologies. We observe that disparate backgrounds, languages, tools, and techniques are a major barrier to effective communication among people, organisations, and/or software s ..."
Abstract - Cited by 518 (3 self) - Add to MetaCart

Formal Ontology and Information Systems
by Nicola Guarino , *1998*
"... Research on ontology is becoming increasingly widespread in the computer science community, and its importance is being recognized in a multiplicity of research fields and application areas, including knowledge engineering, database design and integration, information retrieval and extraction. We sh ..."
Abstract - Cited by 824 (11 self) - Add to MetaCart

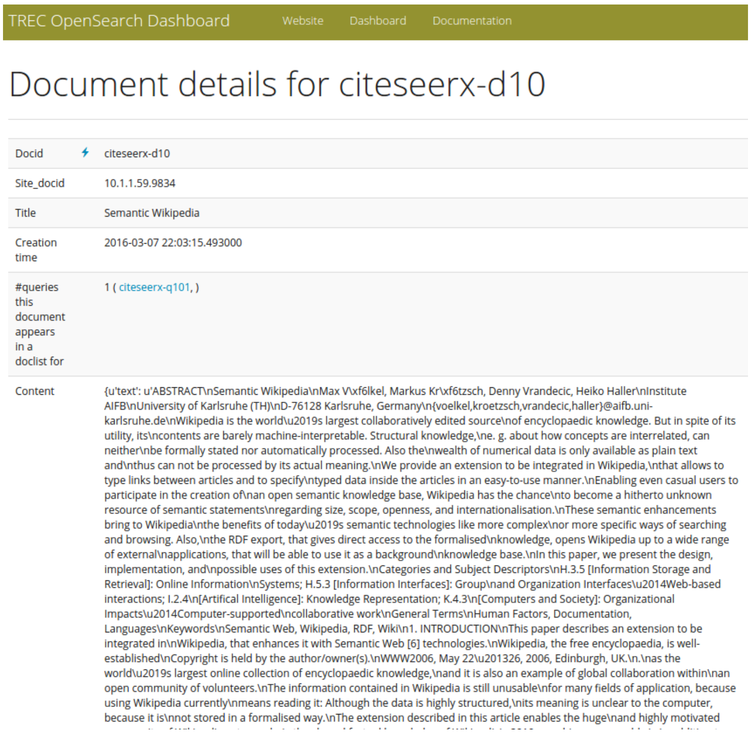
Ontology Development 101: A Guide to Creating Your First Ontology
by Natalya F. Noy, Deborah L. Mcguinness , *2001*
"... In recent years the development of ontologies—explicit formal specifications of the terms in the domain and relations among them (Gruber 1993)—has been moving from the realm of Artificial-Intelligence laboratories to the desktops of domain experts. Ontologies have become common on the World-Wide Web ..."
Abstract - Cited by 707 (5 self) - Add to MetaCart

Towards a Standard Upper Ontology
by Ian Niles, Adam Pease , *2001*
"... The Suggested Upper Merged Ontology (SUMO) is an upper level ontology that has been proposed as a starter document for The Standard Upper Ontology Working Group, an IEEE-sanctioned working group of collaborators from the fields of engineering, philosophy, and information science. The SUMO provides d ..."
Abstract - Cited by 536 (21 self) - Add to MetaCart

Toward Principles for the Design of Ontologies Used for Knowledge Sharing
by Thomas R. Gruber - *IN FORMAL ONTOLOGY IN CONCEPTUAL ANALYSIS AND KNOWLEDGE REPRESENTATION, KLUWER ACADEMIC PUBLISHERS, IN PRESS. SUBSTANTIAL REVISION OF PAPER PRESENTED AT THE INTERNATIONAL WORKSHOP ON FORMAL ONTOLOGY , 1993*
"... Recent work in Artificial Intelligence is exploring the use of formal ontologies as a way of specifying content-specific agreements for the sharing and reuse of knowledge among software entities. We take an engineering perspective on the development of such ontologies. Formal ontologies are viewed a ..."
Abstract - Cited by 1788 (3 self) - Add to MetaCart

Figure 4.14: A Screenshot of CiteSeerx search Page

As a site, CiteSeerx participated in TREC OpenSearch beside SSOAR and Microsoft Academic Search. It shared 200 queries including 100 test and 100 training queries. It has uploaded 10253 candidate documents for these queries (51.27 documents per query). Each candidate document for CiteSeerx has a single key "text" as its content (see Figure 4.15). The value of this attribute is the unicode of the document's full text and includes the document's whole text, i.e. the authors, abstract, keywords, and main text as it shown in the Listing 4.7).



