promise

PROMISE

Participative Research labOratory for Multimedia and Multilingual Information Systems Evaluation

FP7 ICT 2009.4.3, Intelligent Information Management

Deliverable 4.4
Report on operational systems as experimental platforms

Version 1.0, August 2013

SEVENTH FRAMEWORK PROGRAMME

Network of Excellence co-funded by the 7th Framework Program of the European Commission, grant agreement no. 258191

Document Information

Deliverable number:	4.4
Deliverable title:	Report on operational systems as experimental platforms
Delivery date:	August 2013
Lead contractor for this deliverable	UvA
Author(s):	Martin Braschler, David Graus, Melanie Imhof, Tom Kenter, Mihai Lupu, Hendrike Peetz, Florina Piori, Ork de Rooij, Maarten de Rijke
Participant(s):	UNIPD, UvA, IRF/TUW, ZHAW
Workpackage:	4
Workpackage title:	Evaluation Metrics and Methodologies
Workpackage leader:	UvA
Dissemination Level:	PU – Public
Version:	1.0
Keywords:	Living Labs, evaluation, case studies

History of Versions

Version	Date	Status	Author (Partner)	Description/Approval Level
0.9	12/08/2013	Draft	UvA	Circulated to contributing partners
1.0	31/08/2013	Final	UvA	Took final comments on board

Abstract

This deliverable describes the work done on the use of operational systems as experimental platforms in the context of tasks 4.3 and 4.4 of WP4. It presents a set of methodological principles, a list of practical considerations, as well as a small number of legal and ethical concerns that living lab experimenters should address, before discussing an extensive list of case studies. The case studies highlight different aspects of the potential and limitations of living lab experiments for information access evaluation, both algorithmic, organizational and practical.

Table of Contents

Document Information 3
Abstract 3
Table of Contents 4
Executive Summary 5
1 Introduction 6
2 Methodologies 8
2.1 User panels 8
2.1.1 Black box IR application evaluation 8
2.2 Log analysis 9
2.2.1 Click models 10
2.2.2 A/B testing 10
2.2.3 Interleaving methods 11
3 Practical considerations 12
3.1 Logging 12
3.2 Running a live system 14
3.3 Pitfalls 15
4 Legal and ethical considerations 17
4.1 Content presentation and distribution 17
4.2 Data gathering and retention 17
4.3 Age restrictions 17
5 Case studies 19
5.1 User panel/observational: PatOlympics 19
5.2 User panel/experimental: Guerrilla campaigning 20
5.3 Log analysis/observational: Intent-aware click models 22
5.4 Log analysis/observational: Modeling clicks beyond the first result page 24
5.5 Log analysis/observational: Themestreams 25
5.6 Log analysis/experimental: yourHistory 27
5.7 Log analysis/experimental: Evaluating aggregated search 29
5.8 Log analysis/experimental: Lerot 30
6 Conclusions 33
References 34

Executive Summary

In a living lab experiment one exposes a live system in an uncontrolled manner to end users and observes how they interact with the system, or with alternative versions of the system. The feedback one obtains is often implicit in character and large in volume. Living labs are a relatively underexplored methodology for experimental evaluation in academic information access research. However, the methodology is widely being used in industrial research and development into information access. One very common measurement method is A/B testing, in which different groups of a live system are served by alternative versions of the system in such a way that observed differences in user behavior can be related back to these alternatives. Over the years, a solid understanding has been developed of the strengths and weaknesses of A/B testing in industry.

In this deliverable we start by situating the living lab methodology against the background of other evaluation methodologies: offline evaluation (e.g., in the Cranfield tradition), user centered evaluation, and online evaluation. We primarily think of living labs as an online evaluation methodology.

We also situate the living lab experimental methodology in the landscape of methods for gathering user data. There, we distinguish between user studies, user panels and log analysis: living labs are mostly used and seen as a way of generating log data, often with a very implicit signal, but some types of user panel may also be seen as living lab experiments, as we will see below.

As the living lab methodology is relatively underexplored in academic research, the deliverable complements a number of general lessons with a list of case studies aimed at highlighting three distinct aspects. First, we report on activities in PROMISE that relate to living labs. Second, we illustrate the wide range of types of experimental findings that can be obtained using living labs. And third, we list challenges associated with experiments in a living lab.

Our case studies cover a broad range of evaluation activities. We discuss two types of method that may best be characterized as user panels, one aimed at acquiring observational data, the other meant to support contrastive retrieval experiments. Then, we discuss living lab experiments that are aimed at producing and exploiting log data. Three such studies are observational in character and concern click models, relatively rare search engine use behavior, and visualizations. Three further studies are aimed at learning contrastive lessons, concerning entity linking, aggregated search, and online learning to rank.

1 Introduction

In deliverable D3.4 (Continuous evaluation, M30) we made an important distinction between three types of evaluation in information retrieval: offline evaluation, user studies, and online evaluation. Deliverable 3.4 focused on continuous evaluation methods within each of these three types. In this deliverable, we focus on so-called living laboratories ("living labs") as an operationalization of "operational systems as experimental platforms." Living labs mostly fall under "online evaluation" but in the literature, and in the PROMISE project, the living labs approach is sometimes also used in the setting of user studies.

So what is a living laboratory? The concept of living labs is attributed to Jarmo Suominen.[1] It has been argued that a living lab represents a user-centric research methodology for sensing, prototyping, validating and refining complex solutions in multiple and evolving real life contexts. Nowadays, several living lab descriptions and definitions are available from different sources. In the context of information access and evaluation of systems that cater for information access, living labs are about involving and integrating users within the research process, amongst others by observing them while they engage in natural interactions with a live information access system.

As Azzopardi and Balog [2011] put it, a living lab "would, not only, enable the capture of real interaction and usage data, but also provide a context for testing and evaluating IR models, methods and systems." Kelly et al. [2009] put it this way: "A living laboratory on the Web that brings researchers and searchers together is needed to facilitate ISSS [Information-Seeking Support System] evaluation. Such a lab might contain resources and tools for evaluation as well as infrastructure for collaborative studies. It might also function as a point of contact with those interested in participating in ISSS studies." Indeed, it has been claimed that such living labs could act as points of convergence for a range of disciplines around that shared interest of information access [Pirolli, 2009].

To help understand where in the spectrum of user-based evaluation methods, living labs fit, the following typology of user data comes in handy [Dumais et al., 2011]:

	Observational	Experimental
User studies Controlled interpretation of behavior with detailed instrumentation	In-lab behavior observations	Controlled taks, controlled systems, laboratory studies
User panels In the wild, real-world tasks, probe for detail	Ethnography, field studies, case reports	Diary studies, critical incident surveys
Log analysis No explicit feedback but lots of implicit feedback	Behavioral log analysis	A/B testing, interleaved comparisons

Table 1. Types of user data gathering and research focus.

[1] See http://staffnet.kingston.ac.uk/~ku07009/LivingLabs/PapersAndSlides/Day1RichardEnnals.pdf for an explanation and some of the history regarding the concept of living labs.

Living labs cover the bottom two rows of this table: it is key that users (and experimenters) in a living lab interact with a live system. The table highlights that the purpose of conducting a living lab experiment could be *observational* in nature: to develop a picture of user behavior. It may also be more *experimental* in nature: to determine if one approach is better than another approach.

Living labs have been presented not just as a platform for collaborative research, but also as a platform where users co-create the product, application or service (i.e., users are not just subjects of observation, but also part of the creation). Essentially, the users explore emerging ideas and scenarios *in situ*, outcomes of the evaluation process are then fed back into the design of the product to further enhance their user experience.

While living labs have lots of appeal, offering a range of research opportunities and benefits, the development, implementation and deployment of the approach comes with non-trivial challenges that experimenters need to be aware of and tackle beforehand. Indeed, the aim of this deliverable is three-fold: to report on activities in PROMISE that relate to living labs, to illustrate the wide range of types of experimental findings that can be obtained using living labs, and to list challenges associated with experiments in a living lab.

The deliverable is organized as follows. We outline the main experimental methodologies used in living labs (Section 2) as well as the key challenges just mentioned, both practical (Section 3) and ethical or legal (Section 4), and discuss a large number of case studies of living labs that illustrate both the methodology and the challenges (in Section 5). To enable a proper understanding, each of the case studies will be placed in one of the cells in the bottom two rows in Table 1.

2 Methodologies

In this section we briefly mention the main methodologies used by PROMISE partners as part of their living lab studies. We follow the distinctions made in Table 1 and focus exclusively on the data gathering methods in the last two rows in the table: user panels and log analysis.

2.1 User panels

Within PROMISE we have developed a new type of user panel, on which we focus in the section below.

2.1.1 Black box IR application evaluation

Let us define an *IR application* to consist of an IR system, a specific document collection, an application layer, and a configuration set. While log file analysis, A/B testing and interleaving comparisons all require changes in the applications to be enabled, such as writing a log file or interleaving results, to be evaluated, the black box IR application evaluation can be conducted in any operational application. Moreover, the black box IR application evaluation measures the tester's explicit feedback based on a set of quality criteria. Each criterion (e.g., *Freshness*, *Query Syntax*, *Social Aspects* and *Navigational Queries)* evaluates an aspect of functionality that users experience when using the application. The collectivity of quality criteria estimates the overall user perception. The criteria are based on established best practices for information retrieval applications and have been compiled in a board of use case domain stakeholders. Since the application is treated as a black box the criteria must be testable using only the user interface. This is achieved by a tester who follows the step-by-step instructions in the test script. A list of the quality criteria and their test scripts can be found in [Rietberger et al., 2012].

Black box IR application evaluations aim to be applicable for all IR applications; however, the range of IR applications is wide. For example, there are IR applications used for enterprise search, where the user searches the enterprise website, as well as in the cultural heritage domain, where large digital libraries and archives are searched. These different kinds of IR applications are known as the application's use case domain. The requirements and therefore also the applicability of the methodology depends on the use case domain of the IR application. Therefore each criterion in the list has to be checked for its applicability to each use case domain. For the four PROMISE use case domains we already provided adaptations in the deliverable D4.2 "Evaluation in the Wild" [Rietberger et al., 2012]. In order to ensure the applicability to various domains the generalizability of the methodology was investigated and it was found that only eight out of 43 are not generalizable to the three PROMISE use case domains and the enterprise search domain [Imhof et al., 2013]. Mostly the criteria are not applicable due to differences in the usage. While the PROMISE use case domains (Unlocking culture, Search for innovation, Visual clinical decision support) are mainly used in a professional context, the enterprise search applications are often used by lay users and for fun.

