This is a repository copy of *Evaluation: Thinking Outside the (Search) Box*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/93016/

Version: Accepted Version

Proceedings Paper:
Clough, P.D. (2015) Evaluation: Thinking Outside the (Search) Box. In: FIRE '14 Proceedings of the Forum for Information Retrieval Evaluation. FIRE '14 Forum for Information Retrieval Evaluation, 05-07 Dec 2014 , Bangalore, India. ACM , New York, NY, USA , pp. 1-9. ISBN 9781450337557

Reuse
Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Evaluation: Thinking Outside the (Search) Box

Paul Clough
Information School
University of Sheffield
Sheffield, UK
p.d.clough@sheffield.ac.uk

ABSTRACT

Evaluation of IR systems has typically focused on the system and specifically assessing the quality of a ranked list of results with respect to a query. However, IR functionality is typically just one component amongst many that are used to help support users' wider information seeking activities. Many systems that include a search box also provide features, such as faceted lists, subject hierarchies, visualizations and recommendations to help users find information. In this paper I discuss experiences gained from developing a system to support exploration and discovery in digital cultural heritage. In particular I focus on the development of system components to support search and navigation and how the different components were evaluated within the development life-cycle of the project. The importance of taking a holistic approach to evaluation, as well as utilising evaluation approaches from domains other than IR, is emphasized. In short, we need to be thinking outside the (search) box when it comes to evaluation in IR.

CCS Concepts

•Information systems → Evaluation of retrieval results; Retrieval effectiveness;

Keywords

IR evaluation, case studies, component-based evaluation

1. INTRODUCTION

Nowadays search is ubiquitous and often working behind the scenes to power websites and search-based applications. Commonly the search box is just one component of many provided to support users' wider information seeking and encountering behaviors [20, 38]. For example, think about a system such as Amazon.com that includes navigational aids (e.g., facets and subject categories), social interaction features (e.g., likes and sharing) and recommendations (e.g., "people who bought this also bought this"), as well as more traditional support for query formulation (e.g., auto suggest) and query reformulation (e.g., related searches). Similarly, next generation library catalogs also incorporate this kind of search functionality [6]. Increasingly more complex systems are being designed to support more open-ended, varied and complex search tasks [36, 19] and central to developing such systems are questions of evaluation [25]. After all, without evaluation we cannot quantify the performance of an IR system or its value to end users and service providers.

Traditionally the focus of evaluation in IR, particularly with respect to more system-oriented evaluation and large-scale evaluation campaigns (e.g., TREC and CLEF), has been the search box component (i.e., query and ranked list of results) using test collections [33, 37]. This approach, characteristic of intrinsic evaluation, aids system and algorithm development as components of the system can be isolated, performance evaluated and parameters optimized with respect to a gold standard result, often pre-defined by the evaluators. However, user-oriented approaches are also important to enable the system to be evaluated in a more holistic way, including user-system interaction and the user interface. This approach, characteristic of extrinsic evaluation, considers the value of the system in use within more realistic settings; either as an embedded system or serving a precise function for the user (e.g., in helping the user complete a goal or task). This has typically been the focus of Interactive IR (IIR) evaluations [24]. However, this assumes a working prototype system is available and does not readily support evaluation during system (and sub-system) development. In reality we need a combination of approaches as applications consisting of multiple components are developed and integrated into operational systems.

This paper describes experiences gained from developing an information system that incorporates various finding aids to support users' exploration of digital cultural heritage [2, 11]. The system integrated a number of sub-systems or components into an application that involved both intrinsic and extrinsic evaluations during component development (i.e., formative evaluation) and following the implementation of an integrated prototype (i.e., summative evaluation). Various evaluation methods were used from areas, such as taxonomy design, recommender systems and visualization. In particular this paper highlights the need for multiple evaluation methods during system development. Section 2 provides terminology for multi-component systems, Section 3 discusses evaluation; Section 4 describes the PATHS system, Section 5 the various evaluation approaches used during development; and Section 6 concludes the paper with a discussion.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

FIRE '14, December 05-07, 2014, Bangalore, India

© 2015 ACM. ISBN 978-1-4503-3755-7/15/08…$15.00

DOI: http://dx.doi.org/10.1145/2824864.2824890

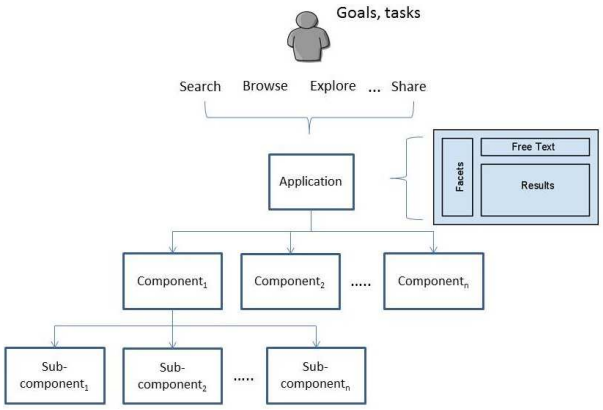

Figure 1: An 'application' consisting of 'components' and 'sub-components'.