Document details for citeseerx-d10

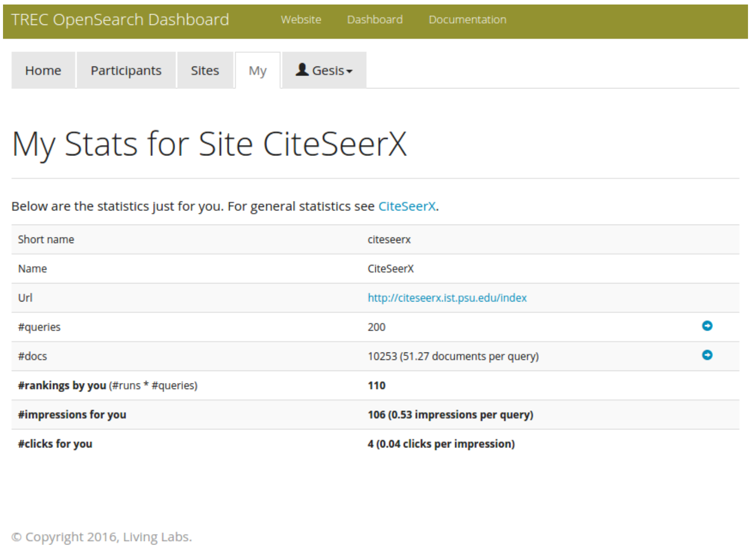
Docid	citeseerx-d10
Site_docid	10.1.1.59.9834
Title	Semantic Wikipedia
Creation time	2016-03-07 22:03:15.493000
#queries this document appears in a doclist for	1 (citeseerx-q101,)
Content	{u'text': u'ABSTRACT\nSemantic Wikipedia\nMax V\xf6lkel, Markus Kr\xf6tzsch, Denny Vrandecic, Heiko Haller\nInstitute AIFB\nUniversity of Karlsruhe (TH)\nD-76128 Karlsruhe, Germany\n{voelkel,kroetzsch,vrandecic,haller}@aifb.uni-karlsruhe.de\nWikipedia is the world\u2019s largest collaboratively edited source\nof encyclopaedic knowledge. But in spite of its utility, its\ncontents are barely machine-interpretable. Structural knowledge,\ne. g. about how concepts are interrelated, can neither\nbe formally stated nor automatically processed. Also the\nwealth of numerical data is only available as plain text and\nthus can not be processed by its actual meaning.\nWe provide an extension to be integrated in Wikipedia,\nthat allows to type links between articles and to specify\ntyped data inside the articles in an easy-to-use manner.\nEnabling even casual users to participate in the creation of\nan open semantic knowledge base, Wikipedia has the chance\nto become a hitherto unknown resource of semantic statements\nregarding size, scope, openness, and internationalisation.\nThese semantic enhancements bring to Wikipedia\nthe benefits of today\u2019s semantic technologies like more complex\nor more specific ways of searching and browsing. Also,\nthe RDF export, that gives direct access to the formalised\nknowledge, opens Wikipedia up to a wide range of external\napplications, that will be able to use it as a background\nknowledge base.\nIn this paper, we present the design, implementation, and\npossible uses of this extension.\nCategories and Subject Descriptors\nH.3.5 [Information Storage and Retrieval]: Online Information\nSystems; H.5.3 [Information Interfaces]: Group\nand Organization Interfaces\u2014Web-based interactions; I.2.4\n[Artifical Intelligence]: Knowledge Representation; K.4.3\n[Computers and Society]: Organizational Impacts\u2014Computer-supported\ncollaborative work\nGeneral Terms\nHuman Factors, Documentation, Languages\nKeywords\nSemantic Web, Wikipedia, RDF, Wiki\n1. INTRODUCTION\nThis paper describes an extension to be integrated in\nWikipedia, that enhances it with Semantic Web [6] technologies.\nWikipedia, the free encyclopaedia, is well-established\nCopyright is held by the author/owner(s).\nWWW2006, May 22\u201326, 2006, Edinburgh, UK.\n.\nas the world\u2019s largest online collection of encyclopaedic knowledge,\nand it is also an example of global collaboration within\nan open community of volunteers.\nThe information contained in Wikipedia is still unusable\nfor many fields of application, because using Wikipedia currently\nmeans reading it: Although the data is highly structured,\nits meaning is unclear to the computer, because it is\nnot stored in a formalised way.\nThe extension described in this article enables the huge\nand highly motivated

