Second International

Workshop on Adaptive Information Retrieval (AIR)

London, United Kingdom

18th October 2008

Proceedings

Second International Workshop on Adaptive Information Retrieval (AIR)

London, United Kingdom

18th October 2008

Editors: H. Joho, F. Hopfgartner, J. M. Jose, C. J. van Rijsbergen

Booklet Editor: F. Hopfgartner

Sponsors:

Semedia project (FP6-041273)

Salero project (FP6-027122)

K-Space project (FP6-027026)

MIAUCE project (FP6-033715)

General Chairs
Joemon M. Jose, University of Glasgow, UK
C. J. van Rijsbergen, University of Glasgow, UK

Programme Chair
Hideo Joho, University of Glasgow, UK

Publicity Chair
Frank Hopfgartner, University of Glasgow, UK

Local Organising Committee
Ioannis Arapakis, University of Glasgow, UK
Leif Azzopardi, University of Glasgow, UK
Desmond Elliott, University of Glasgow, UK
Martin Halvey, University of Glasgow, UK
Yashar Moshfeghi, University of Glasgow, UK
Thierry Urruty, University of Glasgow, UK
Robert Villa, University of Glasgow, UK

Table of Contents

Programme **iv**

Invited Keynotes **1**

D. Harper "Adaptive Search on a Web Scale" 3

N. Fuhr "A Probability Ranking Principle for Interactive IR" . . . 4

Oral Presentations **5**

R. Eckstein and A. Henrich "Reaching the Boundaries of Context-Aware IR: Accepting Help from the User" 7

M. Springmann and H. Schuldt "Using Regions of Interest for Adaptive Image Retrieval" 13

C.-P. Klas, S. Kriewel and M. Hemmje "An Experimental System for Adaptive Services in Information Retrieval" 21

Poster Presentations **25**

P. Bermejo, F. Hopfgartner, and J. M. Jose "Automatically Creating and Comparing Features and Contexts" 27

R. Kanawati "A CBR Framework for Implementing Community-Aware Web Search Engine" 32

M. Palomino, M. P. Oakes, and Y. Xu "An Adaptive Method to Associate Pictures with Indexing Terms" 38

P. Ferguson, H. Lee, C. Gurrin, S. Sav, T. Foures, S. Lacote, A. F. Smeaton, and N. O'Connor "Searching without Text in an Interactive TV Environment" 44

L. Sitbon and P. Bellot "A readability measure for an Information Retrieval Process Adapted To Dyslexics" 48

C. Cullen, B. Vaughan, S. Kousidis, and J. McAuley "Metadata Visualisation Techniques for Emotional Speech Corpora" . . . 52

R. Kaptein and J. Kamps "Improving Information Access by Relevance and Topical Feedback" 58

N. Fuhr, M. Jordan and I. Frommholz "Combining Cognitive and System-Oriented Approaches for Designing IR User Interfaces" 64

A. Diriye, A. Blandford and A. Tombros "Information Search Stage Based Representations" 69

L. Freund "Putting Digital Government in Context" 75

Programme

09:00 – 10:00	*Invited Talk*	
	Prof. David Harper (Google Switzerland)	*Adaptive Search on a Web Scale*
10:00 – 10:30	*Coffee Break*	
10:30 – 12:30	*Oral Presentations*	
	R. Eckstein, and A. Henrich (University of Bamberg, Germany)	*Reaching the Boundaries of Context-Aware IR: Accepting Help from the User*
	M. Springmann and H. Schuldt (University of Basel, Switzerland)	*Using Regions of Interest for Adaptive Image Retrieval*
	C.-P. Klas, S. Kriewel, and M. Hemmje (University of Hagen and University of Duisburg-Essen, Germany)	*An Experimental System for Adaptive Services in Information Retrieval*
12:30 – 14:00	*Lunch Break and Poster Session*	
14:00 – 15:00	*Invited Talk*	
	Prof. Norbert Fuhr (University of Duisburg-Essen, Germany)	*A Probability Ranking Principle for Interactive IR*
15:00 – 16:00	*Break out session*	
16:00 – 16:30	*Coffee Break and Poster Session*	
16:30 – 17:30	*Report back and Discussion*	

