



PROMISE

Participative Research labOratory for Multimedia
and Multilingual Information Systems Evaluation

FP7 ICT 2009.4.3, Intelligent Information Management

Deliverable 4.2

Tutorial on Evaluation in the Wild

Version 1.0, September 2012



Document Information

Deliverable number: 4.2
Deliverable title: Tutorial on Evaluation in the Wild
Delivery date: 31/08/2011
Lead contractor for this deliverable: ZHAW
 Stefan Rietberger, ZHAW
 Melanie Imhof, ZHAW
 Martin Braschler, ZHAW
 Richard Berendsen, UvA
 Anni Järvelin, SICS
 Preben Hansen, SICS
Author(s): Alba García Seco de Herrera, HES-SO
 Theodora Tsikrika, HES-SO
 Mihai Lupu, TUW
 Vivien Petras, UBER
 Maria Gäde, UBER
 Michael Kleineberg, UBER
 Khalid Choukri, ELDA
Participant(s): CELCT, ELDA, HES-SO, SICS, TUW, UvA, ZHAW
Workpackage: 4
Workpackage title: Evaluation Metrics and Methodology
Workpackage leader: UvA
Dissemination Level: PU – Public
Version: 1.0
Keywords: Evaluation, operational systems, application, stakeholders, industry, practical significance

History of Versions

Version	Date	Status	Author (Partner)	Description/Approval Level
0.1	15/11/2011	Outline	ZHAW	Document created
0.2	24/08/2012	Draft	ZHAW	Draft version for content review
1.0	31/08/2012	Finished	ZHAW	Complete for submission

Abstract

This deliverable describes a methodology to perform an application-centric evaluation of operational information access systems. The application is treated as a black-box. Moreover the presented methodology can evaluate aspects of functionality which typical users can actually access and experience.

The methodology estimates the user perception based on a wide range of criteria that cover four categories, namely indexing, document matching, the quality of the search results and the user interface of the system. The criteria are established best practices in the information retrieval domain as well as advancements for user search experience. For each criterion a test script has been defined that contains step-by-step instructions, a scoring schema and adaptations for the three PROMISE use case domains.

The proposed methodology can be used to monitor a single application over time, to conduct a direct comparison of a few applications or to arrange an evaluation campaign. The evaluation requires the tested search application to have text search functionality. Also we recommend to only compare applications within the same use case domain.

To validate the presented methodology an evaluation campaign was conducted, where each participating PROMISE partner evaluated sites either from the PROMISE use case domains or from the enterprise search domain. The results and insights from this campaign served as a basis to further improve and refine the proposed criteria.

In the end of this deliverable a practical step by step tutorial to conduct an evaluation using the methodology as described is given.

Table of Contents

Document Information.....	3
Abstract.....	4
Table of Contents.....	5
Executive Summary	6
1 Introduction	7
2 Related Work	7
3 Definitions	11
4 Black Box Application Evaluation.....	11
4.1 Scope and Limitations	11
4.2 Information Access Application Model	13
4.3 Description of Methodology	14
4.4 Evaluation Scenarios.....	16
4.5 Scoring	17
4.6 Revised Criteria and Test Scripts	18
4.6.1 Index Criteria	19
4.6.2 Matching Criteria.....	30
4.6.3 User Interface Criteria.....	36
4.6.4 Search Results Criteria	58
4.7 Testing Examples.....	62
4.7.1 Freshness.....	62
4.7.2 Sorting of Result List.....	65
4.7.3 Navigational Queries	66
5 Conducting an Evaluation, Step By Step	69
6 Validation Efforts.....	72
7 Outlook.....	72
Acknowledgements.....	74
References.....	74

Executive Summary

A methodology is presented which enables industry practitioners to comparatively evaluate information access applications. This “black box application evaluation” methodology is based on testing application criteria which are modeled after prototypical users’ behaviours and needs. By basing criteria and especially test descriptions on typical users and their behaviour, testers are usually not required to have expert domain knowledge.

The first part of this deliverable deals with the academic background of the methodology. It is based on earlier work and has been significantly improved since. The black box approach and test modeling require tests to be executed on aspects of applications which typical users can actually access and experience. Therefore, test results are directly linked to a notion of user perception, which was chosen as the measure.

A comprehensive catalogue of criteria and associated tests has been elaborated and is presented in the central part of this deliverable. Criteria are described by providing a basic assumption about a user need and/or behaviour. An irregularity is formulated based on the assumption which informs the failure or error condition of associated tests. For technical reference, the most likely root causes of the irregularities are described, giving pointers as to which aspects of an application are in need of correction if the tests were to fail. Lastly, test scripts are given, which contain clear testing procedures and scoring values for evaluation. Depending on the use case domain of applications to be evaluated, a suitable subset of criteria can be selected from that catalogue.

To assist industry practitioners in conducting an evaluation, testing examples and a walk through tutorial are included at the end of this deliverable. The tutorial provides a step by step guideline to conducting a black box application evaluation. The basic evaluation setup as well as tester acquisition and instruction are described.

1 Introduction

This deliverable is based on the PROMISE task 4.5 “Evaluation in the Wild”. The task is concerned with adapting and modifying evaluation methodology as it is used in PROMISE evaluation tasks for limited evaluation outside of PROMISE.

We herein present a methodology called “black box application evaluation” that enables industry practitioners to evaluate operational information access applications. It is based on earlier work [Braschler et al. 2006, Braschler et al. 2009] where the current state of Swiss and German enterprise search at that time was evaluated. That work did not yet examine the methodology’s properties, metrics and had no notion of different use case domains, as the main focus was involving industry representatives rather than producing academic research. The presented methodology is substantially further developed. It was expanded based on current academic research and has been improved by experimentation and validation efforts. This document contains the academic background as well as clear instructions and recommendations on how to conduct an actual evaluation using the methodology.

This methodology employs a black box approach, as the name suggests. As a consequence, it is generically usable on a variety of information access applications and evaluators require but the most basic public access to any application that they need to evaluate. Furthermore, application performance is measured comparatively by an estimate of user perception.

Basically, this measure should be designed to be able to indicate practically significant differences as opposed to merely statistically significant differences between applications [Sanderson & Braschler 2009]. Also, an evaluation model and metrics in the context of information retrieval application evaluation need to provide absolute measures that are comparable across applications. A clear set of standards and measure thresholds serve as indicators of an *estimate of user perception* to evaluators. The aim is to provide corporate decision makers in charge of operational information systems with clear indicators of their applications’ performance and enable them to identify important issues as quickly and clearly as possible. Ideally, the inclusion of best practices would allow for specific recommendations of improvement.

The notion of user perception was chosen since the evaluation methodology focuses on exhaustively testing only aspects which typical users can actually access and experience. Tests are designed to model the behaviour of prototypical users and therefore test outcomes are directly linked to a user’s perception of the tested aspect. Furthermore, the targeted audience of such an evaluation (i.e. corporate decision makers) are expected to have an interest in assessing and improving the user perception of any aspect of any application they are responsible for.

2 Related Work

An evaluation of information retrieval (IR) applications measures the overall quality of an application including the searching behaviour and the interaction of the user with the system.

As mentioned in the introduction, two previous studies [Braschler et al. 2006, Braschler et al. 2009] attempted to identify the current state of Swiss and German enterprise search. The evaluation is based on a large number of mostly independently weighted criteria. These criteria are accumulated in an evaluation grid and the overall score is a weighted sum of the criteria scores. The present tutorial bases its methodology on some of the work presented in

these studies. Specifically, the criteria categories were adopted for this previous work. The criteria themselves were completely done from scratch. The previous work also omits consultation of use case domains and does not directly answer questions about the meaning of the evaluation metrics. Finally, the present report introduces a concrete guide how to carry out the evaluation.

We now briefly review other approaches to evaluating information access applications than the one we report on in this deliverable. We highlight the main similarities and differences between these methods and our methods.

Log file analysis is an evaluation strategy to study web-searching and web search engines. Transaction logs are used to collect significant amounts of searching data on different systems [Jansen 2006]. Every transaction is logged in order to reconstruct user behaviour. Transactions are user signals such as clicking on a retrieved item, entering a query and reformulating a query. To facilitate the comparison with other analyses a standard terminology and a set of metrics was introduced [Jansen and Pooch 2001]. Using the conducted data not only sessions can be analysed but also terms and queries. Log file analysis was criticized since it does not capture the users' perception of the search and only server-side data is collected [Blecic et al. 1998]. Next, we describe two approaches that use statistics from transaction logs to evaluate a pair of systems in order to determine which one is better.

The evaluation and comparison of two information retrieval applications can also be done using *A/B testing*. The basic idea of A/B testing is to assign users randomly to one of the systems; either to the Control, which is the existing version or to the Treatment, which is the new version that is evaluated [Kohavi et al. 2007]. The users' behaviour data is then collected and it can be evaluated by statistical tests if there is a significant difference between the systems. In order to evaluate systems using A/B testing both systems need to be runnable in parallel and a click feedback is implemented.

Instead of splitting the users into two groups as in A/B testing the interleaving search results [Radlinski et al. 2008] method suggests to compare two systems by showing an *interleaved combination of the search result* of the two systems. Therefore the user interface needs to be adjusted to display the results accordingly. The clicks of users are used as implicit feedback, since it is assumed that these clicks indicate the preference between the two systems. Recent improvements in this line of work include work partially supported by PROMISE [Hofmann et al. 2011, 2012]. This work is treated in more detail in deliverable 4.3 [Berendsen et al. 2012b].

For A/B testing and, to a lesser extent perhaps, for interleaving ranked lists it holds that the methods work best when there is a lot of usage data. Therefore, it works best for high frequency queries and extending these methods to also measure performance differences on the long tail of low frequency queries is an open problem. In addition to working best for high frequency queries, these methods work best for documents that are already returned in the top results: these documents receive enough clicks to obtain reliable estimates. Black box testing generates usage data itself and can test systems using rare queries. In contrast to A/B testing and interleaving ranked lists, black box testing allows evaluating other aspects of functionality beside evaluation metrics computed on the ranked list(s). In log analysis, we are interpreting clicks as relevance feedback by modeling click behaviour, but usage data contains noise: we were not there to observe users or ask them why they clicked what they clicked. Still, usage data is typically constructed in an unobtrusive manner: we can expect the user to display spontaneous behaviour. In black box testing we model a real end user with a

protocol of using the system. Behaviour is less spontaneous but the human following the protocol and testing the system is there to report on all aspects of his or her behaviour.

Traditional IR system evaluation concentrates on measuring how “topical” relevant the presented information is with respect to the information needed by the user. When evaluating an information retrieval (IR) system, a distinction is made between user-based and system-based evaluation.

User-based evaluation measures the user’s satisfaction and therefore measures the overall success of an IR system from the perspective of a user. Dunlop [Dunlop 2000] describes an evaluation framework that adapts evaluation techniques from the human-computer interaction domain. An alternative approach to evaluation of interactive information retrieval (IIR) is proposed by Borlund [Borlund 2009]. As further reference, Kelly [Kelly 2009] provides a recent overview over the various interactive information retrieval system evaluation strategies.

System-based evaluation, however, focuses on measuring the performance of the retrieval strategy; this has a long tradition dating back to the Cranfield studies in the 1960s. The Cranfield experiments [Cleverdon 1967] use a small collection together with a set of test queries and relevance judgments for each query document pair. Using these judgments, the recall as well as the precision of an information retrieval system can be determined. By recall, we understand the fraction of relevant documents that are retrieved and precision is the fraction of retrieved documents that are relevant. Since the number of relevance judgements grows with the number of documents in the collection and the number of queries, alternative methods such as pooling have been introduced. In pooling [Spärck Jones and van Rijsbergen 1975] relevance is only assessed over a subset of the collection that contains the top k documents returned by different IR systems. Recently, low-cost evaluation techniques, such as Move-to-Front (MTF) pooling and "Interactive Searching and Judging" (ISJ), have been proposed; they keep the necessary relevance judgments at a minimum. MTF pooling [Cormack et al. 1998] improves pooling by using a variable number of documents from each system depending on performance. In ISJ [Cormack et al. 1998] assessors submit a query, judge the retrieved documents and then reformulate the query until it is unlikely to find any more relevant documents. Furthermore sampling techniques such as the Minimal Test Collection (MTC) [Carterette 2006] and Statistical Average Precision (statAP) [Carterette 2008] compare two or more systems. Also new evaluation measurements that account for incomplete judgements have been presented.

