

Overview of the Living Labs for Information Retrieval Evaluation (LL4IR) CLEF Lab 2015

Anne Schuth¹, Krisztian Balog², and Liadh Kelly³

¹ University of Amsterdam, The Netherlands

² University of Stavanger, Stavanger, Norway

³ ADAPT Centre, Trinity College, Dublin, Ireland

anne.schuth@uva.nl, krisztian.balog@uis.no, liadh.kelly@tcd.ie

Abstract. In this paper we report on the first Living Labs for Information Retrieval Evaluation (LL4IR) CLEF Lab. Our main goal with the lab is to provide a benchmarking platform for researchers to evaluate their ranking systems in a live setting with real users in their natural task environments. For this first edition of the challenge we focused on two specific use-cases: product search and web search. Ranking systems submitted by participants were experimentally compared using interleaved comparisons to the production system from the corresponding use-case. In this paper we describe how these experiments were performed, what the resulting outcomes are, and conclude with some lessons learned.

Keywords: Information retrieval evaluation, living labs, product search, web search.

1 Introduction

Evaluation is a central aspect of information retrieval (IR) research. In the past few years, a new evaluation paradigm known as living labs has been proposed, where the idea is to perform experiments in situ, with real users doing real tasks using real-world applications [12]. The need for more realistic evaluation, involving real users, was reiterated at recent IR workshops [1, 3, 11]. This type of evaluation, however, has so far been available only to (large) industrial research labs [15, 24]. Our main goal with the Living Labs for IR Evaluation (LL4IR) CLEF Lab is to provide a benchmarking platform for researchers to evaluate their ranking systems in a live setting with real users in their natural task environments, similar to the living labs for IR instances proposed in [2, 13]. The lab acts as a proxy between commercial organizations (live environments) and lab participants (experimental systems), facilitates data exchange, and makes comparison between the participating systems. The first edition of the lab focuses on two use-cases and one specific notion of what a living lab is (with a view to expanding to other use-cases and other interpretations of living labs in subsequent years). Use-cases for the first lab are: product search (on an e-commerce site) and web search (through a commercial web search engine).

The LL4IR CLEF Lab contributes to the understanding of online evaluation as well as an understanding of the generalization of retrieval techniques across different use-cases. Most importantly, it promotes IR evaluation that is more realistic, by allowing

researches to have access to historical search and usage data and by enabling them to validate their ideas in live settings with real users. This initiative is a first of its kind for IR. CLEF Newsreel [6]⁴ is a similar initiative, but for a different problem domain: news recommendation. By contrast we are focusing on the very different space of information retrieval, which contains its own unique use-cases, approaches, challenges, and researchers. Major differences between the labs include the presence of a query and, importantly, that our API lifts the real time processing requirements on the part of participants, lowering the participation threshold significantly.

This paper reports on the results obtained during the official CLEF evaluation round that took place between May 1 and May 15, 2015. The positive feedback and growing interest from participants motivated us to organize a subsequent second evaluation round. As this second round is still ongoing at the time of writing, we provide more detailed results and analysis, including those of the second round, in an extended version of this overview paper [22].

In the next section we describe our API architecture and evaluation methodology. We then describe each of the two use-cases of the first edition of the lab in turn in Sections 2 and 4, and provide details and analysis of the submissions received. Finally, in Section 5, we conclude the paper.

2 Living Labs for IR

For the LL4IR CLEF Lab, evaluation is done primarily through an API. We first describe the workings of our API, followed by the setup of our evaluation divided into training and test phases. We then describe how we compute evaluation metrics using interleaved comparisons. Finally, we describe how we aggregate interleaving outcomes.

2.1 Living Labs API

For each of the use-cases, described in Sections 2 and 4, challenge participants take part in a live evaluation process. For this they use a set of frequent queries as training queries and a separate set of frequent queries as test queries. Candidate documents are provided for each query and historical information associated with the queries. When participants produce their rankings for each query, they upload these to the commercial provider use-case through the provided LL4IR API. The commercial provider then interleaves a given participant’s ranked list with their own ranking, and presents the user with the interleaved result list. Participants take turns in having their ranked list interleaved with the commercial providers ranked list. This process of interleaving a single experimental system with the production system at a time is orchestrated by the LL4IR API, such that each participant gets about the same number of impressions. The actions performed by the commercial providers’ system users are then made available to the challenge participant (whose ranking was shown) through the API; i.e., the interleaved ranking, resulting clicks, and (aggregated) interleaving outcomes.