The black box evaluation methodology can be used to monitor a single IR application, to compare IR applications from the same use case domain or in an evaluation campaign, where several IR applications are compared to each other. Therefore, the black box IR application evaluation methodology enables researchers to conduct controlled experiments on an operational system. When monitoring an application, the applications configuration, the IR system, the collection, the user interface and everything else that affect the user's experience can be changed and another evaluation will show the effects.

In our limited testing it was found that the evaluation of an application takes around 2 to 4 hours, since all tests have to be executed manually. In terms of automation of the tests, it was found that most of the tests are hard to automate since intellectual effort is required [Imhof et al., 2013]. For example, some tests ask the tester to formulate queries from a document and others require a distinction between characteristic and not characteristic terms. Another challenge is the handling of different visualizations of the same concept. In the test script of the *Query Term Highlighting* criteria the tester should investigate if the query terms are highlighted in the result list. A human being can answer this question easily since he realizes that bold, italic, colored etc. are all visualizations of the same concept (highlighting).

2.2 Log analysis

One of the ways in which user search behavior may be analyzed is through a transaction log analysis, which over the years has proved an apt method for the characterization of user behavior. Its strengths include its non-intrusive nature—the logs are collected without questioning or otherwise interacting with the user—and the large amounts of data that can be used to generalize over the cumulative actions taken by large numbers of users [Jansen, 2008]. It is important to note that transaction log analysis faces limitations: not all aspects of the search can be monitored by this method, for example, the underlying information need [Rice and Borgman, 1983]. It can also be difficult to compare across transaction log studies of different systems due to system dependencies and varying implementations of analytical methods. Comparability can be improved to some extent by providing clear descriptions of the system under investigation and the variables used [Jansen and Pooch, 2001].

Information science has a long history of transaction log analysis, from early studies of the logs created by users of library online public access catalog systems [Peters, 1993] to later studies of the logs of Web search engines [Jansen and Pooch, 2001]. This was followed by the analysis of more specialized search engines and their transaction logs. For instance, Mishne and de Rijke [2006] study the behavior of users of a blog search engine through a log file analysis and Carman et al. [2009] examine the difference between the vocabularies of queries, social bookmarking tags, and online documents. Three frequently used units of analysis have emerged from the body of work: the session, the query, and the term, though the definition of each unit may vary across studies [Jansen and Pooch, 2001].

As suggested by Table 1, log files are the “raw materials” unearthed by living lab experiments, so to say, that feed into two types of studies: observational studies of user behavior and experimental studies aimed at contrasting alternative ranking or presentation methods. For the former we spent considerable efforts in PROMISE on inferring, and computing, so-called click models; for the latter, we made use of A/B testing and interleaving methods in PROMISE. All three types of approach are briefly explained in the present section and illustrated in case studies in later sections.

2.2.1 Click models

Click data has always been an important source of information for web search engines. It is an *implicit* signal because we do not always understand how user behavior correlates with user satisfaction: user's clicks are biased. Following [Joachims, 2005], who conducted eye-tracking experiments, there was a series of papers that model user behavior using probabilistic graphical models (see [Koller, 2009] for a general introduction). The most influential works in this area include the UBM model by Dupret and Piwowarski [2008], the Cascade Model by Craswell et al. [2008] and the DBN model by Chapelle and Zhang [2009].

A *click model* can be described as follows. When a user submits a query *q* to a search engine she gets back 10 results: u_1, …, u_{10}. Given a query *q* we denote a *session* to be a set of events experienced by the user since issuing the query until abandoning the result page or issuing another query. Note that one session corresponds to exactly one query. The minimal set of random variables used in all click models to describe user behavior are: *examination* of the *k*-th document (E_k) and *click* on the *k*-th document (C_k):

- E_k indicates whether the user looked at the document at rank *k* (hidden variables).
- C_k indicates whether the user clicked on the *k*-th document (observed variables).

In order to define a click model we need to denote dependencies between these variables. For example, for the UBM model we define

$$P(E_k = 1 \mid C_1, \ldots, C_{k-1}) = \gamma_{kd}$$

$$E_k = 0 \Rightarrow C_k = 0$$

$$P(C_k = 1 \mid E_k = 1) = a_{u_k},$$

We refer to [Chuklin et al, 2013c] for further details.

2.2.2 A/B testing

In many realistic search applications the designer of the system wishes to positively influence the behavior of users. We are therefore interested in measuring the change in user behavior when interacting with different ranking algorithms. For example, if users of one system respond more positively, or if some utility gathered from users of one system exceeds utility gathered from users of the other system, then we can conclude that one system is superior to the other, all else being equal.

The real effect of the search engine depends on a variety of factors such as the user's intent (e.g., how specific their information needs are, how much novelty vs. how much risk they are seeking), the user's context (e.g., what items they are already familiar with, how much they trust the system), and the interface through which the search results are presented. Thus, the experiment that provides the strongest evidence as to the true value of the system is an online evaluation, where the system is used by real users that perform real tasks. It is most trustworthy to compare a few systems online, obtaining a ranking of alternatives, rather than absolute numbers that are more difficult to interpret.

For this reason, many real world systems employ an online testing system [Kohavi et al., 2009], where multiple algorithms can be compared. Typically, such systems redirect a small percentage of the traffic to a different alternative search engine, and record the users' interactions with the different systems. There are a few considerations that must be made when running such tests. For example, it is important to sample (redirect) users randomly, so that the comparisons between alternatives are fair. It is also important to single out the different aspects of the search engines. For example, if we care about algorithmic accuracy, it is important to keep the user interface fixed. On the other hand, if we wish to focus on a better user interface, it is best to keep the underlying algorithm fixed.

2.2.3 Interleaving methods

As explained in D3.4, in interleaving comparison methods rankers are assessed using implicit feedback from actual users, such as click behavior, touch behavior, query reformulations, etc. A common approach is to use interleaved comparison methods [Chapelle et al., 2013; Chuklin et al., 2013b; Hofmann et al., 2013b; Joachims, 2003], in which the document lists proposed by two candidate rankers for a given query are interleaved and the resulting list presented to the user, whose clicks are used to infer a noisy preference for one ranker over the other. Recently, interleaving methods have been successfully applied in large-scale settings [Chapelle et al., 2013; Chuklin et al., 2013b]. In comparison to absolute click metrics typically used in A/B testing, interleaved comparison methods reduce variance (briefly, this is because they perform within-subject as opposed to between-subject comparisons), and make different assumptions about how clicks should be interpreted (as relative, as opposed to absolute feedback).

Until recently, it was not clear how interleaved comparison methods could reuse historical data. However, the recently developed probabilistic interleave method bridges this gap [Hofmann et al., 2011; 2013c]. Probabilistic interleave is based on a probabilistic interpretation of interleaved comparisons, which allows it to infer comparison outcomes using data from arbitrary result lists, even if they were obtained in comparisons of rankers different from the current target rankers. In probabilistic interleave, the interleaved document list is constructed, not from fixed lists but from softmax functions that depend on the query. The use of softmax functions ensures that every document has a non-zero probability of being selected by each ranker. As a result, the distribution of credit accumulated for clicks is smoothed, based on the relative rank of the document in the original result lists.

3 Practical considerations

While the idea of running living lab experiments, i.e., using an operational system and its actual users, sounds attractive, there are some practical concerns that should be considered by experimenters before plunging into the world of living labs. Below we list issues that we ran into through the case studies considered within PROMISE; we also describe how we addressed them.

3.1 Logging

Running a living lab experiment is all about collecting data, and especially about collecting user data. As reported in D3.4, while not a formal deliverable of the PROMISE project, a logging facility was created to collect behavioral data in a central location. The design of the logging facility is based on the requirements presented by a small number of case studies (on which we report in Section 5 below).

Briefly, the owner of a live system creates a new project in the logging service. The logging service generates a project specific code that will be used to send events from the live system to the logging system. The owner of the live system has access to a REST API to publish events to the logging service and to submit queries to data already collected. The logging service also comes with a Javascript library to simplify logging of user actions from within web applications.

The architecture of the logging service is summarized in Figure 1 below.

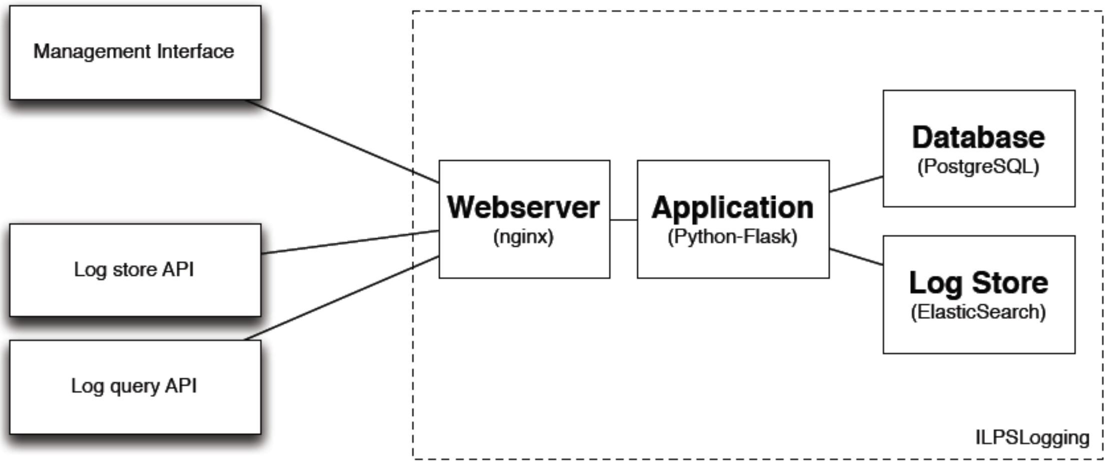
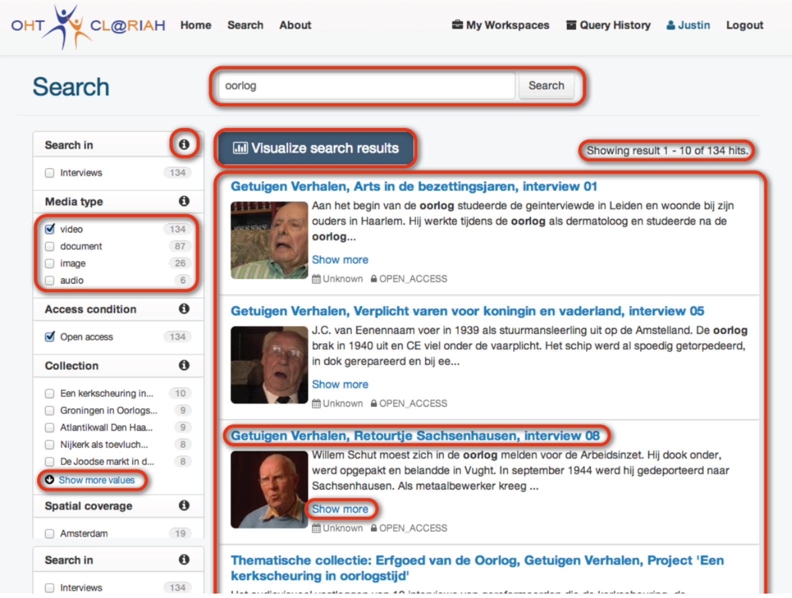


Figure 1. Architecture of logging system for capturing behavioural data.