Cranfield style benchmarking and black box testing complement each other. In black box testing no explicit relevance judgments are being created and doing this in Cranfield style could yield additional insights into system performance. We do not implement that in our black box methodology since it could be done in much the same way as usually, orthogonal to our framework. In black box testing, we can take into account much more aspects than ranking quality alone, from user satisfaction to the quality of metadata to the freshness of content.

3 Definitions

Search Functionality

Information access applications as evaluated by the proposed methodology are required to expose search functionality. Generically, that functionality accepts a query and returns a ranked list of matching documents. In the case of web based applications, for instance, this could be an input field which accepts textual queries and returns a list of matching web pages.

Document

A document in the context of the evaluation is any user-reachable document within an evaluated application. Formats range from simple text files to HTML pages and also binary formats like PDF, office or media documents. All documents are assumed to have been indexed, i.e. retrievable by an application's search functionality. There are tests which challenge that assumption to assess how thoroughly documents are indexed by an application.

Characteristic Phrase

A characteristic phrase is a sequence of words (2+) that is presumably only present in a single document which it is characteristic for. The presumption is necessary due to the fact that, as a human being, one does not have full knowledge of all documents of an application and can therefore not absolutely determine if a phrase (or a document, for that matter) is unique. Such presumably unique and characteristic phrases are required for tests where documents first are identified manually (i.e. by browsing) and then retrieved using the search functionality (known-item retrieval).

4 Black Box Application Evaluation

This evaluation methodology aims to evaluate entire information access applications without any knowledge of their inner workings (hence black box). Since every application is implemented differently, a glass box approach would be detrimental to the generic applicability of the methodology. The principles of this methodology have been described previously on a conceptual level in the PROMISE deliverable D2.2 [Järvelin et al. 2012].

The upcoming subsections describe the methodology in detail. First, the methodology's scope and limitations are explained. Then, an application model is proposed to provide a sense of scope of information access applications which are to be evaluated. Afterwards, the basic steps of the methodology are explained. Following this, evaluation scenarios are defined. The next subsection describes the scoring procedure in actual tests. Thereafter, we present a comprehensive list of criteria and associated tests for evaluations. The last subsection provides three testing examples.

4.1 Scope and Limitations

The established Cranfield paradigm [Voorhees 2002] is excellently developed and well understood, but covers controlled experiments of information retrieval (IR) *systems* only, i.e. feeding a formulated query into a matching system and receiving a ranked list as output.

Operational variables are eliminated in favor of laboratory-like conditions. Since one of the goals of PROMISE is stepping out of the laboratory into operational environments, operational variables are indeed relevant. Within such an operational setting, IR *applications* are used as supporting tools for knowledge-intensive business processes rather than being scrutinized in isolation as in academia. The intention is thus to evaluate applications as a whole as they are employed in the industry, where more components factor into actual performance than only query-document matching.

Concentrating on evaluation of complete information access applications, instead of specific search engine components, emphasizes the importance of high quality information access applications to enterprise (or industrial) information providers. The worth of individual components in a system should be assessed based on their effect on the end result, and the evaluations must incorporate the understanding that optimal performance of a component is not always necessary in face of other demands on the application. Therefore, testing and evaluation must be done not only on system components separately but also on a complete information access application, including the system proper, data and various configuration parameters of value for the service provided by the application to its customers (see application model in section 4.2). This of course implies that system and application evaluation approaches can and should be used complementarily.

The idea of the black box evaluation approach is to provide a generally usable methodology for information access application evaluation of operational and live applications as well as mirrored test environments. The proposed methodology is thus based on 3 premises:

1. *Evaluation is performed on a black box or minimally invasive.*
2. *Evaluation is performed on operational applications.*
3. *Evaluation is performed in a clearly defined use case domain context.*

The first premise is a consequence of the modus operandi of the evaluation execution. Whole applications are to be evaluated which must be accessible through a defined interface, as opposed to a glass box situation where single components could be accessed directly. Because, as previously mentioned, glass boxes would have to be individually handled, a black box approach allows generalization of the methodology. Search applications are thus required to have clearly defined text search functionality in the form of a query input field. The search functionality must be based on an IR system. Applications which only pass query input to a database layer are out of scope of this methodology since indexing, matching and search results criteria either do not apply at all or are based on the assumption of an IR system layer providing query-document matching.

The second premise is based on the goals of the methodology. Live and operational applications are the targets of an evaluation to assess application performance as typical users experience it.

The third premise is a requirement of criteria applicability. If the evaluation is not restricted to a single use case domain and criteria may or may not apply differently for each application, the results are not comparable.

4.2 Information Access Application Model

Information access applications support knowledge intensive business processes. To clarify the scope of an application, we propose to use the following model based on [Peters et al. 2012] in Figure 1:

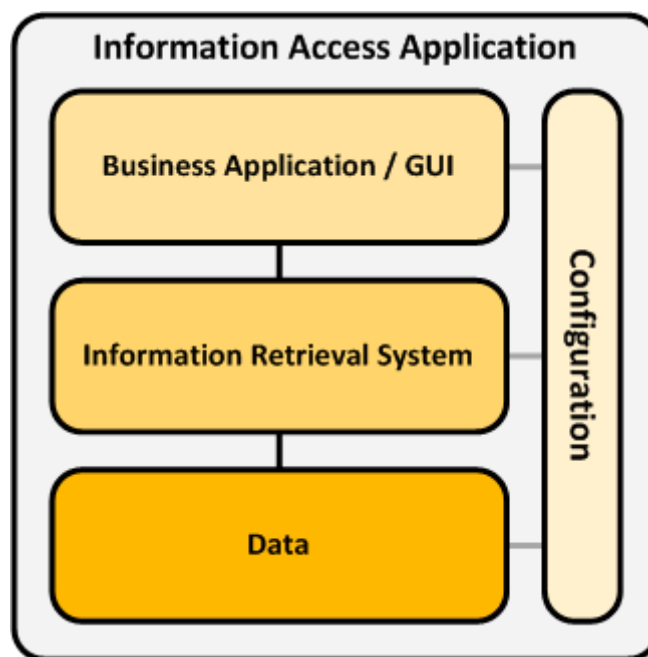


Figure 1: Information Access Application Model

The *business application* layer is composed of the user interface and associated business logic. Through this layer, user input and results presentation is handled. It also provides users with the means to interact with the IR system. Business logic may for example include facilities such as search sessions, search constraints, input validity checks, core entity handling, browsing, etc. These processes are reflected in the business application layer and are a central part of the evaluation.

The *information retrieval system* layer represents the IR systems in the narrower sense as understood in Cranfield style [Voorhees 2002] evaluations. The system layer is concerned with matching queries received from the business application layer to documents in the data layer.

Thirdly, the *data* layer contains the search index and associated data interfacing and transforming functionality.

Parallel to these layers, the application's *configuration* represents operational parameters. The model thus acknowledges the importance of correct parameterization of applications according to the underlying business processes.

Specific users are not incorporated in this model of information access applications. We model prototypical users and their preferences later when we determine which aspects of applications we evaluate (criteria) and how much influence each aspect will have on overall evaluation.

4.3 Description of Methodology

The core of the proposed methodology is a comprehensive set of all known quality criteria and “simple” tests, organized in *hierarchical trees* (see Figure 2 below for a schematic view). Criteria and tests are based on application features and behaviours that are beneficial to a user’s search experience. The aim is to have as many coarse, orthogonal tests as possible, thus covering most aspects of an IR application that may influence the defined evaluation metric. For each use case domain, the trees have to be pruned before an evaluation to only include applicable criteria and tests.

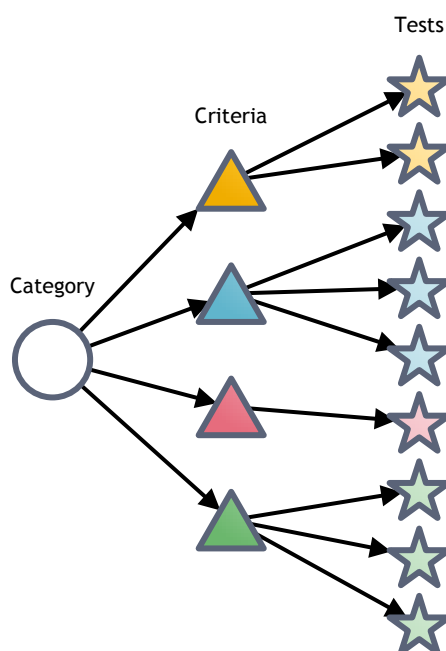


Figure 2: Schema of the hierarchical structure of tests and criteria per category

Categories offer an organizational label to a subset of criteria. Like criteria, they may apply to a use case domain or be omitted for evaluation if not applicable.

Category label	Description
Index	Addresses the indexing component of the IR application, i.e. how documents are processed and stored in order to allow their later retrieval
Query / Document Matching	Covers the matching of queries to documents and handling of mismatching problems
User Interface	Contains user interface criteria which include presentation and usability features
Search Results	Addresses the quality of search results and overall effectiveness

The following Figure 3 shows the coherence of these categories with the aforementioned application model. The previously defined categories are vertically interleaved with the application model in Figure 1. The figure demonstrates which of the application layers influences which criteria categories.

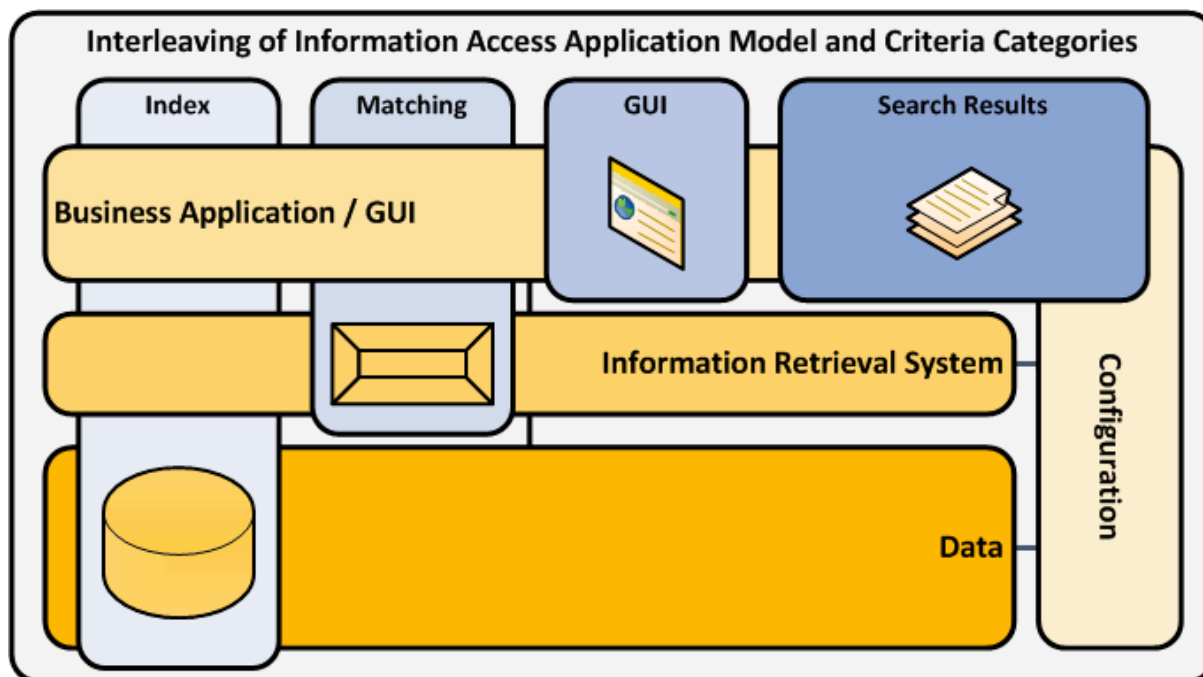


Figure 3: Application Model and Criteria Categories

The actual evaluation is conducted according to the test scripts which are supplied with criteria descriptions. The test scripts contain clear step-by-step instructions for testers and should be sufficiently clear that they do not require creative freedom. Some intellectual effort may be required but should not exceed reading comprehension skills (e.g. formulating queries from document contents) in the languages which are supported by the evaluated applications. The tests must also contain clear definitions of scoring and abort conditions. The underlying assumption and testing behaviour must model an implicit *prototypical user based on applicable use case domains*. By adhering to these requirements, testers usually do not need expert skills and evaluation costs can be kept low. For some expert systems, however, domain knowledge is required.