Figure 1 shows the Living Labs architecture and how the participant interacts with the use-cases through the LL4IR provided API. As can be seen, frequent queries (Q)

⁴ <http://www.clef-newsreel.org/>

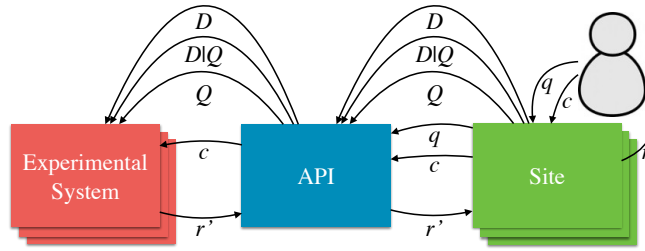


Fig. 1. Schematic representation of interaction with the LL4IR API, taken from [4].

with candidate documents for each query ($D|Q$) are sent from a site through the API to the experimental systems of participants. These systems upload their rankings (r') for each query to the API. When a user of the site issues one of these frequent queries (q), then the site requests a ranking (r') from the API and presents it interleaved with r to the users. Any interactions (c) of the user with this ranking are sent back to the API. Experimental systems can then obtain these interactions (c) from the API and update their ranking (r') if they wish. We provided participants with example code and guidelines to ease the adaptation to our setup.⁵ Our evaluation methodology, including reasons for focusing on frequent queries, is described in more detail in [4].

Training Phase During the training phase, participants are free to update their rankings using feedback information. This feedback information is made available to them as soon as it arrives at the API. Their rankings can be updated at any time and as often as desired. Both click feedback and aggregated outcomes are made available directly and are updated constantly.

Test Phase In the test phase, challenge participants receive another set of frequent queries as test queries. Again, the associated historical click information as well as candidate results for these queries are made available. After downloading the test queries, participants could only upload their rankings until the test phase started or only once after it started. These rankings are then treated in the same way as training queries. That is, they are interleaved with the commercial providers' rankings for several weeks. As for the training phase, in the test phase each challenge participant is given an approximately equal number of impressions. A major difference is that for the test queries, the click feedback is not made available. Aggregated outcomes are provided only after the test phase ends.

2.2 Evaluation Metric

The overall evaluation of challenge participants is based on the final system performance, and additionally on how the systems performed at each query issue. The primary metric used is aggregated interleaving outcomes, and in particular we are interested in

⁵ <http://doc.living-labs.net/en/latest/guide-participant.html>

the fractions of winning system comparisons. There are two reasons for using interleaved comparisons. Firstly, interleaved comparisons ensure that at least half the ranking shown to users comes from the production system. This reduces the risk of showing bad rankings to users. Secondly, interleaved comparisons were shown to be two orders of magnitude more sensitive than other ways of performing online evaluation such as A/B testing [7, 24]. This means that far fewer query impressions are required to make informed decisions on which ranker is better.

Interleaved comparisons Many interleaving approaches have been proposed over the past few years (for instance [9, 10, 18, 19, 21, 23]). By far the most frequently used interleaving algorithm to date is Team Draft Interleaving (TDI) [19] which is also what is used in our living labs. Given a user query q , TDI produces an interleaved result list as follows. The algorithm takes as input two rankings. One ranking from the participant $r' = (a_1, a_2, \dots)$ and one from the production system $r = (b_1, b_2, \dots)$. The goal is to produce a combined, interleaved ranking $L = (a_1, b_2, \dots)$. This is done analogue to how sports teams may be constructed in a friendly sports match. The two team captains take turns picking players. They can pick available documents (players) from the top of the rankings r' and r , these top ranked document are deemed to be the best documents. Documents can only be picked once (even if they are listed in both r and r'). And the order in which the documents are picked determines ranking L . In each round, the team captains flip a coin to determine who goes first. The algorithm remembers which team each documents belong to. If a document receives a click from a user, credit is assigned to the team the document belongs to. The team (participant or production system) with most credit wins the interleaved comparison. This process is repeated for each query. For more details see the original paper describing TDI by Radlinski et al. [19] and a large scale comparison of interleaving methods by Chapelle et al. [7].