The logging system is sufficiently flexible to allow every possible aspect of an interaction with an interface to be logged. To this end, templates have been created to cater for the most common log patterns. Figure 2 below illustrates the principle using the interface of a search engine for a multimedia archive. The areas in red are the typical areas for which actions are captured: the search box, facets, collection choices, follow-through actions, etc.



Getuigen Verhalen, Arts in de bezettingsjaren, interview 01

Aan het begin van de **oorlog** studeerde de geinterviewde in Leiden en woonde bij zijn ouders in Haarlem. Hij werkte tijdens de **oorlog** als dermatoloog en studeerde na de **oorlog**...

Show more

Unknown OPEN_ACCESS

Getuigen Verhalen, Verplicht varen voor koningin en vaderland, interview 05

J.C. van Eenennaam voer in 1939 als stuurmansleerling uit op de Amstelland. De **oorlog** brak in 1940 uit en CE viel onder de vaarplicht. Het schip werd al spoedig getorpedeerd, in dok gerepareerd en bij ee...

Show more

Unknown OPEN_ACCESS

Getuigen Verhalen, Retourtje Sachsenhausen, interview 08

Willem Schut moest zich in de **oorlog** melden voor de Arbeidsinzet. Hij dook onder, werd opgepakt en belandde in Vught. In september 1944 werd hij gedeporteerd naar Sachsenhausen. Als metaalbewerker kreeg ...

Show more

Unknown OPEN_ACCESS

Thematische collectie: Erfgoed van de Oorlog, Getuigen Verhalen, Project 'Een kerkscheuring in oorlogstijd'

Figure 2. Examples of screen areas and actions captured by the logging service.

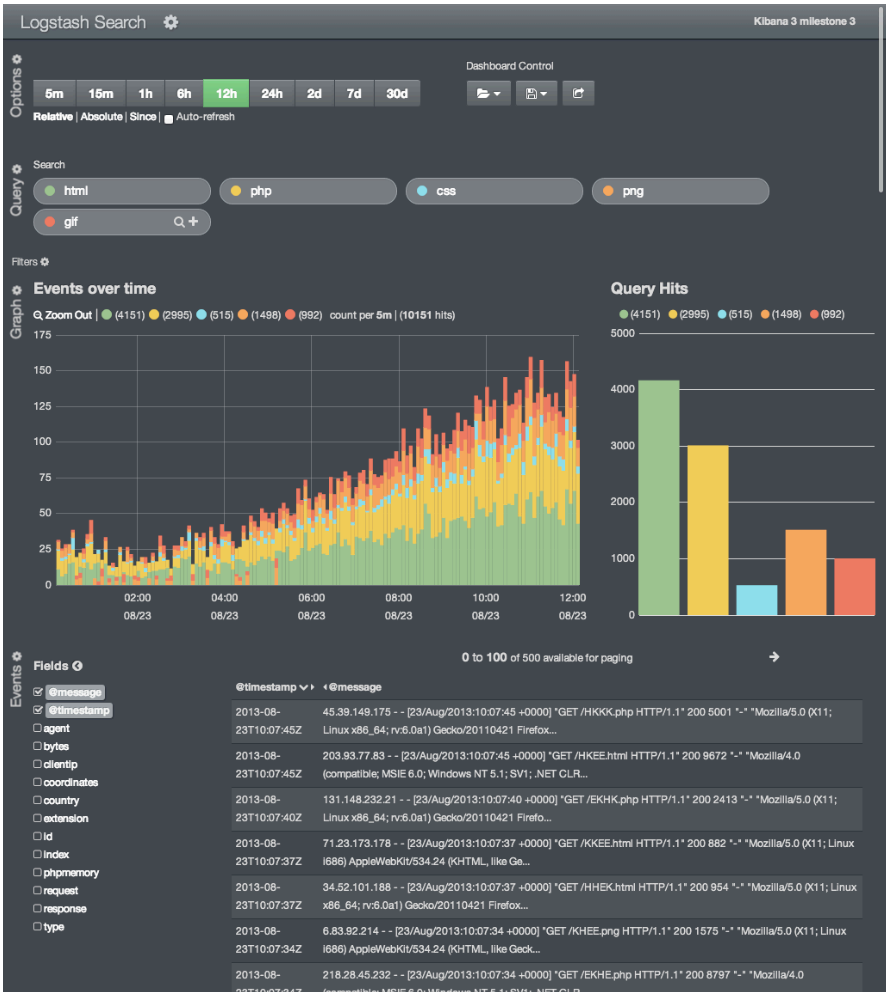
Setting up the Javascript library is rather straightforward. Figure 3 shows an example of code to be included in pages being served.

```
<head>
    <script src="http://logging.ilps.science.uva.nl/static/js/
                 ilpslogging.min.js">
    </script>

    <script>
        var log_options = {
            use_visitor_cookie: true,
            log_current_location: true,
            log_screen_resolution: true,
            log_mouse_movements: true,
            maximal_mouse_sample_freq_ms: 40,
            log_mouse_clicks: true,
            post_events_queue _interval_ms: 3000
        };

        var logging = ILPSLogging('logging.ilps.science.uva.nl',
                                  '7BMgDw1vKQ5VaSBcssCffXO9j9lxaFrZ4h4HbU',
                                  log_options);
        logging.start();
    </script>
</head>
```

Figure 3. Setting the Javascript library at the server end.

Each (user) interaction is an event, which is captured as a JSON object, stored as a document in ElasticSearch, with its own document type. The root level structure is predefined, with templates available for common events, while experimenters can also define their own events. See [de Goede and van Wees, 2013] for further details.

For analysis, it is non-trivial to develop a general purpose interface that will cater for all types of log analysis. We use a general visualization tool, Kibana,[2] to gain general insights in the stream of logs. See Figure 4 for a sample screen shot.



0 to 100 of 500 available for paging

```
@timestamp           @message
2013-08-             45.39.149.175 - - [23/Aug/2013:10:07:45 +0000] "GET /HKKK.php HTTP/1.1" 200 5001 "-" "Mozilla/5.0 (X11;
23T10:07:45Z         Linux x86_64; rv:6.0a1) Gecko/20110421 Firefox...
2013-08-             203.93.77.83 - - [23/Aug/2013:10:07:45 +0000] "GET /HKEE.html HTTP/1.1" 200 9672 "-" "Mozilla/4.0
23T10:07:45Z         (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR...
2013-08-             131.148.232.21 - - [23/Aug/2013:10:07:40 +0000] "GET /EKHK.php HTTP/1.1" 200 2413 "-" "Mozilla/5.0 (X11;
23T10:07:40Z         Linux x86_64; rv:6.0a1) Gecko/20110421 Firefo...
2013-08-             71.23.173.178 - - [23/Aug/2013:10:07:37 +0000] "GET /KKEE.html HTTP/1.1" 200 882 "-" "Mozilla/5.0 (X11; Linux
23T10:07:37Z         i686) AppleWebKit/534.24 (KHTML, like Ge...
2013-08-             34.52.101.188 - - [23/Aug/2013:10:07:37 +0000] "GET /HHEK.html HTTP/1.1" 200 954 "-" "Mozilla/5.0 (X11; Linux
23T10:07:37Z         x86_64; rv:6.0a1) Gecko/20110421 Firefox...
2013-08-             6.83.92.214 - - [23/Aug/2013:10:07:34 +0000] "GET /KHEE.png HTTP/1.1" 200 1575 "-" "Mozilla/5.0 (X11; Linux
23T10:07:34Z         i686) AppleWebKit/534.24 (KHTML, like Geck...
2013-08-             218.28.45.232 - - [23/Aug/2013:10:07:34 +0000] "GET /EKHE.php HTTP/1.1" 200 8797 "-" "Mozilla/4.0
```

Figure 4. Using Kibana for log analysis in the logging service.

3.2 Running a live system

Running and maintaining a live system is not a trivial matter for an academic research group. We list some key dimensions, mostly technical, but also organizational that need to be addressed.

User involvement is one of the key elements of a living lab, and as such should be a focal point of mature living labs. In creating usable systems it is generally accepted that they should be designed according to an iterative approach, and that user involvement is crucial. The focus is on finding out what the relevant experiences, methods, tools that living labs benefit from are. Users are important to define context-aware services, think for example of

[2] http://www.elasticsearch.org/overview/kibana/

cultural differences. Organizational issues include questions like: How to organize user involvement? How to find the right users? What about the validity? How to motivate the users? From a technological point of view: How to get access to large user groups? How to analyze large amounts of data? We are used to analyzing transaction log data in the information retrieval community, but it's not clear a priori which additional data may relevant. Therefore, new analysis and reporting modules might be needed along with scalable, flexible storage and computing resources to cope with large data volumes; see [Schumacher, 2009].

Infrastructure. Within this context, a simple definition of infrastructure can be given as the basic facilities, services, and installations, or underlying framework or features required for the operation of a living lab. There is a natural tension between infrastructure for the "live system" at the core of the living lab (the "production system") and developmental versions of the system in which researchers tune, refine and add alternative methods. Certainly in an academic environment, where infrastructural resources may be constrained, there is a natural tendency to forget about a strict separation between production and test environments—a tendency that we would like to warn against.

Organization and governance. The governance structure of a living lab describes the way it is organized and managed at different levels such as the operational, design, technical and strategic ones. The operational level deals with questions such as maintenance and support and "Who will restart the server after a crash on Saturday evening." The design level (dealing with the interface and interaction design) is often ignored when the focus is mostly on technology development and testing; but users have become used to very high levels of sophistication offered through free online services. The technical level deals with the developmental versions and testing aspects of the system at the core of the living lab. The strategic level deals with issues like: legal issues, ethical issues, and the possible exploitation of results gathered through a production system.

3.3 Pitfalls

The various methodologies listed above, especially A/B testing and interleaved comparisons, are at different stages of theoretical development. Interleaved comparisons are relatively young and less developed than the A/B testing methodology. For instance, interleaved comparison methods able to deal with multiple verticals have only recently been introduced. And while some progress has been made on re-using historical data in the setting of interleaved comparisons, using importance sampling, we still need to realize significant improvements in efficiency of such methods, in terms of the number of impressions needed for convergence. Hence, depending on the research goals, a living lab based on interleaved comparisons may not be an option yet.