For scoring, a spread sheet should be created holding score values and associated weights. The weights should be defined in advance based on the practical significance of any criterion in the evaluated applications' use case domain. Multiple tests within a single criterion are weighted as fractions, e.g. 2 tests are weighted at 0.5 each for a criterion with a weight of 1. However, previous experiments have shown that a uniform weighting of 1 across all criteria already provides useful insight into application performance and works well with the approach of having a large number of coarse tests. If one still uses weights, they should be coarse as well, unless specifically justified. Weighting very finely suggests a level of precision which the methodology does not provide in the general case.

During evaluation, the testers work through the test scripts and note the scores in the spread sheet according to the definitions in the scripts. At the end, scores are aggregated for each category and application. These values may then be used to compare applications.

4.4 Evaluation Scenarios

The proposed methodology can be applied to several evaluation scenarios:

1. *Monitoring* of a single application
2. Direct *comparison* of few applications
3. Evaluation *campaign*

Monitoring

The development of a search application's performance over time can be monitored by regularly re-evaluating the application. This is especially useful for applications under active development which are regularly deployed to the public (or customers). The evaluation is repeated for each release candidate of the application.

Comparison

These are scenarios where a direct comparison of several applications is desirable:

- An established and operational search application is under consideration to be replaced by a new application (or a new version/iteration of the former).
- Several options are to be evaluated for the acquisition of a search application solution.
- Evaluation of competitors' search applications.

Campaign

A campaign is an evaluation effort spanning many applications restricted to a single use case domain. It is mainly used as a tool to assess application performance within a large set of comparable applications. The previous work upon which this methodology is based is an example of a campaign scenario evaluation.

Irrespective of the actual scenario, the evaluation is *always comparative*. Even in the monitoring scenario, evaluation results of two distinct states of the same application are compared. The results are only ever valid in the defined context of an evaluation because of differing use case domains and the subsequent selection of applicable criteria. Also, while great care was taken to provide clear test scripts, evaluators may interpret tests differently from others. Ensuring consistency and preventing bias is an important responsibility of evaluation organizers.

4.5 Scoring

The following Figure 4 shows the grid structure of an evaluation. Each test is executed on each application and results in a score.

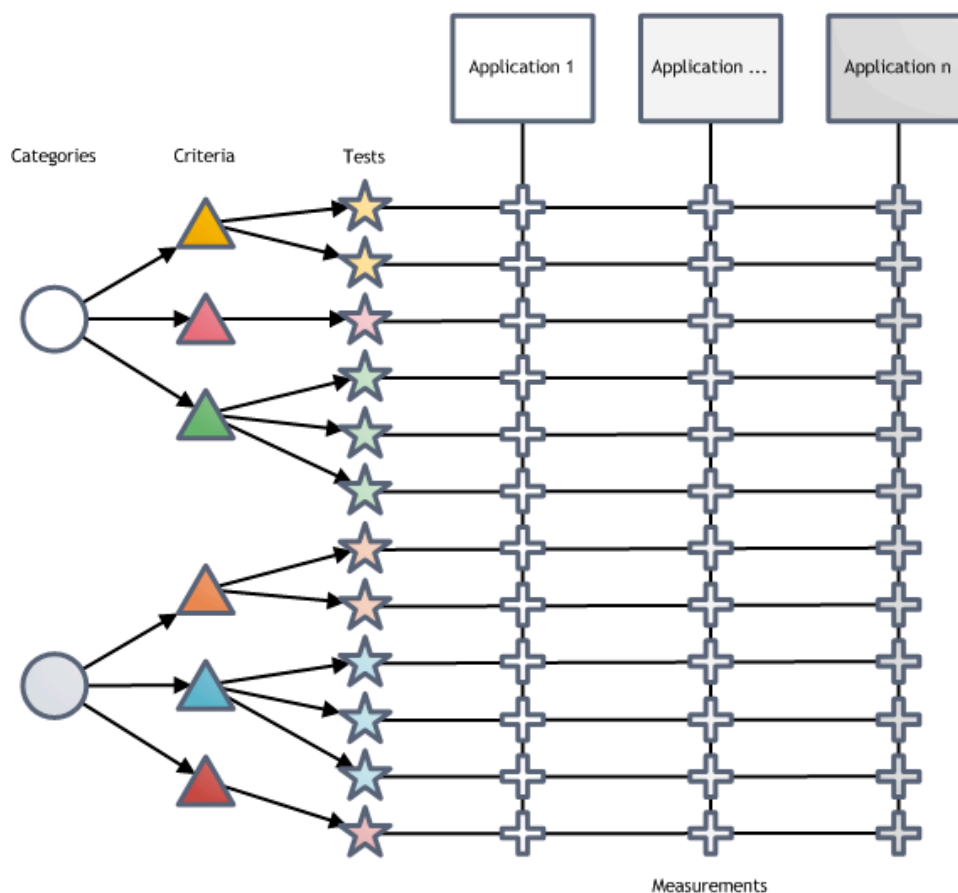


Figure 4: Evaluation Grid for Multiple Applications

The test descriptions in the following section specify how scores are to be given. Valid score values are provided in round brackets.

Most tests contain abort conditions for cases where it is currently impossible to assess a criterion based on the formulated test. It would, however, be possible to assess the criterion using a different method or at another time. For these occurrences, the score shall be noted as “*ABORT*”.

In contrast to regularly executed tests which fail and yield a score of 0, aborted tests shall not be considered when calculating the final score. Aborts are widely spread events within an evaluation. They signify an exception which may point to shortcomings of a test script in a single aspect. A large number of aborts on the same test within an evaluation provide evaluation organizers with evidence as to which criteria should be left out in later evaluations within the same use case domain as they might not be generally applicable in the context.

4.6 Revised Criteria and Test Scripts

This section lists criteria and tests as they have been elaborated in PROMISE. They are informed by the PROMISE use case domains, a generic use case of “enterprise search”¹ and experiences from conducting a full campaign with several PROMISE partners. While the methodology’s description states that criteria may contain multiple tests, all of the elaborated criteria are incidentally mapped to just one test each. This does not affect the evaluation, however, and may change in the future when the catalogue is expanded.

Underlying assumptions, irregularities, root causes and testable features are given for each criterion. Criteria may be validated against use case domains by their associated assumptions. Use case domain adaptations are given for the PROMISE use case domains, where available.

¹ Enterprise Search is understood as a use case domain wherein companies provide information about themselves as a communication channel and provide search functionality for it.

4.6.1 Index Criteria

4.6.1.1 Completeness

Assumption

Users expect to potentially find all documents that can be publicly accessed in any way on the site (namely, through browsing the site) when using the search functionality.

Irregularity

Publicly accessible documents (known through browsing or obtaining a direct link) cannot be found using the search functionality.

Root causes

- The index is incomplete – documents/sets of documents are missing
- The index is incomplete – the index is out of date (→ Freshness)
- The index is incomplete – documents of certain types are missing (→ Format support)

Test

The content scope of the test is decidedly narrow. No linked resources in the application are expected to be accessible through the search. E.g. if a retailer owns another shopping outlet, the latter's products need not be found.

1. Locate 3 documents which match the following criteria:
 - a. Takes at least 5 clicks to locate document
 - b. Document is at least 3 levels from root as determined by URL (optional if URL-rewriting is used in the application)
 - c. URL is at least 100 characters long (optional as above)
2. If no documents matching these criteria are found, abort
3. For each document
 - a. Extract a characteristic phrase (2-4 words) from a central location within the document
 - b. Search for the document using that phrase
4. Score according to the number of documents which can be located within the top 10 search results (0, 1, 2, 3)

Use Case Domain Adaptations

Search for Innovation: the notion of completeness is defined by a user's expectation of the presence of specific content rather than the inclusion of all browsable content. Therefore, there is no way to objectively and easily test this feature.

4.6.1.2 Freshness

Assumption

Users expect to find all publicly available documents, no matter how recently published.

Irregularity

Publicly accessible documents (known through browsing or obtaining a direct link) cannot be found using the search functionality.

Root cause

- The index is incomplete or out of date.

Test

1. Locate a news section / feed on the website
2. Identify two news items newer than 48 hours
 - a. Abort if no suitable news items can be found
3. For each news item
 - a. Extract a characteristic phrase (2-4 words)
 - b. Search the news items using that phrase
4. Score according to the news items retrieved in the top 10 search results (0, 1, 2)

Use Case Domain Adaptations

Search for Innovation: the freshness is determined by what the website claims. Therefore, tests are replaced with the following procedure:

1. Locate the claims regarding the coverage of the website's data
2. If the claimed coverage indicates a particular date (absolute or relative), note the most recent possible date
 - a. Abort if no claim of date coverage can be determined explicitly or implicitly
3. Using a patent office website (uspto.gov, epo.org) locate 2 patent publications on the most recent possible date identified above
4. For each publication
 - a. Extract the publication identifier from the document
 - b. Search for the publication using the identifier
5. Score according to the publications retrieved in the top 10 search results (0, 1, 2)

4.6.1.3 Special Characters

Assumption

Users need to find documents containing search terms which contain characters not usually used in their language.

Irregularity

Queries containing diacritics and special characters do not match character-normalized documents or plain ASCII queries do not match documents containing diacritics and special characters.

Root causes

- No character normalization in indexing process
- No character normalization in query processing

Tests

1. Abort if the search application's data only contains text in languages which do not have diacritics
2. Enter a query containing a word/words which contains diacritical characters
3. Enter a query that contains the same words without diacritics, either by transcribing them (e.g. German umlauts ö->oe) or by using the base characters (ä->a). E.g. "Öffentlich" -> "Oeffentlich", "léger" -> "leger"
4. Score success (1) if both the original word and the normalized one yield the same results using either transcription or base characters; otherwise score failure (0). If number of results is displayed, compare that for convenience.

4.6.1.4 Tokenization

Assumption

User need to find documents containing complex entity names depending on the use case domain. Special characters (e.g. apostrophe, hyphens) should be part of the terms depending on the context.

Irregularity

Users cannot find documents containing complex terms (e.g. "O'Neill", "F/A-18"). Domain specific technical terms like a machine type "MT-201" are not retrievable.

Root causes

- Tokenization only considers alphanumeric characters as part of terms and splits at all other characters

Test

1. Identify 3 documents containing complex (domain-specific) terms as per the above examples.
 - a. Abort if not enough documents can be found in 5 minutes.
2. Search the documents using queries containing little more than these terms
3. Score success (0, 1, 2, 3) in relation to correctly returned documents within the top 10 results

4.6.1.5 Decomposing

Assumption

Applications operating with agglutinative languages, where words may be compounded to form new words with different meanings, should handle such compounds correctly. Users expect to be able to search for parts of a compound word and receive relevant results.

Irregularity

Searching for parts of a compound word does not match documents containing the full compound word. Words are incorrectly decomposed when they should not be.

Root causes

- No or faulty implementation of decomposing

Test

1. Abort if the application's data does not contain languages which can be decomposed
2. Pick 3 compound words from the application's content
3. For each word:
 - a. Enter a query using the meaningful parts of the compound word
 - b. Score success (0 or 1) if documents containing the full compound word rank higher than documents containing only a part of it
4. Score in relation to successful queries (0, 1, 2, 3)

4.6.1.6 Named Entities

Assumption

Users want to search for named entities where the respective entity (e.g. a person) is very clearly defined within the context of the application. Inability to find documents pertaining to the entities at a high rank in the result list is disruptive to the user experience.

Irregularity

Clearly defined entities from the application context cannot be directly found using their names as a short query.

Root causes

- The document indexing process does not consider named entities and thus tokenizes them in less informative bits.

Test

1. Identify 5 named entities (preferably composed of 2 or more terms) based on the applications context. Usually you are able to deduce these from the content. For a news site, this may be VIPs or place names, for a cultural heritage site, it may be artists etc. An example might be "George W. Bush", who is not the same person as "George Bush", or "Bill Clinton", which should not return documents for "George Clinton", and "Martin Luther", which is not the same person as "Martin Luther King"
 - a. Abort if less than 5 named entities can be found
2. Search for the entities using only their name
3. Score success (0, 1, 2, 3, 4, 5) for each query which returns results that clearly refers to the correct entity, and not to other entities that share parts of the name.

Use Case Domain Adaptations

Search for innovation: use name of inventors or assignees, application numbers, publication numbers and publication dates.

4.6.1.7 Stemming

Assumption

Users enter queries based on their intent as a set of key words or as a full question. If a term is entered as a noun, adjective, verb or adverbially may differ from session to session while the intent may not. Stemming counteracts by reducing different grammatical word forms to single stem forms, thereby increasing the probability of matching the intended word irrespectively of its form.

Irregularity

Different word forms and plurals in queries yield different results.

Root causes

- Stemming incorrectly or not implemented.

