Journal of Universal Computer Science, vol. 17, no. 6 (2011), 831-858
submitted: 10/4/10, accepted: 25/3/11, appeared: 28/3/11 © *J.UCS*

A Framework to Evaluate Interface Suitability for a Given Scenario of Textual Information Retrieval

Nicolas Bonnel
(ThinkCollabs, Rennes, France
nico@thinkcollabs.org)

Max Chevalier
(IRIT/SIG, Université de Toulouse, UMR 5505, France
Max.Chevalier@irit.fr)

Claude Chrisment
(IRIT/SIG, Université de Toulouse, UMR 5505, France
Claude.Chrisment@irit.fr)

Gilles Hubert
(IRIT/SIG, Université de Toulouse, UMR 5505, France
Gilles.Hubert@irit.fr)

Abstract: Visualization of search results is an essential step in the textual Information Retrieval (IR) process. Indeed, Information Retrieval Interfaces (IRIs) are used as a link between users and IR systems, a simple example being the ranked list proposed by common search engines. Due to the importance that takes visualization of search results, many interfaces have been proposed in the last decade (which can be textual, 2D or 3D IRIs). Two kinds of evaluation methods have been developed: (1) various evaluation methods of these interfaces were proposed aiming at validating ergonomic and cognitive aspects; (2) various evaluation methods were applied on information retrieval systems (IRS) aiming at measuring their effectiveness. However, as far as we know, these two kinds of evaluation methods are disjoint. Indeed, considering a given IRI associated to a given IRS, what happens if we associate this IRI to another IRS not having the same effectiveness. In this context, we propose an IRI evaluation framework aimed at evaluating the suitability of any IRI to different IR scenarios. First of all, we define the notion of IR scenario as a combination of features related to users, IR tasks and IR systems. We have implemented the framework through a specific evaluation platform that enables performing IRI evaluations and that helps end-users (e.g. IRS developers or IRI designers) in choosing the most suitable IRI for a specific IR scenario.

Keywords: Textual Information Retrieval Systems, Interface Suitability for IR scenario, Visual Information Retrieval.
Categories: H.3.3

1 Introduction

Retrieving information is generally done through an interface which enables the user to specify his query and to visualize the search results. In this context, a particular

attention should address the last step of the ad-hoc[1] textual Information Retrieval (IR) process. This step consists in visualizing search results in a specific manner through an Information Retrieval Interface (IRI). This latter component is very important in the IR process since it aims at facilitating the process of search results by users (finding documents relevant to his information needs). An important issue is the choice of the most suitable IRI for a specific IR scenario.

This paper deals with IRI evaluation which is a difficult task. The proposal aims at evaluating the suitability of any IRI for the various possible IR scenarios. Up to now, no real investigation in this direction has been done. Evaluating interfaces exist but is mainly based on a usability study even though many evaluation levels can be considered [Thomas and Cook 2005]. This type of evaluation is not sufficient from our point of view. For example, it does not allow evaluating if a given IRI is really suitable for a specific IR scenario (a specific IR task carried out by a specific user using a specific system). In this paper, we suggest a complementary approach to classical evaluation techniques in order to evaluate this suitability. More precisely, we define a specific evaluation framework aiming at evaluating any IRI in the same and replicable way. This work requires the definition of IR scenarios (described by specific features) in order to identify if a specific user using a specific IR system achieves a specific IR task through a specific IRI. Statistical techniques are then used to compute the suitability scope of an IRI. As a result, we also propose a specific and complete platform that allows one to evaluate or select IRIs according to various scenarios.

This article is organized as follows. **Section 2** presents the motivations of our approach. We then present in **section 3** some examples of IRIs to illustrate the context of our work, coming from general contexts (e.g. the Web) or from more specific ones (e.g. Digital Libraries). So, although several interfaces described are applicable to the web, this section is not limited to this context. Then, **section 4** deals with current evaluations of IRIs and identifies some limits. **Section 5** defines IR scenarios based on specific features. We describe in **section 6** the different parts of our evaluation framework and the possible individual and global analyses. **Section 7** provides details on the framework implementation, on the evaluation process, and on the way to evaluate the suitability of IRIs for IR scenarios. We also present, in this section, the resulting prototype aiming at making easy the choice of any IRI. Finally, **sections 8 and 9** present conclusions and future work.

2 Motivations

Numerous interfaces appear in the literature as presented in section 3. Search result visualization and more generally interaction with the user are a key point of information retrieval process. In view of this proliferation and diversity of IRIs, evaluation is a mean to distinguish between them. As described in section 4, the literature contains various evaluation methodologies to measure interface usability. In IR domain, besides measuring usability, it is necessary to evaluate the contribution of

[1] "In ad-hoc retrieval an IR system normally relies on a user's query as a clue to select documents and rank them for output to satisfy users' needs" (Kwok, 1996).

an interface in information search. The question is: Are interfaces with a good usability suitable for all the information searches. This question was discussed in InfoVis [Plaisant et al., 2008] in the general context of data visualization. One originality of our approach is that it is domain-specific (i.e. IR domain) and thus has to take into account characteristics of this domain. However, this approach is applicable to other domains. In IR, information searches vary according to user, goal, and document characteristics, for example. This raises a more precise question: Is a given interface suitable for a given IR scenario?

Measuring suitability of a given interface to a given scenario will be interesting for IRS developers, IRI designers, and IR community:

- It will help IRS developers to choose among common IRIs the one suitable for future users and uses of their systems;
- It will help IRI designers to identify IR scenarios for which their IRI is suitable and to improve their IRI in order to be suitable to other IR scenarios. Moreover, it will help IRI designers to determine the prior target of their system;
- It will help IR community to know the benefit of a given IRI in the IR process. Moreover, it will enable deeper analyses to identify IRI features that are suitable for a given IR scenario.

In order to offer an answer to these expectations, we propose a framework to evaluate suitability of IRIs according to various IR scenarios, as detailed in sections 5 and 6.

3 Variety of Information Retrieval Interfaces

Many IRIs have already been proposed. Some of them are used whereas others did not pass the prototype stage. However, the major problem remains the great disparities between these various interfaces and their evaluations. Due to the diversity of baselines [Julien et al. 2008], comparisons and meta-analysis still remain difficult. To illustrate our matter, we cite below some IRIs. Many other systems exist for example applied to the Web. The reader may for instance read the list of "Top Visual Search Engines[2]."

Common IRIs, and probably the most used, consist in a linear display of results as a list ordered according to a "(system) relevance" criterion (e.g. Google): a ranked list visualization. This visualization has many drawbacks [Berenci et al. 1998]. An evolution of this result list is the display of document groups as done in Grouper [Zamir and Etzioni 1999] or in the search engine Clusty[3]. The latter one clusters on-the-fly the first 100 results to obtain categories. The clustering method applied depends on the expected result. Clustering can be hierarchical, which enables to obtain a finer distribution of the results in the classes, as proposed by Grokker[4] in its

[2] http://www.masternewmedia.org/top-visual-search-engines-the-most-interesting-ways-to-visually-explore-search-engine-results/
[3] www.clusty.com
[4] www.grokker.com

2D interface. Other kinds of 2D IRIs use also this clustering approach, more precisely cartographic 2D interfaces such as IRIs based on self-organizing maps (e.g. the WEBSOM project [Lagus et al. 1996]). Other interfaces use this kind of cartographic 2D visualizations, generally based on links that exist between the various results. This is the case of the metasearch engine KartOO[5] that displays results on a map. In this kind of interfaces, users can visualize links between results and, for example, some common "keywords". However, KartOO presents results as a succession of maps, so the linear aspect persists. Many other works related to metasearch can be found in [Spoerri 2006, Koshman et al. 2006].

Other interfaces go a step further in result restitution (more graphical or metaphoric) with the use of "3D spatialization" of the retrieved results. There are two main approaches: the first one is "simple 3D" oriented where the third dimension aims to increase the display space, and the second one can be described as a 3D virtual environment in which the user walks in the results. Concerning the first approach, we highlight various proposals. A first proposal refers to the 3D visualization of a tree such as in Cat-a-Cone [Hearst and Karadi 1997] which uses a Cone Tree [Robertson et al. 1991] to display simultaneously obtained results and a hierarchy of predefined categories. Other proposals aim at distributing documents according to different features such as weights of query terms, like for example in [Rohrer and Ebert 1999, Houston and Jacobson 2000], NIRVE [Cugini et al. 2000], or in Easy-DoR [Chevalier et al. 2004]. The second approach is based on 3D cartographic metaphors, such as in the ViOS[6] interface. The prototype SmartWeb [Bonnel et al. 2005, Bonnel et al. 2006] uses also this concept to represent search results in a 3D virtual city. Unlike ViOS, SmartWeb organizes search results in a 3D space according to a self-organizing map, which enables grouping (and placing) results according to word distribution and thus to have a semantic proximity.