Figure 4.15: A Screenshot of CiteSeerx's candidate document

```
{ u'text':
    u'ABSTRACT
    Semantic Wikipedia
    Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller
    Institute AIFB
    University of Karlsruhe (TH)
    D-76128 Karlsruhe, Germany
    {voelkel,kroetzsch,vrandecic,haller}@aifb.uni-karlsruhe.de
    Wikipedia is the world s largest collaboratively edited source
    of encyclopaedic knowledge. But in spite of its utility, its
    contents are barely machine-interpretable
    ...
    H.3.5 [Information Storage and Retrieval]: Online Information
    Systems; H.5.3 [Information Interfaces]: Group
    and Organization Interfaces-Web-based interactions; I.2.4
    [Artifical Intelligence]: Knowledge Representation; K.4.3
    [Computers and Society]: Organizational Impacts-Computer-supported
    collaborative work
    General Terms
    Human Factors, Documentation, Languages
    Keywords
    Semantic Web, Wikipedia, RDF, Wiki
    1. INTRODUCTION
    This paper describes an extension to be integrated in
    information for agents and other programs is hardly possible
    right now: only complete articles may be read as blobs of
    ...
'}
```

Listing 4.7: A Candidate Document of CiteSeerX's Query Term Semantic Wikipedia

For the CiteSeerx, we uploaded 100 runs for the testing queries and 10 runs for the training queries. We extracted the number of documents that have cited each



My Stats for Site CiteSeerX

Below are the statistics just for you. For general statistics see CiteSeerX.

Short name	citeseerx
Name	CiteSeerX
Url	http://citeseerx.ist.psu.edu/index
#queries	200
#docs	10253 (51.27 documents per query)
#rankings by you (#runs * #queries)	**110**
#impressions for you	**106 (0.53 impressions per query)**
#clicks for you	**4 (0.04 clicks per impression)**

© Copyright 2016, Living Labs.

Figure 4.16: Our Stats for Site CiteSeerX

candidate document in our corpus from CiteSeerx. Nevertheless, there are documents among candidate's lists which don't have popularity data as citation. We considered the amount of null for such documents in the list. Having the citation number and the CiteSeerx ranking, we then re-ranked the candidate documents for each query according to their usage data (citations) and their relevancy position in the CiteSeerx. Our present performance statistics in Open Search is represented in Figure 4.16. At the moment (ten days after uploading the runs) our runs have gained 106 impressions.

Chapter 5

Evaluation

The Text REtrieval Conference (TREC) is a series of workshops centered on different IR research areas or tracks. It is co-sponsored by the National Institute of Standards and Technology (NIST) [1] and the Intelligence Advanced Research Projects Activity (part of the office of the Director of National Intelligence). Its aim is to provide the required infrastructure for large-scale evaluation of text retrieval methodologies and development of new evaluation techniques to support and improve the research within the information retrieval community (both industry and academia) The TREC has run annually since 1992. By hosting annual IR research workshops, TREC has supported the improvement in research and development on IR systems.

Different studies investigated the impact of TREC on IR development and research, e.g. a study by RTI International on commission from NIST [35] presents a variety of qualitative benefits of TREC's activities in economic and none-economic terms. According to the NIST's claim, the effectiveness of retrieval systems approximately doubled within the first six years of the TREC workshops. In 2008, Hal Varian, Chief Economist at Google, declared that TREC's activities of IR research have headed to the creation of new IR evaluation programs internationally. Another research suggests that TREC's existence had a high impact on approximately one-third of an improvement of more than 200% in web search products between 1999 and 2009 [37]. Since TREC-3 in 1994, this conference has begun to recognize more the importance of the user in information seeking of interactive retrieval systems [38].