Test

1. Enter a few single term queries using singular words, plurals, different verbal and adjective forms
2. Score success (1) if different forms of the same word in a query return the same results. Otherwise score failure (0).

4.6.1.8 Meta-Data Quality

Assumption

Users benefit greatly from rich meta-data and suffer from incomplete or incorrect meta-data.

Irregularity

Documents cannot be found based on their known meta-data fields. The result list shows wrong titles or missing information for documents.

Root causes

- Bad automatic meta data processing
- Even worse manual meta data processing
 - e.g. copying Word documents and leaving the document properties unchanged, resulting in many documents sharing the same meta-data but having very different content

Test

1. Check if structured search facilities are available, i.e. searching in meta-data fields
 - Abort if no structured search is available
2. Identify 3 documents within the application which have clearly defined meta-data
 - Abort if no such documents can be found
3. Search for the identified documents using a combination of available meta-data only (search in respective fields)
4. Score success (0, 1, 2, 3) for each query which returns the intended document (within top 20) and displayed the correct meta-data i.e. did not just find that information in the document text.

4.6.1.9 Office Document Handling

Assumption

Binary office documents (PDF, Microsoft Office formats, etc.) can contain relevant information and should be suitably parsed and indexed by an application. Users expect to find binary documents using queries describing their content and / or meta-data.

Irregularity

Binary documents which are otherwise reachable within the application (e.g. by browsing) are not retrievable using the provided search functionality and a suitable query.

Root causes

- Faulty or non-existent binary document processing

Test

1. Identify some textual binary documents (PDF, MS Office, etc.) in the application and note their title and some characteristic words of content
 - Abort if the application contains no binary documents or none are to be expected
2. For each identified document
 - Search for title of document
 - Search for characteristic words
3. Score success (1) if you are confident that binary documents are retrievable and being processed correctly. Otherwise score failure (0).

Use Case Domain Adaptations

Search for Innovation: look for documents that contain tables or flowcharts. Use keywords from them to perform the tests described above, and to identify whether the images depicting mentioned tables or flowcharts have been OCRed.

4.6.1.10 *Separation of Actual Content and Representations*

Assumption

Structural elements (header, footer, etc.) within documents are not relevant information and matches on these elements are disruptive to the user experience.

Irregularity

Relevant documents in the result list are obscured by a multitude of other documents which match only on structural elements.

Root causes

- Indexing does not remove recurring and / or structural elements

Tests

1. Find documents which contain recurring structural information, i.e. headers, footers, navigational elements or similar.
 - a. Abort if no such documents can be found with reasonable effort (5-10 min.)
2. Search for terms which are present in structural information (e.g. "Copyright", "Page Number", header / footer content, "home" link, etc.) and add a word from the application's domain which returns lots of results
3. Score success (1) if the top results are not primarily composed of documents containing structural information. Otherwise score failure (0).

Use Case Domain Adaptations

Search for innovation: e.g. of terms "Consisting of" "assignee" "inventor" ,"patent" or their non-English equivalents.

Cultural Heritage: e.g.: rights, licenses, providers, publishers, etc.

4.6.1.11 Duplicate (Content) Documents

Assumption

Users gain no information from duplicate documents, while very similar documents and versions of the same document may be useful or even critical in the case of some use case domains.

Irregularity

The result list contains identical documents.

Root causes

- Indexing process does not check for document redundancy

Test

1. Search with a number of very broad terms (2-4, applicable to the website)
2. Analyze the result lists - are there obvious duplicates?
3. Quickly repeat the test for up to 5 minutes with different terms until you are confident with your findings
4. Score success (1) if no duplicates are found. Otherwise score failure(0).

Use Case Domain Adaptations

Search for innovation: does not apply. Different version of the documents may appear extremely similar, but if they are published with different identifiers (kind codes), they are legally different.

Cultural Heritage: does not apply. Information Systems within the CH domain can and sometimes should contain "duplicates" since some researchers are especially interested in slightly different version of one object.

4.6.2 Matching Criteria

4.6.2.1 Query Syntax

Assumption

Experienced search users expect to be able to enhance and specify their queries by using query operators such as the Booleans "OR", "AND", "NOT" or similarity operators such as "LIKE".

Irregularity

- The query syntax is weak, allowing none or only the most basic operations
- The implementation of the query operators is faulty

Root causes

- Missing or faulty implementation of query operators

Test

1. Identify 2 query terms from the context of the application which are each going to yield a reasonable amount of results (> 100).
2. For the following queries, use the appropriate equivalent operator for the implemented search functionality. In the case of the "AND" operator for example use the "+", "&" or the corresponding field in the advanced search options. Always note the number of returned results.
 - a. Abort if the search application does not specify the number of returned results or an estimate thereof
 - b. Retrieve query 1 with term 1
 - c. Retrieve query 2 with term 2
 - d. Retrieve query 3 as "<term 1> AND <term 2>"
 - e. Retrieve query 4 as "<term 1> AND NOT <term 2>"
 - f. Retrieve query 5 as "<term 1> OR <term 2>"
3. Score in relation to the amount of the following conditions being met (0, 1, 2, 3). The "lqueryl" notation is shorthand for comparing the number of returned results for the queries:
 - a. lquery 1l >= lquery 3l
 - b. lquery 1l >= lquery 4l
 - c. lquery 1l <= lquery 5l

4.6.2.2 Phrasal Queries

Assumption

Users may enter queries as complete phrases using quotes (" ") and expect to find the complete phrase reflected in the search result.

Irregularity

Using quotes does not yield results containing the phrase and instead, results containing only one or more of the phrases' words are returned.

Root causes

- Quotes operator is not or insufficiently implemented

Test

1. Identify 3 phrases with at least 3 words in the context of the evaluated application
2. For each phrase, enter a query with the phrase in quotes
3. Score success (1) if phrases in quotes return fewer results than phrases without quotes. Otherwise score failure (0).

4.6.2.3 Over- and Under-Specified Queries

Assumption

Users feel irritated if long queries return very few or no results and short queries return almost the entire collection.

Irregularity

Missing the application's unknown "sweet spot" in terms of query length returns an undesirable number of results. Users receive no indication of what went wrong.

Root causes

- No user guidance when result set has an unusual number of hits
- Matching model punishes verbose descriptions

Test

1. Copy and paste a sentence from any document within the application into a query and add some out-of-context terms
2. Score success (1) if the document can still be found, score failure (0) otherwise
3. Use 2 terms from the application's context as a query, which should return a very large number of results
4. Score success (1) if the application offers suggestions or facilities to improve your search, e.g. further terms, browsing, etc. Score failure (0) otherwise for a total of (0, 1, 2)

4.6.2.4 Feedback

Assumption

User feedback mechanisms help users find more relevant information if they are willing to do several search iterations in the application.

Irregularity

Iterative searches have no influence on the matching and / or no facilities for explicit user feedback are present in the application.

Root causes

- No relevance feedback implemented in application.

Tests

1. Score success (1) if any of these features or similar ones are present in the application, score failure (0) otherwise:
 - Result list assessment by user (e.g. an option saying "this result was useful for me")
 - Result item assessment by user (e.g. star rating or checkbox)
 - Result list reordering (not sorting!) by user as a means of feedback

Use Case Domain Adaptations

Recall-oriented use cases will want to emphasize this criterion while precision-oriented ones may opt to omit it.

4.6.2.5 Multimedia

Assumption

Users want to find different types of objects such as text, videos, images and audio using a single query.

Irregularity

The system does not contain different object types or does not distinguish between those.

Root causes

- No result representation sorted by media type
- No facet for media type

Test

1. Check the availability of the following features:
 - a. Search results contain multimedia content
 - b. Query by example of a multimedia document
 - c. Restriction of search to specific media types
2. Score according to the number of present features (0, 1, 2, 3)

Use Case Domain Adaptations

Cultural Heritage: Omit feature 1a, score accordingly (0, 1, 2)

4.6.2.6 Cross-Language Information Retrieval

Assumption

Users want to find results in different languages by querying in their native language; they are not willing or able to repeat queries in several languages.

Irregularity

The system does not offer cross-language information retrieval, only monolingual search is available.

Root causes

- Cross-language features are not implemented

Test

1. Abort if the application's content only contains documents in a single language (quickly browse the site to check).
2. Score success (1) if the application contains cross-language functionality (e.g. translations aids, automatic translation, etc.). Otherwise score failure (0).

4.6.3 User Interface Criteria

4.6.3.1 Performance / Responsiveness

Assumption

Users usually require queries to run and return near instantaneously. Delays are perceived as a flaw, produce a large amount of frustration and lead to quick dismissal of the application in cases where the information retrieval application may be substituted by external applications (e.g. Google).

Irregularity

Queries run slowly and return results after a perceivable delay.

Root Causes

- Unsuitable implementation of application
- Insufficient computational capacity
- Insufficient bandwidth capacity

Test

1. Score success in relation to response time after issuing a query in the application (0, 1, 2)
 - a. < 1s: excellent, score 2
 - b. < 5s: acceptable, score 1
 - c. > 5s: unacceptable, score 0

4.6.3.2 Browsing

Assumption

Users in some cases want to get an overview of an application's content prior to or instead of using the search functionality.

Irregularity

Navigation within the application's content is inconvenient or impossible without the usage of search functionality.

Root Causes

- Documents are not ordered in organizational units and/or topical structures.

Test

1. Score success (1) if the application offers usable browsing functionality which allows access to documents. Otherwise score failure (0).

Use Case Domain Adaptations

Cultural Heritage: browsable lists of authors, lists of creators, lists of locations/dates, timelines, (virtual) exhibitions, etc.

4.6.3.3 Field search (Facets)

Assumption

Users benefit from the possibility of exploring the search results via topically similar categories automatically extracted from the set of results, either from meta-data or the text itself.

Irregularity

Only the entire content can be searched.

Root causes

- No facet functionality implemented

Test

Score success (1) if the search can be topically (or otherwise) filtered or score failure (0) otherwise.

Use Case Domain Adaptations

Facet examples:

- Search for Innovation: publication date, issuing authority
- Cultural Heritage: author, media type, provider, creator, country, language, publication year

4.6.3.4 Query Term Highlighting

Assumption

Highlighted query terms in a result list help users to preliminarily assess the relevance of documents.

Irregularity

Query terms are not highlighted or otherwise marked in the result list.

Root Causes

- Feature not implemented

Test

Score success (1) if query terms marked in any way in the result list Otherwise score failure (0).

4.6.3.5 Document Summarization

Assumption

Users can quickly assess document relevance if a suitable document summary is provided in the result list. For factual queries, a summary may already contain the sought-after fact, thereby satisfying the information need at first glance.

Irregularity

No document summary is provided, only the most basic document identifiers or meta-data.

Root Causes

- Feature not implemented

Test

Score according to quality of the provided document summaries in the results list (0, 1, 2):

- None → 0
- Summary contains the first n words of the document → 1
- Summary is directly relevant to query and contains most or all of the query terms → 2

4.6.3.6 Result List Presentation

Assumption

A visually pleasing and well organized result list presentation is very helpful to the user.

Irregularity

The visual presentation makes it difficult for the user to get any further hints of relevance and is detrimental to usability.

Root Causes

- Aesthetics
- Usability

Test

Score according to your impression of the result list presentation (0, 1, 2):

1. 2 for good (useful layout, visually pleasing)
2. 1 for sufficient / decent (practical, functional, basic)
3. 0 for bad (unwieldy layout, cluttered, confusing)

4.6.3.7 Exception Handling

Assumption

Users are confused if the application encounters an error and does not explain what went wrong.


Irregularity

The application encounters an error but does not provide any insightful hint to the user why the error occurred and how it can be avoided.

Root Causes

- Syntactically incorrect queries
- Internal application exception

Test

1. Score success (1) if the application neither crashes nor shows obscure technical error codes on rubbish queries like:
 - a. fill the query field with all “A”s
 - b. Control characters which are usually escaped (e.g. “” = ASCII character 178)

4.6.3.8 Term Suggestions

Assumption

Users benefit from term suggestions if an application's use case domain is specialized. Usually, the taxonomy of a use case domain is clearly defined and failing to formulate queries using terms from within that taxonomy leads to document matching problems. Terms suggestions also help to correct spelling mistakes.

Irregularity

Applications with specialized domain taxonomy have matching problems.

Root Causes

- Feature not implemented

Test

Score success (1) if terms are suggested when entering a query or alternatively on the result list. Otherwise score failure (0).

4.6.3.9 User Guidance

Assumption

Users appreciate application feedback and guidance mechanisms in case very few (and especially zero) or too many results are returned.