Aggregated Outcomes We report the following aggregated interleaving metrics, where *Outcome* serves as the primary metric for comparing participants rankings. These aggregations are constantly updated for training queries. For the test phase they are only computed after the phase is over.

#Wins is defined as the number of wins of the participant against the production system, where a *win* is defined as the experimental system having more clicks on results assigned to it by TDI than clicks on results assigned to the production system;

#Losses is defined as the number of losses against the production system;

#Ties is defined as the number of ties with the production system;

#Impressions is the total number of times when rankings (for any of the test queries) from the participant have been displayed to users of the production system; and

Outcome is defined as the fraction of wins, so $\#Wins / (\#Wins + \#Losses)$.

An *Outcome* value below the *expected outcome* (typically 0.5) means that the participant system performed worse than the production system (i.e., overall it has more losses than wins). Significance of outcomes is tested using a two-sided binomial test which uses the expected outcome, p-values are reported.

Note that using these metrics, we are in theory only able to say something about the relationship between the participant’s system and the production system. However, Radlinski et al. [19] show experimentally that it is not unreasonable to assume transitivity. This allows us to also draw conclusions about how systems compare to each other. Ideally, instead of interleaving, we would have used multileaved comparison methods [21, 23] which would directly give a ranking over rankers by comparing them all at once for each query.

3 Use-case 1: Product Search

3.1 Task and Data

The *product search* use-case is provided by REGIO Játék (REGIO Toy in English), the largest (offline) toy retailer in Hungary with currently over 30 stores. Their webshop⁶ is among the top 5 in Hungary. The company is working on strengthening their on-line presence; improving the quality of product search in their online store is directed towards this larger goal. An excerpt from the search result page is shown in Figure 2.

As described in Section 2, we distinguish between training and test phases. Queries are sampled from the set of frequent queries; these queries are very short (1.18 terms on average) and have a stable search volume. For each query, a set of candidate products (approximately 50 products per query) and historical click information (click-through rate) is made available. For each product a structured representation is supplied (see below). The task then is to rank the provided candidate set.

Product Descriptions For each product a fielded document representation is provided, containing the attributes shown in Table 1. The amount of text available for individual products is limited (and is in Hungarian), but there are structural and semantic annotations, including:

- Organization of products into a two-level deep topical categorization system;
- Toy characters associated with the product (Barbie, Spiderman, Hello Kitty, etc.);
- Brand (Beados, LEGO, Simba, etc.);
- Gender and age recommendations (for many products);
- Queries (and their distribution) that led to the given product.

Candidate Products The candidate set, to be ranked, contains all products that were available in the (recent) past. This comprises all products that are considered by the site’s production search engine (in practice: all products that contain any of the query terms in any of their textual fields). One particular challenge for this use-case is that the inventory (as well as the prices) are constantly changing; however, for challenge participants, a single ranking will be used throughout the entire test period of the challenge, without the possibility of updating it. The candidate set therefore also includes products that may not be available at the moment (but might become available again

⁶ <http://www.regiojatek.hu/>

REGIO JÁTÉK Ahol a vásárlás gyerekjárték!

INFORMÁCIÓ | BEJELENTÉZÉS

angry birds | Keresés

KATEGÓRIÁK | ÉLETKOR | MÁRKÁK | MESEHŐS | AKCIÓK | ÁRUHÁZAK

Kosár 0

Találatok szűkítése | Találatok | 26 termék elérhető

Kategóriák

- Matrac, szőrf, ráolós állatok (4)
- Készletfejlesztő (3)
- Papír, írószer (2)
- Űszógumi, karószó (2)
- Akciófigurák (2)
- [további kategóriák](#)

Márkák

- Hasbro (2)
- Bastwey (1)
- Bestway (1)

Mesehősök

- Angry Birds (12)
- Star Wars (2)

Nem

- mindegy (26)

Életkor

0 - kortalan

Ár

1 - 100000

- Csak akciós
- Csak új termékek

Ügyfélszolgálat

- 06 (30) 206-1000
- Online chat
- Hívj Skype-on!
- Írj nekünk!