[1] http://www.nist.gov/

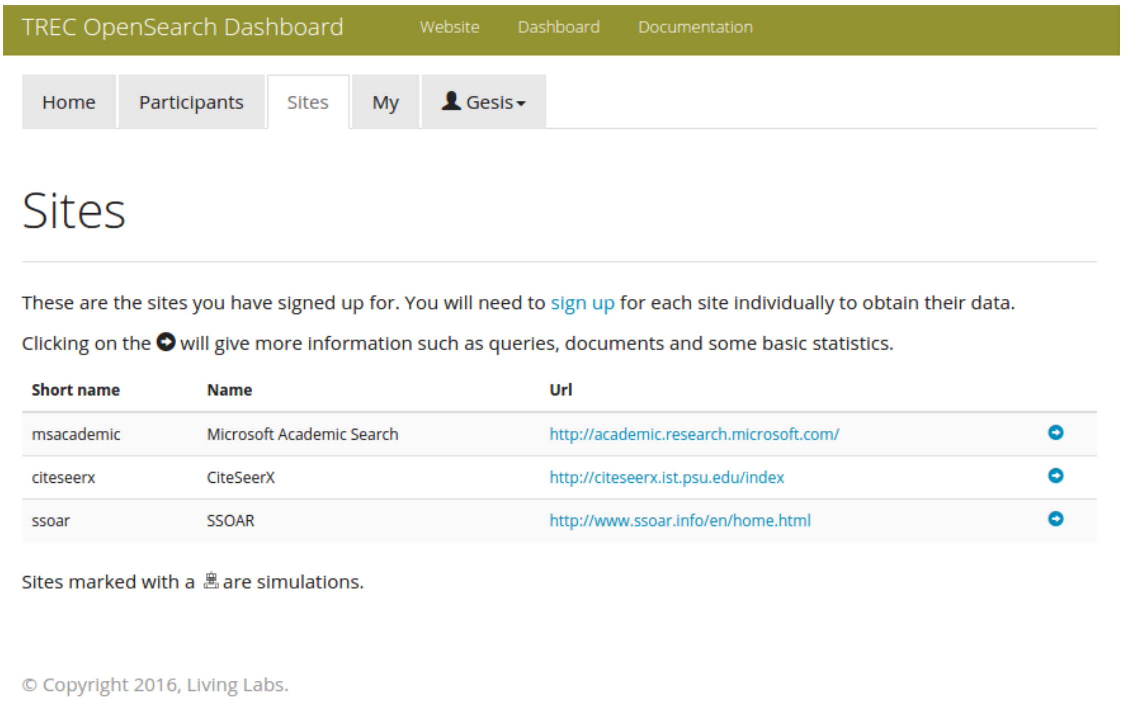
A TREC workshop consists of a set of tracks which are tasks that focuses on a special problem of information retrieval. OpenSearch runs for the first time as a track at TREC in 2016 with the same style and structure as CLEF LL4IR. OpenSearch is defined as a new model for evaluation of IR systems (ad hoc academic search engines) that allows the IR researchers to replace the components of the search engines (ranking) with their own components and observe the user interaction within these components. This helps researchers of IR to evaluate their system in a live setting using the user of search engines.

We have added a living lab functionality to the DSpace platform to suit the lab's needs. Our living lab extension for SSOAR enables search engine developers to obtain feedback about SSOAR end users' acceptance of their rankings.

The extension contains six classes, which can be added to DSpace (overlay and additions) directories of Discovery and Usage packages in a DSpace platform. Accordingly, other DSpace-based repositories are able to add the Living Lab's extension to their repository and participate in a living labs challenge by (1) adding these six classes to the Discovery package of XMLUI extension, (2) creating one table in a PostgreSQL database of DSpace, which is used as a cache mechanism for recording the interleaved result of a particular session, (3) adding the living lab's config file to the config file's position, (4) adding logging configuration in log4j.properties for Living Labs activities, information, and errors, (5) execute the bash file, which first runs a feedback provider to extract the feedback from the living labs log files, it then uploads the feedback to the API (this can be done by a by cronjob), and then finally makes a report of these processes. We have installed the extension on both vanilla DSpace and SSOAR. The whole set is easy to install and adaptable to DSpace-base sites (RQ4).

It is notable that by employing of our application, we were able to involve our use-case, SSOAR, in the living lab evaluation track along with two famous and popular sites (CiteSeerx and Microsoft Academic Search) who also participating in this evaluation paradigm using their own implemented application.

We took part as a participant in the TREC Open Search to (1) evaluate our living lab extension for SSOAR (represented in Section Section 4.5), (2) to evaluate the DSpace ranking system–that can be concluded at the end of the evaluation test phases in 15 November 2016–, and (3) to propose a ranking system that possibly outperforms the actual SSOAR ranking system (in Section 4.6.1).



Sites

These are the sites you have signed up for. You will need to sign up for each site individually to obtain their data.

Clicking on the ➜ will give more information such as queries, documents and some basic statistics.