Invited Keynotes

Adaptive Search on a Web Scale

David Harper

Google Switzerland GmbH,
harper@google.com

Abstract. This talk provides an overview of research opportunities in adaptive search, with particular emphasis on web search and the consequent challenges of scale. Examples of adaptive search will be drawn from web search engines, and based in part on these examples, I will explore how adaption applies to search processes and search interfaces. The implications of web scale for research will be considered from various viewpoints: types of user, types of search, and search engine performance. Evaluation of adaptive web search remains a challenge, and I will briefly cover types of experiment, including access to collections and research tools. Some concluding remarks will be made on possible research directions and activities for the information retrieval research community.

A Probability Ranking Principle for Interactive IR

Norbert Fuhr

University of Duisburg-Essen
Information Systems
Department of Computational and Cognitive Sciences
Duisburg, Germany
norbert.fuhr@uni-due.de

Abstract. The classical Probability Ranking Principle (PRP) forms the theoretical basis for probabilistic Information Retrieval (IR) models, which are dominating IR theory since about 20 years. However, the assumptions underlying the PRP often do not hold, and its view is too narrow for interactive information retrieval (IIR). In this talk, a new theoretical framework for interactive retrieval is proposed: The basic idea is that during IIR, a user moves between situations. In each situation, the system presents to the user a list of choices, about which s/he has to decide, and the first positive decision moves the user to a new situation. Each choice is associated with a number of cost and probability parameters. Based on these parameters, an optimum ordering of the choices can the derived - the PRP for IIR. The relationship of this rule to the classical PRP is described, and issues of further research are pointed out.

References

1. N. Fuhr. A probability ranking principle for interactive information retrieval. *Inf. Retr.*, 11(3):251–265, 2008.

Oral Presentations

Reaching the Boundaries of Context-Aware IR: Accepting Help from the User

Raiko Eckstein and Andreas Henrich

University of Bamberg, 96052 Bamberg, Germany,
{raiko.eckstein,andreas.henrich}@uni-bamberg.de,
http://www.uni-bamberg.de/en/minf/

Abstract. The paper outlines an *interactive retrieval model for complex search situations*. The highly creative process of constructing products in the engineering domain is often characterised by difficult information needs which cannot be fulfilled easily by current search engines. We introduce a user-centred approach which allows the visual statement of complex context-augmented search queries. Additionally, the user is guided through high-dimensional data according to the *exploratory searching* paradigm. We propose the usage of parallel coordinates plots known from multi-dimensional data visualisation as a means for issuing faceted search queries.

1 Motivation

This research investigates search mechanisms in product development processes in the mechanical engineering domain. Mechanical engineers face a variety of different artefact types during their work which manifests itself in different information needs. The user not only wants to retrieve simple *documents*, but his information needs also centre around *products*, *projects* and *persons / experts*.

We examined one scenario, where an engineer was looking for a certain product. Not the product itself was his main objective, but the supplier who delivers it. The engineer wanted to know which other products were offered by that supplier to achieve higher order sizes. Those kind of complex search situations are difficult to support proactively by a context-aware search engine.

In a development project many different document types exist which capture different aspects of the product developed, like textual documents describing requirements, technical drawings and 3D-CAD[1] models. Not all documents describing a product are equally relevant in every process phase. The information needs are specific to the task the engineer is working on. To leverage reuse of existing parts or products in the company, the engineers need an efficient way to get access to artefacts from past projects.

We want to support the user in retrieving the existing information by utilizing context information about the user and the documents in the company. Our context model consists of the following seven dimensions: *user*, *document*,

[1] Computer-Aided Design

product, *process*, *task*, *project*, and *company*[1]. That includes domain specific measures like the degree of maturity for products.