Irregularity

Too few or too many results are returned from a query, making a useful interpretation of the results unfeasible.

Root Causes

- Over-/underspecified queries
- Spelling mistakes
- Domain specific terminology unknown to searching user

Test

1. Formulate a query to provoke zero results, e.g. by entering a large number of terms completely unrelated to the application's context
 - a. Abort if there are always results
2. Score in relation to these features being present (0, 1, 2, 3) when generating zero results
 - a. Term / phrase suggestions present and suitable
 - b. Spell checker and suggestions
 - c. Suggestion of similar/other queries having good (e.g. as assessed by users through feedback) results

4.6.3.10 *Related Content*

Assumption

Users may benefit from being shown content related to their current query.

Irregularity

No additional content than the result list is displayed.

Root causes

- Feature not implemented

Test

Score success if the application suggests other content which is mostly similar or relevant to any entered query. (0 or 1)

4.6.3.11 *Context Information*

Assumption

User may benefit from further information about their query from the application's context. The information may be chosen from corpus statistics or the interaction of users with the result list, for instance.

Irregularity

No additional information than the result list is displayed.

Root causes

- Feature not implemented

Test

Score success (1) if the application presents any contextual information (besides and not including content meta-data) such as the number of views of pages or the number of citations (if applicable), etc. Otherwise score failure (0).

4.6.3.12 *Personalization*

Assumption

Users benefit from customization options and contextual information in their search interface by accommodating for their tastes of interaction with the application. They also benefit from search results which take their preferences, previous queries or similar queries by other users into account. This test is strictly about personalization for SEARCH. If the site offers other personalization features, e.g. features that are tied to the business (loyalty programme etc.), these are not applicable to this test.

Irregularity

A user's continued usage of the application has no influence of the outcome of queries and/or additionally offered content.

Root Causes

- User profiles are not implemented.

Test

1. Check if there is a user-based search profile which enables users to modify the search behaviour and/or the visual presentation.
2. Score if the feature is present (0 or 1)

Use Case Domain Adaptations

Lay user applications are expected to emphasize personalization more than expert applications.

4.6.3.13 *Localization*

Assumption

Users appreciate or require an application which allows interaction based on their own language, reading direction and cultural conventions.

Irregularity

The application looks and behaves the same, no matter where the user originates from.

Root causes

- Missing user interface localization

Test

1. Find localization functionality and switch to a different language
 - a. Abort if only one language is supported
2. Score success (1) if these conditions are met or failure (0) otherwise:
 - a. All of the application company country's national languages are selectable
 - b. Do a quick browse and check if the localization is consistent across the application, i.e. no English words for elements or exception messages where it is not expected (e.g. IT specific terminology)

4.6.3.14 *Result List Import / Export*

Assumption

Users may wish to export a result list with the given query for further processing or later re-importing into the application to review previous results.

Irregularity

Result list and query cannot be exported or imported.

Root causes

- User interface does not provide import/export functionality.

Test

Score success (1) if query import / export functionality is present, score failure (0) otherwise.

4.6.3.15 *Sorting of Result List*

Assumption

For structured documents, users want to sort the result list by specific fields of the documents or by any available meta-data.

Irregularity

The result list cannot be sorted.

Root causes

- No document structure
- Missing meta-data
- Sorting not implemented

Test

1. Enter a query which yields a reasonable amount of results (> 100)
2. Score success (1) if the results can be sorted by suitable criteria. Otherwise score failure (0).

Use Case Domain Adaptations

Search for Innovation: sorting criteria: publication dates, IPC codes, assignee name.

Cultural Heritage: date, creator, provider, country, etc.

4.6.3.16 *Justification of Results*

Assumption

Users want to know how the application generated the retrieved result, especially when they get a result set which does not meet their expectations.

Irregularity

No information is given about how the retrieved result was generated.

Root causes

- No description is given by the application.

Test

1. Score success (1) for each of these features being present at all, to a maximum of (3) in total (0, 1, 2, 3):
 - a. Result list document summarization, "Snippets"
 - b. Key word highlighting (in title or snippet)
 - c. Display of number of results of any query
 - d. Other features which justify the results by providing different presentations of (additional) information

4.6.3.17 *Monitoring*

Assumption

Users want to issue standing queries and identify changes in the results over time.

Irregularity

Queries can only be issued one at a time and the application offers no possibility to update and display changes.

Root causes

- Feature not implemented.

Test

Score success (1) if any given query can be monitored over time using a subscription or similar functionality. Otherwise score failure (0).

Use Case Domain Adaptations

This criterion is directly opposed to stability because the former requires regular change in the data collection. Omit the test if stability is a requirement for the application.

4.6.3.18 *System Override / User Control*

Assumption

Users do not always want spelling correction, stemming or relevance feedback where these or similar features are present, especially if these do not happen to enhance the users' search effectiveness.

Irregularity

Corrective or "helpful" features disrupt the user's experience by malfunctioning or correcting actually valid input.

Root causes

- Search features which modify user input cannot be manually disabled

Test

Score success (1) if user input modifying features can be overridden or disabled. Score failure (0) otherwise.

Settings like these usually reside in an "advanced search" configuration.

4.6.3.19 *Navigational Aids*

Assumption

Users want to quickly navigate back to previous queries and then forward again to compare results of refined queries, for example.

Irregularity

Previous query results have to be retrieved by entering the previous query again.

Root causes

- Feature not implemented

Test

Score success (1) if the application offers a button or link (not just browser shortcuts) to navigate back and forth for quick access to previously entered queries. Score failure (0) otherwise.

4.6.3.20 Social Aspects

Assumption

Sharing and annotating of search results allows users to see what their friends and colleagues are searching for, possibly leading to new discoveries within an application's content.

Irregularity

Search results cannot be shared with other users.

Root Causes

- No implementation of social aspects.

Test

1. Score in relation to these features being present (0, 1, 2) in the search functionality (not the whole site)
 - a. Search results can be annotated and shared with other users, either locally on the site (e.g. tagging) or by using social networks (e.g. Google +1, Facebook like, Flickr, etc.)
 - b. Content suggestions based on searches by other known users and similarities to one's own searches are available

Use Case Domain Adaptations

Search for Innovation: does not apply.

4.6.3.21 Entertainment / Fun

Assumption

A user's experience is influenced by the entertainment value of the application.

Irregularity

The user interface is bland and boring.

Root causes

- Aesthetics
- Interactiveness of UI

Test

Score in relation to *objective* entertainment value of the entire search application (0, 1, 2):

- entertaining, engaging and visually pleasing : score 2
- decent usability and aesthetics : score 1
- boring, cluttered und unwieldy: score 0

4.6.3.22 *Mobile Access*

Assumption

Users may want to access the information retrieval application from a mobile device with different display and input capabilities than common PCs.

Irregularity

There are no adaptations in the user interface when accessing the application from a mobile device and there is no different version of the application for mobile devices. **IMPORTANT:** there may be sites that offer mobile versions, but these contain no search functionality. For our purposes, these are to be scored as if having no mobile version.

Root causes

- Mobile device adaptation not implemented.

Tests

1. Access the application from a mobile device or set your browser's user agent accordingly
2. Use a specified alternate entry point for mobile devices if present
3. Score success (81) if the application has suitable adaptations (mainly layout) to mobile devices AND search functionality is present, score failure (0) otherwise.

4.6.4 Search Results Criteria

4.6.4.1 Navigational Queries

Assumption

Users enter queries to find an entry point to topically structured content of the application.

Irregularity

The application's search functionality does not prominently return entry points to topically structured content.

Root causes

- Freshness and completeness of index insufficient
- Bad meta-data quality

Test

1. Quickly browse the site and identify an entry point to some topically similar documents, e.g. using the application site's navigation bar
2. Build a query with a few words using the information on the entry point
3. Score success (1) if entry point could be retrieved in the top 10 results or failure (0) otherwise

Use Case Domain Adaptations

Cultural Heritage: not applicable

4.6.4.2 Factual Queries

Assumption

Users enter queries to find a single fact. A single trustworthy document is sufficient to satisfy the information need.

Irregularity

Factual information cannot be found by suitable queries.

Root causes

- Freshness and completeness of index are lacking
- Bad treatment of binary documents (e.g. PDF)
- Missing document summaries or snippets in result list

Test

1. Pick 5 facts from the application's content, examples:
 - a. Company's year of incorporation
 - b. Number of branches
 - c. Revenue
 - d. CEO
 - e. Product lines
 - f. etc.
2. Build short queries for these facts from the context
3. Score success for each query which retrieved the sought for fact in the top 10 results (0, 1, 2, 3, 4, 5)

Use Case Domain Adaptations

Search for innovation: not applicable.

Cultural Heritage: not applicable

4.6.4.3 Known / Suspected Item Retrieval

Assumption

Users want to quickly find a document in an application which they have accessed before or expect to be present.

Irregularity

A known or expected document is not retrievable by a query based on previous knowledge of the document.

Root causes

- Lacking freshness and completeness of index

Test

This test is very similar to the one from Factual Queries. The difference is that in factual queries, the presence of the fact is expected and the fact is searched for. In this test, the facts and the document are already known and the document has to be retrieved.

1. Pick 5 unique (assumed to be!) documents from the application's content, examples:
 - a. Last year's business report
 - b. Product specification
 - c. etc.
2. Abort if not enough uniquely identifiable documents can be found
3. Formulate queries based on the documents' main topic (in terms of quantity)
4. Score success for each query which retrieved the sought for document within the top 10 results (0, 1, 2, 3, 4, 5)

Use Case Domain Adaptations

Search for Innovation: use publication numbers.

Cultural Heritage: use very well-known artefacts (e.g. "The Scream" - this is in all likelihood the search for the picture by Munch)

4.6.4.4 Diversity

Assumption

Users expect different aspects of an ambiguous query to be represented if available.

Irregularity

The result list is dominated by very similar items, even if query is ambiguous or allows a rich set of possible results.

Root causes

- Feature not implemented

Test

1. Formulate 5 queries using very broad search terms (e.g. “Paris”)
2. For each query, check if the results are suitably diverse (Paris: facts, history, travel, maybe Mrs Hilton, etc.)
3. Score in relation to query results which provided diverse results (0, 1, 2, 3, 4, 5)

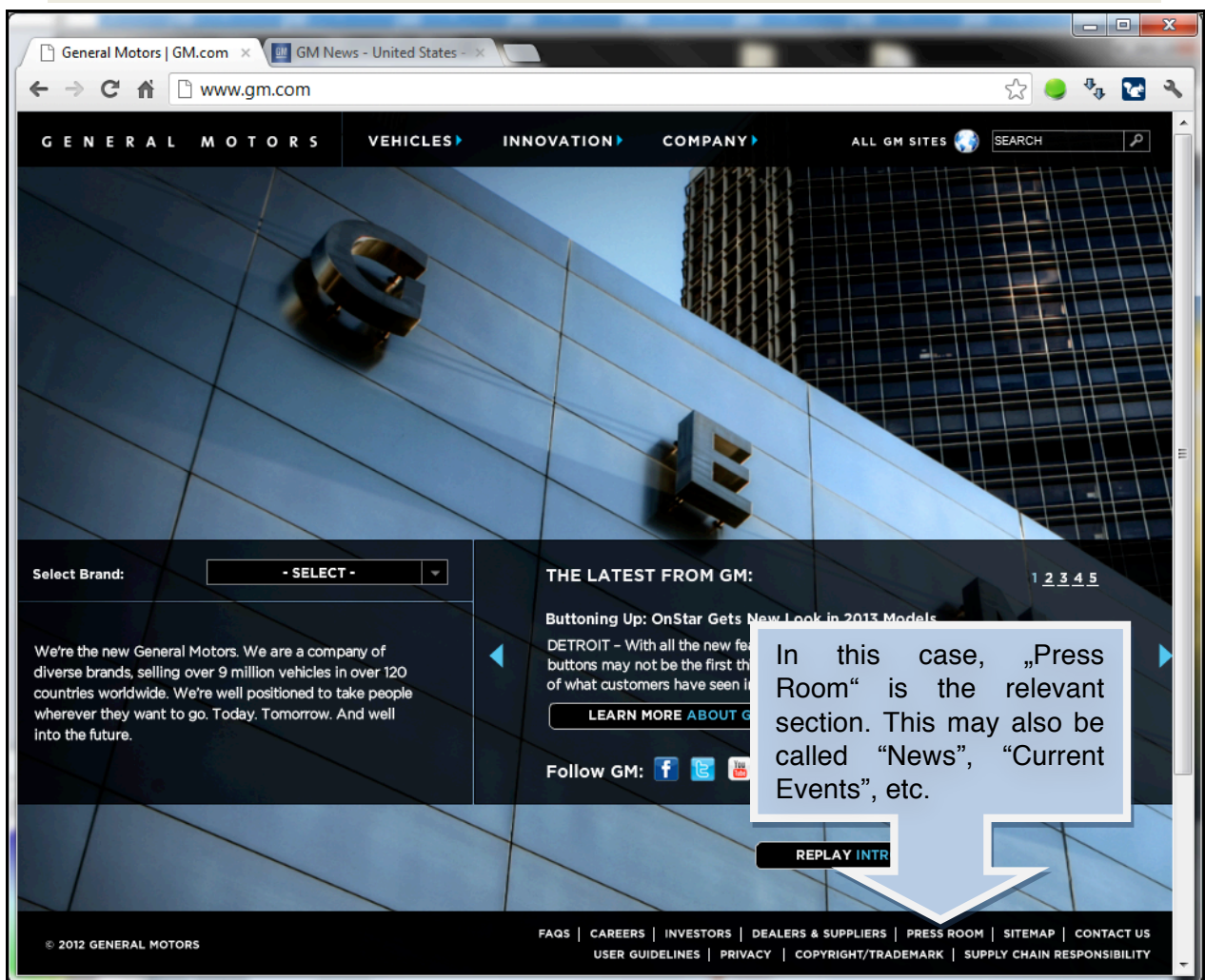
4.7 Testing Examples

This section contains three walk-through examples of tests. They were selected as prime examples of more complex and also simple tests. These examples can be provided to testers in order to clarify the testing procedure. The test script steps are given with screenshots depicting the relevant elements and actions.