To avoid getting lost in these existing IRIs, we propose a taxonomy of IRIs (Figure 1). This classification presents the main advantage to integrate all IRIs which can be textual, 2D or 3D based. The various elements are not exclusive and many of them can appear in a same IRI.

However the major problem in the evaluation of these IRIs is the fact that they are not all based on the same retrieval system and that they do not all propose the same processing (such as clustering or result filtering). So identifying the "best" IRI is rather difficult especially when this judgment depends on various elements. We identified these elements, which allow us to build an evaluation and comparison platform. However, before presenting these various elements the next section presents usual evaluation approaches of interface suitability in order to identify their limits.

[5] www.kartoo.com
[6] http://en.wikipedia.org/wiki/ViOS

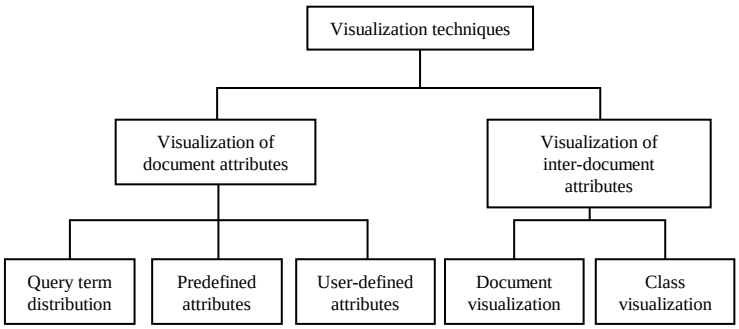


Figure 1: IRI classification based on visualized characteristics

4 Usual evaluation of interface usability

To evaluate interfaces many approaches may be used. These evaluations mostly aim at evaluating interface usability. We can split them into two main trends: analytic methods and empiric methods. Following sections are related to most common ways to evaluate interfaces. Many other trails could be found. In [Canas 2008] for instance, Canas argues that an interface should be analyzed through "the mutual dependency between interface functions and user functions and the cognitive level of interaction." Such analysis could be also used to evaluate interfaces.

4.1 Analytic Methods

Analytic methods use a simulation of user activities without real implication of users. Experts do this simulation.

4.1.1 Heuristic Evaluation

Nielsen and Molich in 1990 proposed an heuristic evaluation [Nielsen and Molich 1990]. This evaluation is based on an analysis done by experts who use heuristic approaches to identify ergonomic issues and rate their severity. To achieve this evaluation, Nielsen and Molich proposed nine heuristic approaches among which are:

- Use a simple and natural language,
- Minimize user cognitive overload,
- Prevent errors,
- Feature feedback to user…

The effectiveness of heuristic evaluation mainly depends on the number of experts and the interface itself. Indeed, some interfaces are simpler to evaluate with heuristic principles than others. It is also important to highlight the high cost of such evaluations due to the high number of experts required.

4.1.2 Cognitive Walkthrough (Mental Simulation)

Cognitive walkthrough [Wharton et al. 1994] consists in simulating the use of an interface by a user characterized by a specific profile. Experts still do this simulation. This evaluation follows three main steps:

- Define required data (listing actions, describing the interface…),
- Execute actions,
- Explain obtained results.

This evaluation is effective because it is based on actions and allows one to identify a high number of issues. A drawback of this method is the fact that expert must simulate the user's behavior using his profile.

4.1.3 Other Analytic Methods

- *Guidelines review*: Guidelines correspond to the interface directives (how an interface should be to be effective…). Experts when verifying if guidelines are applied in the interface identify the issues. Experts in interface design propose these guidelines. Some guidelines can be found in [Brown 1988, Smith and Mosier, 1986].
- *Formal usability inspection*: Dumas and Janice in 1999 underlined that to achieve a good evaluation experts should meet some representative users to discuss formally on interface issues and evolutions [Dumas and Janice 1999].

4.2 Empiric Methods

Empiric evaluation is based on observations of real users' behavior and attitudes when using an interface.

4.2.1 Questionnaires and Interviews

A well-known method to evaluate an interface consists in using questionnaires. For instance, Chin et al in 1988 proposed a standard questionnaire called QUIS (Questionnaire for User Interaction Satisfaction) [Chin et al 1988]. This questionnaire is complete since it refers to all the issues that can be identified when using an interface. As it is complete, some parts of this questionnaire can be ignored if the interface does not have some of the actions.

Another way to collect explicit user feedback consists in individual interviews. This method aims at pointing out critical issues identified by a user during real interactions with interfaces. Kuhn in 2000 indicates that it is also important to make a meeting with all users to homogenize the way users describe issues [Kuhn 2000].

4.2.2 Acceptability Test

Shneiderman and Plaisant in 2005 identified that testing the acceptability of an interface is difficult [Shneiderman and Plaisant 2005]. Indeed, evaluation requires knowing all the user's needs and skills… Therefore, to simplify this test they propose 5 criteria to evaluate interface acceptability: learning time, execution speed of actions,

error rate, "remanence" time (how long a user reminds how to use the interface?), and subjective satisfaction.

4.2.3. Observation

Such evaluation is based on the observation of users when interacting with an interface. For instance, observations can be made with a video camcorder and/or a microphone. Badre et al in 1993 [Badre et al 1993] justified that "Video offers a number of advantages including recording of the entire context of an interaction (e.g., use of documentation, environmental distractions...), the ability to hear verbal or thinking-aloud style comments [Lewis & Mack 1982] made by the user, and simply the ability to see exactly what the user is doing at a given point". Then, to identify issues and potential evolutions, experts analyze such observations.

4.3 Discussion

The aforementioned evaluations (analytic and empiric) aim at evaluating interface usability. Some studies have been done in order to evaluate performance of such usability evaluation methods [Hartson et al., 2003][Grice, 2003][Koutsabasis et al., 2007].

However, from the IR point of view even if such evaluation methods are applicable to evaluate IRI usability, one key issue has not been studied yet: Is the interface suitable for a given IR scenario? Indeed, even if an interface has a good usability, it may not meet a specific user doing a specific information retrieval task.

In the IR domain, evaluation is usually performed with regards to system effectiveness. To carry out such evaluation, some benchmarks exist such as TREC[7], CLEF[8] or INEX[9]. However, IRI is not considered as a component which should be evaluated independently even in specific tasks qualified as "interactive" proposed in these benchmarks. Indeed, even if user interaction is focused, the impact of the IRI on the IR process is lost in the intrinsic performance of the IRS. These benchmarks are not designed to evaluate the IRI impact on the IR process. This way of evaluating IRIs does not make it possible to identify if an IRI is suitable for a given IR scenario. We can define an IR scenario as a set of features characterizing the parts involving in the IR process (i.e. system, user, and task).

In this context, the main issues are:

- Identifying every characteristic related to "components" implied in the IR process;
- Varying the different characteristics to provide a wide range of IR scenarios to obtain a global evaluation of every IRI;
- Preserving the same and replicable way to evaluate any IRI.

As a solution the evaluation should rely on a unique IRS whose behavior can be controlled in order to vary performances. So, in our approach, we propose a

[7] Text REtrieval Conference - http://trec.nist.gov
[8] Cross-Language Evaluation Forum - http://clef.isti.cnr.it
[9] Initiative for the Evaluation of XML Retrieval - http://www.inex.otago.ac.nz

framework that aims at evaluating IRI by using a single and specific IRS to identify the IR scenarios where an IRI is appropriate (the user achieves or fails his IR task). To do such an evaluation, the framework relies on a virtual IRS that provides the search results related to every IR scenario. The proposed virtual IRS offers a lot of services to any IRI and contains specific test data (a list of documents associated to rsv[10] values and relevance judgments). First of all, we detail features characterizing any IR scenario and then we describe the evaluation framework.

5 Characterizing IR Scenarios

In order to evaluate the suitability of any given IRI for different IR scenarios, a first issue is to define the concept of IR scenario. We model a scenario of use as a triplet (**system**, **user**, **task**). We define a scenario as a set of features, each related to a member of this triplet. Varying each feature defines a particular scenario. Many features can be identified in [Lainé-Cruzel 1999]: "Who is the user?", "What is he looking for?", and "What is his aim?". Hölscher and Strube in 2000 confirm this vision in characterizing the user via two main kinds of knowledge (domain knowledge and practical knowledge) [Hölscher and Strube 2000].

Thus, in our evaluation framework, features are split in three categories:

- Those associated to the user (Who is the user?),
- Those associated to the information retrieval task (What is he looking for? What is his aim?),
- Those associated to the IR system and more particularly to search results. We introduce this category to characterize the difficulty for a user to manage some kinds of search results.

This section presents various features that can be useful to characterize IR scenarios. The list of features may be considered as a first proposal and might be extended. Adding features will offer both more precise scenarios and a wider range of scenarios. It will offer a better fine-grained way to distinguish IRIs with regards to suitability.