2 Description of the Problem

Our search prototype covers the main document types occurring in product development which comprises CAD models, technical drawings and different textual documents amongst others. Unlike in text retrieval engines, our scenario demands several mechanisms to determine similarity. In text retrieval several mechanisms exist for the determination of similarity. We also included different similarity measures for domain specific document types like 3D-CAD models and technical drawings. Here the contained information cannot be represented as simple text and therefore bears the necessity of additional similarity measures. That step is necessary to offer the user a single point of entry to start compound searches. The search engine is embedded into a workflow engine which guides the engineer through the development process. The integration of the search engine into a project planning portal is elaborated in [2]. The engine stores project and product dependent information which manifests itself as contextual information that can be used in search queries. The inclusion of this kind of information in a similarity search leads from *single-criteria matching* to *multi-criteria matching*, which introduces several difficulties.

This includes the determination of weights for the different search criteria, a description of the dependencies between the criteria and the decision, whether a criterion acts as a simple filter or if it influences the document ranking. Different solutions are conceivable including methods of machine learning and manual determination of the weights by experts. But since the researched domain and the scope of the search engine are quite broad, we claim that no automatic definition of the weights can be achieved as the information needs in that special scenario are too diverse. We are of the opinion that the user himself can be decisive about the search criteria, if the user interface is intuitive for the user. He needs navigational access to the available information including the context information. That access can be subsumed under the broader notion of *exploratory search* which depicts a more interactive approach to find relevant information[3].

The user interface should not deliver the classical *user experience*, but act as a means to support the user in stating complex queries involving queries-by-example with a customizable weighting of contextual information and other facets.

3 Interactive Retrieval Model for Complex Searches

This section describes the interactive retrieval model and introduces a graphical user interface which enables the user to state complex search queries necessary in the product development domain.

3.1 The Layer Approach

As we want to support the engineer in retrieving the above described artefact types, we distinguish different layers. A layer is defined by a single artefact type and contains all indexed artefacts from this type. In our scenario we can distinguish the layers *document*, *product*, *project*, and *person*. This step is necessary, since a mixture of different artefact types within the ranking of one search result is not possible assuming different similarity measures and would also be difficult to convey to the user.

Different types of connections between the different artefact types and the actual artefacts exist. The elements of one layer can be connected. In the product layer, *is-part-of* relations can be established between products and their components. The linkage between the layers defines the anchor points which enable the traversal through the layers. For example, *documents* are linked to the *project*, in which they were created. Our search paradigm allows advancing from one layer to another through those inter-layer connections. For example, a requirements document — found in the documents layer — is connected to a project from the project layer. Additionally, the different projects consist of subprojects where different links might be propagated. For each project the user easily can switch to the person layer and find people working in that project. In our model every layer can be connected to every other layer and is not restricted to only connecting the layer on top or below that one.

The engineer usually searches on one layer and is navigating in one type of artefacts. He should have the possibility to navigate between the different layers (i.e. the different artefact types) based on interim search results. Those could be derived from initial searches or results of browsing through the stock of artefacts, i.e. the user can take one result artefact and switch to another layer by using its facets. Alternatively, the user might use the whole result set and use it as a filter on other layers of artefacts. That filtering can be adjusted later if necessary and therefore enables a roundtrip between the different layers.

3.2 The Visual Approach

Our graphical user interface expands the search paradigm of *faceted searching*[4] in a visual way which eases the definition of more complex search queries. Additionally, our search prototype aims at making dependencies and correlations between different facets of an artefact visible.

Figure 1 shows a screenshot of our current graphical user interface search prototype showing a text query for "manual" with the facets *user*, *project*, *phase*, *group* and *document type* with four selected projects. The upper area contains a *parallel coordinates plot* [5] as the main part of the user interface. The visualisation consists of a changeable number of parallel axes representing different facets. Each axis can be swapped, removed or used for filtering the facets. A list box with additional available facets allows the adding of new facets which are applicable according to the selections made up to now in the current query. If the user selects documents of type *CAD*, the facet values for the document format should be restricted. The bottom of the search interface shows the result list which lists the currently selected search results according to the chosen facets and their values. In future versions that result list will be enhanced by additional advanced result representations of the document's facets.