While the theoretical aspects of A/B testing have been well studied and documented, the practical aspects of running them in online settings, such as web sites and services, are still being developed. As the usage of A/B testing grows in these online settings, it is becoming more important to understand the opportunities and pitfalls one might face when using them in practice. A survey of A/B testing in the context of the web and lessons learned was extensively documented in *Controlled Experiments on the Web: Survey and Practical Guide* [Kohavi, et al., 2009]. Various pitfalls were identified, such as assuming that common statistical formulas used to calculate standard deviation and statistical power can be applied and ignoring robots in analysis (a problem unique to online settings). Online experiments allow for techniques like gradual ramp-up of treatments to avoid the possibility of exposing many customers to a bad (e.g., buggy) Treatment. With that ability, it was discovered that it is easy to incorrectly identify the winning Treatment; see [Crook et al., 2009] for a detailed and colorful discussion.

But even when either an interleaved comparison or A/B testing methodology may be available and suitable to in principle realize a researcher's experimental goals, there may be reasons to forego a full-blown living lab experiment. One important example that we encountered in the PROMISE project has to do with a mixture of three underlying factors: human, legal and organizational. Let us explain and illustrate this using the CLEF-IP task setting.

The retrieval methods that resulted following the organization of the CLEF-IP tasks in recent years, are novel in an environment where intellectual property (IP) experts use boolean queries (often enhanced with sophisticated word proximity operators). Measuring their effectiveness in live patent retrieval systems by, for example, specifically designed user studies on modified live search systems, is, however, not possible. The main reason for this is that professional patent search/retrieval systems are behind pay walls, like for example Thomson Reuter's Patent Search Systems [Thomson, 2013], developed by privately held companies that are reluctant to open their retrieval systems for such experiments. A second reason for the impracticability of using live retrieval systems to evaluate new retrieval components in a controlled environment for patent retrieval is that such experiments involving a third party—researchers—is seen as a possible breach of confidentiality agreements. Which prior art is searched for by an IP expert at a company or patent office may be valuable information for a competitor, and is usually kept under restricted access policies. Another reason for the impossibility to test new patent retrieval methods using living laboratories is that the new methods are not (yet) trusted. More specifically: boolean search is understood and trusted by patent experts due to its simplicity and—more importantly—its reproducibility of search results at later times. Modern (statistical) retrieval methods are less trusted to reproduce search results at a later time, when the same queries are used.

4 Legal and ethical considerations

When running a living lab experiment one typically runs a public-facing service. Because of this, there are obvious ethical and/or legal dimensions that should be addressed. We do not offer generic solutions in this section, but merely draw the experimenter's attention to these dimensions.[3]

If an academic researcher runs a living lab experiment inside an external non-academic organization, we advise that he/she interacts with the organization's legal department to address the dimensions below: content presentation and distribution, data gathering and retention, age restrictions. If the living lab is being set up inside an academic organization, it is essential to obtain approval from the organization's ethical review board (ERB); given the novel nature of living labs as an experimental methodology (and given the fact that not all computer science departments actually have an ERB, certainly not in Europe), we advise that approval is sought as early as possible in the planning of the experimental trajectory—delays are far more likely in an academic setting than in a non-academic organization that is used to offering public-facing services.

4.1 Content presentation and distribution

Content used for living lab experiments is often made available by third parties through an API. Think, e.g., of tweets made available by Twitter or news made available by the New York Times. Typically, it is not permitted to reproduce this content verbatim in an interface or to (re)distribute it. This may have obvious consequences for the design of interfaces and the functionality of the system at the heart of a living lab experiment.

But there is more. For academic research projects where reproducibility is a key dimension, this can be an important issue; if content cannot be redistributed to fellow researchers, a frequent solution is to make sure the relevant content is retained for a sufficiently long period so that fellow researchers can come and "visit the content."

4.2 Data gathering and retention

For A/B testing so-called cookies are often used to distinguish between the "control" en "treatment" groups. Depending on the country of the experimenter, or sometimes of the host of the living lab, user permission needs to be requested.

It is good practice to ask user permission if and when data needs to be gathered—for instance, for reading and storing a user's timeline on Facebook or for sharing this information with fellow researchers. If permission to gather and store user data is granted, it is also good practice to let the user indicate a data retention period (e.g., one day, one month, one year, indefinitely).

4.3 Age restrictions

Within an academic setting obtaining permission for working with human subjects younger than 18 years is rarely trivial. On some platforms, age restrictions are relatively easy to

[3] Disclaimer: the comments and suggestions below should not be construed as legal advice.

impose, e.g., on Facebook, through the relevant API calls. In other cases, the data being gathered is sufficiently generic, such as clicks, age does not play a direct role. We advise academic experimenters to think this through prior to submitting a formal request to their ERB.

5 Case studies

In this section we summarize case studies in the use of operational systems as experimental platforms. We have labeled and grouped the studies using the terminology introduced in Table 1 and highlight studies based on user panels and those based on log analysis as their data gathering method, either with an observational or an experimental overall aim.

5.1 User panel/observational: PatOlympics

As an alternative to conducting scaled user-based patent retrieval experiments in living laboratories we have organized an event that brought together patent experts and information retrieval experts: PatOlympics.

Generally, Intellectual Property (IP) experts and Information Retrieval (IR) specialists do not meet, as the conference events of one community do not address the needs of the other. IR conferences are too technical for an IP expert, while IP events are centered on higher level needs of groups of patent users, usually of no interest for IR researchers. PatOlympics was organized as an event where the two communities could directly interact and understand each other's needs. The feedback given on PatOlympics was extremely positive, both kinds of event participants (IR and IP community members) declaring that the meeting was very useful in understanding 'the other side'.

PatOlympics were organized in 2010 and 2011. The main idea behind them was to allow professional IP searchers to test various IR systems participating in the event, on a particular request for information [Lupu 2011]. PatOlympics was a competition with two patsports: CrossLingual Retrieval and ChemAthlon. CrossLingual Retrieval targeted those IR systems that answered queries in one language with relevant documents in other languages. The ChemAthlon patsport involved systems that were specialized in searching chemical compounds. The queries and the data to search for in the ChemAthlon competition were always in English. From the organizational point of view, the two patsports are similar.

In PatOlympics, each participating IP expert had his or her request for information and worked together with the IR team demoing a retrieving system for 20-25 minutes to find answers to the information request. All IR teams had to work with the same initial data sets, which were distributed well ahead of the time that the event took place, to allow participants to index and process the patent collections. The data collections distributed were the CLEF-IP collection, for the CrossLanguage Retrieval task, and the TREC-CHEM collection, for the ChemAthlon task. The retrieval results obtained in the 20 minutes sessions of IP expert-IR team collaboration to answer information requests were submitted to the PatOlympics infrastructure, which computed and displayed scores in real time. After 5–6 rounds, where patent experts moved from one tested IR system to test another, the competitions closed and the final winners were displayed on a scoreboard. More details on the infrastructure employed in the PatOlympics events can be found in [Lupu 2011].

The metrics computed in the PatOlympics competition were Precision at 200 retrieved documents (the maximum number of documents that IR teams were allowed to submit). This measure—displayed on the scoreboard as 'Number of documents retrieved'—was chosen because we aimed to have an easy to understand measure for everybody in the room. In 2012, PatOlympics was organized not as a competition, but as a study on the use of search systems specialized on patent data. PatOlympics 2012 was organized as a demo session at IRFC 2012 where 7 systems doing patent search were demoed. Conference and demo-session participants were allowed to use the systems and user-system interactions were logged.

In sum, evaluation campaigns organized in the frame of NTCIR, CLEF or TREC have little impact on patent professionals. One can argue that this is a result of the researchers' incapacity of communicating their results to the IP communities. It is, however, hard for researchers to get IP experts to take a look at a novel retrieval system, and—when they do—it is often discovered that features were already available in commercial systems, to which academics do not have access. The PatOlympics experiment provided an environment for patent experts and information retrieval researchers to understand the needs and work of the each other, the event being a highlight of the IRFS 2010 and 2011, as well as of the IRFC 2012 conference.

5.2 User panel/experimental: Guerrilla campaigning

As a validation of the black box evaluation methodology inside PROMISE, a campaign was conducted where each participating PROMISE partner was asked to identify ten target sites that they would evaluate. In terms of Table 1, this is an example of "user panels" with an experimental focus. The sites were required to fit in the PROMISE use case domains and/or belong to well-known or economically strong organizations (implicit "enterprise search" use case). Partners were provided with test scripts and an accompanying scoring sheet.

It was found that the overall score of each application correlates with the testers' subjective impressions. A correlation coefficient of 0.53 in the range [-1, 1] was achieved. This result should not be over-interpreted at this point, but taken as an indication that the scores should be useful for their intended purpose. Figure 5 shows the overall results of the campaign as a boxplot.

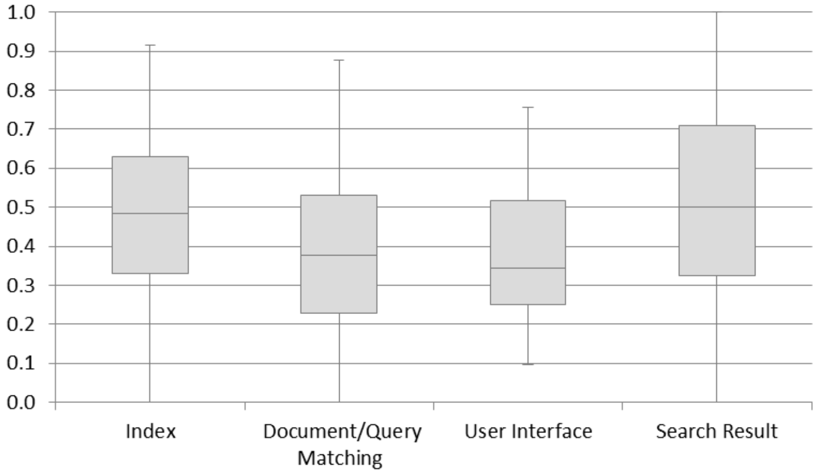


Figure 5: Overall results from guerrilla campaign

The following table (Table 2) provides a more detailed view on the campaign results. The tests were divided into three categories: (i) good results, where most of the sites passed the test, (ii) poor results, where most of the sites failed the test and (iii) neutral results, where no general tendency was found.