5.1 Features Related to Search Results

The following features characterize the set of documents retrieved by the IRS. The main interest of using these features is that most of them are well-known criteria used to evaluate IRSs, notably in TREC[11] campaigns. Thus, this set of features represents the quality of an IRS result.

- **Number of Retrieved Documents.** This criterion refers to the number of documents retrieved by the IRS. Indeed, an IRI can be effective for a low number of retrieved documents and on the contrary be ineffective when this number grows too high.

[10] rsv = retrieval status value. It corresponds to the relevance measure that any search engine computes according to a specific query.
[11] http://trec.nist.gov

- **Content Homogeneity of Retrieved Documents.** This criterion measures how much the content of retrieved documents deals with the same topic. It allows the system to evaluate if an IRI is effective when topics are numerous in the set of retrieved documents. In our approach we used the standard deviation of similarity between each retrieved document and the average document.
  Let $rd = \{d_1, d_2 \ldots, d_i\}$ be the set of retrieved documents;
  Where $d_i = \{(t_1, w_{dn,t1}), \ldots (t_m, w_{dn,tm})\}$ is the set of weighted terms corresponding to the document d_i content. The weight of every term conveys its importance in the document. Some additional information related to this vector based representation of document content can be found in [Manning et al. 2008].
  The content homogeneity is measured as:

$$ch(rd) = \sqrt{\frac{1}{|rd|}\sum_{i=1}^{|rd|} sim(d_i, \bar{d})^2}$$

  Where $\bar{d}$ corresponds to the "average document" described by a vector containing all the terms, each one associated to a weight that corresponds to:

$$w_{\bar{d},t_k} = \sqrt{\frac{1}{|rd|}\sum_{i=1}^{|rd|} w_{d_i,t_k}}$$

  Sim corresponds to a similarity measure such as the cosine measure. The more the *ch(rd)* value is near 1, the more the content is homogeneous.
- **Precision of Retrieved Documents.** This criterion indicates the ratio between relevant documents and the total number of document retrieved by the IRS. It can be computed using the following equation 1. A high value of this criterion indicates that many retrieved documents are relevant.

$$\text{Precision} = \frac{\#\text{RelevantDocuments}}{\#\text{RetrievedDocuments}} \quad (1)$$

- **TopN Precision.** This criterion gives the proportion of relevant documents in the N documents returned by the IRS as the most relevant ones. N can also be called the document cut-off value (DCV). The TopN Precision can be computed using equation 2. A high value of this criterion when N is low (e.g. N=10), it indicates that many relevant documents are in the first 10 returned by the IRS.

$$\text{TopN Precision} = \frac{\#\text{RelevantDocumentsInTheNFirstDocuments}}{\text{N}} \quad (2)$$

- **Standard Deviation of Relevant Document rsv.** This criterion conveys the distribution of relevant documents in the search result. It measures if documents are all situated around a specific rsv value or not. The standard deviation is computed using the rsv value of the relevant documents present in the search result.

5.2 Features Related to the User

Hölscher and Strube in 2000 pointed out that the success of information retrieval depends on two main features: domain knowledge level and practical knowledge level [Hölscher and Strube2000].

- **Domain Knowledge Level.** This knowledge influences, among others, the way the user manages search results. Indeed, the higher the domain knowledge is, the better the identification of relevant documents in the search result is. The explanation is that users must know the domain of their information need to identify relevant information.
- **Practical Knowledge Level.** This criterion gives information on the ability of the user to use a computer framework for IR. This knowledge also influences the way the user manages search results. Intuitively, the higher the practical knowledge is, the better the user's possibilities to find relevant documents among the search results are. This knowledge gathers various notions such as computer use, IRS manipulation and manipulation of 3D interfaces. An adequate questionnaire may be used to estimate these levels.

These two features are characterized by qualitative values such as neophyte, average and expert, for example. For a given user, it is possible to identify the values of these two features using appropriate questionnaires (via closed questions and Likert-type scales) [Brajnik et al. 1996].

5.3 Features Related to the Search Task

The features related to search tasks consist in identifying what kind of search can be performed by the user.

Task type. Different types of search tasks can be associated to users' information needs. Depending on the task, IRIs should not return necessarily the same result. For example, based on the list of expectations mentioned by Rosenfeld and Morville in 1998 [Rosenfeld and Morville 1998], in information retrieval, the following tasks have been identified:

- **Known-item searching**: In this task, the user knows of a particular document. The goal is not to exhaustively find documents about a topic, but to find a single correct document [Ogilvie and Callan 2003].
- **Existence searching**: In this task, the user knows of a particular topic. The goal is not to exhaustively find documents about the topic, but to find at least one document dealing with the exact theme.
- **Exploratory searching**: The goal of this task is not to exhaustively find documents about a given topic. The user is essentially poking around the topic.
- **Comprehensive searching**: The goal of this task is to find exhaustively documents about a given topic.

As aforementioned, the set of features presented above is not restrictive and many others can be added. For instance, additional features characterizing users may be physical characteristics such as age, gender, vision limitations or cultural characteristics such as reading level or known languages [McCracken et al. 2003], or information seeking strategies [Belkin 1996]. Regarding search task, one can add

features like available time to do the task… Once the IR scenarios defined by the selected features, the next step is to involve them in an evaluation framework.

6 Evaluation Framework

The proposed evaluation framework includes a unique search engine on which the search task will be processed. Indeed, every interface has to be independent from the IRS (i.e. from data) to avoid the bias of evaluating an IRI and an IRS as a whole. In order to compare two IRIs, evaluation process must use the same document collection [Hearst 2009, p.61]. In our approach, we respect this recommendation and go further in using a unique IRS to evaluate every IRIs on the same baseline as suggested in [Julien et al. 2008]. The proposed framework follows guidelines similar to those presented in [Hearst 2009, chap. 2].

Such an evaluation framework must be based on different parts:

- *Evaluation criteria*. These evaluation criteria enable measuring the capacity of an IRI to perform a given test. (section 6.1)
- *Test data*. The various test data describes IR scenarios that involve an IRS result, a task description and a user expected for interacting with an evaluated IRI. (section 6.2)
- *Evaluation results*. These results correspond to a set of tests. Every test gathers a value for every evaluation criteria using a specific test data (scenario). (section 6.3)
- *Evaluation result analyses*. It relies on methods to analyze evaluation results. (section 6.4)

The following sections detail these different parts.

6.1 Evaluation criteria

To measure the suitability of an IRI for a context, we need some criteria related to the success of the scenario. In our proposal we use only, as a first attempt, a single criterion that corresponds to the success or the failure of the search (task achievement).

Task achievement. This criterion gives information on the task fulfillment. The interest of this criterion is for example to determine the combinations of (system, user, task) for which a given IRI is suitable for and those for which it is not appropriate. For this criterion, the user can judge whether he succeeded or not in completing the search task. This achievement can be estimated in a binary manner (success, failure), according to a Likert scale [Likert, 1932] (note from 1 to 5 for example), or even linguistic values [Zadeh, 1975].

Related to task achievement, a complementary criterion could measure time required to achieve the task. This criterion might consist in measuring how long the user uses the IRI to carry out a task.

6.2 Evaluation test data

To be compared according to the same baseline, IRIs have to use the same search result sets i.e. documents retrieved by the retrieval system must be the same for all the interfaces. Indeed, this principle guarantees coherence of results between interfaces, in order to make sure that each interface is evaluated the same way. This rule does not exclude that each interface manage differently the documents (i.e. execute additional indexation). However, to allow such management, the retrieval system shall provide enough information on each document (keywords, raw content, links…).

Moreover a specific attention must also be paid to the variety of search result sets used during the evaluation process. In order to make varying scenarios, search result sets must cover all the possible combinations of features related to the system (low or high number of documents, low precision or high precision…). Building such result sets is a time consuming task. To limit these drawbacks, search result sets may be automatically built from existing ones like TREC or CLEF collections for instance. In this case, the first step consists in selecting or adding *topics* in order to build result sets for different tasks (different query types). The second step consists in adapting search result contents (based on *qrels*[12]) in order to respect all the possible combinations of system features (removing, adding, changing the position of documents in the result set…). To do this, we implemented a specific application to generate any search result sets corresponding to any combination of system feature values.

Furthermore, evaluators have to be carefully chosen to construct a representative panel of users. This panel must cover the different combinations of user features (domain and practical knowledge). To cover in the most effective way the different possibilities, the system can automatically select a specific test data to be used in the evaluation.