Hírlevél

Ne maradj le akcióinkról, iratkozz fel hírlevelünkre!

Név:

E-mail cím:

Feliratkozás

Termékek száma 15

Rendezés alapértelmezett










 <p>Angry Birds - Star Wars kártya</p> <p>745 Ft</p>	 <p>Angry Birds matricák ANG</p> <p>150 Ft</p>	 <p>Angry Birds kártyagyűjtő album ANG</p> <p>695 Ft 245 Ft</p>
 <p>ANGRY BIRDS gyűjthető figurák, 2 db /cs</p> <p>2 130 Ft</p>	 <p>Angry Birds SW. szivacsdobáló 4 féle A</p> <p>5 995 Ft</p>	 <p>2x90 db Angry Birds - Star Wars puzzle</p> <p>1-245 Ft 745 Ft</p>
 <p>Puzzle "4in1" Star Wars - Angry Birds</p> <p>2 255 Ft</p>	 <p>Űszógumi Angry Birds 56cm</p> <p>745 Ft</p>	 <p>Angry Birds GO matrica ANG</p> <p>80 Ft</p>

Fig. 2. Screenshot of REGIO, our product search use-case.

in the future). Participating systems were strongly encouraged to consider all products from the provided candidate set. Those that were unavailable at a given point in time were not displayed to users of the REGIO online store. Further, it may happen (and as

Table 1. Fielded document representation of products in the product search use-case.

Field	Description
age_max	Recommended maximum age (may be empty, i.e., 0)
age_min	Recommended minimum age (may be empty, i.e., 0)
arrived	When the product arrived (first became available); only for products that arrived after 2014-08-28
available	Indicates if the product is currently available (1) or not (0)
bonus_price	Provided only if the product is on sale; this is the new (sales) price
brand	Name of the brand (may be empty)
category	Name of the (leaf-level) product category
category_id	Unique ID of the (leaf-level) product category
characters	List of toy characters associated with the product (may be empty)
description	Full textual description of the product (may be empty)
main_category	Name of the main (top-level) product category
main_category_id	Unique ID of the main (top-level) product category
gender	Gender recommendation. (0: for both girls and boys (or unclassified); 1: for boys; 2: for girls)
photos	List of photos about the product
price	Normal price
product_name	Name of the product
queries	Distribution of (frequent) queries that led to this product (may be empty)
short_description	Short textual description of the product (may be empty)

we show in [22] it indeed does happen) during the test period that new products arrive; experimental systems are not able to include these in their ranking (this is the same for all participants), while the production system might return them. This can potentially affect the number of wins against the production system (to the advantage of the production system), but it will not affect the comparison across experimental systems.

3.2 Submissions and Results

Two organizations submitted a total of four runs. In addition, a simple baseline provided by the challenge organizers is also included for reference. Table 2 presents the results.

Approaches The organizers’ baseline (BASELINE in Table 2) ranks products based on historical click-through rate. Only products that were clicked for the given query are returned; their attributes are not considered. In case historical clicks are unavailable (this happened for a single query R-q97), (all) candidate products are returned in an arbitrary order (in practice, in the same order as they were received from the API via the `doclist` request).

The University of Stavanger [8] employed a fielded document retrieval approach based on language modeling techniques. Specifically, building upon the Probabilistic Retrieval Model for Semistructured Data by Kim et al. [14], they experimented with three different methods (UIS-*) for estimating term-field mapping probabilities. Their

Table 2. Results for the product search use-case. The expected outcome under a randomly clicking user is 0.28. P-values are computed using a binomial test.

Submission	Outcome	#Wins	#Losses	#Ties	#Impressions	p-value
BASELINE	0.4691	91	103	467	661	< 0.01
UIS-MIRA [8]	0.3413	71	137	517	725	0.053
UIS-JERN [8]	0.3277	58	119	488	665	0.156
UIS-UIS [8]	0.2827	54	137	508	699	0.936
GESIS [20]	0.2685	40	109	374	523	0.785

results show that term-specific field mapping in general is beneficial, but their attempt at estimating field importance based on historical click-through information has met with limited success.