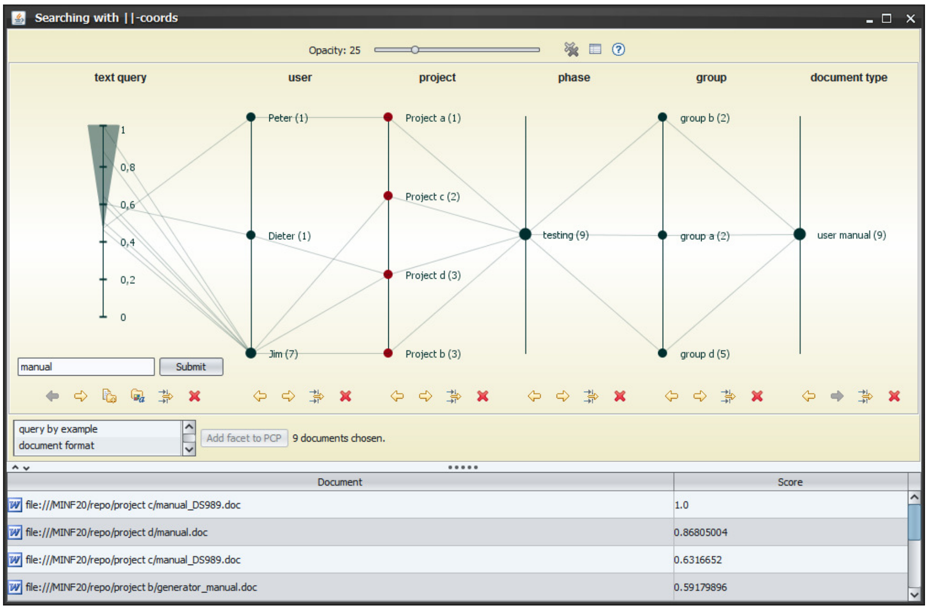


Fig. 1. Prototype showing Parallel Coordinates as a search interface

The classic input field for text queries was omitted to allow a unification of the query statement. Text queries are viewed as a document facet. The left axis in the screenshot shows a query for the term "manual". The specific facet values consist of the score, the text search component returned for each document. The same approach works for queries-by-example.

The user can start a retrieval task classically by stating a query which might consist of a textual query or by an example file which is used as query input. The user might quickly sketch a drawing of a component he needs in his current task and ask the search engine for similar components in a geometric way. Alternatively, the user can start browsing all available documents and then refine the search results by restricting them by choosing facets.

Facets in a faceted search usually are used for filtering. A selection of a classical facet value filters out all documents which do not belong to that facet. With the parallel coordinates visualisation we expand this notion and enable multi-selections of facet values and use facets for ranking purposes.

The query options depend on the different types of facets. The facets with ordinal and nominal measurements allow the selection of single or multiple facet values. The selections act as a filter on the result documents, i.e. a binary decision is made, if the result list should contain the document or not. The result list then consists of an unordered set of documents. Ratio measurements allow the selection of intervals where certain sections can be overlayed by a function which ranks documents with certain values for that facet higher. A use case for this type of query can be the selection of products which have a high degree of maturity. The user might accept products ranging from 100% to 80% but wants to have the more mature documents on top of the ranking. Figure 2 shows examples of functions which can be applied to facets of that type.

This custom ranking functionality can be used with several facets in one query. The user can add weights to the ranking facets to set the weighting for the single facets. If no weighting is set, simply an equal weighting is assumed.