Short name	Name	Url	
msacademic	Microsoft Academic Search	http://academic.research.microsoft.com/	➜
citeseerx	CiteSeerX	http://citeseerx.ist.psu.edu/index	➜
ssoar	SSOAR	http://www.ssoar.info/en/home.html	➜

Sites marked with a 🤖 are simulations.

© Copyright 2016, Living Labs.

FIGURE 5.1: A Screenshot of TREC Open Search Dashboard Page
http://trec-open-search.org/

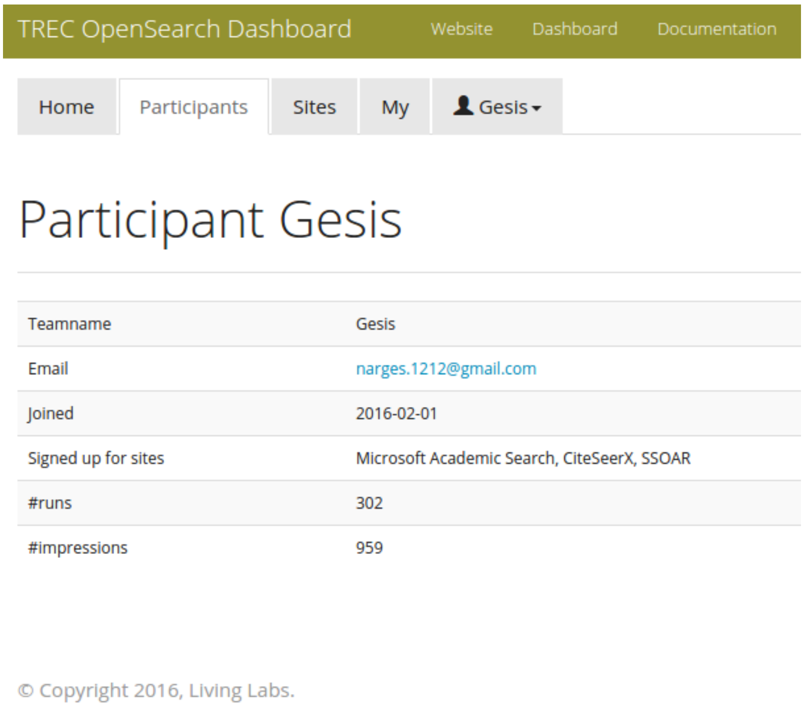
Figure 5.1 is a screenshot of the TREC Open Search website, and presents the sites for which we have signed up (to get the permission to upload our ranking for these sites).

We presented a solution to overcome the lack of high-frequency simple term queries that are required by the living lab. We supplied a part of experimental queries with SSOAR's high-frequency browsing queries and we implemented our method such that it can cover both types of queries.

Through our query frequency analysis and our implementing Collection Provider, we were able to prepare an experimental collection (Figure 4.11) containing 130 queries - the maximum number of 100 candidates' documents for each query (overall, 12,049 documents with the average of 91,98 documents per query).

Through the modification and employment of the Living Labs client, we could communicate with the living lab's API in order to share our collection with the TREC evaluation campaign.

Through the modification and the employment of the Living Labs client, we could communicate with the living lab's API in order to share our collection with the TREC evaluation campaign. The final task was to run the feedback extractor



Participant Gesis

Teamname	Gesis
Email	narges.1212@gmail.com
Joined	2016-02-01
Signed up for sites	Microsoft Academic Search, CiteSeerX, SSOAR
#runs	302
#impressions	959

© Copyright 2016, Living Labs.

FIGURE 5.2: Our Participation Details in TREC Open Search Dashboard Page http://trec-open-search.org/

TABLE 5.1: Gesis Statistics for Site SSOAR

# **Queries**	130
# **Documents**	12050 (91.98 documents per query)
# **Gesis Ranking**	194
# **Impressions for Gesis**	1527 (11.66 impressions per query)
# **Clicks for Gesis**	24 (0.02 clicks per impression)

on the log files generated during the evaluation phases and uploading the generated feedback to the API regularly. To automate this periodic process, we implemented a shell script which runs the module for analyzing and extracting the feedback from the log files, uploading them to the API and documenting a daily report. The shell script executed as a Cron job every 24 hours. Looking at the daily reports, we are able to check if the procedures are operating appropriately.

As a participant, we generated a new ranking system and used the provided collection to create experimental runs. We uploaded the runs via the living lab API. Our extension interleaved the runs with SSOAR ranking and showed them to the end users. We were able to observe our ranking approach feedback produced by SSOAR's users.