Team GESIS [20] also used a fielded document representation. They used Solr for ranking products and incorporated historical click-through rates, if available, as a weighting factor.

Dealing with Inventory Changes As mentioned in Section 3.1, the product inventory is subject to changes. Not all products that were part of the candidate set were available at all times. If all products were available, the expected probability of winning an interleaved comparison (assuming a randomly clicking user) would be 0.5. However, on average, 44% of the products were actually unavailable. These products were only ever present in the participants ranking (the site’s ranking never considered them). And, only *after* interleaving were these products removed from the resulting interleaved list. We note that this is undesired behavior, as they should have been filtered out *before* interleaving. The necessary adjustments have been made to the implementation for the next round of the challenge. As for interpreting the current results, this means that the chances for products from the participants ranking to be clicked were reduced. This in turn reduces the expected probability to win to:

$$P(\text{participant} > \text{site}) = (1 - 0.44) \cdot 0.5 = 0.28.$$

Consequently, if a participant’s system wins more than in 28% of the impressions, then this is more than expected. And thus the participant’s system can be said to be better than the site’s system if the outcome is (significantly) more than 28%.

Results We find that at least 3 submissions are likely to have improved upon the production system’s ranking. Somewhat surprisingly, the simple baseline performed by far the best, with an outcome of 0.4691. This was also the only system that significantly outperformed the production system. The best performing participant run is UIS-MIRA, with an outcome of 0.3413. A more in-depth analysis of the results is provided in the extended lab overview paper [22].

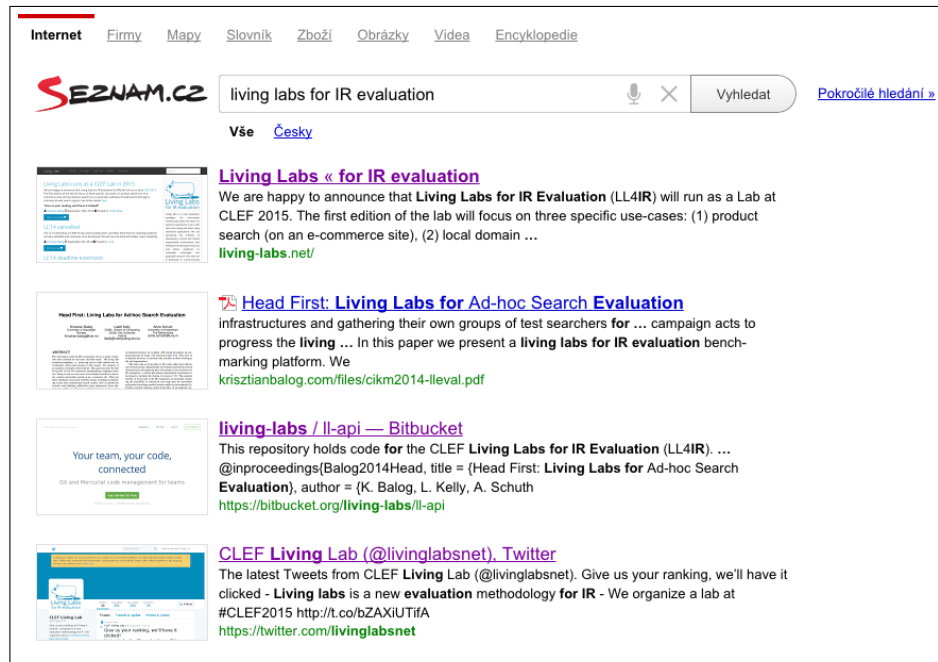


Fig. 3. Screenshot of Seznam, our web search use-case.

4 Use-case 2: Web Search

4.1 Task and Data

The *web search* use-case is provided by Seznam,⁷ a very large web search engine in the Czech Republic. See Figure 3 for a screenshot of the user interface.

Seznam serves almost half the country’s search traffic and as such has very high site traffic. Queries are the typical web search queries, and thus are a mixed bag of transactional, navigational and transactional [5]. In contrast to the product search use-case, apart from the scale and the query types, Seznam does not make raw document and query content available, rather features computed for documents and queries. This is much like any learning to rank dataset, such as Letor [17]. Queries and documents are only identified by a unique identifier and for each query, the candidate documents are represented with sparse feature vectors. Seznam provided a total of 557 features. These features were not described in any way. The challenge with this use-case then is a learning to rank challenge [16].