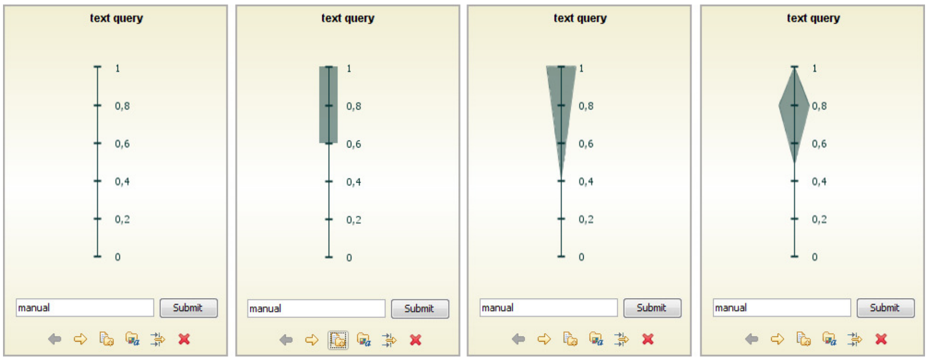


Fig. 2. Query types for ratio measurements

After the user has issued and refined the query and retrieved a set of result artefacts, he can choose to state a new search query based on the obtained results. For example, if the engineer is searching for technical drawings based on facet filtering, he can choose to submit a new query (by-example) based on a found technical drawing from the result set. This step includes a geometric similarity search among all technical drawings and returns all geometrically similar technical drawings to the one from the query. Alternatively, the user can choose to search for documents with similar facet values. This multi-level querying enables the user to do exploratory searches.

Our search interface incorporating a parallel coordinates plot supports experts in searching for documents and artefacts in the mechanical engineering domain. Those experts should be enabled to save certain searches as "templates" so that the engineer/non-expert can use the search engine without being overwhelmed with the higher complexity the search interface offers in comparison to present search engines. Those templates reduce the number of degrees of freedom so that the search space can be reduced accordingly. Still it seems important to make this added search functionality available for product developers without revealing the whole complexity, but leaving some parameters, e.g. search terms, to adjust the search.

4 Future Work

Obviously, the resulting search interface itself may become complex and inappropriate for casual users. Therefore, we consider the presented approach as an interface for power users which can store a created search as a project phase specific template for casual users which can apply search templates for the actual phase they are working in a much easier way.

For this template mechanism also more advanced result representations (e.g. including term clouds) have to be included in order to visualise the overall content of the whole result set including its heterogeneous structure.

Acknowledgements

The work described in this paper is part of the Bavarian Research Cooperation FORFLOW consisting of six Bavarian universities working in the fields of engineering design and computer science. It collaborates with 21 companies and is funded by the Bavarian Research Foundation (Bayerische Forschungsstiftung BFS). [2]

References

1. Eckstein, R., Henrich, A.: An integrated context model for the product development domain and its implications on design reuse. In: Proc. of the 10th Intl. DESIGN Conference, Dubrovnik, Croatia. (2008)
2. Eckstein, R., Henrich, A.: Integration possibilities of a context-based search engine into a project planning portal in the mechanical engineering domain. In Hinneburg, A., ed.: Lernen-Wissen-Adaption, Workshop Information Retrieval. Halle/Saale. (2007)
3. White, R.W., Drucker, S.M., Marchionini, G., Hearst, M., Schraefel, M.C.: Exploratory search and hci. In: CHI '07: Human factors in computing systems, New York, NY, USA, ACM (2007) 2877–2880
4. Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., Yee, K.P.: Finding the flow in web site search. Communications of the ACM **45**(9) (2002) 4249
5. Inselberg, A., Dimsdale, B.: Parallel coordinates: a tool for visualizing multidimensional geometry. In: Proc. of the First IEEE Conf. on Visualization. (1990)

[2] For more details see http://www.abayfor.de/forflow/en/

Using Regions of Interest for Adaptive Image Retrieval

Michael Springmann and Heiko Schuldt

Database and Information Systems Group, University of Basel, Switzerland
{michael.springmann, heiko.schuldt}@unibas.ch

Abstract. Content-based image retrieval mainly follows a Query-by-Example approach and therefore requires well selected examples to start an initial search. This position paper describes how *Regions of Interest* (ROI) can be used to better adapt the system to the user's information needs. In particular, it highlights how novel input devices such as Interactive Paper or TabletPCs can be used to capture much more details about what the user is precisely looking for already at the time the query is defined or when relevance feedback is specified.