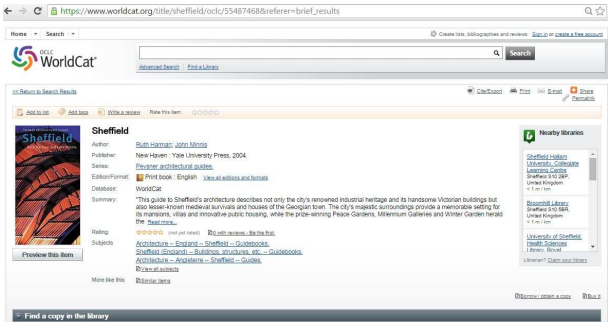

Figure 2: An example multi-component application: Worldcat.org (item-level page).

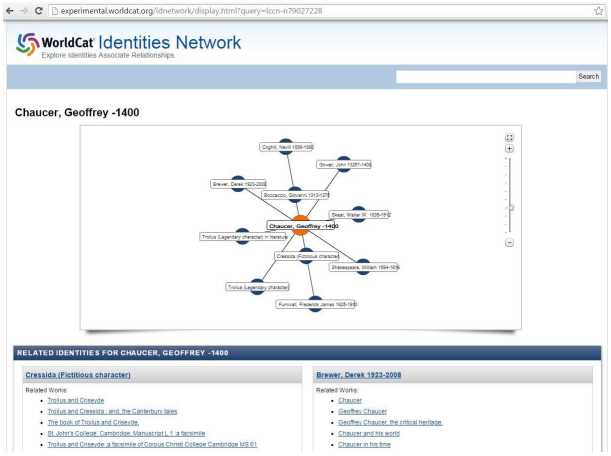

Figure 3: An example multi-component application: Worldcat.org (author network view).

2. MULTI-COMPONENT SYSTEMS

Increasingly systems are becoming more complex, consisting of various sub-systems or components that implement specific functionality, for example to support users' information seeking activities [25]. This paper uses the notion of *component* to refer to units that can be integrated to form an application. Figure 1 illustrates this: an application consists of components that work together to help the user accomplish some overall goal or task. The application is available to users through the User Interface (UI), which acts as mediator between system-user, surfacing component functionality.

From this perspective, an IR system is a component that provides support for querying and relevance ranking. Of course the IR system may also form the basis for other features (e.g., browsing aids) depending on how components are implemented. Functionalities to support users' wider information seeking activities could include recommender systems, subject hierarchies and visualizations. Figures 2 and 3 show an example of a multi-component system: the Worldcat.org universal library catalog. This includes the typical search box, advanced search features, library finding aids, related/similar items and reviews (Figure 2) and a network visualization of related authors (Figure 3).

In addition, components may also be sub-divided into sub-components (Figure 1). For example, an IR system component could be sub-divided into discrete units, such as indexing, stemming, query translation (in the case of CLIR), relevance feedback, etc. These are often chained together to implement component functionality. From an evaluation perspective it would be possible to evaluate the outputs at any level and investigate the effects between component and sub-component performance.

At this point it is worth considering terminology used because the notion of 'system' and 'component' is used in various ways within IR evaluation literature. For example, Hanbury and Ḿuller [18] describe the notion of component-level evaluation, but their use of the term 'component' relates to what Fig. 1 calls sub-component. Similarly, Ferro and Harman [14] describes the GridCLEF track at CLEF, an initiative to investigate the effects of various system components for multilingual information access systems with respect to language (e.g., retrieval models, merging strategies, stemmers and translation resources) on overall system performance. In this paper, component reflects some atomic aspect of specific functionality that can be integrated into an application and the outputs of which can be evaluated.

3. EVALUATION

3.1 Evaluation in IR

To evaluate means to ascertain the value of something or to appraise it and in the case of IR enables the success of an IR system to be quantified. This could involve evaluating characteristics of the IR system itself, such as its retrieval effectiveness, or assessing consumers' acceptance or satisfaction with the system. When preparing an evaluation, key questions to ask include [32]: (i) what to measure (i.e., the evaluation criteria); (ii) how to measure (i.e., measures to quantify the criteria); and (iii) the methodology to use (e.g., benchmarks, simulations or human-centered studies). For decades the primary approach to IR evaluation has been system-centered, focusing on performance, such as retrieval effectiveness or efficiency based on using test collections and the Cranfield paradigm or methodology. However, user- or human-centered approaches are increasingly being used and take into account the user, the user's context and situation, and their interactions with an IR system, perhaps in a real-life operational environment [24, 8].

Evaluation in IR is often distinguished between various levels [35]: (1) evaluation within the IR system context; (2) evaluation within the information seeking context; and (3) evaluation within the work context. Mandl [28] also describes four levels at which Geographic IR (GIR) systems can be evaluated: (i) at the component level (evaluating individual sub-systems, such as a toponym recognizer); (ii) at the system level (evaluating the outputs of a complete IR system, perhaps using test collections); (iii) at the user-system interaction level (assessing an entire GIR system including interfaces and visualizations in a controlled laboratory setting); and (iv) at the user performance level (assessing an operational system in use and its impact on daily work tasks). In practice it is common to make use of various evaluation approaches throughout the development of an IR system and at these various levels. This can range from using test collections to develop, contrast and optimize search algorithms; to conducting lab-based user experiments for improving the design of the user interface; to evaluation carried out in situ as the IR system is deployed and used.