As described in Section 2, the web search use-case also consists of a training and test phase. For the test phase, there were 97 queries, for the training phase 100 queries were provided. On average, for each query there were about 179 candidate documents. In total, there were 35,322 documents.

⁷ <http://search.seznam.cz/>

Table 3. Results for the web search use-case. The expected outcome under a randomly clicking user is 0.5. P-values were computed using a binomial test.

Submission	Outcome	#Wins	#Losses	#Ties	#Impressions	p-value
EXPLOITATIVE BASELINE	0.5527	3030	2452	19055	24537	< 0.01
UNIFORM BASELINE	0.2161	430	1560	1346	3336	< 0.01

4.2 Results

The web search use-case attracted 6 teams that submitted runs for the training queries. However, none of them submitted runs for the test queries. Therefore, we can only report on two baseline systems, provided by the challenge organizers. Baseline 1, titled EXPLOITATIVE BASELINE in Table 3, uses the original Seznam ranking and was therefore expected to produce an outcome of 0.5.⁸ Baseline 2, titled UNIFORM BASELINE in Table 3, assigned uniform weights to each feature and ranked by the weighted sum of feature values. This baseline was expected to not perform well.

Over the past months, there have been over 440K impressions on Seznam through our Living Labs API. On average this amounts to 2,247 impressions for each query. Approximately 6% of all impressions were used for the testing period. As can be seen in Table 3, the EXPLOITATIVE BASELINE outperformed the production system. An outcome (outcome measure described in Section 2) of 0.5527 was achieved, with 3,030 wins and 2,452 losses against the production system, and 19,055 ties with it. As expected, the UNIFORM BASELINE lost many more comparisons than it won. Both outcomes were statistically significant according to a binomial test. Again, we refer to the extended lab overview paper [22] for full details.

5 Discussion and Conclusions

The living labs methodology offers great potential to evaluate information retrieval systems in live settings with real users. The Living Labs for Information Retrieval Evaluation (LL4IR) CLEF Lab represents the first attempt at a shared community benchmarking platform in this space. The first edition of LL4IR focused on two use-cases, product search and web search, using a commercial e-commerce website, REGIO, and a commercial web search engine, Seznam. A major contribution of the lab is the development of the necessary API infrastructure, which is made publicly available.

The LL4IR CLEF Lab attracted interest from dozens of teams. There were 12 active participants, but only 2 teams ended up submitting results for the official evaluation (excluding the baseline systems, provided by the organizers). We found that, while many researchers expressed and showed their interest in the lab, our setup with an API, instead of a static test collection, was a hurdle for many. We plan to ease this process of adapting to this new evaluation paradigm by providing even more examples and by organizing tutorials where we demonstrate working with our API.

⁸ If use-cases uploaded their candidate documents in the order that represented their own ranking, then this was available to participants. We plan to change this in the future.

Overall, we regard our effort successful in showing the feasibility and potential of this form of evaluation. For both use-cases, there was an experimental system that outperformed the corresponding production system significantly. It is somewhat unfortunate that in both cases that experimental system was a baseline approach provided by the challenge organizers, nevertheless, it demonstrates the potential benefits to use-case owners as well. One particular issue that surfaced and needs addressing for the product search use-case is the frequent changes in inventory. This appears to be more severe than we first anticipated and represents some challenges, both technical and methodological.

The API infrastructure developed for the LL4IR CLEF Lab offers the potential to host ongoing IR evaluations in a live setting. As such, it is planned that these “challenges” will continue on an ongoing basis post-CLEF, with an expanding number of use-cases as well as refinements to the existing use-cases.⁹ In fact, a second round of our evaluation challenge is already underway at the time of writing, with some modifications to the initial setup. A more detailed analysis of the use-cases, including results from the second evaluation round, and a discussion of ideas and opportunities for future development is provided in the extended lab overview paper [22].

Acknowledgements

We would like to acknowledge the support of (in alphabetical order): the CLEF Initiative; the Dutch national program COMMIT; the REGIO Játék online toy store; and the Seznam commercial search engine. We would also like to thank the participants for their submissions and interest in the lab.