1 Introduction

In traditional keyword-based text retrieval, the user poses a query by choosing the subset of terms that are relevant for his information need. Each document in a collection to be searched, in turn, is expected to contain significantly more words than the query. Therefore, the user already provides a good starting point by defining what he considers relevant – and can gradually refine his search by either adding new terms or adjusting the weights of individual terms by means of relevance feedback.

In content-based image retrieval (CBIR), the user starts by selecting one or more images to pose a query by example (QbE). The example may contain several regions, and it is rather natural to assume that some of them may be more relevant for the user's information need than others (e.g., the foreground or the center of the image might be more relevant than the background or elements towards the boundary of the image). Systems like Blobworld [1] allow the user to select the region of interest from the automatically segmented image for formulating a query. However, automatic segmentation does not always give desired results for CBIR, e.g., if the object is partially occluded or consists of several parts. The latter is quite common, for instance in medical CBIR, where the region of interest might be the complete fracture of a bone and thus encompasses two disconnected regions as well as the part in between.

Relevance feedback has been proposed to overcome the well-known semantic gap between high-level semantics intended by the user and low-level features used for retrieval (e.g., in MARS [2]). However, little attention has been put so far on the perspective to use relevance feedback also to lower the misalignment of relevant query parts to irrelevant parts of the retrieved images.

One reason for this is that selecting appropriate regions from images can be more time-consuming and therefore less user friendly. In the context of the Query-by-Sketching (QbS) project [3], we aim at overcoming this limitation by using input devices like Interactive Paper [4] and Tablet PCs. Although these devices, in particular Tablet PCs, are available since several years, most applications do not exploit their full potential – in particular, they are rather used as a replacement for mouse and keyboard and no particular attention is paid to the question which tasks can be made more user-friendly with the special support offered by these devices. We believe that visual tasks, especially image retrieval, will be candidates for significant improvements.

In this paper, we present first first results of an ongoing project aiming on providing adaptive image retrieval using regions of interest. In order to make use of the special character of novel input devices in a best possible way, the overall retrieval process is performed using several steps, (which are also reflected in the structure of this paper):

1. The user poses a query by *selecting one or more regions of interest* (ROI) in a way tailored to exploit Interactive Paper or Tablet PCs. This step is further detailed in Section 2.
2. *Corresponding regions* of images in the database are determined. Automatically derived classifications and constraints may be used to reduce the search space and therefore reduce the overall execution time (see Section 3).
3. The *ranking of results* is based on the similarity of image regions that have been identified in the previous step. Customized distance measures are used to ensure that only relevant parts of the image contribute to the computed score (see Section 4).
4. *Relevance feedback* gives the user the chance to refine the search. This iterative process might affects all steps of the retrieval. The related parts are therefore described directly in each of the sections.

The paper concludes with a summary in Section 5.

2 Selecting Regions of Interest

Conventional approaches to region-based image retrieval assume that the user's PC is equipped with keyboard and mouse. Therefore the two selections that can easily by performed are (a) select regions by clicking on a pre-segmented area of an image or and (b) to draw rectangular bounding boxes. Proper image segmentation on-the-fly is non-trivial and time consuming, even if tools like magic wand or magnetic lasso are used. Thus, existing solutions trade segmentation quality for speed of execution and convenience for the user.

In our approach, we assume that the user is equipped with a digital pen as input device that resembles a regular pen. The digital pen can determine its positions on the interactive paper and perform the actions defined for that position (see [3, 4] for more details on this technology). In our case, this is used for selecting regions on printed images and issuing queries. It is therefore straightforward

(a) (b) (c) (d)

Fig. 1. An example query image (a), a rectangular bounding box of the cat (b), a manually segmented region containing the same cat (c) and the car in the background (d).

to select individual regions of interest (ROI) in the same easy and flexible, yet possibly imprecise way as one would do that on regular paper. At the same time, this is still likely to be more accurate than using an ill-segmented image or the rectangular selection of non-rectangular objects. Fig. 1 shows examples of such selections. If needed, the initial selection can be refined using semi-automatic segmentations like color invariant snakes [5] or GrabCut [6].