Table 5.1 shows the current statistics of our participation as “Gesis”, including the number of runs that we uploaded (194), the number of impressions we got(1527), with an average of 11.66 per query, and the number of clicks, which is 24.

Chapter 6

Conclusions

Living labs for IR enabled the researchers to evaluate their ranking approach in a live environment with real users. It connected sites and researchers via the living lab's API. We showed how we could participate as a researcher in the living labs hosted by CLEF and we reported our first experience with this methodology as IR researchers. We could learn from and improve our ranking system within the lab.

We also presented an implementation of an extension that enables our DSpace-based repository (SSOAR) to participate in the living labs for IR hosted by TREC Open Search as a site. Our Collection Provider generated a dataset of 130 queries and 7,612 documents. We shared this data using the living lab's site client. Using the installed living lab's extension for obtaining the experimental ranking from the API, interleaving, and logging tasks we were able to track and record the user's search activities.

We could extract and share the feedback from the log files using the feedback extractor in order to form the feedback, as well as to use the site client to upload the feedback.

For the purpose of demonstrating our approach's capability, we took part in the TREC Open Search Challenge as an IR researcher and produced a new ranking system based on characteristic scale and score of SSOAR usage data combined with Solr TF-IDF scores.

By employing our DSpace extension we were able to participate in OpenSearch

TREC 2016 with a DSpace-based repository (SSOAR) parallel to two other use-cases (CiteSeerx and Microsoft Academic Search)

The first outcome of the ranking system's evaluation will be announced by TREC Open Search in the middle of July. We expect that this work will help (1) us in further progress to improve our portal ranking system and (2) other sites that use DSpace to join this research cooperation.

To sum up, this contribution is valuable because we were able to employ our approach in an existing application (SSOAR). We are delighted to share the components of our retrieval system with the participating IR researchers and believe that this work will hopefully benefit these researchers to evaluate their ranking algorithms using our users' feedback. Our work facilitates DSpace-based repositories to participate in the living labs for IR evaluation paradigm. It can also help other sites that are interested in participating in a living lab for IR to get a general overview and the idea of the whole structure and the requirements for transforming a site to such an experimental environment. We also expect to gain a more precise assessment of the performance of our own IR system. Furthermore, we hope that this contribution encourages further research which leads to an improved approach.

6.1 Limitations and Future Work

In its current form, we proved that our approach for TREC OpenSearch is useful when we used it on our production system (SSOAR). However, it does have some limitations that could benefit from the attention of researchers and developers.

By employing our approach on a DSpace-based repository (SSOAR), were able to participate in OpenSearch TREC 2016 as a site. We have shown that our extension for DSpace works appropriately. However, the use of this approach is limited to the Living Labs for IR evaluation campaign by CLEF or TREC. Living Labs for other potential components of the sites, like recommendation system or user interface design, need to be implemented separately.

Another limitation is that our approach follows the user's interactions to the level in which the user clicks (views) the items. We could also go further to consider and record the download actions as a stronger evidence of user information satisfaction.

Since we couldn't find enough high-frequency queries, we added browsing queries to the lab. Consequently, we implemented a method that can cover both simple searches and browse searches on the assumption that if SSOAR's number of users grow in the future and we have more traffic and receive more user's queries with higher frequency, we can easily add simple queries as well. As we mentioned before in Section 4.3.1, the types of queries can be clarified in the query site's IDs of the list of experimental queries.

The experiment was limited to the search engine's frequency of enough queries that are listed and written in the specified JSON file. Our JSON list of the experimental queries in SSOAR consists of a mix of simple queries and browsing queries. This file is generated manually. Therefore, all the queries' metadata (qstr, type, and site_qid) should be listed in the JSON file one by one. For future work, it would be recommended to implement a method that produces this list of the standard DSpace log files.

During the training phase, we confronted an unusually long waiting time for our server's response due to the OpenSearch API's unavailability. The search and browsing functionalities were waiting for the experimental rankings, and this caused delays when the API's server was slow or not responding at all. To prevent such a condition we should have reduced the server response time to under 500ms. We implemented a timeout of 500ms and registered in HostTracker [1] to monitor the API server and be notified by email in case the server went down. In such a condition, we deactivated the living lab's mode.