Good results	Neutral results	Poor results
- Completeness - Phrasal Queries - Performance/Responsiveness - Browsing - Known Item Retrieval - Diversity	- Office Document Handling - Separation of Actual Content and Representations - Special Characters - Duplicate Documents - Metadata Quality - Tokenization - Named entities - Query Syntax - Over- and Under-Specified Queries - Cross-Language IR - Exception Handling - Result List Presentation - Entertainment - Localization - Facets - Sorting of Result List - Justification of Results - Navigational Queries	- Freshness - Synonyms - Stemming - Feedback - Multimedia - User Guidance - Personalization - Social Aspects - Result List Import/Export - Monitoring - System Override - Related Content - Context Information - Navigational Aids - Mobile Access - Geo-Location

Table 2: Criteria Results

As can be seen, the result is decidedly mixed, although it should be noted that in specific cases the operator of the application would probably assign different weights to the criteria (however, please also note, that as pointed out, the overall scores with the coarse equal weighting employed so far showed a good correlation with the tester's subjective impressions—see the preceding discussion).

Below, we describe our preliminary conclusions of the detailed test results. The quality of an IR application can easily be improved by regular updates of the index to ensure its freshness. Also stemming should be applied to ensure that the user's information need can be fulfilled even though the user does not know the appropriate word form.

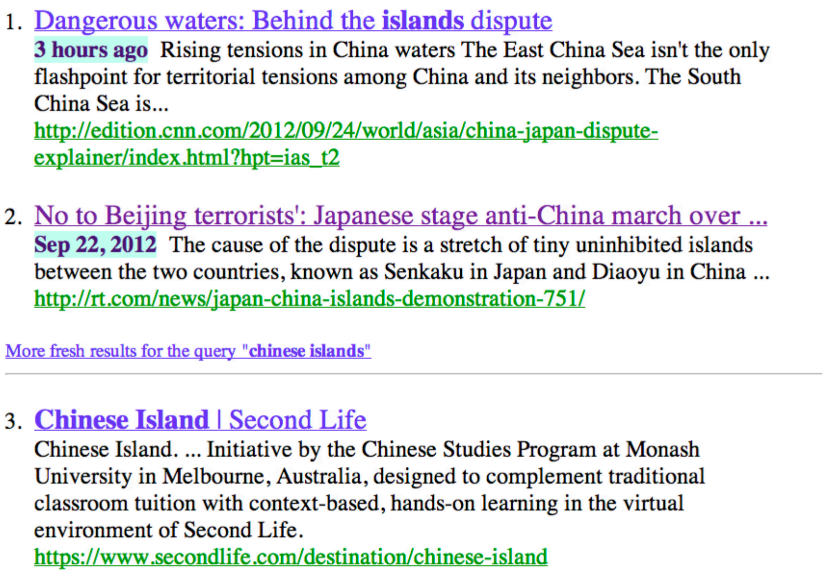
In the document matching category we realized that it is rarely the case that the user is enabled to give feedback about the search results. Furthermore, multimedia retrieval is not yet widely applied. Depending on the use case domain, this may not be an actual requirement, however. To guide the user, term suggestions or spell checking algorithms should be included. Users also greatly benefit from other users when social aspects such as sharing results are available. This can additionally be used to suggest related content. Some advanced search functionalities such as result list exports, monitoring and system overrides, are only sporadically implemented in the evaluated applications. The search results category's tests were passed quite successfully. However, considering geo-locations for the search could further improve the retrieval results for some use case domains.

Furthermore, we can note that currently known practices, as described in D2.3 Best Practices Report [Braschler et al. 2012], are able to achieve high scores as shown by the fact that the maximal score was above 0.75 in all the categories. However, most of the applications exhibit poor results for the following tests: Freshness, Stemming, Multimedia, Related Content, Separation of Actual Content and Representation, Tokenization, Cross-Language Information Retrieval, Result List Presentation, and Sorting of Result List.

5.3 Log analysis/observational: Intent-aware click models

The idea of search result diversification appeared several years ago in the work by Radlinski and Dumais [2006]. Since then all major commercial search engines addressed the problem of ambiguous queries either by a technique called federated/vertical search (see, e.g., [Arguello et al., 2009]) or by making result diversification a part of the ranking process [Agrawal et al., 2009; Styskin et al., 2011].

In PROMISE we have been particularly interested in one particular vertical: fresh results, i.e., recently published webpages (news, blogs, etc.). Figure 6 shows part of a search engine result page (SERP) in which fresh results are mixed with ordinary results in response to the query “Chinese islands”. We say that every document has a presentation type, in our example “fresh” (the first two documents in the figure) or “web” (the third, an ordinary search result item). We refer to the list of presentation types for the current result page as a layout. We assume that each query has a number of categories or intents associated with it. In our case these will be “fresh” and “web.”



1. Dangerous waters: Behind the **islands** dispute
   3 hours ago Rising tensions in China waters The East China Sea isn't the only flashpoint for territorial tensions among China and its neighbors. The South China Sea is...
   http://edition.cnn.com/2012/09/24/world/asia/china-japan-dispute-explainer/index.html?hpt=ias_t2

2. No to Beijing terrorists': Japanese stage anti-China march over ...
   Sep 22, 2012 The cause of the dispute is a stretch of tiny uninhibited islands between the two countries, known as Senkaku in Japan and Diaoyu in China ...
   http://rt.com/news/japan-china-islands-demonstration-751/

More fresh results for the query "**chinese islands**"

3. **Chinese Island** | Second Life
   Chinese Island. ... Initiative by the Chinese Studies Program at Monash University in Melbourne, Australia, designed to complement traditional classroom tuition with context-based, hands-on learning in the virtual environment of Second Life.
   https://www.secondlife.com/destination/chinese-island

Figure 6. Group of fresh results at the top followed by an ordinary search result item.

The main problem that we aimed to address in this observational study in PROMISE related to the topic of intent-aware retrieval is the problem of modeling user behavior in the presence of vertical results. In order to better understand user behavior in a multi-intent environment we proposed to exploit intent and layout information in a click model so as to improve its performance. Unlike previous click models our proposed model uses additional information that is already available to search engines. We assume that the system already knows the probability distribution of intents/categories corresponding to the query. This is a typical setup for the TREC diversity track [Clarke et al., 2011] as well as for commercial search systems. We also know the presentation type of each document. We argue that this presentation may lead to some sort of bias in user behavior and taking it into account may improve the click model's performance. The main questions, then, that we sought to answer through the living lab methodology were

- How do intent and layout information help in building click models?
- How does the performance change when we use only one type of information or both of them?
- How does the best variation of our model compare to other existing click models?

In order to test our ideas and answer our research questions, we were allowed to use a click log of the Yandex search engine and then used the Expectation-Maximization algorithm to infer model parameters; see [Chuklin et al., 2013c]. For our main experiment we used a sample of sessions with fresh results from a period of 30 days in July 2012. We discarded sessions with no clicks and did not take into account clicks on positions lower than ten. Fresh results were also counted and could appear at any position. We had 14,969,116 sessions with 2,978,309 different queries.

This living lab experiment allowed us to arrive at novel observations about vertical search, observations that could subsequently be consolidated in a novel framework of intent-aware click models, which incorporates both layout and intent information. Our intent-aware modifications can be applied to any click model to improve its perplexity. One interesting feature of an intent aware click model is that it allows us to infer separate degrees of relevance for different intents from clicks. These degrees of relevance can be further used as features for specific vertical ranking formulas. Another important property of intent-aware additions to click models is that by analyzing examination probabilities we can see how user patience depends on his/her intent and SERP layout. Put differently, it allows us to use a click model as an ad-hoc analytic tool.

Apart from these concrete scientific results facilitated by the living lab experiment in this case study, a valuable lesson learned concerns data cleaning and data collection. Selecting a sample from a log file is a non-trivial task; external factors beyond control of the experiments (e.g., a world cup football match) may cause unusual behavioral patterns to be reflected in the logs. Thus, multiple samples usually need to be created to ensure that no outlier phenomena are mistaken for regular phenomena. This requirement may come with non-trivial demands on the resources to be made available within the experimenter's environment.

5.4 Log analysis/observational: Modeling clicks beyond the first result page

In web search, many of the ranking functions, and, hence, many of the metrics and evaluation settings focus on the first result page, i.e., the first ten items. The next ten search results are usually available in one click. These documents either replace the current result page or are appended to the end. Hence, in order to examine more documents than the first 10 the user needs to explicitly express her intention. Although click-through numbers are lower for documents on the second and later result pages, they still represent a noticeable amount of traffic [Chuklin et al., 2013b].

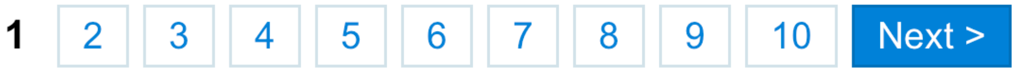


Figure 7. Result page switching (pagination) buttons.

Figure 7 shows an example of such buttons. There, the user can switch to the next result page either by using the page number (e.g., "2") or by clicking the "Next" button. In one particular case study carried out within the PROMISE project, we have a similar setup in our experiments. By analyzing the click log of the Yandex search engine we learned that one third of all users uses the pagination buttons at least once a week. At the query level, with probability 5–10%, a user will go to the second result page. This number is even bigger for further result pages—once she has switched to the second page, a user often continues to the third and fourth pages, and this probability is at least five times bigger than the probability of switching from the first to the second page. On average, our users examine 1.1 pages. These facts suggest that we need to pay more attention to the ranking of documents below the first result page—such documents have a non-trivial click pattern and are examined by a substantial number of users.

The scientific advances facilitated by this case study include the introduction of new click models on top of the widely used dynamic Bayesian network model and showing that by explicitly adding pagination buttons into a click model we can achieve better results in predicting clicks beyond the first result page. As an immediate application of the click models we can follow the procedure outlined in [Chuklin et al, 2013d] to build a more accurate evaluation metric; see [Chuklin et al., 2013b]. The click models have been implemented, with the code available as open source through the Bitbucket platform.[4]

A valuable, more methodological lesson learned concerns the size of the data collected: to be able to study relatively rare phenomena such as next result page visits, a significant amount of data needs to be collected; in a live system, there may be regular as well as unannounced changes; these may cause undesired side-effects in the data. To minimize the chance of observing unwanted effects, we found that it is useful to collect as much data in as short a period of time as possible and to extend the data collection as much as possible.

5.5 Log analysis/observational: Themestreams

Our next example concerns a living lab experiment aimed at understanding how humanities researchers want to interact with large streams of social media data. Specifically, we set up a living lab experiment to test alternative views of aggregates of tweets of stakeholders in the setting of political discourse analysis: Who brings in a topic for discussion in the public debate? Who owns it? Etc. Over the past couple of years, politics and politicians have discovered social media as important means for communicating with voters and for influencing public opinion. Keeping track of the many discussion forums and other outlets is no trivial matter. Typical politically relevant themes include: the economy, healthcare, defense, foreign policy. According to a leading communication agency, during recent national elections in The Netherlands discussions revolved around approximately 500 issues, with differing levels and patterns of attention.