4.7.1 Freshness

Here we demonstrate a test of the index category criterion “Freshness” on the web search application of General Motors².

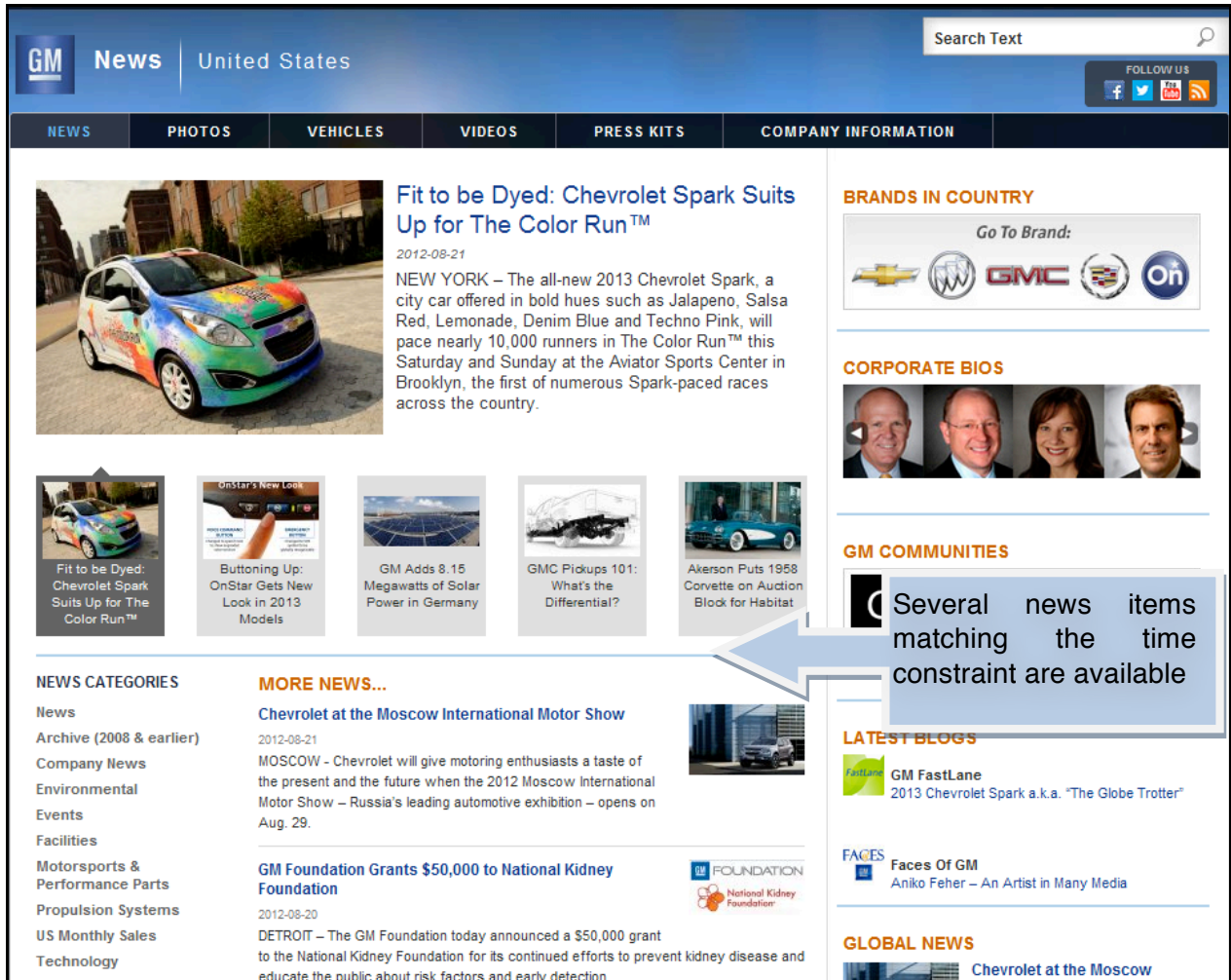
1. Locate a news section / feed on the website



² All screenshots and contents © General Motors 2012

2. Identify two news items newer than 48 hours

a. Abort if no suitable news items can be found



The screenshot shows the GM News United States website. The main headline is "Fit to be Dyed: Chevrolet Spark Suits Up for The Color Run™" dated 2012-08-21. Below it, there are several smaller news items in a grid, including "Buttoning Up: OnStar Gets New Look in 2013 Models", "GM Adds 8.15 Megawatts of Solar Power in Germany", "GMC Pickups 101: What's the Differential?", and "Akerson Puts 1958 Corvette on Auction Block for Habitat". On the right side, there are sections for "BRANDS IN COUNTRY" (Go To Brand: Chevrolet, Buick, GMC, Saturn, OnStar), "CORPORATE BIOS" (four portraits), "GM COMMUNITIES" (Several news items matching the time constraint are available), "LATEST BLOGS" (GM FastLane, Faces Of GM), and "GLOBAL NEWS" (Chevrolet at the Moscow).

Fit to be Dyed: Chevrolet Spark Suits Up for The Color Run™
2012-08-21
NEW YORK – The all-new 2013 Chevrolet Spark, a city car offered in bold hues such as Jalapeno, Salsa Red, Lemonade, Denim Blue and Techno Pink, will pace nearly 10,000 runners in The Color Run™ this Saturday and Sunday at the Aviator Sports Center in Brooklyn, the first of numerous Spark-paced races across the country.

BRANDS IN COUNTRY
Go To Brand:
Chevrolet Buick GMC Saturn OnStar

CORPORATE BIOS
Four portraits of GM executives.

GM COMMUNITIES
Several news items matching the time constraint are available

NEWS CATEGORIES
News
Archive (2008 & earlier)
Company News
Environmental
Events
Facilities
Motorsports & Performance Parts
Propulsion Systems
US Monthly Sales
Technology

MORE NEWS...
Chevrolet at the Moscow International Motor Show
2012-08-21
MOSCOW - Chevrolet will give motoring enthusiasts a taste of the present and the future when the 2012 Moscow International Motor Show – Russia's leading automotive exhibition – opens on Aug. 29.

GM Foundation Grants \$50,000 to National Kidney Foundation
2012-08-20
DETROIT – The GM Foundation today announced a \$50,000 grant to the National Kidney Foundation for its continued efforts to prevent kidney disease and educate the public about risk factors and early detection.

LATEST BLOGS
FastLane GM FastLane
2013 Chevrolet Spark a.k.a. "The Globe Trotter"

FACES Faces Of GM
Aniko Feher – An Artist in Many Media

GLOBAL NEWS
Chevrolet at the Moscow

3. For each news item

a. Extract a characteristic phrase (2-4 words)



Fit to be Dyed: Chevrolet Spark Suits Up for The Color Run™
Multicolor version of brand's first U.S. mini car will pace urban 5K races nationally
2012-08-21

NEW YORK – The all-new 2013 Chevrolet Spark, a city car offered in bold hues such as Jalapeno, Salsa Red, Lemonade, Denim Blue and Techno Pink, will pace nearly 10,000 runners in The Color Run™ this Saturday and Sunday at the Aviator Sports Center in Brooklyn, the first of numerous Spark-paced races across the country.

Chevrolet and The Color Run today announced a sponsorship deal that makes the Spark – Chevrolet's first mini car for the U.S. and Canadian markets – the official vehicle and pace car for The Color Run through 2013.

The Color Run is a popular, nationwide series of urban 5K races in which thousands of participants are doused from head to toe in different colors for each kilometer. Participation in and buzz for The Color Run have exploded since its debut last year, with nearly a half-million "likes" on Facebook.

"The Spark and The Color Run are perfect running mates because both appeal to style-conscious, high-energy urbanites who don't live life in neutral," said Cristi Landy, marketing director, Chevrolet Spark. "We hope Color Runners will like what they see when they meet the Spark."

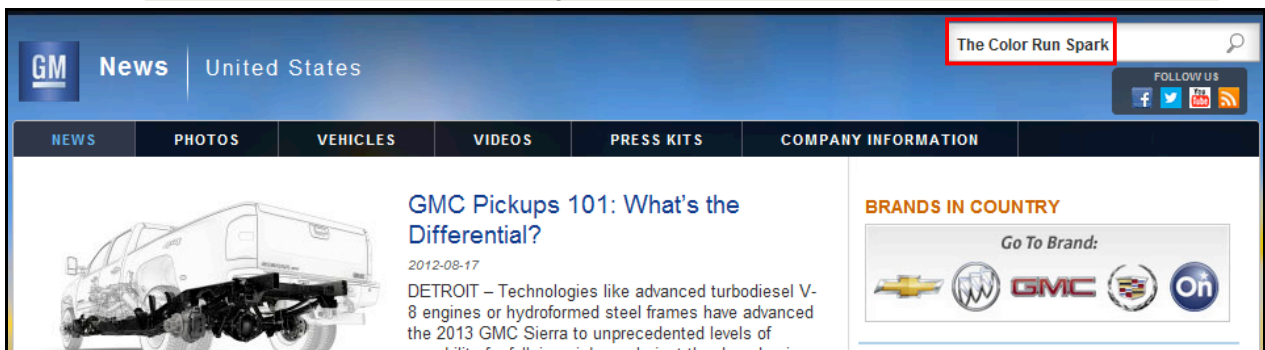
Pacing The Color Run events in New York is a specially designed Chevrolet Spark with a multicolor paint scheme over a white base that replicates the look of a color-saturated runner at the finish line. The theme continues inside with splashes of bright color throughout an

"Our team wanted **The Color Run Spark** to be a pace car that will energize and inspire the crowd to take a closer look," said GM's senior design manager who worked on the production version of the Spark.

This phrase is presumed to be unique to this document and therefore characteristic.