For relevance feedback, the digital pen is not only used to select relevant or irrelevant result images, but also to select the regions that make them relevant. Notice, that this might be much easier and intuitive to end users than assigning numeric values or preference judgements to images, since the task can be easily described as: "Select all relevant regions of the presented result images."

Retrieval using region selection can be enhanced by annotations (for instance also automated image analysis algorithms such as face detection) in order to use additional features which might help in determining similarity and adding constraints on extracted image attributes. The advantages are twofold: Firstly, users do not need to specify all relevant concepts in the query which can be quite challenging (e.g., in [7] medical annotations with 116 distinct classes are used). Secondly, class membership of images determined at insertion time can be stored and evaluated very efficiently and therefore reduce the search space for nearest neighbors based on visual features.

3 Matching Regions in Images

Since our approach does not rely on pre-segmented images in the database, it is more flexible in adapting to the user's information need. The user can, for instance, also select areas high contrast in itself, but low contrast from the background like the dark parts of the fur of the cat in Fig. 1(c) or partly covered objects like the car in Fig. 1(d). The latter is the common case where automatic image segmentation fails as already mentioned in [1]. On the downside, more processing time is needed during query execution, since the regions of images of the database corresponding to the ROI still need to be identified. The following section describes a generic solution used in our approach. However, it should be

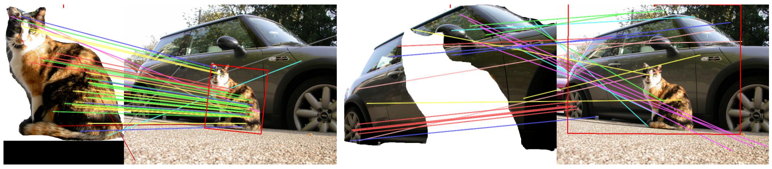
(a) (b)

Fig. 2. The 1481 SIFT keypoints extracted from the query image (a) and a very similar image of same size with 2536 keypoints (b) to which we will later on compare it.

noted that for particular image domains, there might exist much more sophisticated solutions that incorporate more knowledge about the objects shown in the images, e.g., model-based image registration for medical images or face detection/recognition. Such techniques should be preferred whenever available and applicable. Nevertheless, in the general CBIR setting, arbitrary images might be stored in the system and user queries may also vary significantly. Therefore such generic approaches are still of great importance.

For images in the database we extract salient keypoints and descriptor using SIFT [8], which has been proposed for object identification and is invariant to scale, rotation, and to a certain extent to variations of the illumination or viewpoint. For every such keypoint SIFT extracts a 128-dimensional descriptor [9]. In addition to this vector, the main orientation of the gradient and the scale at which it was detected are computed and stored with the keypoint. Fig 2 shows an arrow for each of the 1481 keypoints that have been extracted from the 682×512 pixel image in Fig. 1(a).

Matching keypoints in the ROI with corresponding keypoints in the database uses the squared Euclidean distance of the SIFT descriptors. For each keypoint in the query image the best match in the compared image needs to be determined and also the second best match, if the ratio between the distance of the two is used as quality measure as proposed in [9]. This computation is quite intensive – in the particular example when matched with the image shown in Fig. 2(b) it requires 1481×2536 computations of the distance between 128-dimensional vectors. Therefore this takes about 924 ms on a Intel Core 2 Duo with 2.33 GHz. Taking a closer look at the keypoints in our query image, we see that many of them are located outside the region of interest. In fact, the area containing the cat as in Fig. 1(c) contains only 684 keypoints and matching only these takes 438 ms. The car shown in Fig. 1(d), despite of covering an area significantly bigger in size, contains only 536 keypoints and matching takes 350 ms. Notice also, that both image, Fig. 1(a) and the original image of 2(b), have exactly the same size and very similar content, yet SIFT detected approximately 70% more keypoints in Fig. 2(b). Many of these additional keypoints are located in the lower part of the image, where there is additional road visible, which is in our case of little interest. When the rectangular selection shown in Fig. 1(b) is used,



(a) (b)

Fig. 3. Result of determining the corresponding regions. Lines connect keypoints which are matched, color of lines indicate cluster. The red lines have been used to determine the affine transformation which maps the query image inside the red bounding box.

also additional (irrelevant) keypoints are contained, 900 in total, and matching takes 576 ms and therefore more than 30% longer than our region of interest.