6.3 Evaluation Results

The previous guidelines allow performing various simulations of IRI utilization. This makes it possible to sum up the evaluation results into a summarized table (a sample is given in Table 1). In this table we simplified the name of every criterion as follows:

[12] *qrels* is a list of documents judged relevant for a given topic

Short name	***Corresponding scenario feature***
NbDocs	Number of Retrieved Documents
ContentH	Content Homogeneity of Retrieved Documents
Prec	Precision of Retrieved Documents
NPrec	TopN Precision
Rank SD	Standard Deviation of Relevant Document rsv
Domain	Domain Knowledge Level *In following examples values can be "neo" (neophyte), "avg" (average) or "exp" (expert)*
Practical	Practical Knowledge Level *In following examples values can be "neo" (neophyte), "avg" (average) or "exp" (expert)*
Type	Task type *In following examples values can be "known" (Known-item searching), "exist" (Existence searching), "explo" (Exploratory searching), "comp" (Comprehensive searching)*
Short name	***Corresponding evaluation criterion***
Result	Task achievement *In following examples values can be "Yes" (success) or "No" (failure).*

Eval. ID	**IR Scenario**								**Evaluation** Result
	System (documents retrieved by the IRS)					**User** (Knowledge level)		**Task** (Goal)	
	NbDocs	***ContentH***	***Prec***	***Nprec***	***Rank SD***	***Domain***	***Practical***	***Type***	
1	24	0.8	0.75	0.9	2.3	neo	neo	known	**Yes**
2	250	0.3	0.6	0.2	4.0	neo	neo	known	**No**
…	…	…	…	…	…	…	…	…	…

Table 1: Samples of evaluation results

The process used to carry out scenarios and obtaining evaluation results is detailed in the implementation section (section 7).

6.4. Evaluation Result Analyses

Analyses on collected evaluation results can be conducted at two levels: IRI individual analysis aims at mapping the suitability of a given IRI for various contexts (i.e. IR scenarios) and meta-analysis (based on IRI individual analyses) intends to identify and rank IRIs suitable for a given context.

6.4.1 IRI individual analysis

To precisely make possible the interpretation of the suitability of an IRI for various contexts (scenarios) we propose to analyze the table of results (Table 1); for instance in order to obtain a specific decision tree. A decision tree aims at identifying scenarios in which the evaluated IRI is effective through a learning process. To build such decision trees, any algorithm can be used (i.e. C4.5 [Quinlan 1996]). Figure 2 illustrates an example of a decision tree resulting from an evaluation of a ranked list

visualization of search results like the one provided by Google. To facilitate the interpretation of such a tree, real numeric values (Table 1, e.g. *Nprec*=0.9, 0.2…) have been discretized (Table 2, e.g. *Nprec*: 0.9→high, 0.2→low…) for every system feature. Note that discretization may be processed differently for each feature, according to the upper and lower bounds for instance. For example, considering a range from 0.0 to 1.0 for ContentH discretization may lead to the following Likert scale: low ([0.0;0.33]), average (]0.33;0.66]), and high (]0.66;1.0]). Since raw values (Table 1) are stored it is possible on-demand to obtain another Likert scales in order to improve the interpretation of the suitability of an IRI.

Eval. ID	IR Scenario								Evaluation Result
	System (documents retrieved by the IRS)					User (Knowledge level)		Task (Goal)	
	NbDocs	***ContentH***	***Prec***	***Nprec***	***Rank SD***	***Domain***	***Practical***	***Type***	
1	<30	high	high	high	medium	neo	neo	known	**Yes**
2	>=30	low	average	low	high	neo	neo	known	**No**
…	…	…	…	…	…	…	…	…	…

Table 2: Discretized system feature values

In this figure, we observe that the ranked list based IRI is effective, for instance, when the number of relevant documents is high in the search result and the task is a known-item task.

IRI#1 “ranked list” suitability analysis

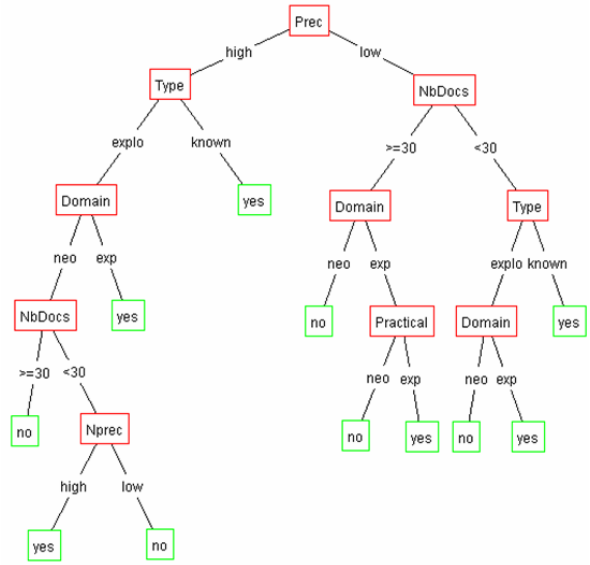


Figure 2: Example of decision tree resulting from many evaluations of a ranked list based IRI (i.e. display like Google does)

6.4.2 Meta-analysis: Ranking IRIs for a given context

The previous process aiming at generating the IRI decision tree allows IRI designers to understand the suitability of their IRI for different contexts (scenarios). To go further in the analysis we also propose to rank IRIs for a given scenario with respect to their suitability. To do this all the rules leading to a successful task achievement are extracted from every decision tree associated to IRIs. These rules are then sorted according to their confidence and support values. From these values, IRIs are then ranked to indicate the most suitable ones for a given context. More details related to the implementation of these functionalities are given in the following.

7 Framework Implementation

The implementation of the proposed framework is based on a software tool that supports this evaluation framework and ensures easier and reproducible experiments (using shared test data). Figure 3 shows a situation in which two evaluation processes are done for two different interfaces (client-side). Every evaluation process uses an instance of the same virtual search engine. This figure also shows the different parts of the implemented centralized software (server-side).

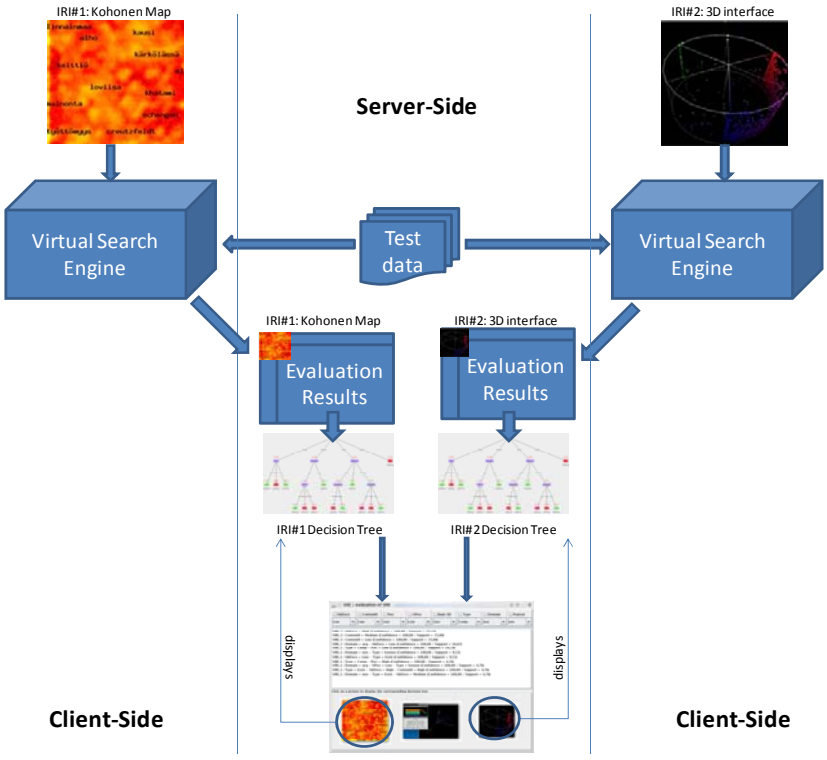


Figure 3: General overview of the evaluation framework

To achieve its implementation, our proposal has been developed using the Java language. In this section, we only describe the main functionalities of this prototype (i.e. evaluation process) that are illustrated by the following scenario (Figure 4).

The evaluation framework is so based on a “virtual” search engine (VSE) used by every evaluated IRI. In order to enable the communication between the evaluated IRI and the evaluation framework, the VSE provides a set of services presented in the appendix A.