Bibliography

- [1] J. Allan, B. Croft, A. Moffat, and M. Sanderson. Frontiers, challenges, and opportunities for information retrieval: Report from SWIRL 2012 the second strategic workshop on information retrieval in lorne. *SIGIR Forum*, 46(1):2–32, 2012.
- [2] L. Azzopardi and K. Balog. Towards a living lab for information retrieval research and development. A proposal for a living lab for product search tasks. In *CLEF’11*, 2011.
- [3] K. Balog, D. Elswailer, E. Kanoulas, L. Kelly, and M. D. Smucker. Report on the CIKM workshop on living labs for information retrieval evaluation. *SIGIR Forum*, 48(1):21–28, 2014.
- [4] K. Balog, L. Kelly, and A. Schuth. Head first: Living labs for ad-hoc search evaluation. In *CIKM’14*, 2014.
- [5] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, Sept. 2002. ISSN 01635840.
- [6] T. Brodt and F. Hopfgartner. Shedding light on a living lab: The CLEF NEWS-REEL open recommendation platform. In *IiX’14*, 2014.
- [7] O. Chapelle, T. Joachims, F. Radlinski, and Y. Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions on Information Systems (TOIS)*, 30:1–41, 2012.

⁹ See <http://living-labs.net/> for details.

- [8] A. B. Ghirmatsion and K. Balog. Probabilistic field mapping for product search. In *CLEF 2015 Online Working Notes*, 2015.
- [9] K. Hofmann, S. Whiteson, and M. de Rijke. A probabilistic method for inferring preferences from clicks. In *CIKM'11*, page 249, 2011.
- [10] T. Joachims. Evaluating retrieval performance using clickthrough data. In J. Franke, G. Nakhaeizadeh, and I. Renz, editors, *Text Mining*, pages 79–96. Physica/Springer, 2003.
- [11] J. Kamps, S. Geva, C. Peters, T. Sakai, A. Trotman, and E. Voorhees. Report on the SIGIR 2009 workshop on the future of IR evaluation. *SIGIR Forum*, 43(2): 13–23, 2009.
- [12] D. Kelly, S. Dumais, and J. O. Pedersen. Evaluation challenges and directions for information-seeking support systems. *Computer*, 42(3):60–66, 2009.
- [13] L. Kelly, P. Bunbury, and G. J. F. Jones. Evaluating personal information retrieval. In *ECIR'12*, 2012.
- [14] J. Kim, X. Xue, and W. B. Croft. A probabilistic retrieval model for semistructured data. In *ECIR'09*, 2009.
- [15] R. Kohavi. Online controlled experiments. In *SIGIR'13*, 2013.
- [16] T.-Y. Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, 2009.
- [17] T.-Y. Liu, J. Xu, T. Qin, W. Xiong, and H. Li. LETOR: Benchmark dataset for research on learning to rank for information retrieval. In *LR4IR'07*, 2007.
- [18] F. Radlinski and N. Craswell. Optimized interleaving for online retrieval evaluation. In *WSDM'13*, 2013.
- [19] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In *CIKM'08*, 2008.
- [20] P. Schaer and N. Tavakolpoursaleh. GESIS at CLEF LL4IR 2015. In *CLEF 2015 Online Working Notes*, 2015.
- [21] A. Schuth, F. Sietsma, S. Whiteson, D. Lefortier, and M. de Rijke. Multileaved comparisons for fast online evaluation. In *CIKM'14*, 2014.
- [22] A. Schuth, K. Balog, and L. Kelly. Extended overview of the Living Labs for Information Retrieval Evaluation (LL4IR) CLEF lab 2015. In *CLEF 2015 Online Working Notes*, 2015.
- [23] A. Schuth, R.-J. Brintjes, F. Büttner, J. van Doorn, C. Groenland, H. Oosterhuis, C.-N. Tran, B. Veeling, J. van der Velde, R. Wechsler, D. Woudenberg, and M. de Rijke. Probabilistic multileave for online retrieval evaluation. In *SIGIR'15*, 2015.
- [24] A. Schuth, K. Hofmann, and F. Radlinski. Predicting search satisfaction metrics with interleaved comparisons. In *SIGIR'15*, 2015.