3.2 Evaluation beyond IR

As alluded to previously, the scope of IR systems is widening and going beyond just query-ranked list functionality to include different features to support information searching and seeking. As discussed in Section 2, an application may include an IR system component to support retrieval via querying. However, the addition of features beyond an IR system requires re-thinking how evaluation is carried out. Ultimately it may require learning methods and best practices from communities traditionally outside of IR, such as recommender systems, HCI and information visualization. Additionally, in the development of IR applications, evaluation from the perspective of the software developer, user interface and end user must also be considered (and in context) and reflected in the overall evaluation strategy. Therefore, the following provides some insights on evaluation from these various fields, outside the focus of of traditional IR.

Evaluation of recommender systems: a recommender system guides users to specific items of interest based on a given item (or items) and a user profile [1]. Various approaches have been proposed to evaluate recommender systems, although most are conducted off-line and based on determine predictive accuracy – the extent to which a system can accurately predict a user's rating [21]. Accuracy is often assessed using a 'leave-n-out' approach: a user generated rating is withheld and the system is required to predict its value. The variation between the actual and predicted rating can be measured, for example using Mean Accurate Error (MAE), classification accuracy metrics, precision and recall measures and error rate measures. Similar to IR, the use of techniques that take into account users are becoming increasingly common. For example, the Human Recommender Interaction (HRI) framework proposed by McNee et al. [29].

Evaluation of subject hierarchies: items in a collection are often mapped to subject categories (e.g., thesauri or classification schemes), arranged for navigation hierarchically or as facets. Regardless of whether the hierarchies are created manually or automatically, they must be evaluated [26]. Evaluation approaches can be grouped into the following [17]: (i) the comparison of a hierarchy with an existing gold standard; (ii) the comparison of the hierarchy against a set of pre-defined criteria (e.g., consistency, completeness or clarity); (iii) evaluation of the hierarchy by a group of domain experts; and (iv) the use of statistical measures to automatically evaluate and compare hierarchies. Increasingly, there is a need to evaluate the hierarchies in work and task contexts.

Evaluation of visualizations: information visualizations are inherently complex to evaluate and multiple evaluation approaches exist [31, 22]. Plaisant [31] identifies four areas of focus for evaluation: (i) controlled experiments comparing different design elements (e.g., specific interface widgets), (ii) usability evaluation of a tool; (iii) controlled experiments comparing two or more tools; and (iv) case studies of tools in real settings. The most common approaches tend to focus on assessing user experience based on usability testing and controlled experiments whereby users complete set tasks. Increasingly there are calls for information visualizations to be evaluated in real life settings rather than in artificial lab-based environments.

User-oriented evaluation: approaches for evaluation that involves users in some way typically fall within the area of Interactive IR [24, 8, 23]. Criteria used to assess the system are typically concerned with how well users achieve their goals or tasks and their success and satisfaction with the outputs. Additional criteria can include efficiency, utility, informativeness, usefulness and usability. Many measures are used in IIR evaluation, including task success; time spent completing the task, number of documents viewed/saved, number of interactions, user satisfaction and engagement. Methods for conducting user-oriented evaluation typically include task-based studies conducted in a controlled laboratory setting, side-by-side evaluation and online or live testing. Data collection can involve logs, questionnaires, observations, think-aloud and focus groups [24].

Usability evaluation: usability is a criterion that considers how easy a user interface is to use and is often defined by the following components: learnability, efficiency, memorability, errors and satisfaction[1]. Commonly used questionnaires for gathering feedback on usability include: the System Usability Scale (SUS) [9], the Computer System Usability Questionnaire (CSUQ) [27] and the Questionnaire for User Interface Satisfaction (QUIS) [10]. The NASA-TLX[2] is also often used to rate users' perceived workload in order to assess a task. Usability testing is typically conducted in controlled lab-based environments and may include specialist facilities, such as video recording and eye-tracking.

Software testing: during development various forms of testing can take place including acceptance testing (checking the overall system functions as required), unit testing (assessing a single function, procedure or class), integration testing (assessing whether units tested in isolation work properly when put together), system testing (assessing whether the entire system can cope with real data and heavy loading) and regression testing (checking that the system preserves it functionality in operation). Typical software quality criteria include correctness, reliability, usability, integrity, maintainability and efficiency [30].

[1]Jakob Nielsen provides many useful articles on his site, e.g.: http://www.nngroup.com/articles/usability-101-introduction-to-usability/

[2]Overview of the NASA TLX tool: http://www.nasatlx.com/

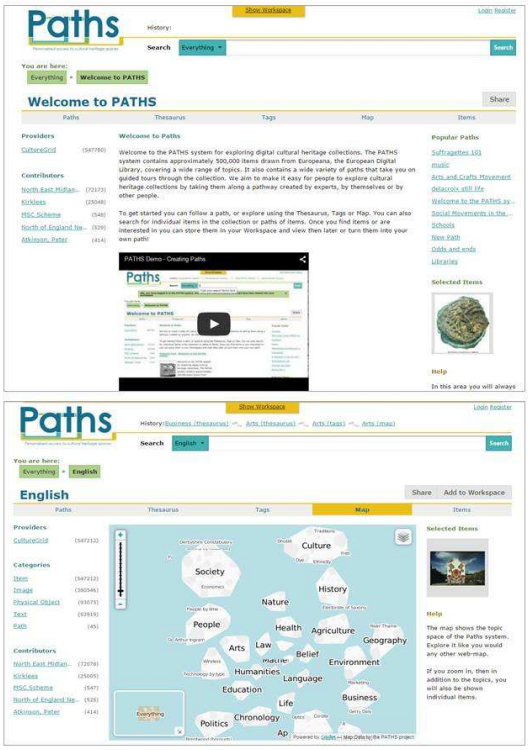

Figure 4: Example screenshots of the PATHS system: the main landing page (top) and collection overview visualization (bottom).