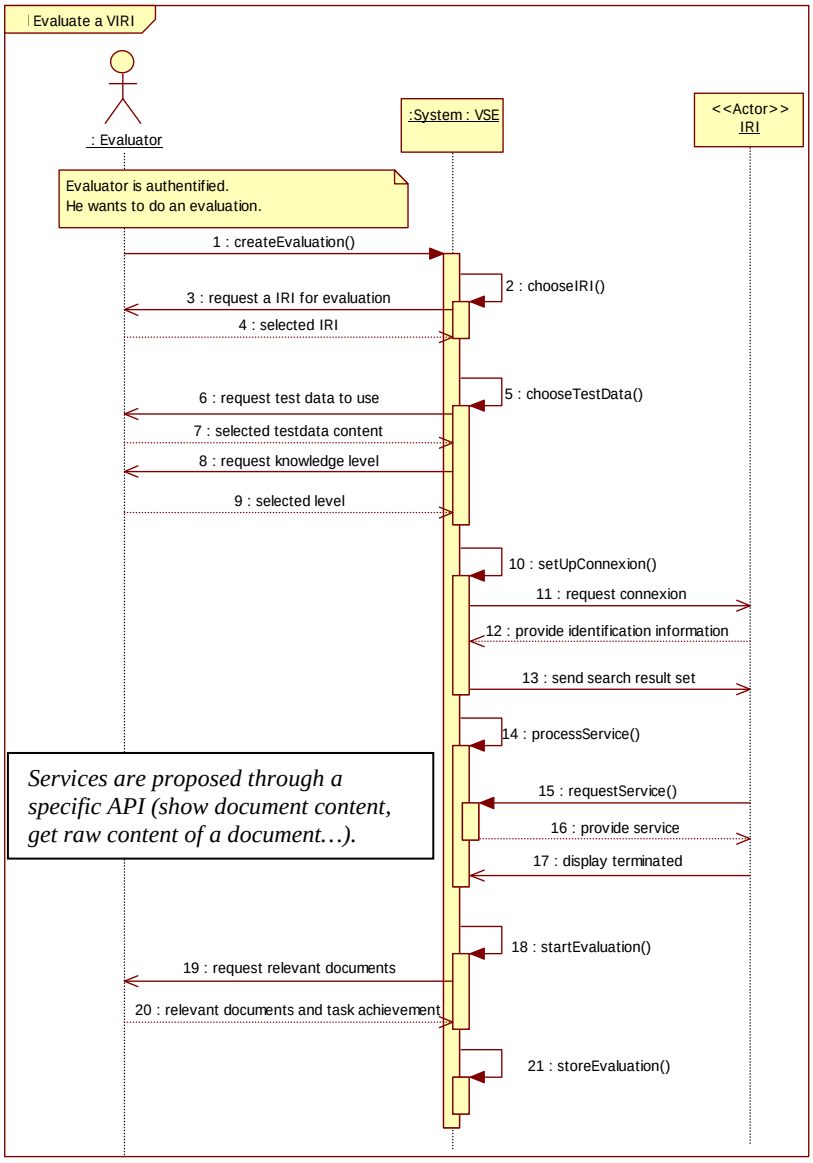


Figure 4: IRI evaluation scenario

Figure 5 describes the UML class diagram corresponding to the VSE. For comprehension, this diagram is split into three parts:

- Part A represents data about users, IRIs and the related evaluation results,
- Part B describes data about criteria implied in evaluation process,

Part C represents test data used for evaluation. Note that we associate several test data to each single query in order to evaluate a same query (information need) in various scenarios since each test data has a specific combination of system feature values.

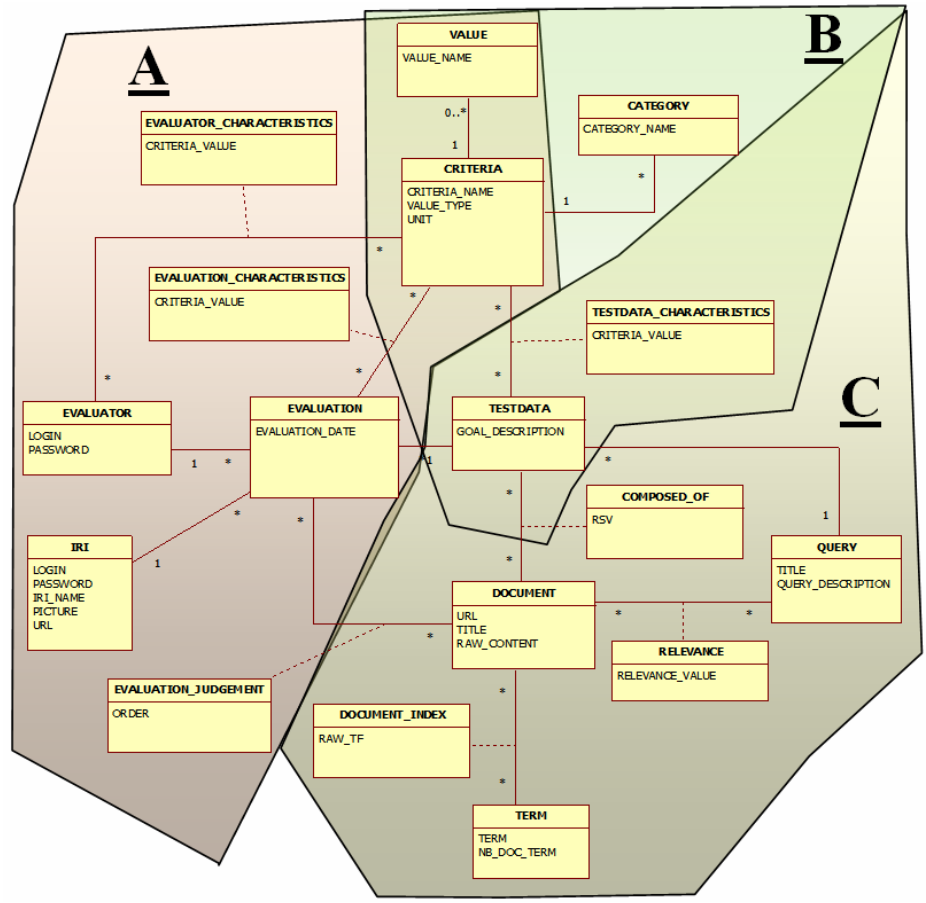


Figure 5: UML class diagram of the VSE

Through the main window of the VSE users can register as a new evaluator, register a new IRI to be evaluated, or sign in indicating the role they want to play (evaluator, IRI developer or analyst). An analyst can access all the evaluation results (Figure 12) whereas an IRI developer can only access his own IRI evaluation results (Figures 10 and 11).

As an example, we describe a real scenario corresponding to an evaluation of a ranked list IRI for a known-item search task by a user having average domain knowledge and being neophyte with regards to practical knowledge. The result of this evaluation adds a new row in the table of this IRI evaluation results as illustrated in figure 10 (see the highlighted row).
When logged in, the evaluator must choose the evaluation (i.e. the query) he wants to perform (Figure 6). When chosen, he selects his domain knowledge level related to the evaluation.

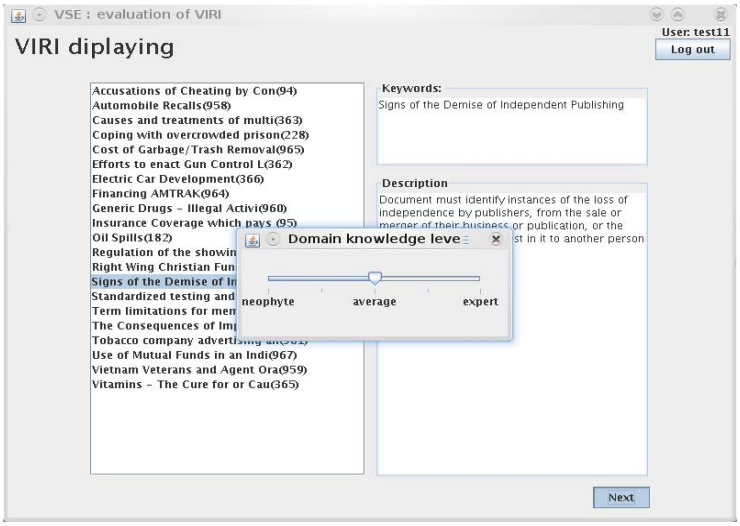


Figure 6: Window for query selection

Then, the system displays the goal of the evaluation (i.e. the query) to the user (Figure 7). It explains the expected search results related to the task (e.g. known-item search).

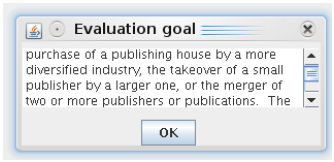


Figure 7: Evaluation goal explanation

Then, while exploring results through the evaluated IRI (e.g. the ranked list shown in the left window of Figure 8), the user can access document contents in a specific window that display the content of the selected document this is done by the IRI that calls a service of the VSE (right window of Figure 8).