A characteristic phrase³ in this context is a phrase which is only present in a single document. It is important to identify such a phrase to be able to retrieve the intended document. In IR terms, this is called *known-item retrieval*.

b. Search the news items using that phrase



GM News | United States

Search: **The Color Run Spark**

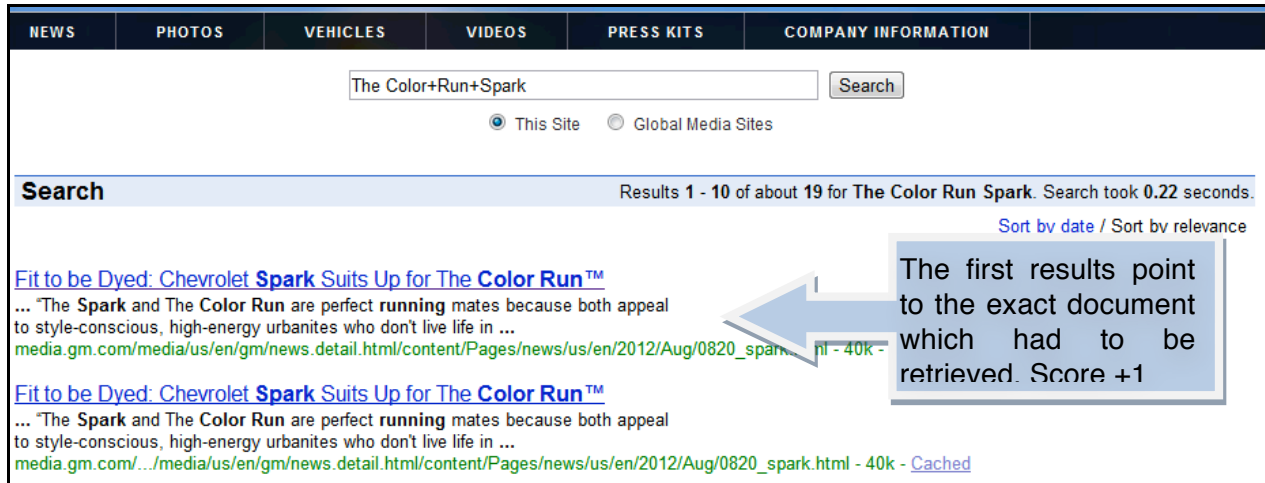
NEWS | PHOTOS | VEHICLES | VIDEOS | PRESS KITS | COMPANY INFORMATION

GMC Pickups 101: What's the Differential?
2012-08-17
DETROIT – Technologies like advanced turbodiesel V-8 engines or hydroformed steel frames have advanced the 2013 GMC Sierra to unprecedented levels of capability for full-size pickups. In just the decade since

BRANDS IN COUNTRY
Go To Brand:
Chevrolet GMC OnStar

³ Needed in criteria: Completeness, Freshness, Office Document Handling, Known / Suspected Item Retrieval

4. Score according to the news items retrieved in the top 10 search results (0, 1, 2)



NEWS PHOTOS VEHICLES VIDEOS PRESS KITS COMPANY INFORMATION

The Color+Run+Spark Search

☒ This Site ☐ Global Media Sites

Search Results 1 - 10 of about 19 for The Color Run Spark. Search took 0.22 seconds. Sort by date / Sort by relevance

[Fit to be Dyed: Chevrolet Spark Suits Up for The Color Run™](#)
... The Spark and The Color Run are perfect running mates because both appeal to style-conscious, high-energy urbanites who don't live life in ...
[media.gm.com/media/us/en/gm/news.detail.html/content/Pages/news/us/en/2012/Aug/0820_spark.html - 40k -](#)

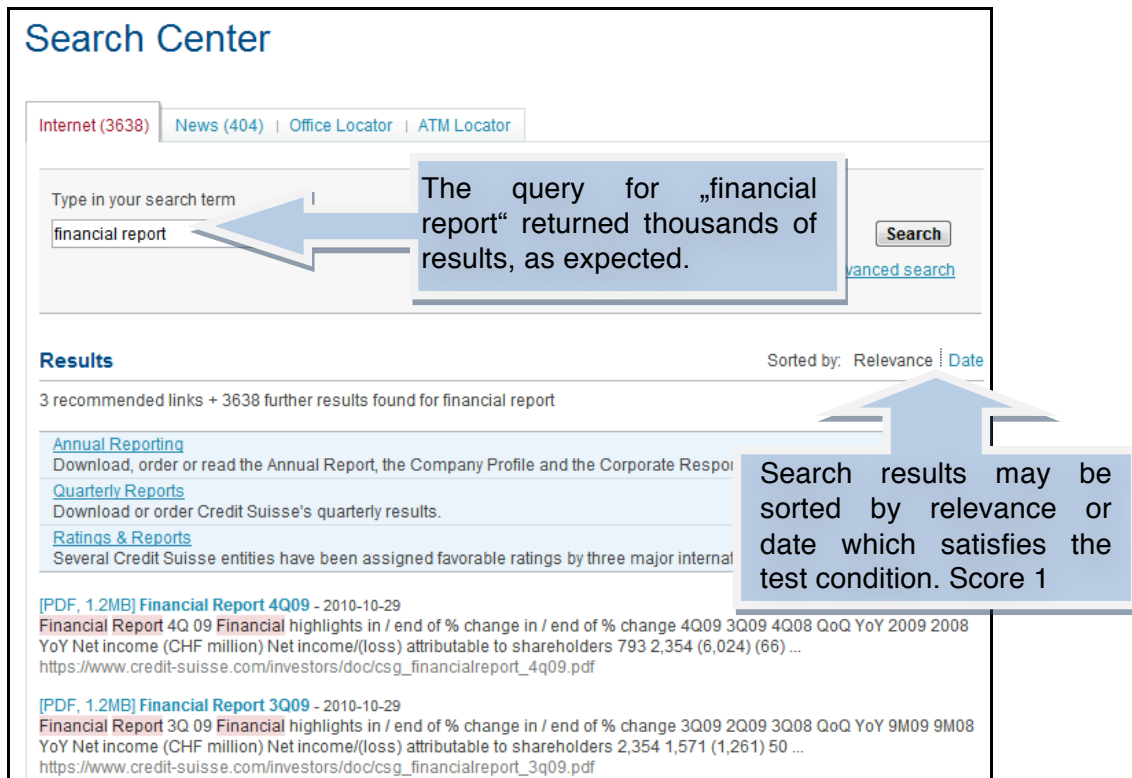
[Fit to be Dyed: Chevrolet Spark Suits Up for The Color Run™](#)
... The Spark and The Color Run are perfect running mates because both appeal to style-conscious, high-energy urbanites who don't live life in ...
[media.gm.com/.../media/us/en/gm/news.detail.html/content/Pages/news/us/en/2012/Aug/0820_spark.html - 40k - Cached](#)

Incidentally, this example also showed duplicates in the result list. If duplicates are also being tested, one would already note the negative score for that test as well.

4.7.2 Sorting of Result List

Using the user interface criterion of “Sorting of Result List” we demonstrate a simple feature checking test on the web search application of Credit Suisse⁴.

1. Enter a query which yields a reasonable amount of results (> 100)
2. Score success if the results can be sorted by suitable criteria (0 or 1)



Search Center

Internet (3638) News (404) Office Locator ATM Locator

Type in your search term
financial report Search

[Advanced search](#)

Results Sorted by: Relevance | Date

3 recommended links + 3638 further results found for financial report

[Annual Reporting](#)
Download, order or read the Annual Report, the Company Profile and the Corporate Responsibility Report.

[Quarterly Reports](#)
Download or order Credit Suisse's quarterly results.

[Ratings & Reports](#)
Several Credit Suisse entities have been assigned favorable ratings by three major international rating agencies.

[PDF, 1.2MB] [Financial Report 4Q09](#) - 2010-10-29
Financial Report 4Q 09 Financial highlights in / end of % change in / end of % change 4Q09 3Q09 4Q08 QoQ YoY 2009 2008
YoY Net income (CHF million) Net income/(loss) attributable to shareholders 793 2,354 (6,024) (66) ...
https://www.credit-suisse.com/investors/doc/csg_financialreport_4q09.pdf

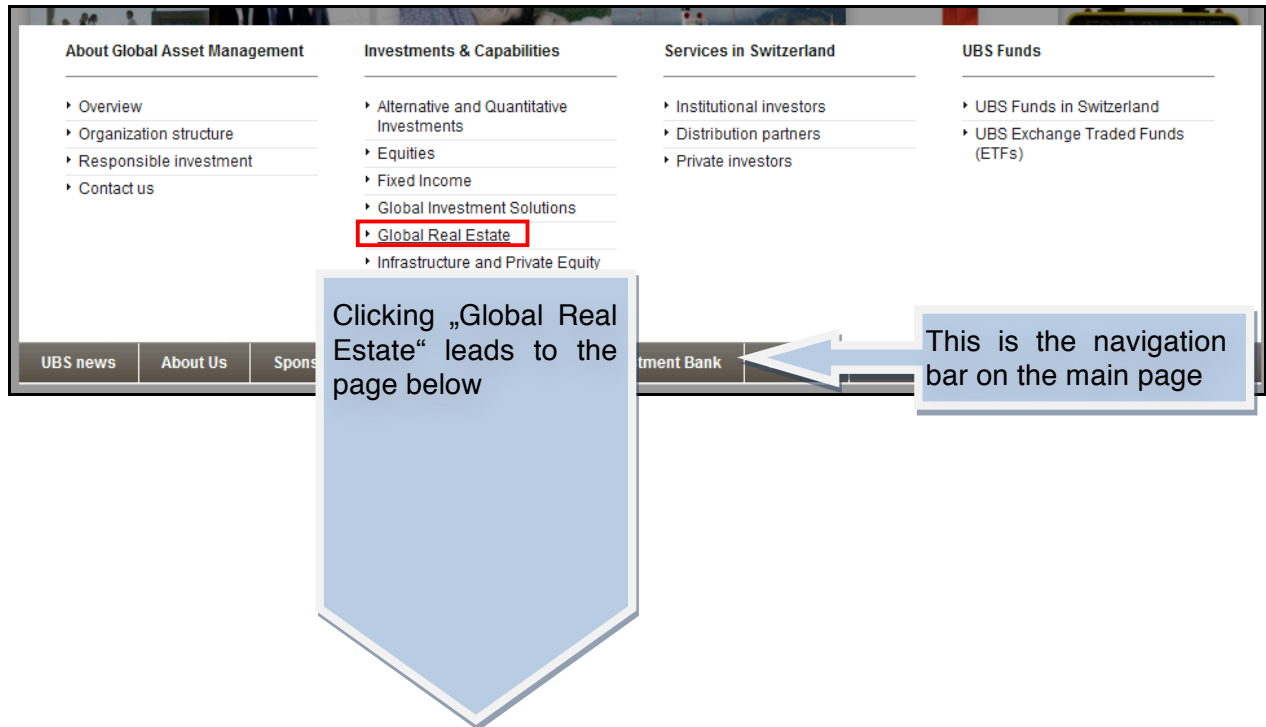
[PDF, 1.2MB] [Financial Report 3Q09](#) - 2010-10-29
Financial Report 3Q 09 Financial highlights in / end of % change in / end of % change 3Q09 2Q09 3Q08 QoQ YoY 9M09 9M08
YoY Net income (CHF million) Net income/(loss) attributable to shareholders 2,354 1,571 (1,261) 50 ...
https://www.credit-suisse.com/investors/doc/csg_financialreport_3q09.pdf

⁴ All screenshots and material © 1997 - 2012 CREDIT SUISSE GROUP AG and/or its affiliates. All rights reserved.


4.7.3 Navigational Queries

This is a demonstration of the test for the search results criterion “Navigational Queries” on the web search application of UBS⁵.

1. Quickly browse the site and identify an entry point to some topically similar documents, e.g. using the application site's navigation bar



⁵ All screenshots and material © UBS 1998-2012. All rights reserved.



[Global homepage](#)
[Country ▾](#)
[Contact ▾](#)
[English ▾](#)
[Mobile](#)
[Sitemap](#)

[Global home ▾](#) > [Global Asset Management ▾](#)

Global Real Estate


[About UBS Global Asset Management](#)

Global Real Estate

- [History](#)
- [Global Multi-Managers](#)
- [Global Real Estate Securities](#)
- [Global Customized Client Mandates](#)
- [Sustainable investing](#)
- [Research](#)
- [Contacts](#)
- [Offices](#)


Global Real Estate

Global Real Estate invests in properties in Continental Europe, Japan, UK and US and in publicly traded real estate securities worldwide. It actively manages investments in property including office, industrial, retail, multi-family residential, hotel and farmland real estate. Capabilities include core and value-added private strategies on a global, country and regional basis. These are offered through a variety of investment vehicles.




Americas

- [United States](#)



Asia Pacific

- [Australia](#)
- [Japan](#)



Europe

- [Germany](#)
- [Luxembourg](#)
- [Switzerland](#)
- [United Kingdom](#)

2. Build a query with a few words using the information on the entry point

3. Score success if entry point could be retrieved in the top 10 results (0 or 1)

1586 search results, of which [244 in news](#)

Results 1 - 20 all ubs.com in all l

Searching for „Global Real Estate“ returned the same page as previously navigated to as the top item. Score 1

Sort by Best Match

Global Real Estate

Global Real Estate invests in properties

Capabilities include core...value-added

Global home > Global Asset Management > Global Real Estate

and farmland real estate.

ry and regional basis...

Global Real Estate - Australia

Global Real Estate - Australia provides listed property investment...measured over rolling three-year periods.

More Global Real Estate Securities Global Real Estate Securities enable...