The participants of political discussions can often be mapped to a select number of so-called influencer groups. Specifically, one can identify the following four groups. First, there are those who currently have an (important) position within the governing body, the politicians. Second, there are those who lobby for (specific) important issues, the lobbyists. Third, there are journalists who specialize in politics as well as other high profile media influencers such as television stars or columnists. Fourth and finally, all other people taking part in political discussions we group together as the rest: the public. In this living lab, our technological aim was two-fold. First, to test the responsiveness of the interface under natural interactions, especially of the language and search technology that feeds the interface. Second, to contrast the usefulness of two types of summarizing online discussions. Figure 8 shows a screendump of the interface, called ThemeStreams, with one of the summaries of a discussion shown at the bottom (the large, colorful term cloud).[5]

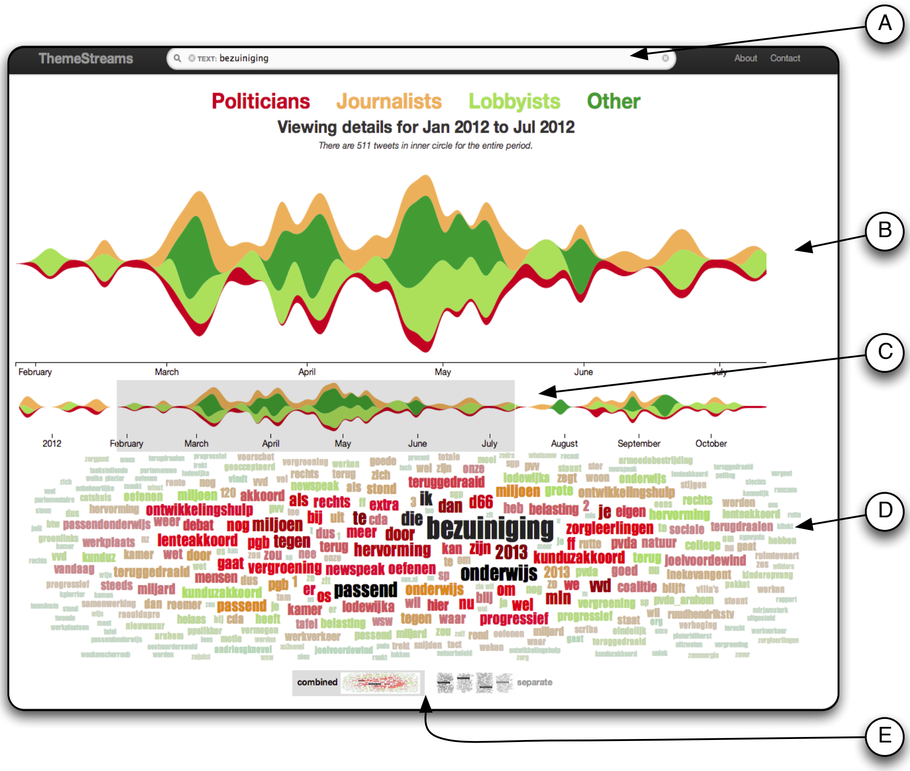
Users can gain insights in the development of messages around a topic in one of two ways. From the ThemeStreams home page they can access a fixed list of predefined themes and then explore streams of tweets around a theme they select [de Rooij et al., 2013]. Alternatively, they can enter a topic in a search box (item A in Figure 8). In response to a topic submitted by a user (either predefined or ad-hoc), ThemeStreams displays a zoomable stream graph at the top of the page (item B in Figure 8), depicting the number of tweets in

[4] https://github.com/varepsilon/clickmodels

[5] http://themestreams.xtas.net

the inner circle of four influencer groups retrieved for the topic. The thickness of the stream at each point in time is weighed by their “lifetime” (as determined by the number of retweets and mentions these tweets have received). In this way, we provide insight into how influential a group has been throughout the development of a theme, who finds a particular theme important and who were the first to talk about a particular theme.

Users can dive into more detail by zooming in using the focus + context principle [Card et al., 1999]. In part C in Figure 8, users can select a specific temporal interval, for instance because they know about important events related to their topic or because they observe interesting phenomena in the zoomable stream graph in part B of the interface. This allows users to not only see how important a theme was for an influencer group, but also what words one group used that other groups did not. To provide context, the stream graph for the entire period is also visualized (in part C of the interface); this enables rapid re-inspection of time periods close to the current focus. The user’s selection (indicated with a grey area, see part C), triggers the following events in the interface: (1) the zoomable stream graph in part B is restricted to the selected period and (2) in part D a term cloud is generated based on the tweets in the selected period. We offer two types of term cloud visualization selectable through the buttons in part E: one with a separate cloud for each of the influencer groups and one with a combined representation with different colors indicated which influencer group was most influential for the term shown. In order to comply with the Twitter ToS, the publicly accessible version of the demonstrator does not give access to tweets from which term clouds are generated.



Viewing details for Jan 2012 to Jul 2012

There are 511 tweets in inner circle for the entire period.

Figure 8. ThemeStreams: giving insight into the streams of themes being discussed in politics; the circled letters are explained in the text.

The interface was made part of the workflow of media analysts working for a communication agent. Implicit feedback allowed us to optimize the underlying language technology, especially the term extraction to for generating term clouds. Implicit signals, complemented with a user study, revealed that ThemeStreams was intuitive to understand, and inspection of parts of any query was easy to do. The combined cloud proved to be more insightful for fast overviews of the data. The individual clouds proved to be more useful for inspecting relative word usage between the groups. We also found a need for also depicting most represented speakers for any one group. A more detailed user study is currently in the works, and will be presented at a later time.

An important experimental lesson learned is that live demonstrators will lead to feature requests, which are sometimes perceived (by the target group) to be essential, and thus need to be added. In the case of ThemeStreams the additional feature request that we could not ignore concerns the ability to click through from the visual summaries down to the tweets from which they are generated; while this is technically feasible, and no problem in terms of the responsiveness of the interface, the Twitter ToS disallow the public display of tweets from a private index; a "private" version of ThemeStreams, that requires a user to log on, is in development to address the limitation.

5.6 Log analysis/experimental: yourHistory

This case study explores a platform that has so far been little used for information retrieval evaluation: Facebook. Within PROMISE we experimented with an application called yourHistory. The technological aim of yourHistory is to evaluate event linking and ranking algorithms. The public functionality offered is best explained using the public description:[6]

> "In history we often study dates and events that have little to do with our own life. yourHistory makes history tangible by showing historic events that are personal and based on your own interests (your Facebook profile). Often, those events are small-scale and escape history books. By linking personal historic events with global events, we link your life with global history: it's like we're writing your own personal history book. We represent your Facebook profile as a bag of concepts, by extracting raw text from your profile and applying state-of-the-art entity linking techniques. By leveraging the structured nature of DBPedia we extract historic event entities. We map the DBPedia entities to their corresponding Wikipedia pages. To generate your personal timeline, we match your profile entities to the events by applying a variety of similarity metrics. The final selection of historic events you are presented with is realized through a mix of your personal profile, the timespans of your own and your parents' lives, and statistical properties concerning the events. By the way, the events in the global history are based on the timeline of modern history.[7]"

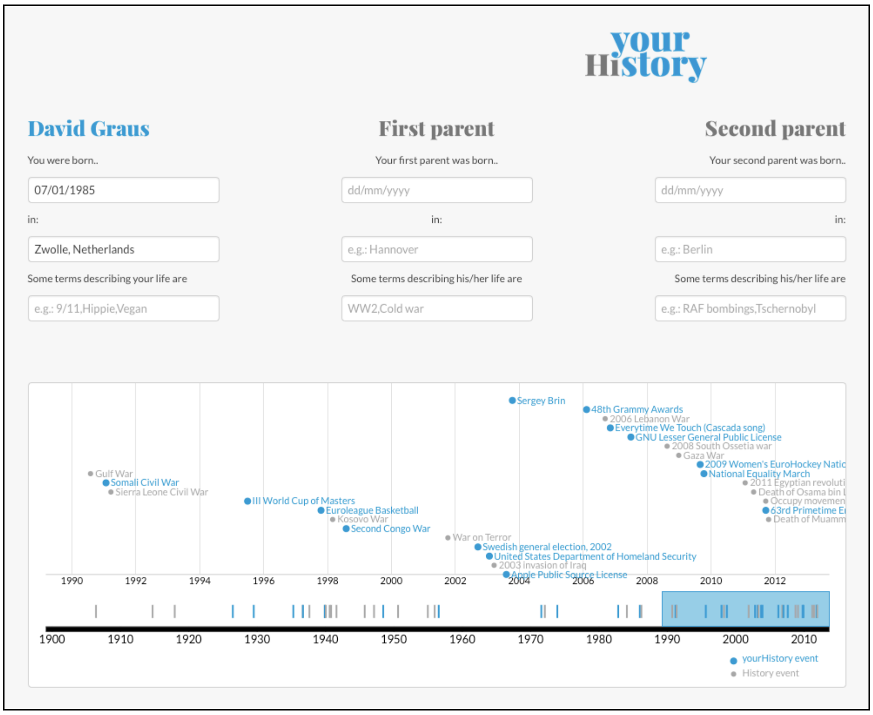
A screen dump of the app can be seen in Figure 9. At the top of the page, the user provides additional information about him/herself. At the bottom of the page, a personalized timeline of historic events is displayed: highlighted (in blue) are the events that we think are of personal interest to the user, and in grey, key historical events are displayed. Users can "promote" and "demote" events by clicking (or not) on them and visiting related Wikipedia

[6] http://apps.facebook.com/yourhistory

[7] http://en.wikipedia.org/wiki/Timeline_of_modern_history

pages, thereby giving the experimenters implicit feedback about the level of interestingness. A future design of the visualization will have more explicit facilities for feedback.

At the time of writing the experiment is still ongoing. Preliminary findings, based on interactions with several hundreds of users, suggest that personalized events generate a higher click-through-rate than general historic events. Moreover, the clicks on personalized events, on average are longer in terms of dwell time at the target page than those on general historic events.



your History

David Graus
You were born..
07/01/1985
in:
Zwolle, Netherlands
Some terms describing your life are
e.g.: 9/11,Hippie,Vegan

First parent
Your first parent was born..
dd/mm/yyyy
in:
e.g.: Hannover
Some terms describing his/her life are
WW2,Cold war

Second parent
Your second parent was born..
dd/mm/yyyy
in:
e.g.: Berlin
Some terms describing his/her life are
e.g.: RAF bombings,Tschernobyl

Figure 9. yourHistory in action. Suggested events for the user David Graus shown at the bottom of the screen. Accordig to the yourHistory application, the events in blue are the personalized events that are estimated to be of interest to David Graus. The events in grey are historic events that are estimated to be of lesser interest to the user. By clicking on an event, the user can express his interest or disinterest, thus generating implicit feedback.