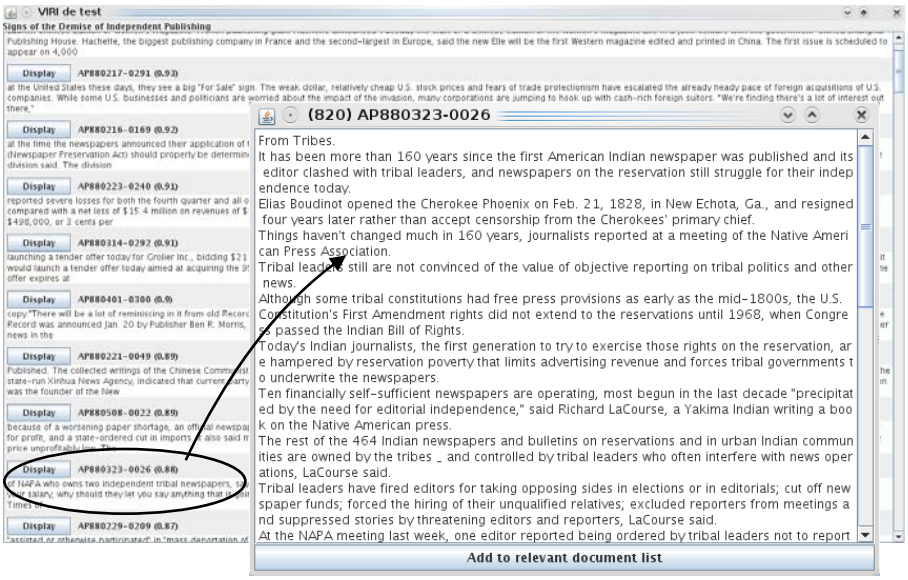


Figure 8: A ranked list based IRI (left window) and a window resulting from a VSE service displaying document content (right window)

Moreover, the evaluator must select documents he considers as relevant to his assigned task. In order to do this, he either clicks on the button named “Add to relevant documents” in the window displaying the content of a given document or selects documents through the evaluation related window opened by the VSE (Figure 9). When selecting a relevant document, the user has to include it in a list ordered by relevance (right side of Figure 9). For the moment we only considered binary relevance for documents selected by users. The reason is twofold: we intended to not overwhelm users with cumbersome assessments and we had IR collections with only binary judgements to generate our test data. Gradual relevance might be used on the condition to have gradual judgements of documents for each proposed query in the *qrels* used by the framework (cf. section 6.2).

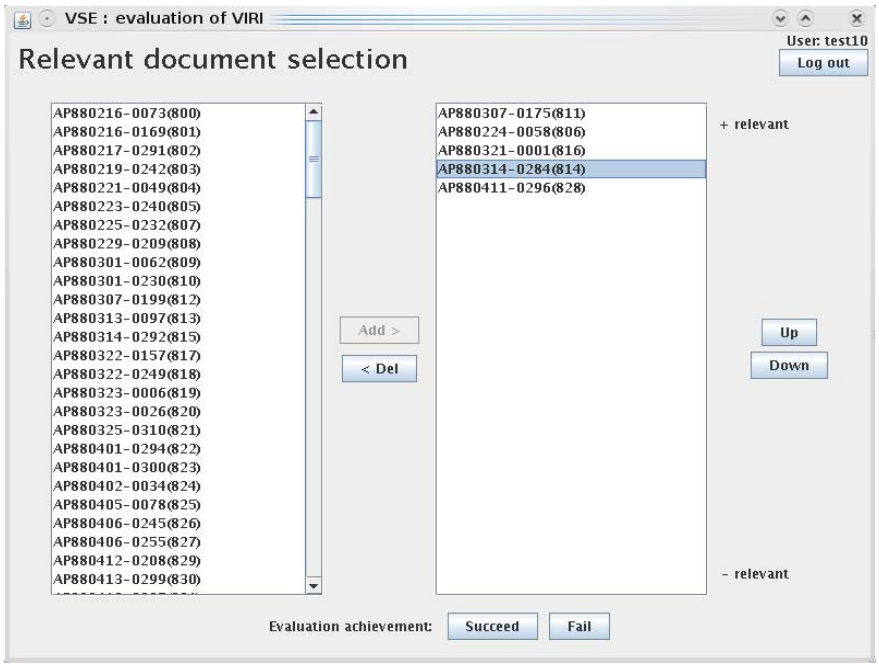
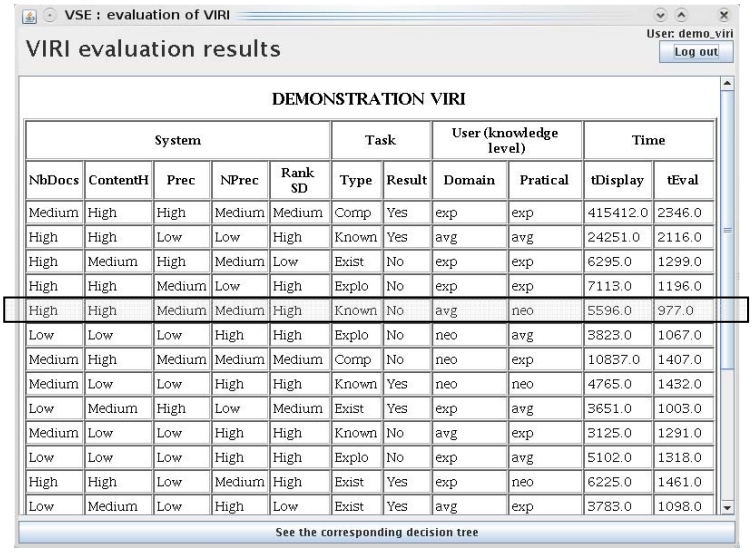


Figure 9: The main evaluation window of the VSE

Finally, in order to complete his evaluation, the user must indicate to the VSE if he considers his task achieved or not (bottom of Figure 9). Once completed, the evaluation results are stored in the database.

Since evaluation results are stored related to each IRI, every IRI developer can access to the results about his IRI (Figure 10). In the result tables, we have added the time used to display the IRI (*tDisplay*) and the time needed to carry out the scenario (*tEval*). In addition to this result display, the IRI developer can access the corresponding decision tree (Figure 11) based on the achievement values. In our implementation, decision trees are generated with the GINNet[13] tool.

[13] http://ginnet.gforge.inria.fr



System					Task		User (knowledge level)		Time	
NbDocs	ContentH	Prec	NPrec	Rank SD	Type	Result	Domain	Pratical	tDisplay	tEval
Medium	High	High	Medium	Medium	Comp	Yes	exp	exp	415412.0	2346.0
High	High	Low	Low	High	Known	Yes	avg	avg	24251.0	2116.0
High	Medium	High	Medium	Low	Exist	No	exp	exp	6295.0	1299.0
High	High	Medium	Low	High	Explo	No	exp	exp	7113.0	1196.0
High	High	Medium	Medium	High	Known	No	avg	neo	5596.0	977.0
Low	Low	Low	High	High	Explo	No	neo	avg	3823.0	1067.0
Medium	High	Medium	Medium	Medium	Comp	No	neo	exp	10837.0	1407.0
Medium	Low	Low	High	High	Known	Yes	neo	neo	4765.0	1432.0
Low	Medium	High	Low	Medium	Exist	Yes	exp	avg	3651.0	1003.0
Medium	Low	Low	High	High	Known	No	avg	exp	3125.0	1291.0
Low	Low	Low	High	High	Explo	No	exp	avg	5102.0	1318.0
High	High	Low	Medium	High	Exist	Yes	exp	neo	6225.0	1461.0
Low	Medium	Low	High	Low	Exist	Yes	avg	exp	3783.0	1098.0

Figure 10: Table of evaluation results related to an IRI

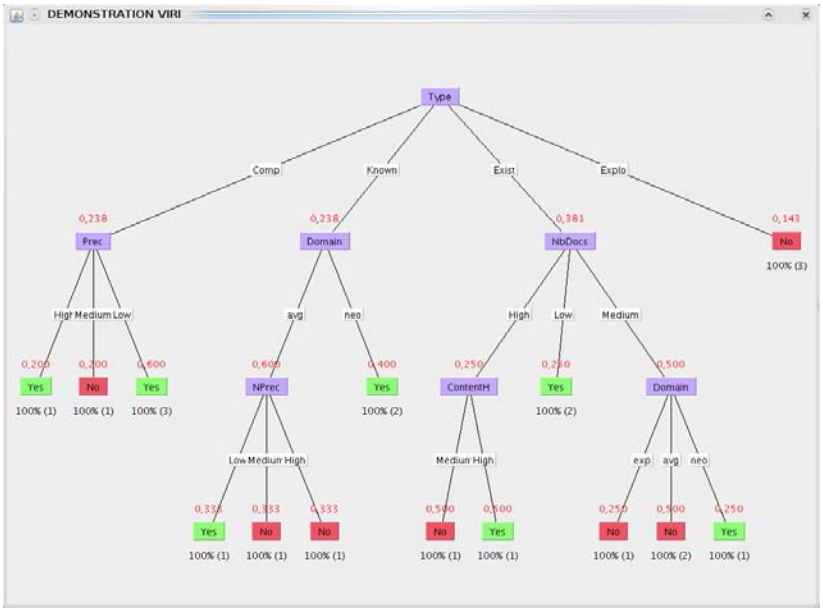


Figure 11: Decision tree displaying the evaluation results related to an IRI

These decision trees (Figure 11) characterize the behavior of a single evaluated IRI. To aggregate many evaluation results in order to identify which interfaces are suitable for a specific scenario, all the rules leading to a successful task achievement

are extracted from the decision trees. Next, all these rules are displayed in a specific window (Figure 12) (rules and IRI screenshots are only given as example).