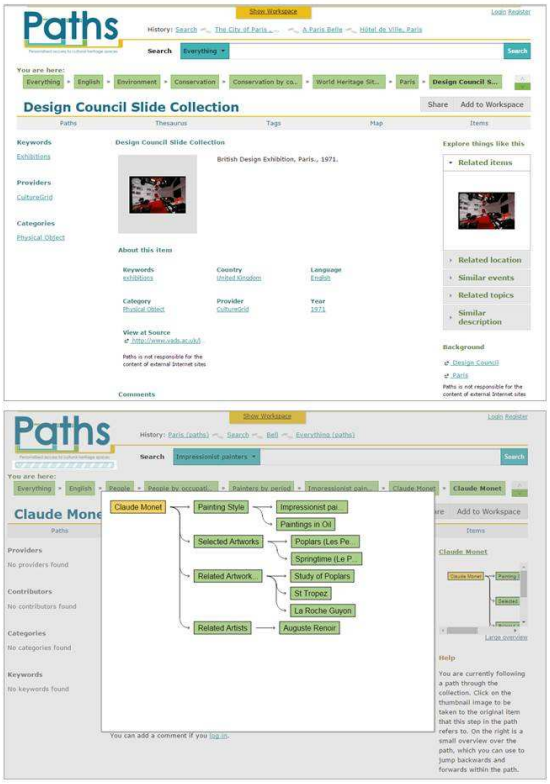

Figure 5: Example screenshots of the PATHS system: item-level view (top) and path editor view (bottom).

4. THE PATHS SYSTEM

The PATHS (Personalized Access To cultural Heritage Spaces) project[3] was funded under the European Commission's FP7 programme and explored various ways of supporting discovery and exploration in large and heterogeneous cultural heritage collections [2, 11]. Prototype systems based on content from Europeana[4], Europe's aggregator for cultural heritage, were developed using a fairly standard user-centered development process: identify requirements, prototype and evaluate. Software development was carried out in an iterative fashion resulting in the production of two main prototypes. These were desktop applications, but components were also re-used to create mobile applications. A central aspect of the system was the use of paths/trails to allow users of the system to organize digital content into guided pathways resembling exhibitions and tours commonly provided for physical collections, such as those found in museums.

The PATHS system might best be described as an information seeking support system or exploratory search system in that it supports multiple finding modes and peoples' wider information activities and making sense of the information found. The basic mantra behind the system was to support: 'Find', 'Collect' and 'Use'. Find includes modes, such as search, browse, explore and discover; during collect users can gather materials for later use, supported by a workspace or bookmarking functions; and use supports making subsequent use of collected items, for example creating learning objects, and being able to share them (i.e., social interaction). The PATHS system made use of multiple UI components for supporting users: (i) standard search box and facets, (ii) map-based visualization, (iii) a thesaurus based on a data-driven subject hierarchy, (iv) links to related items, (v) item-level recommendations, and (vi) features for creating, editing, publishing and following paths. Figures 4 and 5 show example screenshots of the system .

5. EVALUATING THE PATHS SYSTEM

Using the terminology from Figure 1, the overall PATHS system is an application that consists of a number of components. Development was undertaken in a distributed manner with one lead partner responsible for overall system architecture design, software development and system integration, and academic institutions leading R&D activities to develop system components, carry out evaluation activities, design the user interfaces and develop UI components.

The overall application development process followed an Agile approach whereby prototypes were incrementally developed, which in our case included the design and testing of underlying components. These components were developed in parallel with the system architecture and user interface, and following requirements gathering. The functionality of components, such as related/similar items and recommen-

[3]Download project reports from: http://www.paths-project.eu/

[4]Europeana portal: http://www.europeana.eu/portal/

dations, was evaluated by researchers during their development, making use of various approaches including test collections, crowdsourcing and controlled lab-based studies (Section 5.1). Components were also evaluated through the evaluation of the overall integrated system (Section 5.4) whereby users were asked to rate the usefulness of and satisfaction with system components (e.g., recommendations, related items, etc.) when carrying out various tasks. When considering the development of applications, such as the PATHS system, what is striking is the wide variety of evaluation methods used throughout the life-cycle - some more formal than others - and by different stakeholders in the project. Beyond the evaluation activities previously mentioned, this also includes system and integration testing by the system architects (Section 5.2), and testing of user interface designs (Section 5.3).

5.1 Evaluating the components

Following requirements gathering and the creation of a functional specification, components were identified that encapsulated specific functionalities, such as search and recommendation. Typically components were evaluated in isolation, for example to identify the most effective algorithm or optimize parameters, before implementation in the architecture and integration into a working prototype. Component evaluation went far beyond typical IR evaluation to consider other finding aids, such as subject hierarchies and visualizations to provide collection overviews.