An important lesson about running this Facebook-based experiment concerns special requirements imposed by the ethical review board (ERB) at the home institute of the experimenters. In particular, the following requirements were imposed:

- To use the Facebook API to only offer the application to users aged 18 or higher.
- To explicitly ask for permission for using and/or sharing users' profile data, in one of three modalities: only by the application, by authors of the application, by other researchers at the authors' university.
- To explicitly ask for permission for storing the profile data, again in three modalities: for 1 day, for 3 months, or indefinitely.

In addition, at the request of the ERB additional sanity checks, based on heuristics, were implemented to help ensure that the application is only offered to Facebook users aged 18 or older.

5.7 Log analysis/experimental: Evaluating aggregated search

The next example case study concerns online experiments to improve online evaluation metrics, i.e., interleaved comparison methods. In a result page returned by a modern search system some results look different and may be more visually attractive than others. Moreover, results from different sub-collections (e.g., News, Images, Finance, Mobile) are usually grouped (i.e., presented adjacent in the ranking) to improve the search result browsing experience. These results are often called vertical documents. If the vertical results are grouped, we call such group a vertical block. In Figure 10 we provide a schematic picture of two document lists containing vertical blocks; the vertical block occupies positions 3 to 5 in ranking A and positions 4 to 5 in ranking B.

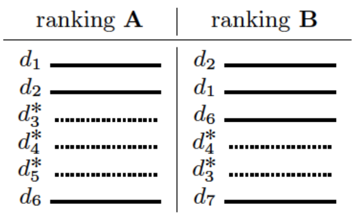


Figure 10. Two rankings with a vertical block present. Vertical documents are shown as dotted lines and also marked with *.

As pointed out in Section 2, there is an efficient way of comparing two rankings called interleaving: it produces an interleaved ranked list out of rankings A and B, shows it to the user and then infers user preferences from their clicks. However, if we want to interleave ranked lists from Figure 10 using existing interleaving methods (balanced, team-draft or probabilistic interleaving), we may end up in a situation where the resulting interleaved ranking has vertical documents mixed with regular documents. That is, those interleaving methods do not respect the grouping of vertical results. As was found by Dumais et al. [2001], this can significantly alter the user experience, which violates one of the core principles of user-based evaluations formulated by Joachims [2003].

In [Chuklin et al., 2013a] we have proposed the first vertical-aware interleaved comparison method, VA-TDI. In contrast to previous interleaved comparison methods, VA-TDI is designed to account for the placement of vertical result lists as one contiguous block, thus preserving this important aspect of the user experience.

Interestingly, we validated this method in two sets of experiments, first using real-life click log data (i.e., the living lab experiment that justifies the inclusion of the study here), and second using simulations. This combination of experimental methodologies enabled us to validate the proposed interleaving method both in a specific realistic search setting, and in a broader simulation setup. Limitations of the log approach include that only one specific type of vertical could be tested. Future work should validate the approach in additional search settings, possibly including results with several vertical blocks from a variety of vertical search engines. The simulation approach is based on a state of the art federated click model, but as new insights are gained into users' click behavior with and without vertical results, the simulations should be further refined. Nevertheless, we found no qualitative differences between our experiments on log data and using the simulation setup. This suggests that the obtained results are reliable.

VA-TDI preserves the quality of the user experience. Our living lab experiments on click log data showed that the user behavior (as captured by click metrics) on vertical-aware interleaved lists falls between that on the original rankings A and B. Our simulations confirmed that, in contrast to non vertical-aware interleaving, VA-TDI consistently produces one coherent block of vertical results. In addition, VA-TDI is able to reliably detect preferences in the quality of web-only, vertical-only, or overall result list quality. On click log data, we observed good correlations with commonly-used click metrics. In our simulations, we found that VA-TDI achieves the same accuracy as TDI, while preserving the quality of the user experience. Finally, our simulation experiments showed that VA-TDI preserves unbiasedness under random clicks. Our results confirm that VA-TDI opens up the way for applying interleaved comparison methods to search engine results with vertical or aggregated results, removing a major limitation of previous methods.

The methodological take home message from this case study concerns the dual methodology used: living lab experiments and a simulation. We recommend that to support exploratory research and evaluation issues, settings that face real users and, hence, are limited by the constraint to produce “reasonable” results, simulations should be considered in addition to a living lab experiment so as to facilitate broader explorations. In an ideal world, the outcomes of both types of experimental methodology confirm each other.

5.8 Log analysis/experimental: Lerot

Our final example concerns the creation of a software package, Lerot, that bundles all ingredients needed for experimenting with online learning to rank for information retrieval, thereby filling an important hiatus in the availability of experimental tooling.[8]

Adapting IR systems to a specific user, group of users, or deployment setting has become possible and popular due to learning to rank techniques [Liu, 2009]. Generally speaking, a learning to rank method learns the weights of a function that maps a document-query pair described by a feature vector to a value that is used to rank documents for a given query. We refer to such a function with instantiated weights as a ranker. Most current approaches learn offline, i.e., before deployment rankers are estimated from manually annotated training data.

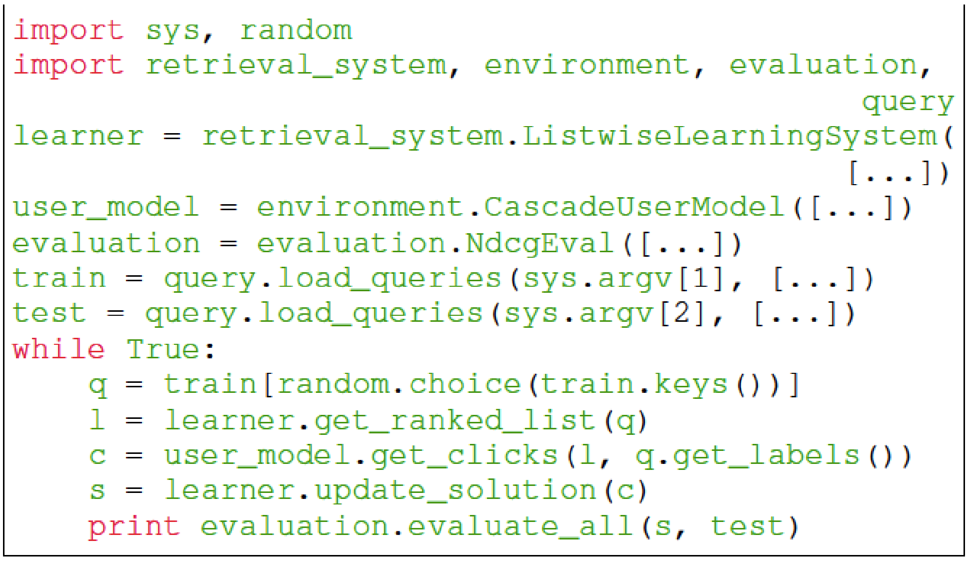
In contrast, an online learning to rank method learns directly from interactions with users, e.g., using click feedback. For instance, the current state-of-the-art online learning to rank approach uses dueling bandit gradient descent (DBGD) [Hofmann et al., 2013b; Yue and Joachims, 2009] to find a high quality ranker. In each step, the current best ranker is perturbed, and then both the original and perturbed rankers are compared using an interleaved comparison method [Radlinski et al., 2008]: the rankings proposed by the two rankers are interleaved and presented to the user, whose clicks determine which ranker wins the comparison. If the perturbed ranker wins, the original ranker is adjusted slightly in its direction.

Lerot, the framework presented here, offers a solution for evaluating and experimenting with online learning to rank algorithms in living labs and simulations. As we pointed out above, living labs represent a user-centric research methodology that seeks to test and evaluate

[8] Available at https://bitbucket.org/ilps/lerot.git

emerging technologies in real-world contexts. Therefore, they form the ideal environment for prototyping and assessing online learning to rank methods. Lerot is designed to support such experiments with online learning to rank algorithms, or with components of such algorithms in a living lab setup. Lerot also offers the next best thing: simulations of users interacting with a search engine. In contrast to experiments run in a full-blown living lab environment, simulation experiments make it possible to generate a wide range of candidate result lists, without the risk of adversely affecting user experience in a production system, as we saw in the previous case study, on aggregated search. Thus, simulation experiments with Lerot may complement or precede experimentation in a living lab setup for online learning to rank.

In very broad terms, Lerot can be used to run two types of experiment: learning experiments and evaluation experiments. Learning experiments operate in a continuous space of possible solutions and evolve rankers over time to find the optimal one. Evaluation experiments, on the other hand, operate on a fixed set of rankers and are designed to identify the best ranker among this set using, for instance, interleaved comparisons. In our use of Lerot we have mostly focused on describing the learning experiments so far.



```
import sys, random
import retrieval_system, environment, evaluation,
                                               query
learner = retrieval_system.ListwiseLearningSystem(
                                             [...])
user_model = environment.CascadeUserModel([...])
evaluation = evaluation.NdcgEval([...])
train = query.load_queries(sys.argv[1], [...])
test = query.load_queries(sys.argv[2], [...])
while True:
    q = train[random.choice(train.keys())]
    l = learner.get_ranked_list(q)
    c = user_model.get_clicks(l, q.get_labels())
    s = learner.update_solution(c)
    print evaluation.evaluate_all(s, test)
```

Figure 11. Minimal example of an online learning experiment that uses a list wise learning algorithm and a cascade user model to simulate clicks.

A minimal example of a learning algorithm embedded in a simulation with a user model is shown in Figure 11. The example defines a learner, a user model, an evaluation method, and lists of training and test queries with labels. If real users are available, they are the source of the training queries and the clicks. In their absence, the queries come from a dataset and the clicks from a click model that uses relevance judgments. The queries *q* are observed in a random order, a ranked list *l* is produced by the learner, this ranking is sent to the click model and the clicks *c* it produces, in turn, are observed by the learner so that it can update the solution. The updated solution *s* is then evaluated on the test queries. In theory, this process continues indefinitely.

Lerot fills an important niche in the world of online learning to rank. The framework has all batteries included (except for the data), to replicate experiments; no code needs to be written. Lerot has been used to verify the findings in numerous publications [Chuklin 2013a; Hofmann et al., 2013a–d] at major venues. The framework is easily extensible to compare the implemented methods to new online evaluation and online learning approaches.