This window is composed of three specific parts:

- The top of the window allows users to build a specific scenario corresponding to their needs. The selected checkboxes (each one corresponding to one feature) define the specific scenario and are used to filter the whole rules,
- The middle part of the window displays available rules (those corresponding to the scenario). Rules (upper part of the interface) and so interface thumbnails (lower part of the interface) are ordered according to decreasing rule confidence and support,
- The bottom part of the window displays thumbnails of evaluated interfaces for which the rules are successful for the specified scenario. When clicking on one thumbnail, users can have access to the decision tree representing the selected interface behavior.

Thus, any user (e.g. IRS developer or IRI designer) can define interactively the scenario for which he expects to find effective IRIs. To define such scenario he selects and ranks the different feature values using the widget located at the top of the window. Through this system, users (e.g. IRS developers or IRI designers) can choose among evaluated interfaces those that correspond to their application needs.

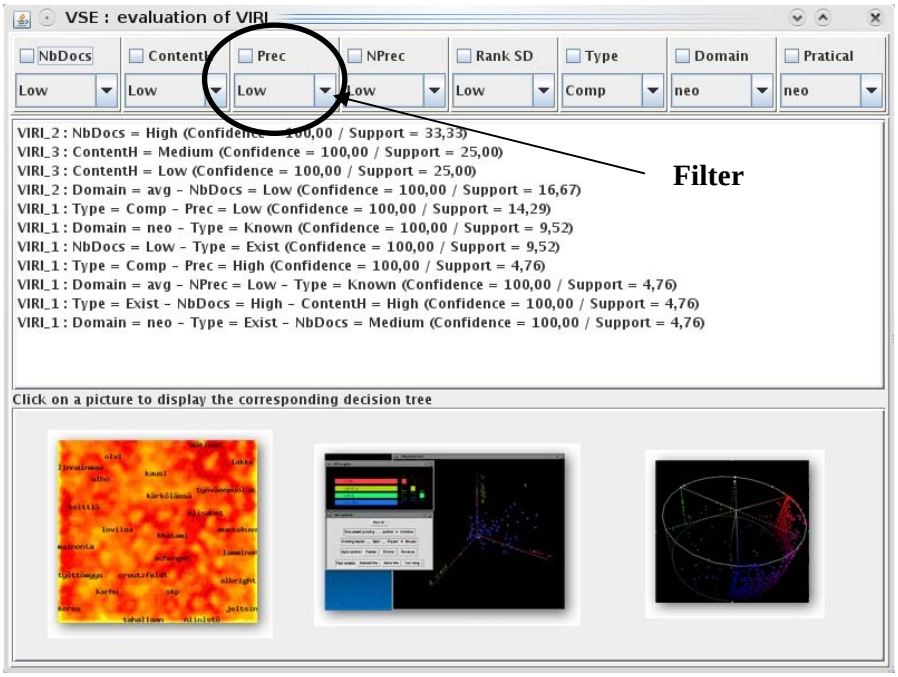


Figure 12: Specific navigation tool to identify IRIs suitable for a given IR scenario

8 Conclusion

Information Retrieval Interface is a critical component of Information Retrieval Systems. Indeed, with such visualization interface, users can manage search results in a more effective way. Since numerous IRIs are available it is difficult to choose those that are the most suitable for a specific purpose.

This paper deals with IRI evaluation which is a delicate and difficult task. Evaluation methods of interfaces exist but focus on the study of usability. In IR domain, evaluation is usually performed with regards to system effectiveness without considering IRI as a component which should be evaluated independently to measure its impact on IR process. Such evaluations do not allow one to evaluate if a specific IRI is really suitable for a specific IR scenario (a specific IR task done by a specific user thanks to a specific system). Our proposal is a complementary approach to usability evaluation methods. It aims at evaluating the suitability of an interface for various possible IR scenarios. Up to now, no real investigation in this direction has been done. We define in this paper a framework for evaluating the suitability of an IRI for IR scenarios. We base our framework on characterizing IR scenarios, defining evaluation criterion, and methods to collect and analyze evaluation results. This framework aims at providing a wide range of IR scenarios to obtain a global evaluation of an IRI and preserving the same and replicable way to evaluate their own IRIs. It relies on a unique virtual IRS whose behavior can be controlled in order to vary performances.

We implemented this framework through a fully functional platform for textual information retrieval context. This platform uses common features of information retrieval and a binary evaluation criterion (task achievement). This platform is also flexible since any feature and criterion can be easily added in the evaluation framework. It is an OS independent application based on Java Web Start technology[14] (i.e. no installation is required and execution can be done via the web using the JNLP protocol).

We are now waiting for IRI developers who wish to evaluate their IRI with the proposed Java API. This API should be developed, in the future, in many other programming languages.

9 Future work

We identify many perspectives for this work. First of all, the evaluation we propose can be combined with other evaluation frameworks. Since our approach may be considered as a static evaluation (through IR scenarios) we can combine it with a dynamic evaluation. This dynamic evaluation may, for instance, rely on usage criteria, user action history, and user preferences. Thanks to this combination, users could select the most appropriate interface according to their preferred evaluation aspects (cognitive, static, dynamic…).

[14] http://java.sun.com/javase/technologies/desktop/javawebstart/index.jsp

Moreover, on one hand we plan to integrate a new category of features related to "queries". Indeed, features related to query ambiguity prediction like Jensen-Shannon divergence [Carmel et al. 2006] or coherence [He et al. 2008] could be used to improve search achievement analysis. On the other hand, evaluation criteria could be added, for instance, on qualitative aspects of the IR process related to feelings of evaluators when processing tasks. All the evaluation criteria (Task achievement and new criteria) may also be characterized via linguistic labels that could be then processed [Martínez, 2007] and used in the decision-tree construction.

Then, our framework may be adapted to any other information retrieval application like catalogs, desktop, domain specific IR (i.e. Entrez-PubMed[15])… This adaptation implies the identification of features characterizing the different evaluation scenarios related to the different users' tasks involved in such applications.

In a more general way, we are convinced that our approach complements the existing usability evaluations. Thus, the results obtained with our framework and with usability tests could lead to:

- Measuring correlation between usability and suitability,
- Making IRS visualization of search results adaptive. An IRS could so automatically suggest the most usable and suitable IRI(s) for a specific context.

References

Badre, A. N., Hudson, S. E., & Santos P. J. (1993) An Environment to Support User Interface Evaluation Using Synchronized Video and Event Trace Recording, Technical Report, GIT-GVU-93-16, Graphics, Visualization and Usability Center, Georgia Institute of Technology, USA. http://hdl.handle.net/1853/3626

Belkin, N. J. (1996). Intelligent information retrieval: Whose intelligence? *Fifth International Symposium for Information Science (ISI-96)* (pp. 25-31).

Berenci, E., Carpineto, C., & Giannini, V. (1998). Improving the effectiveness of web search engines using selectable views of retrieval results. Journal of Universal Computer Science, vol. 4, no. 9 (1998), (pp. 737-747). DOI 10.3217/jucs-004-09-0737.

Bonnel, N., Cotarmanac'h, A., & Morin, A. (2005). Meaning Metaphor for Visualizing Search Results. *9th International Conference on Information Visualisation (IV'05)* (pp. 467-472), IEEE Computer Society.

Bonnel, N., Lemaire, V., Cotarmanac'h, A., & Morin, A. (2006). Effective Organization and Visualization of Web Search Results. *IASTED International Conference on Internet and Multimedia Systems and Applications (EuroIMSA 2006)* (pp. 209-216), Acta Press.

Brajnik, G., Mizzaro, S. & Tasso, C. (1996). Evaluating user interfaces to information retrieval systems: a case study on user support. *19th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '96)* (pp. 128-136), ACM, New York, NY, USA.

Brown, C. M. (1988). Human-Computer Interface Design Guidelines. Ablex Publishing Corp.

[15] http://www.ncbi.nlm.nih.gov/pubmed/

Canas, J.J. (2008). Cognitive Ergonomics in Interface Development. Evaluation. Journal of Universal Computer Science, vol. 14, no. 16, (pp. 2630-2649). DOI 10.3217/jucs-014-16-2630.

Carmel, D., Yom-Tov, E., Darlow, A., & Pelleg, D. (2006). What makes a query difficult?. In *Proceedings of the 29th Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (Seattle, Washington, USA, August 06 - 11, 2006). SIGIR'06. ACM, New York, NY (pp. 390-397). DOI = http://doi.acm.org/10.1145/1148170.1148238

Chevalier, M., Chrisment, C., & Julien, C. (2004). Helping People Searching the Web: Towards an Adaptive and a Social System. *Proceedings of the IADIS International Conference WWW/Internet 2004 (Madrid, Spain)*, Volume 1, (pp. 405-412).