Search box: this component made use of a standard Solr back-end implementation. The index comprised approx. 1.4 million items, a subset of Europeana. Although a standard IR test collection approach could have been adopted to tune and evaluate retrieval performance of the search component, this was problematic in our case as no suitable test collection existed. Therefore, the approach taken was to judge the top 10 ranked results were for (topical) relevance given a set of queries and compute retrieval effectiveness using $P@10$. Representative queries were sampled from a Europeana search log collected during the project.

Recommendations: non-personalized recommendations were implemented using a content-based approach through mining item co-occurrence information from a sample of search logs from the Europeana portal [13]. Items viewed consecutively in a session were extracted and made available to users along with links to "related items" based on identifying similar items in the collection. Related items were evaluated separately (see below) and the usefulness of non-personalized recommendations was considered as part of evaluating the integrated prototype.

We also experimented with session-based personalized recommendations implemented using Personalized PageRank [12]. The idea was to extract information from items viewed during a session to create a user profile that could be used to offer suggestions to further items of potential interest to the user. To evaluate various methods of producing recommendations we used a sample of Europeana search logs to create an evaluation dataset (gold standard) of users' interactions. A set of 1,000 sessions containing at least 5 viewed items was randomly selected from the logs. Sequences of 5 viewed items were then extracted and used for evaluation with the goal being to predict the nth item from the log training on the prior $n-1$ items. This is similar in spirit to the recent TREC Session Track activity to facilitate Cranfield-style evaluation across sessions rather than a single query-response action. A precision score was used to indicate whether the relevant item (the 5th item in the session) was present in the first 10 items shown to the user ($P@10$). A variant of $P@10$ was also used that scored a recommendation as correct if from the same Wikipedia category as the relevant item ($Pcat@10$). This followed the assumption that items within the same topical category might also be of interest to users.

Visualizations: the application included a novel approach for exploring document collections using a map-based visualization [16]. Evaluating visualizations is inherently complex and can include aspects such as the user experience, visual data analysis and reasoning, collaborative data analysis or work practices [22]. A preliminary evaluation of the map-based visualization component was carried out with the goal of gathering feedback from users regarding the intuitiveness, utility, and usability of the map for an undirected task in which users were instructed to gain an overview of items in a document collection. An experiment was carried out in a controlled lab-based setting in which 10 participants were asked to explore the document collection for up to 10 minutes. Participants then rated the system using a modified version of the System Usability Scale that we adapted for evaluation of interactive digital maps. The SUS asks participants to rate a system for 10 questions (e.g., "I thought the system was easy to use" and "I felt very confident using the system") on a 5-point Likert scale. Participants were also asked to provide qualitative feedback on the visualization.

Related or similar items: a component was developed to compute the similarity between pairs of items [3]. This basic functionality was used within a range of features in the PATHS system, such as clustering related items, forming intra-collection links and providing non-personalized recommendations. Various methods were investigated for computing the similarity between pairs of metadata records in Europeana in order to estimate the similarity between items. Similarity measures are often evaluated by comparing the scores they generate against human judgments.

To generate a suitable dataset for the document collection in the PATHS system, based on Europeana, we ran a crowdsourcing experiment. A total of 295 random pairs were sampled from our dataset and shown to workers from CrowdFlower[5], together with a 5-point Likert scale (0=completely unrelated items; 5=completed related items). To evaluate the effectiveness of methods to identify similar items, the scores generated by each method were compared with the 295 gold standard pairs. Pearson and Spearman correlation coefficients were computed between the similarity estimates produced by the automatic approaches and the average score obtained from the human judgments. In addition, methods to compute various 'types' of similarity were also developed, such as identifying pairs of items with similar author, similar people involved, similar time period, similar location, similar events, similar location and similar description. Additional datasets were generated to evaluate methods for computing typed similarity.

Subject hierarchies: to aid browsing and navigation of the collection the PATHS system included a subject hierarchy [16]. Data-driven approaches were developed to induce hierarchical topic structures automatically and enable

[5]http://www.crowdflower.com/

the mapping of items from the collection onto the hierarchy. Similar to evaluating the related items component, the evaluation of methods to generate subject hierarchies required human judgments to assess criteria, such as consistency and completeness. Specifically, two aspects were considered: whether categories in the hierarchy were cohesive (i.e., items in a category were closely related) and whether parent-child relationships were sensible (i.e, that the parent and child categories were obviously related). In addition we assessed whether the hierarchies were perceived to provide an overview of items in a collection and whether items were well-placed in the hierarchy. To evaluate these criteria, an online experiment was created whereby an interactive hierarchy (one hierarchy randomly selected from the different generation methods) was shown to participants (225 in total) who were able to explore a subset of items from the PATHS system and provide feedback on different aspects using a 7-point semantic differential scale (-3 to +3). The online experimental system also recorded users' interactions (e.g., clicks and dwell time) which could be combined with the participants' ratings to score the hierarchies.