Global home > Global Asset Management > Global Real Estate > Global Real Estate - Australia

Global Real Estate (US - Securities - REITS)

...markets. We also offer actively managed global real estate securities portfolios. Contact

information...Thorpe-Apps Chief Investment Officer Global Real Estate Securities +44-20-7901 5567

Global home > Global Asset Management > us > gre > Global Real Estate - UBS Global Real Estate Securities

5 Conducting an Evaluation, Step By Step

Step 1: Define the Context

You already have a clear idea of which application(s) you need to evaluate. Identify your evaluation scenario (more in section 4.4) accordingly:

- *Monitoring*: Evaluation of a single application with the intention of repeating the same evaluation at a later time to observe performance developments
- *Comparison*: Evaluation of multiple applications to assess their relative performance, e.g. comparing several product options for acquisition
- *Campaign*: Evaluation of many applications for market or academic research

Identify the use case domain you are going to operate in. Putting a label on the use case domain for each evaluation helps you to identify shifts in the applications' served domains. Results are only comparable if the evaluation has been done in the context of the same use case domain. When monitoring a single system, domain shifts become immediately apparent and previous evaluation results should not be compared against anymore.

If you need to evaluate several applications, consider carefully if they really fit in a single domain. Discard any applications which do not fit for the purposes of the evaluation. Otherwise you risk invalidating the results.

Examples for use case domains include the PROMISE use case domains of Cultural Heritage, Medical Image Retrieval and Search for Innovation (patent retrieval). Other more generic examples:

- Enterprise Search: Search functionality for information needs about a company, used as a communication tool (e.g. banks, insurances, etc.)
- Retailer: Search on products and their descriptions (e.g. Amazon)
- Media Provider: Search on media contents (e.g. news outlets)

Results at the end of this step:

- Definition of evaluation scenario
- Definition of use case domain
- List of applications to be evaluated within that use case domain

Step 2: Select Applicable Criteria

With the previously defined evaluation scenario, use case domain and application list, you have the required information to select applicable criteria from the catalogue in section 4.6. Consult the catalogue and note all applicable criteria. The methodology is insensitive to criteria which are falsely deemed applicable. During evaluation, the associated tests will be marked as not applicable by testers. These considerations are mainly done for evaluation efficiency reasons.

Define the weights of your selected criteria. Keep in mind that tests are coarse and simple. Assigning finely grained weights will not provide more specific results. If you assign weights other than 1, make sure there is ample reason to do so and document it.

Results at the end of this step:

- List of applicable criteria and their weights

Step 3: Create Score Spread Sheet and Test Script

Create a spread sheet which can be used to note scores. Enter the tests associated with the applicable criteria as rows and application names as columns. Scores will be noted in the cross point cells of tests and applications. Optionally add score range information to each test as a helpful quick glance feature for testers⁶.

Prepare the test script using the test catalogue in section 4.6. Take care to include the assumption and any examples given outside of the test or assumption sections for a maximum of clarity.

Results at the end of this step:

- Score spread sheet
- Test script

Step 4: Acquire and Instruct Testers

Testers need to have reading comprehension skills in the languages which are present in each of the evaluated applications and be able to interact with the applications without any handicaps. No further skills or knowledge should be required. This is primarily due to the way tests are defined. They implicitly model how a prototypical user would act and experience the application. The tests are detailed enough not to leave room for misinterpretation of the test procedure if some further instruction is provided.

⁶ Make sure to copy the correct values. Previous tests have shown strong irritation in testers when score ranges on the score sheet differ from those in the test script.

However, some use case domains require some domain knowledge or familiarity with domain specific applications. Especially when the formulation of queries is required, domain knowledge is a prerequisite to avoid mismatches.

Distribute the score sheet and test script to your testers. Let them read all the tests first and then define a standard procedure for any unclear tests. Consistently communicate standard procedures to all testers⁷.

Here are some important definitions of the terminology used in the tests from section 4.6. For more comprehensive descriptions, please consult section 3.

- *Document*: this can be any document unit within an application, e.g. HTML pages, PDF documents, office format documents or media format documents
- *Characteristic phrase*: a sequence of words (2+) which is presumably only present in a single document for which it is characteristic

To estimate the time effort and costs, expect a single evaluation run to take about 4 hours on average per application (may take less if many criteria are deemed not applicable).

Results at the end of this step:

- Testers acquired
- Test script and score sheets distributed to testers
- Testers instructed
- Time and cost estimate of evaluation

Step 5: Run the Evaluation

Let testers begin the evaluation and collect their results. If testing procedures change during the evaluation based on new experiences or knowledge, communicate any new procedure and let the affected criteria be re-tested.

For comparison and campaign scenarios, let multiple testers evaluate each application to lessen tester bias. In the monitoring scenario, make sure you employ the same group of testers in each iteration of the evaluation.

Results at the end of this step:

- Collected evaluation results, ready for comparison and further processing

Step 6: Iterate

Especially in the case of the monitoring scenario, you will need to evaluate the applications again at a later point of time to observe changes. Use the same test script and employ all standard procedures you may have defined in earlier iterations.

⁷ Also, in case of a monitoring evaluation, document the procedure to retain consistency over time.

6 Validation Efforts

As a validation inside PROMISE, a campaign was conducted where each participating PROMISE partner was asked to identify ten target sites which they would evaluate. The sites were required to be based in the partners' country, would fit in 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.

A fifth category was introduced in the campaign where testers were asked to give their subjective impressions of the tested applications. The goal of the criteria contained therein was to correlate the testers' general impression with the overall score of the application in order to validate the plausibility of *user perception* as the chosen measure.

The results and experiences made during the campaign have served as ground work to further improve on the methodology. These improvements are already incorporated into this document.

After the campaign the criteria were revised according to the insights from the evaluation of the campaign as well as the feedback from the participating PROMISE partners. Overlapping criteria have been removed or their differences were explained in more detail. Some of the criteria resulted in plenty of aborts. After consultation with the testers, it was realized that the test processes were not described clearly enough. Therefore criteria and test descriptions have been overhauled and in some cases even split up into several smaller tests.

The evaluation of the campaign showed that the overall score of each application correlates with the testers' subjective impressions. Unsurprisingly, a large variance was noted for the scores of all evaluated applications from different use case domains. For applications within the same use case domain, variance was significantly smaller. Therefore, it is necessary to compare only applications from the same use case domain.

7 Outlook

This deliverable covers only one methodology which can be used outside of PROMISE. As part of the PROMISE task 4.5, tutorials will be given for industry practitioners to enable them to conduct in-house evaluations. The tutorials will be based on the material in this deliverable as well as further work, wherein other methodologies are adapted to be usable in operational settings.

Considering the black box application evaluation methodology, the measure of user perception is to be explored more thoroughly. Furthermore, the criteria and test catalogue needs to be expanded to cover more aspects which can be tested in black box settings. Additionally, criteria can be based on and connected to best practices from the PROMISE best practices report D2.3 which is being done in parallel to this deliverable.

Some of the tests lend themselves to automation. This depends on the used scenario: for monitoring, for comparison or for a campaign. If automating a test requires programming work for each evaluated search service, automation is a good idea mainly for monitoring and comparison, where few sites are evaluated. But some tests could be automated in a rather site-independent way. Let us look at example test algorithm:

Category: Index
Criterion: Completeness

Automated test:

- Step 1: obtain a crawl starting from the URL of the search service, limiting URLs to the same top level domain (TLD)
- Step 2: Select three random pages from this crawl that have a shortest path from the root URL of at least 5 clicks
- Step 3: For each document, extract a characteristic phrase, using e.g. statistically improbable phrases⁸, or log-likelihood ratio [Berendsen et al. 2012a]
- Step 4: For each phrase, search for it and crawl the search result page (SERP)
- Step 5: Score 0-3 according to the number of phrases for which the document from which it was taken is found.

Note that this idea of automation of tests uses similar ideas as work we report on in deliverable 4.3 [Berendsen et al. 2012b], where we generate pseudo test collections for training and evaluating retrieval algorithms. Other tests that may either partially or completely analysed include:

Category: Index
Criteria: Freshness, Tokenization, Named Entities, Separation of Actual Content and Representation

Category: User Interface
Criteria: Performance / Responsiveness, User Guidance

Automating these tests raises interesting challenges, for example performing named entity recognition, extracting “meaningful” phrases, “characteristic phrases”, near duplicate detection, and so on. In future work, we intend to explore the possibilities outlined above.

⁸ <http://www.amazon.com/gp/search-inside/sipshelp.html>

Acknowledgements

We thank the authors of the studies of the Swiss and German enterprise search portals [Braschler et al. 2006] [Braschler et al. 2009] which have inspired the methodology as presented in this deliverable.

We also thank all PROMISE partners and their associates who were able to assist us in the execution of the guerrilla evaluation campaign.

References

- | | |
|--------------------------|--|
| [Berendsen et al. 2012a] | Berendsen R., Tsagkias E., de Rijke M., Meij E., <i>Generating Pseudo Test Collections for Learning to Rank Scientific Articles</i> , CLEF 2012: Conference and Labs of the Evaluation Forum, Rome, Italy, Springer, September, 2012 |
| [Berendsen et al. 2012b] | Richard Berendsen, Maria Gäde, Michael Kleineberg, Mihai Lupu, Vivien Petras, Stefan Rietberger, <i>Deliverable 4.3. Final Report on Alternative Evaluation Methodology</i> , PROMISE Network of Excellence, 2012. |
| [Braschler et al. 2006] | Braschler, M.; Herget J.; Pfister, J.; Schäuble P.; Steinbach, M.; Stuker, J. <i>Evaluation der Suchfunktion von Schweizer Unternehmens-Websites</i> . 2006 |
| [Braschler et al. 2009] | Braschler, M.; Heuwing, B.; Mandle, T.; Womser-Hacker, C.; Herget, J.; Schäuble, P.; Stuker, J. <i>Evaluation der Suchfunktion deutscher Unternehmenswebsites</i> . 2009 |
| [Cleverdon 1967] | Cleverdon, CW <i>The Cranfield tests on index language devices</i> . 1967 |
| [Cormack et al.1998] | Cormack, GV.; Palmer CR.; Clarke CLA <i>Efficient construction of large test collections</i> . 1998 |
| [Hofmann et al. 2011] | Hofmann K., Whiteson S., de Rijke M., <i>A Probabilistic Method for Inferring Preferences from Clicks</i> , 20 th ACM Conference on Information and Knowledge Management (CIKM 2011), Glasgow, ACM, pp. 249-258, October, 2011 |
| [Hofmann et al. 2012] | Hofmann K., Whiteson S., de Rijke M., <i>Estimating Interleaved Comparison Outcomes from Historical Click Data</i> , CIKM 2012: 21 st ACM Conference on Information and Knowledge Management: ACM, October, 2012 |
| [Jansen 2006] | Jansen, B.J. <i>Search log analysis: What it is, what's been done, how to do it</i> . 2006 |
| [Jansen and Pooch 2001] | Jansen, B.J; Pooch, U. <i>Web user studies: A review and framework for future work</i> . 2001 |
| [Järvelin et al. 2012] | Järvelin, A.; Eriksson, G.; Hansen, P.; Tsikrika, T.; Garcia Seco de Herrera, A.; Lupu, M.; Gäde, M.; Petras, V.; Rietberger, S.; Braschler, M.; Berendsen, R. <i>Deliverable D2.2 - Revised Specification of Evaluation Tasks</i> , PROMISE Network of Excellence, 2012 |

- | | |
|--|---|
| [Kelly 2009] | Kelly, D., <i>Methods for Evaluating Interactive Information Retrieval Systems with Users</i> , In: Foundations and Trends in Information Retrieval, 2009 |
| [Kohavi et al. 2007] | Kohavi, R.; Henne, R.; Sommerfield, D. <i>Practical Guide to Controlled Experiments on the Web: Listen to Your Customers not to the HiPPO</i> . 2007 |
| [Peters et al. 2012] | Peters, C.; Braschler, M.; Clough, P.: <i>Multilingual Information Retrieval: From Research To Practice</i> , Springer, 2012, ISBN 3642230075 |
| [Radlinski et al. 2008] | Radlinski, F.; Kurup, M.; Joachims, T.; <i>How Does Clickthrough Data Reflect Retrieval Quality?</i> 2008 |
| [Sanderson and Braschler 2009] | Sanderson, M.; Braschler, M. <i>Best Practices for Test Collection Creation and Information Retrieval System Evaluation</i> . November 2009 |
| [Spärck Jones and van Rijsbergen 1975] | Spärck Jones, K.; Van Rijsbergen, C.J. <i>Report on the need for and provision of an 'ideal' information retrieval test collection</i> . 1975 |