Online learning to rank is a rapidly evolving area in information retrieval. While several libraries exist for offline learning to rank, Lerot is the first framework for online learning to rank. The framework has been used in many recent publications and reproducing results from those papers only requires a user of the framework to run it with the appropriate configuration file. In sum, Lerot is easy to use and extensible. We have described all functions that need to be implemented in order to do so.

In the context of living labs, Lerot supports two directions of development. First, it allows for experiments with simulated users. The user models it currently implements reflect our current understanding of user behavior; they can easily be extended or replaced by evaluations under different sets of assumptions. Second, Letor provides components that implement complete online learning to rank solutions for use as part of complete living lab evaluation setups.

6 Conclusions

One can think of information retrieval as a field consisting of three main branches of activity: *analysis* (of content, structure, user behavior), *synthesis* (of the outcomes of the analysis so as to gather, store and retrieve information in an effective and efficient manner), and *evaluation* (of the synthesized decisions). In other words, evaluation is a key ingredient of the field. The spectrum of evaluation methods available for researchers in industry and academia continues to expand, as new tasks, new types of data, and new types of evaluation resources become available. The goal of the deliverable has been to gain a better understanding of the strengths and weaknesses of one particular evaluation methodology: the use of operational systems as experimental platforms.

To gain this improved understanding, we situated the living labs methodology against the broader background of offline evaluation methods, user centered evaluation and online evaluation methods. In addition, we indicated the key technological, organizational and legal dimensions of running experiments on operational systems.

The methodology was illustrated with a representative set of examples, organized along two orthogonal dimensions: user panel vs log analysis (i.e., data gathering) and observational vs experimental (i.e., purpose for which the methodology is being used). Rather than simply running existing methods out of the box, or repeating experiments previously published in the literature, we innovated both in terms of algorithmic lessons and in terms of tooling (such as logging and online learning to rank). We hope that these innovations may inspire others to consider using operational systems as an experimental platform, as there is a lot that can be learned about information access in this manner that cannot be learned through alternative evaluation methods, as our case studies have illustrated.

References

[Agrawal et al., 2009] R. Agrawal, S. Gollapudi, A. Halverson, S. Ieong. Diversifying search results. In: *WSDM*'09. p. 5. ACM, 2009.

[Arguello et al., 2009] J.Arguello, F. Diaz, J. Callan, J. Crespo. Sources of evidence for vertical selection. In: *SIGIR* '09. pp. 315–322. ACM, 2009.

[Azzopardi and Balog, 2011] L. Azzopardi, K. Balog. Towards a Living Lab for Information Retrieval Research and Development. A Proposal for a Living Lab for Product Search Tasks. *Conference on Multilingual and Multimodal Information Access Evaluation (CLEF 2011)*, pages 26-37, September 2011.

[Braschler et al., 2012] M. Braschler, S. Rietberger, M. Imhof, A., Järvelin, P. Hansen, M. Lupu, M. Gäde, R. Berendsen, R. García Seco de Herrera: PROMISE deliverable 2.3: Best Practices Report, 2012.

[Carman et al., 2009] M. Carman, M. Baillie, R. Gwadera, F. Crestani. A statistical comparison of tag and query logs. In *SIGIR'09*, pages 123–130, Boston, USA, 2009. ACM.

[Chapelle et al., 2013] O. Chapelle, T. Joachims , F. Radlinski, Y. Yue, Y. Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions on Information Systems* 30(1):1–41, 2013.

[Chapelle and Zhang, 2009] O. Chapelle, Y., Zhang. A dynamic bayesian network click model for web search ranking. In WWW'09. ACM, 2009

[Chuklin et al., 2013a] A. Chuklin, A. Schuth, K. Hofmann, P. Serdyukov, M. de Rijke. Evaluating aggregated search using interleaving. *CIKM '13*, ACM, October 2013

[Chuklin et al., 2013b] A. Chuklin, P. Serdyukov, M. de Rijke. Modeling clicks beyond the first result page. *CIKM '13*, ACM, October 2013.

[Chuklin et al., 2013c] A. Chuklin, P. Serdyukov, M. de Rijke. Using intent information to model user behavior in diversified search. *ECIR'13*, March 2013.

[Chuklin et al., 2013d] A. Chuklin, P. Serdyukov, M. de Rijke. Click-model based information retrieval metrics. *SIGIR 2013*, ACM, July 2013

[Clarke et al., 2011] C.L.A. Clarke, N. Craswell, I. Soboroff. A comparative analysis of cascade measures for novelty and diversity. In: *WSDM '11*. pp. 75–84. ACM, 2011.

[Crook et al., 2009] S. Crook, B. Frasca, R. Kohavi, R. Longbotham. Seven Pitfalls to Avoid when Running Controlled Experiments on the Web. *KDD'09*, ACM, 2009

[Dumais et al., 2001] S. Dumais, E. Cutrell, H. Chen. Optimizing search by showing results in context. In *CHI'01*, ACM, 2001

[Dumais et al., 2011] S. Dumais, R. Jefflies, D.M Russell, D. Tang, J. Teevan. Design and analysis of large scale log studies. A CHI 2011 course. *CHI '11*, May 2011

[Dupret and Piwowarski, 2008] G. Dupret, B., Piwowarski. A user browsing model to predict search engine click data from past observations. In SIGIR'08. ACM, 2008

[de Goede and van Wees, 2013] B. de Goede, J. van Wees. ILPS logging service. Internal status report, University of Amsterdam, August 2013

[Jansen, 2008] B.J. Jansen. The methodology of search log analysis. In B.J. Jansen, A. Spink, I. Taksa, editors, *Handbook of Research on Web Log Analysis*, pages 99–121. Information Science Reference, 2008

[Jansen and Pooch, 2001] B.J. Jansen, U. Pooch. A review of web searching studies and a framework for future research. *Journal of the American Society for Information Science and Technology*, 52(3):235–246, 2001

[Hofmann, 2013] K. Hofmann. Fast and Reliable learning to rank for information retrieval. PhD thesis, University of Amsterdam, 2013

[Hofmann et al., 2011a] K. Hofmann, S. Whiteson, M. de Rijke. A probabilistic method for inferring preferences from clicks. In *CIKM '11*. ACM, 2011

[Hofmann et al., 2011b] K. Hofmann, S. Whiteson, M. de Rijke. Contextual Bandits for Information Retrieval. *NIPS '11*, 2011

[Hofmann et al., 2012] K. Hofmann, S. Whiteson, M. de Rijke. Estimating Interleaved Comparison Outcomes from Historical Click Data. In *CIKM '12*, 2012

[Hofmann et al., 2013a] K. Hofmann, A. Schuth, S. Whiteson, M. de Rijke. Reusing historical interaction data for faster online learning to rank for IR. In WSDM'13. ACM, 2013

[Hofmann et al., 2013b] K. Hofmann, S. Whiteson, M. de Rijke. Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. *Information Retrieval* 16(1):63–90, 2013

[Hofmann et al., 2013c] K. Hofmann, S. Whiteson, M. de Rijke. Fidelity, soundness, and efficiency of interleaved comparison methods. *ACM Transactions on Information Systems 31(4)*, 2013

[Imhof et al., 2013] M. Imhof, M. Braschler, P. Hansen, S. Rietberger. Evaluation for Operational IR Applications – Generalizability and Automation. Proceedings of CIKM 2013 Living Labs for Information Retrieval Evaluation Workshop. San Francisco: ACM, 2013.

[Joachims, 2003] T. Joachims. Evaluating retrieval performance using clickthrough data. In *Text Mining*, 2003.

[Kelly et al., 2009] D. Kelly, S. Dumais, J.O. Pedersen. Evaluation challenges and directions for information seeking support systems. *Computer* 42(3), 60–66, 2009.

[Kohavi et al., 2009] R. Kohavi, R. Longbotham, D. Sommerfield, R.M. Henne. Controlled experiments on the web: survey and practical guide. Data Mining and Knowledge Discovery, 18:140–181, 2009.

[Lui, 2008] T.-Y. Lui. Learning to rank for information retrieval. Foundations and Trends in Information Retrieval, 3(3):225–331, 2009.

[Lupu, 2011] M. Lupu. PatOlympics – An Infrastructure for Interactive Evaluation of Patent Retrieval Tools. *Proceedings of DESIRE'11*, 2011.

[Mishne and de Rijke, 2006] G. Mishne, M. de Rijke. A study of blog search. In Lalmas, M., MacFarlane, A., Rüger, S., Tombros, A., Tsikrika, T., and Yavlinsky, A., editors, *ECIR '06*, LNCS 3936, pages 289–301, 2006. Springer.

[Peters, 1993] T.A. Peters. The history and development of transaction log analysis. *Library Hi Tech*, 11(2):41–66, 1993.

[Pirolli, 2009] P. Pirolli. Powers of 10: Modeling complex information-seeking systems at multiple scales. *Computer* 42, 33–40, 2009.

[Radlinski and Dumais, 2006] R. Radlinski, S. Dumais. Improving personalized web search using result diversification. In: *SIGIR '06*. ACM, 2006.

[Radlinski et a.l, 2023] F. Radlinski, M. Kurup, T. Joachims. How does click-through data reflect retrieval quality? In *CIKM '08*, ACM 2008.

[Rice and Borgman, 1983] R.E. Rice, C.L. Borgman. The use of computer-monitored data in information science and communication research. *Journal of the American Society for Information Science* and Technology , 34(4):247–256.

[Rietberger et al., 2012] S. Rietberger, M. Imhof, M. Braschler, R. Berendsen, A. Järvelin, P. Hansen, A. Garcia Seco de Herrera, T. Tsikrika, V. P. M. Lupu, M. Gäde, M. Kleineberg, and K. Choukri. Promise deliverable 4.2: Tutorial on evaluation in the wild. Promise Consortium, 2012.

[de Rooij et al., 2013] O. de Rooij, D. Odijk, M. de Rijke. ThemeStreams: Visualizing the Stream of Themes Discussed in Politics. *SIGIR '13*, ACM, 2013.

[Schumacher, 2009] J. Schumacher. Living labs: Definition, harmonization cube indicators & good practices. Deliverable 3.1, Alcotra Innovation Project. Alcotra Innovation Project, 2009.

[Schuth et al., 2013]	A. Schuth, K. Hofmann, S. Whiteson M. de Rijke. Lerot: An online learning to rank framework, *Living Labs'13: Workshop on Living Labs for Information Retrieval Evaluation*, November 2013. ACM.
[Styskin et al., 2011]	A. Styskin, F. Romanenko, F. Vorobyev, P. Serdyukov. Recency ranking by diversification of result set. In: *CIKM '11* pp. 1949–1952. ACM, 2011.
[Thomson, 2013]	Thomson Reuters Patent Search Services. http://thomsonreuters.com/ip-search/. Last retrieved: August 2013.