Another type of limitation to consider is the limitation of the TREC Open Search application itself. Living labs for IR has been a means for making the evaluation of retrieval systems through the live search engines more realistic. However, many extensions of this paradigm still deserve further consideration. For instance, it is worthy of further investigation to see whether an alternative to the interleaving method like AB testing improves the reliability and the confidence level of the assessment result of the comparison between ranking algorithms. One way to demonstrate this would be a comparison between the evaluation results of ranking systems using AB testing versus interleaving. One of the disadvantages of AB testing is that it may destroy the user experience with the system by showing him

[1] http://www.host-tracker.com/

a bad ranking. One way to avoid this would be to restrict the number of items in the run to just a small SERP. Another limitation of AB testing is that it is applicable to sites that have high traffic (For a comparison of two rankings, it is required that the two rankings be shown to the users separately, whereas in the interleaving method, two or more runs can be shown to the user at the same time in one ranking result).

Another argument is the interleaving process itself. It is currently the site's task to interleave the site's result with the experimental result, it might cause some ambiguities for the sites or the living lab's organizers, if the site interleaves the rankings as it should. Accordingly, it is essential for each site to check its implementation with the lab's organizer. One proposal would be to make this process unique and to transfer the task to the living lab application instead of the site. It could be feasible for the site to get the already interleaved result from the lab's API. The API could also take the task of filtering the old items - which are not among the actual result because of a reason, such as unavailability - and adding the new (arrived) items. It could be possible that it calculates it ranking and sends it (maybe as a list of IDs to the API) for each query. The API does the filtering and interleaving, and then sends the interleaved ranking to the site in which each item is labeled with the name of its identity (experimental run id).

Bibliography

[1] Social science open access repository (ssoar). http://www.ssoar.info/en/leitlinien.html. Accessed: 2016-06-02.

[2] James Allan. Hard track overview in trec 2003 high accuracy retrieval from documents. Technical report, DTIC Document, 2005. URL http://trec.nist.gov/pubs/trec12/papers/HARD.OVERVIEW.pdf.

[3] Leif Azzopardi and Krisztian Balog. Towards a living lab for information retrieval research and development. In *Multilingual and Multimodal Information Access Evaluation*, pages 26–37. Springer, 2011.

[4] Krisztian Balog, David Elsweiler, Evangelos Kanoulas, Liadh Kelly, and Mark D Smucker. Cikm 2013 workshop on living labs for information retrieval evaluation. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2557–2558. ACM, 2013.

[5] Krisztian Balog, Liadh Kelly, and Anne Schuth. Head first: Living labs for ad-hoc search evaluation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1815–1818. ACM, 2014.

[6] Andrzej Bialecki. Click-through relevance ranking in solr lucid works enterprise, 2011. URL http://www.slideshare.net/lucenerevolution/bialecki-andrzej-clickthrough-relevance-ranking-in-solr-lucid-works-enterprise.

[7] Jamie Callan, James Allan, Charles LA Clarke, Susan Dumais, David A Evans, Mark Sanderson, and ChengXiang Zhai. Meeting of the minds: an information retrieval research agenda. In *ACM SIGIR Forum*, volume 41, pages 25–34. ACM, 2007.

[8] Alberto Díaz, Antonio García, and Pablo Gervás. User-centred versus system-centred evaluation of a personalization system. *Information Processing & Management*, 44(3):1293–1307, 2008.

[9] Lieven Drogmans, Valorie Hollister, and Tim Donohue. Dspace institutional repository platform. 2011. URL http://hdl.handle.net/123456789/7641.

[10] Susan T Dumais and Nicholas J Belkin. The trec interactive tracks: Putting the user into search. *TREC: Experiment and evaluation in information retrieval*, pages 123–152, 2005.

[11] Agathe Gebert. Social science open access repository (ssoar) status, visions and challenges, 2013. URL https://erc.europa.eu/sites/default/files/content/events/Speaker_profiles_abstracts_session_2.pdf.

[12] Wolfgang Glänzel. The application of characteristic scores and scales to the evaluation and ranking of scientific journals. *Journal of Information Science*, 37(1):40–48, 2011.

[13] Trey Grainger and Timothy Potter. *Solr in Action*. Manning Publications, 2014. ISBN 1617291021. URL http://www.amazon.com/Solr-Action-Trey-Grainger/dp/1617291021%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D1617291021.

[14] David Hawking, Paul Thomas, Tom Gedeon, Timothy Jones, and Tom Rowlands. New methods for creating testfiles: Tuning enterprise search with c-test. In *Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation*, pages 5–6, 2009.

[15] Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–48. ACM, 2000.

[16] Thorsten Joachims et al. Evaluating retrieval performance using clickthrough data. pages 79–96, 2003. Text Mining.