Additionally, the results of the four metrics were compared to a second task-based evaluation to investigate whether the simpler metrics could be used as predictors of task performance. A two-stage online experiment was created and carried out by 64 participants. In the first stage three hierarchies were selected and shown alongside each other and participants asked to rate each hierarchy for certain characteristics (e.g., understandability of the headings and the organization of concepts) and make preference judgments (i.e., state their preferred hierarchy). In addition, participants also completed given search tasks, in our case using a Simulated Leisure Task methodology, using one of the hierarchies. In this experiment, 5-point semantic differential scales were used to gather feedback from users on aspects such as their satisfaction with the hierarchy. Qualitative data were also gathered and analyzed in order to evaluate and rank the hierarchies.

5.2 Testing the System Architecture

The PATHS system was developed as a distributed architecture based on web-service technologies (HttpGet, HttpPost, Soap and JSON). Components and data were surfaced through APIs enabling development of various front-end applications, including the main web-based desktop application (integrated prototype) and a simpler tablet-based version. From the perspective of the system architecture evaluation focused on testing web services and the API through load testing: simulating calls to the API and monitoring the system speed and overall performance, verifying API responses and comparing system functionality with the system requirements laid out in the functional specification. Testing from the system architecture or software integration perspective included unit and integration testing.

5.3 Evaluating Interface Designs

The user interface was designed using a typical user-centered design approach involving gathering requirements, prototyping and evaluation [15]. The UI design included three main stages: (i) low-fidelity sketch-based storyboards, (ii) high-fidelity interaction designs, including design of UI components and modeling interaction flows, and (iii) implementation of the prototype front-end. Feedback was gathered at each stage and the final working UI was integrated into the final working prototype and evaluated in lab-based and naturalistic settings (Section 5.4).

However, following the interface design, but prior to the lab-based evaluations, an expert evaluation was conducted based on the *Cognitive Walkthrough* technique. This seeks to identify any usability issues that may arise from users carrying out core tasks with the system and also highlight areas of the interface and integrated system for potential improvement. The evaluation was conducted by a usability expert for various tasks and during interaction the following questions were posed and rated (yes, maybe and no): Will users know what to do? Will users see how to do it? Will users understand whether their action was correct or not?

5.4 Evaluating the Integrated Prototype

Following component development and system infrastructure, an integrated prototype (or application) was created based on an iterative design [15]. The overall system consisted of two main prototype development cycles; each evaluated using a laboratory-based evaluation methodology and field trials.

Lab-based evaluation: evaluation of the application in the lab employed a human-centered approach based on an Interactive IR paradigm and in particular the use of Simulated Work Tasks [7]. This enabled evaluating the efficiency and effectiveness of the application in supporting users' information needs and work tasks under controlled conditions, together with gathering feedback from participants regarding the usability and satisfaction of using the application. Figure 6 shows the protocol used in the lab-based evaluations, including the main activities involved, the inputs to these activities and the outputs.

Data were captured using various mechanisms: questionnaires involving 5/7-point Likert scales, semantic differentials and qualitative feedback; audio recordings of post-study interviews; screen recordings and interaction logs from the Morae system; customized user-system interaction logs; and results from a cognitive styles test. In the evaluation of the first interactive prototype, participants (22 in total) carried out tasks to simulate four information seeking modes: (i) simple fact-finding, (ii) extended fact-finding; (iii) open-ended browsing; and (iv) exploration. Participants were also given a fifth task unstructured task in which they had to construct a path using the system. In the evaluation of the second prototype system, participants (34 in total) were presented with five short structured tasks: (i) finding and following a path; (ii) gaining an overview of the collection; (iii) fact-finding; (iv) open-ended browsing; and (v) exploration.

The initial four tasks were presented to users in a different order based on a Latin Square design to counter-act learning and task order effects. The study was carried out in a controlled usability lab (known as the iLab) at Sheffield University. To evaluate the application, various criteria and measures were used. Data from logs were used to indicate features used by participants during the tasks who were also asked to provide feedback on their task performance, usability, usefulness and satisfaction for different interface components, as well as their overall satisfaction with the system.

Field trials: were carried out to assess the usability and usefulness of the desktop and mobile versions of the application. Field trials allow participants to use the system in their own time, preferred environment and using their own

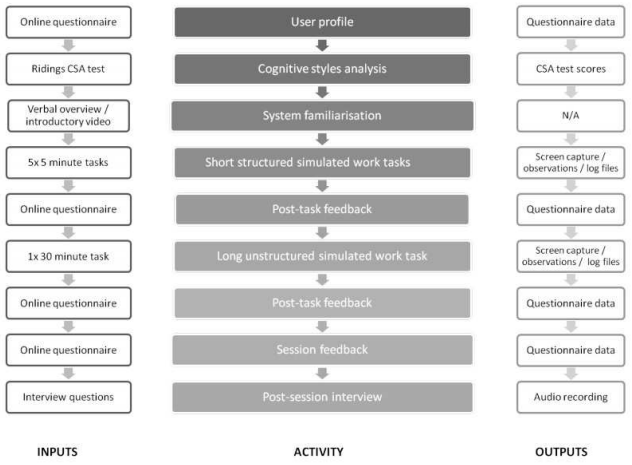

Figure 6: Protocol used for the lab-based prototype evaluations.

technology. This more naturalistic form of testing complemented the findings of the lab-based study. Participants (34 in total) were given a brief demonstration of the PATHS system, to use it to meet any ongoing information requirements, to create a path of their own design and to meet a realistic information need. Participants were asked to use the system over an extended period of time (no more than two weeks). Feedback from participants was collected using questionnaires, interviews and focus groups at the end of the field trial period. In addition, a method based on using online dairies was also experimented with as a mechanism for recording participants' experiences during the study.