Chin, J. P., Diehl, V. A., & Norman, K. L. (1988). Development of an instrument measuring user satisfaction of the human-computer interface. *Proceedings of SIGCHI '88*, ACM/SIGCHI, New York (pp. 213-218).

Cugini, J., Laskowski, S., & Sebrechts, M. (2000). Design of 3D Visualization of Search Results: Evolution and Evaluation. *IST/SPIE's 12th Annual International Symposium: Electronic Imaging 2000: Visual Data Exploration and Analysis VII (SPIE 2000)* (pp. 198-210).

Dumas J., & Janice R. (1999). *A Practical Guide to Usability Testing*, revised edition. Bristol, UK: Intellect Books.

Fekete, J.D., & Plaisant, C. (2004). Les leçons tirées de deux compétitions de visualisation d'information. *16th Conference on Association Francophone d'interaction Homme-Machine (IHM 2004)* (pp. 7-12). ACM Press.

Grice, R. A. (2003). Comparison of cost and effectiveness of several different usability evaluation methods: a classroom case study. Professional Communication Conference, 2003. IPCC 2003. ISBN 0-7803-7949-7, DOI 10.1109/IPCC.2003.1245482 , pp. 140-144.

Hartson, H. R., Andre, T. S. & Williges, R. C. (2003). Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer Interaction*, 15(1), pp. 145-181.

He, J., Larson, M., & de Rijke, M. (2008). Using coherence-based measures to predict query difficulty. Advances in information retrieval: 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008: Proceedings (pp. 689-694).

Hearst, M.A. (2009). Search User Interfaces, Cambridge University Press, September, ISBN 978-0-521-11379-3.

Hearst, M.A., & Karadi, C. (1997). Cat-a-Cone: an interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. *20th International Conference on Research and Development in Information Retrieval (ACM SIGIR'97)* (pp. 246-255). Philadelphia: ACM.

Hölscher, C. & Strube, G. (2000). Web search behavior of internet experts and newbies. *9th International Conference on the World Wide Web: the international journal of computer and telecommunications networking* (pp. 337-346).

Houston, B. & Jacobson, Z. (2000). A simple 3D Visual Text Retrieval Interface. *RTO IST Workshop on "Multimedia visualization of massive military datasets"*. published in TRO-MP-050 (2002) (pp. 24.1-24.5)

Julien, A. C., Leide, E. J., & Bouthillier, F. (2008). Controlled User Evaluations of Information Visualization Interfaces for Text Retrieval: Literature Review and Meta-Analysis, 59(6): pp. 1012-1024.

Koshman, S., Spink, A., Jansen, B. J., Blakely, C., & Weber, J. (2006). Metasearch result visualization: an exploratory study. *Canadian Association for Information Science Conference*, York University, Toronto, Ontario.

Koutsabasis, P., Spyrou, T. & Darzentas, J. (2007). Evaluating usability evaluation methods: criteria, method and a case study. *Human-Computer Interaction. Interaction Design And Usability*, Lecture Notes in Computer Science, 2007, Volume 4550/2007, DOI: 10.1007/978-3-540-73105-4_63, pp. 569-578.

Kuhn, K. (2000). Problems and Benefits of Requirements Gathering With Focus Groups: A Case Study. *International Journal of Human-Computer Interaction*, 12(3&4), pp. 309-325.

Lagus, K., Kashi, S., Honkela, T. & Kohonen, T. (1996). Browsing Digital Libraries with the Aid of Self-Organizing Map. In *5th International World Wide Web Conference* (pp. 71-79).

Lainé-Cruzel, S., (1999). ProfilDoc – Filtrer une information exploitable. *Bulletin des bibliothèques de France (BBF)*, 44(5), pp. 60-64.

Lewis, C., & Mack R., (1982). Learning to Use a Text Processing System: Evidence from "Thinking Aloud" Protocols. *Proceedings of Human Factors in Computer Systems* (p. 387-392).

Likert, R. (1932). A technique for the measurement of attitudes, Archives of Psychology, 22(140), 1-55.

Manning, C.D., Raghavan, P. & Schütze, H., 2008. *Introduction to Information Retrieval*, Cambridge University Press., ISBN 0521865719, 2008.

Martínez, L. (2007). Sensory evaluation based on linguistic decision analysis. *International Journal of Approximate Reasoning*. Volume 44, Issue 2, February 2007, Fuzzy Decision-Making Applications, pp. 148-164.

McCracken, D.D., Spool, J. M., & Wolfe, R. J. (2003). *User-Centered Web Site Development: a Human-Computer Interaction Approach*. Pearson Education.

Molich, R., & Nielsen, J. (1990). Improving a Human-Computer Dialogue. *Communication of the ACM*, 33(3), 338-348.

Nielsen, J., & Mack, R.L. (1994). Usability Inspection Methods. New York, NY, USA: John Wiley & Sons.

Nielsen, J., & Molich, R. (1990). Heuristic Evaluation of User Interfaces. *SIGCHI Conference on Human Factors in Computing Systems: Empowering People (CHI'90)* (pp. 249-256). New York, NY, USA: ACM Press.

Ogilvie, P., & Callan, J. (2003). Combining Document Representations for Known-Item Search. *26th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'03)* (pp. 143-150). New York, NY, USA: ACM Press.

Plaisant, C., Fekete, J.D., & Grinstein, G. (2008). Promoting Insight Based Evaluation of Visualizations: From Contest to Benchmark Repository. IEEE Transactions on Visualization and Computer Graphics, 14(1), (pp. 120-134).

Quinlan, J. R. (1996). Improved Use of Continuous Attributes in C4.5. Journal of Artificial Intelligence Research 4 (1996), (pp. 77-90).

Robertson, G., Mackinlay, J. & Card, S. (1991). Cone Trees: Animated 3D Visualizations of Hierarchical Information. *SIGCHI Conference on Human Factors in Computing Systems: Reaching Through Technology (CHI'91)* (pp. 189-194). ACM Press, New York.

Rohrer, M., & Ebert, D.S. (1999). A shape-based Visual Interface for Text Retrieval. *IEEE Computer Graphics and Applications*, 19(5), (pp. 40-46).

Rosenfeld, L., & Morville, P. (1998). Information Architecture for the World Wide Web. O'Reilly & Associates, Inc.

Shneiderman, B. (1998). *Designing the User Interface.* 3rd edition. Addison-Wesley.

Shneiderman, B., & Plaisant, C. (2005). *Designing the User Interface*. 4th edition. Addison-Wesley.

Smith, S. L. & Mosier, J. N. (1986). Design Guidelines for Designing User Interface Software. Technical Report ESD-TR-86-278. Bedford, MA: The MITRE Corporation.

Spoerri, A. (2006). Visualizing Meta Search Results: Evaluating the MetaCrystal toolset. *69th Annual Meeting of the American Society for Information Science and Technology (ASIST 2006)*, Austin, TX.

Thomas, J. & Cook, K. (2005). Illuminating the Path: The Research and Development Agenda for Visual Analytics, IEEE CS Press (2005), http://nvac.pnl.gov/agenda.stm.

Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994). The Cognitive Walkthrough Method: A Practitioner's Guide. New York, NY, USA: John & Wiley.

Zadeh, L.A. (1975). The concept of linguistic variable and its application to approximate reasoning - I. Information Sciences, 8(3), pp. 199-249.

Zamir, O., & Etzioni, O. (1999). Grouper: A Dynamic Clustering Interface to Web Search Results exploitable. *Computer networks*, 31(11-16), pp. 1361-1374.

Appendix A – The minimal VSE API

In order to provide VSE services to developers who want to evaluate IRIs, we propose a minimal API:

VSE_API class	**VSE_API(login, password)** *Connects the IRI to the VSE and blocks the API until it receives the document list corresponding to the selected test data.*
	endDisplay() *Indicates that the IRI has displayed the document result list to the user. It initiates the evaluation process.*
	getDocList(): List<Document> *Returns the document list to the IRI.*
	getQuery(): String *Returns the query terms related to the selected test data.*
Document class	**display()** *Displays the document content through the VSE (Figure 8, right window).*
	getContent(): String *Returns the full document content to the IRI.*
	getContentWithoutMarkup(): String *Returns the raw document content without any tags (e.g. HTML tags) to the IRI.*
	getTermList(): List<Term> *Returns the document indexed terms to the IRI.*
	getRSV():Float *Returns the retrieval status value (the relevance value) of the document to the IRI.*
	getTitle(): String *Returns the document title to the IRI.*
Term class	**getDocCount(): Integer** *Returns the number of documents containing this term to the IRI.*
	getRawTF(): String *Returns the raw term frequency in a given document (see the Document class – getTermList()) to the IRI.*
	getTerm(): String *Returns the string corresponding to this term to the IRI.*