[17] Jaap Kamps, Shlomo Geva, Carol Peters, Tetsuya Sakai, Andrew Trotman, and Ellen Voorhees. Report on the sigir 2009 workshop on the future of ir evaluation. In *ACM SIGIR Forum*, volume 43, pages 13–23. ACM, 2009.

[18] Diane Kelly and Cassidy R. Sugimoto. A systematic review of interactive information retrieval evaluation studies, 1967-2006. *Journal of the American Society for Information Science and Technology*, 64(4):745–770, 2013. ISSN 1532-2890. doi: 10.1002/asi.22799. URL http://dx.doi.org/10.1002/asi.22799.

[19] Ron Kohavi. Online controlled experiments: Lessons from running a/b/n tests for 12 years. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1–1, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3664-2. doi: 10.1145/2783258.2785464. URL http://bit.ly/KDD2015Kohavi.

[20] Ron Kohavi, Randal M Henne, and Dan Sommerfield. Practical guide to controlled experiments on the web: listen to your customers not to the hippo. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 959–967. ACM, 2007.

[21] Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M Henne. Controlled experiments on the web: survey and practical guide. *Data mining and knowledge discovery*, 18(1):140–181, 2009.

[22] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[23] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[24] Lynne McKechnie, George R Goodall, Darian Lajoie-Paquette, and Heidi Julien. How human information behaviour researchers use each other's work: a basic citation analysis study. *Information research*, 10(2):10–2, 2005.

[25] Kim Plassmeier, Timo Borst, Christiane Behnert, and Dirk Lewandowski. Evaluating popularity data for relevance ranking in library information systems. In *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*, page 125. American Society for Information Science, 2015.

[26] Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 43–52. ACM, 2008.

[27] Stephen Robertson. Richer theories, richer experiments. In *Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation*, page 4, 2009.

[28] Mark Sanderson. Test collection based evaluation of information retrieval systems. *Foundations and TrendsÂő in Information Retrieval*, 4(4):247–375, 2010. ISSN 1554-0669. doi: 10.1561/1500000009. URL http://dx.doi.org/10.1561/1500000009.

[29] Tefko Saracevic. Effects of inconsistent relevance judgments on information retrieval test results: A historical perspective. *Library Trends*, 56(4):763–783, 2008.

[30] Philipp Schaer and Narges Tavakolpoursaleh. Historical clicks for product search: Gesis at clef ll4ir 2015. 2015. URL https://nbn-resolving.org/urn:nbn:de:0074-1391-8. In Toulouse, France.

[31] Anne Schuth, Krisztian Balog, and Liadh Kelly. Overview of the living labs for information retrieval evaluation (ll4ir) clef lab 2015. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 484–496. Springer, 2015.

[32] Anne Schuth, Katja Hofmann, and Filip Radlinski. Predicting search satisfaction metrics with interleaved comparisons. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 463–472. ACM, 2015.

[33] MacKenzie Smith, Mary Barton, Mick Bass, Margret Branschofsky, Greg McClellan, Dave Stuve, Robert Tansley, and Julie Harford Walker. Dspace: An open source dynamic digital repository. 2003.

[34] Karen Sparck Jones and Cornelis Joost van Rijsbergen. Information retrieval test collections. *Journal of documentation*, 32(1):59–75, 1976.

[35] Gregory Tassey, Brent R Rowe, Dallas W Wood, Albert N Link, and Diglio A Simoni. Economic impact assessment of nistâĂŹs text retrieval conference (trec) program. *National Institute of Standards and Technology, Gaithersburg, Maryland*, 2010. URL www.nist.gov/director/planning/impact_assessment.cfm.

[36] Giannis Tsakonas and Christos Papatheodorou. Exploring usefulness and usability in the evaluation of open access digital libraries. *Information processing & management*, 44(3):1234–1250, 2008.

[37] Ellen M Voorhees, Paul Over, and Ian Soboroff. Building better search engines by measuring search quality. *IT Professional*, 16(2):22–30, 2014.

[38] Ryen W. White. *Interactions with Search Systems*. Cambridge University Press, 2016. ISBN 1107034221. URL http://www.amazon.com/Interactions-Search-Systems-Ryen-White/dp/1107034221%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D1107034221.

[39] Thomas D Wilson. Human information behavior. *Informing science*, 3(2):49–56, 2000.

An Appendix – Source Code

The source code modules presented in this thesis are available at https://github.com/EIS-Bonn/Theses/tree/master/2015/Narges%20Tavakolpoursaleh