6. SUMMARY

Search-based applications typically involve multiple components and require evaluation beyond what we traditionally do in IR. In this paper I have highlighted some of the wider issues involved in evaluating multi-component systems, especially with respect to components other than the IR system and within the development life-cycle. The example case study, the PATHS application, demonstrates the use of multiple approaches to evaluation throughout the life-cycle, including the use of system- and user-oriented methods for developing components and evaluating prototype applications that combine various components through a user interface to support users' information seeking and information use more widely.

Evaluating IR systems and search-based applications will continue to be an important area of research interest within the community, but we need to be thinking outside the (search) box when it comes to evaluation in IR. Many issues and challenges still lie ahead that include the following:

- **User- and system-oriented approaches**: these should be viewed as complementary rather than at odds with each other. Developing systems will typically require a range of evaluation methods from both approaches and the use of simulations [4] and living labs [25] may be ways of bringing the fields together and supporting evaluating of more complex multi- component systems.
- **To inform and predict**: ultimately the results of user studies should be used to inform the design of system-oriented evaluation studies (e.g., in the design of appropriate test collections and effectiveness measures); in turn the results of system-oriented evaluations should be able to predict the performance of systems in real life operation. This requires computer scientists and information scientists working together.
- **Relationships between criteria**: the relationship between system performance measures and user-oriented criteria, such as usability and user satisfaction, need to be studied and better understood [34]. The effects of users' context (e.g., task and individual differences) on criteria need to be better understood and modeled.
- **Sharing evaluation practices**: we need to learn methodologies and best practices in evaluation from fields outside of IR (and vice-versa – we can inform other fields about the best practices in IR evaluation). For example, through networking events (e.g., workshops and summer schools). This also includes developing and making available example IR system and application evaluation case studies.
- **Beyond ad hoc search**: the focus for benchmarks in IR is typically ad hoc search. However, user interactions with IR systems and applications are far richer [36, 19]. Therefore, we must consider how to develop benchmarks for other modes of interaction, such as browsing. In addition, test collection-style resources must be created for the evaluation of components other than the IR system, such as subject hierarchies and recommender systems.
- **Whole=sum of parts?**: typically components and sub-components are evaluated in isolation, but how these would affect each other when integrated is not well understood. This may require more evaluation exercises, such as GridCLEF [14], to determine relationships and effects between (sub-) components.
- **Whole-page relevance**: there has been recent work on evaluating beyond a single ranked list and evaluating a whole results page that may include, for example, different verticals and ads. Tools, such as SASI, should be further explored that enable isolating and assessing UI components on a search interface [5].

7. ACKNOWLEDGMENTS

The research leading to these results was carried out as part of the PATHS project funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 270082. The author gratefully acknowledges the contributions of all partners in the PATHS project, and all participants in the user requirements and evaluation activities. Thanks also to the European Science Foundation and the Evaluating Information Access Systems (ELIAS) Research Networking Programme for providing travel support.

8. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.

[2] E. Agirre, N. Aletras, P. D. Clough, S. Fernando, P. Goodale, M. M. Hall, A. Soroa, and M. Stevenson. PATHS: A system for accessing cultural heritage collections. In *51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, Proceedings of the Conference System Demonstrations, 4-9 August 2013, Sofia, Bulgaria*, pages 151–156, 2013.

[3] N. Aletras, M. Stevenson, and P. Clough. Computing similarity between items in a digital library of cultural heritage. *J. Comput. Cult. Herit.*, 5(4):16:1–16:19, Jan. 2013.

[4] L. Azzopardi, K. Järvelin, J. Kamps, and M. D. Smucker. Report on the sigir 2010 workshop on the simulation of interaction. *SIGIR Forum*, 44(2):35–47, Jan. 2011.

[5] P. Bailey, N. Craswell, R. W. White, L. Chen, A. Satyanarayana, and S. Tahaghoghi. Evaluating whole-page relevance. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 767–768, New York, NY, USA, 2010. ACM.

[6] T. Ballard and A. Blaine. User search-limiting behavior in online catalogs: Comparing classic catalog use to search behavior in next-generation catalogs. *New Library World*, 112(5/6):261–273, 2011.

[7] P. Borlund. The iir evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research. An International Electronic Journal*, 8(3), 2003.

[8] P. Borlund. *User-Centred Evaluation of Information Retrieval Systems*, pages 21–37. John Wiley and Sons, Ltd, 2009.

[9] J. Brooke. SUS: A quick and dirty usability scale. In P. W. Jordan, B. Weerdmeester, A. Thomas, and I. L. Mclelland, editors, *Usability evaluation in industry*. Taylor and Francis, London, 1996.

[10] J. P. Chin, V. A. Diehl, and K. L. Norman. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '88, pages 213–218, New York, NY, USA, 1988. ACM.

[11] P. Clough, P. Goodale, M. M. Hall, and M. Stevenson. Supporting exploration and use of digital cultural heritage materials: the paths perspective. In I. Ruthven and G. Chowdhury, editors, *Cultural Heritage Information Access and management*. 2015.

[12] P. Clough, A. Otegi, and E. Agirre. Personalized page rank for making recommendations in digital cultural heritage collections. pages 49–52, 2014.

[13] P. Clough, A. Otegi, E. Agirre, and M. M. Hall. Implementing recommendations in the paths system. In u. Bolikowski, V. Casarosa, P. Goodale, N. Houssos, P. Manghi, and J. Schirrwagen, editors, *Theory and Practice of Digital Libraries – TPDL 2013 Selected Workshops*, volume 416 of *Communications in Computer and Information Science*, pages 169–173. Springer International Publishing, 2014.

[14] N. Ferro and D. Harman. Clef 2009: Grid@clef pilot track overview. In *Proceedings of the 10th Cross-language Evaluation Forum Conference on Multilingual Information Access Evaluation: Text Retrieval Experiments*, CLEF'09, pages 552–565, Berlin, Heidelberg, 2009. Springer-Verlag.

[15] P. Goodale, P. Clough, N. Ford, M. Hall, M. Stevenson, S. Fernando, N. Aletras, K. Fernie, P. Archer, and A. de Polo. User-centred design to support exploration and path creation in cultural heritage collections. In *Proceedings of EuroHCIR2012*, volume 909, pages 75–78, 2012.

[16] M. M. Hall and P. D. Clough. Exploring large digital library collections using a map-based visualisation. In *Research and Advanced Technology for Digital Libraries*, volume 8092 of *Lecture Notes in Computer Science*, pages 220–231, 2013.

[17] M. M. Hall, S. Fernando, P. Clough, A. Soroa, E. Agirre, and M. Stevenson. Evaluating hierarchical organisation structures for exploring digital libraries. *Information Retrieval*, 17(4):351–379, 2014.

[18] A. Hanbury and H. Ḿuller. Automated component-level evaluation: Present and future. In M. Agosti, N. Ferro, C. Peters, M. de Rijke, and A. F. Smeaton, editors, *CLEF*, volume 6360 of *Lecture Notes in Computer Science*, pages 124–135. Springer, 2010.

[19] A. Hassan Awadallah, R. W. White, P. Pantel, S. T. Dumais, and Y.-M. Wang. Supporting complex search tasks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 829–838, New York, NY, USA, 2014. ACM.

[20] M. A. Hearst. *Search User Interfaces*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[21] J. L. Herlocker and J. A. Konstan. Content-independent task-focused recommendation. *IEEE Internet Computing*, 5(6):40–47, 2001.

[22] T. Isenberg, P. Isenberg, J. Chen, M. Sedlmair, and T. Moller. A systematic review on the practice of evaluating visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2818–2827, 2013.

[23] K. Järvelin, P. Vakkari, P. Arvola, F. Baskaya, A. Järvelin, J. Kekäläinen, H. Keskustalo, S. Kumpulainen, M. Saastamoinen, R. Savolainen, and E. Sormunen. Task-based information interaction evaluation: The viewpoint of program theory. *ACM Trans. Inf. Syst.*, 33(1):3:1–3:30, 2015.

[24] D. Kelly. Methods for evaluating interactive information retrieval systems with users. *Found. Trends Inf. Retr.*, 3(1—2):1–224, Jan. 2009.

[25] D. Kelly, S. Dumais, and J. O. Pedersen. Evaluation Challenges and Directions for Information-Seeking Support Systems. *Computer*, 42(3):60–66, Mar. 2009.

[26] D. Lawrie, W. B. Croft, and A. Rosenberg. Finding topic words for hierarchical summarization. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 349–357, New York, NY, USA, 2001. ACM.

[27] J. R. Lewis. Ibm computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *Int. J. Hum.-Comput. Interact.*, 7(1):57–78, Jan. 1995.

[28] T. Mandl. Evaluating gir: Geography-oriented or

user-oriented? *SIGSPATIAL Special*, 3(2):42–45, July 2011.

[29] S. M. McNee, J. Riedl, and J. A. Konstan. Making recommendations better: An analytic model for human-recommender interaction. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1103–1108, New York, NY, USA, 2006. ACM.

[30] G. J. Myers and C. Sandler. *The Art of Software Testing*. John Wiley and Sons, 2004.

[31] C. Plaisant. The challenge of information visualization evaluation. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '04, pages 109–116, New York, NY, USA, 2004. ACM.

[32] C. J. V. Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[33] S. Robertson. On the history of evaluation in IR. *J. Information Science*, 34(4):439–456, 2008.

[34] M. Sanderson, M. L. Paramita, P. Clough, and E. Kanoulas. Do user preferences and evaluation measures line up? In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 555–562, New York, NY, USA, 2010. ACM.

[35] T. Saracevic. Evaluation of evaluation in information retrieval. In *SIGIR'95, Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington, USA, July 9-13, 1995 (Special Issue of the SIGIR Forum)*, pages 138–146, 1995.

[36] G. Singer, U. Norbisrath, and D. Lewandowski. Ordinary search engine users carrying out complex search tasks. *Journal of Information Science*, 39(3):346–358, 2013.

[37] E. M. Voorhees and D. K. Harman. *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press, 2005.

[38] M. L. Wilson, B. Kules, m. c. schraefel, and B. Shneiderman. From keyword search to exploration: Designing future search interfaces for the web. *Found. Trends Web Sci.*, 2(1):1–97, Jan. 2010.