In order to speed up finding the matches, we apply an early termination strategy to the computation of the squared Euclidean distance similar to [10]. This reduces the computation time to 629 ms for the entire image, 311 ms for the cat region, 236 ms for the car region and 401 ms for the rectangular bounding box without changing the result. We did not yet exploit multi-threading or any kind of indexing. For the time being, our main focus has been on the exploitation of keypoints not only for object recognition where SIFT was originally used and a correct match is defined as matching the same point in the same or identical object, but apply a more relaxed rule, where also points in similar objects can be identified and thus, make also use of it in more general retrieval settings.

Using not only the number of matched keypoints, but also their location and the resulting rotation, scaling and translation of the image, an affine image transformation can be constructed, which maps the query region to the relevant region of the image from the database. When the query image is provided by the user and not part of the database of known images, it might not be possible to define a single threshold on distance or nearest neighbor ratio as proposed in [8] for the selection of matched keypoints. In particular, if the threshold is tight to find only very accurate matches, the system may return few or not even a single match for the region of interest. Instead, an ordered list of matches can be computed such that it always contains enough entries even in unfavorable cases, e.g., where illumination is very different or only similar objects are present. These matches can be clustered based on the scale and rotation in which this transformation would result. Selecting only the cluster with the most consistent transformation and filtering with RANSAC, the transformation gets more robust to many mismatched keypoints as long as at least three good ones remain [11]. The results of this matching step are displayed in Fig. 3. Notice if entire images containing several objects like in Fig. 1(a) were used, either only one region can be matched or both regions just a little - with the user having no control to which will happen.

As a result, the matching region of the ROI in the images of the database can be determined. Using regions in relevance feedback on result images, the

(a) (b)

Fig. 4. Images placed next to each other after affine transformation was applied.

corresponding keypoints in the query can be identified and weighted or excluded. Also certain types of transformations can be excluded, e.g., if the user does not want to allow matches of the ROI being upside-down or scaled to small size. Such constraints can already be enforced within the loop over each keypoint since we have stored their orientation. If the orientation of a keypoint in the matched image differs too much from the query keypoint, we can already reject it and do not even need to compute the distance between the descriptors. This further reduces the matching time, in our experiments with an allowed angle of 10 degrees for the cat region to 152 ms without noticeable changes in the result, because the determined transformation would stay within that limit anyway. At this point, further improvements might also consider optimization of the clustering and RANSAC, since those two steps together require approximately the same time after matches have been determined. Furthermore, any available knowledge about the desired content should be evaluated to limit the number of images for which correspondence needs to be evaluated. The similarity between the corresponding regions itself does not need to be based on SIFT, but can be any appropriate feature(s) and distance measure computed on regions.

4 Similarity Between Corresponding Regions

Many image retrieval systems focus on features that are to a certain extent robust to transformations, e.g., global color histograms are invariant to rotation and not very sensitive to slight variations of the viewpoint or displacement of the observed object. But the tradeoff for that robustness is that such features may not be specific enough to capture the user's perception of similarity. Since our approach can start computing the similarity with the assumption that we have a transformation for corresponding regions, we can evaluate similarity down to the pixel level. We can therefore apply comparison directly on the pixels or areas of reduced resolution and allow small deformations, for instance, with an image distortion model (IDM) of appropriate size [10].

Simple extensions of this model would also allow enforcing strict spatial constraints between several unconnected regions by transforming the two regions into a single one with a mask or alpha channel defining the weight of each pixel within similarity computation. All pixels within a ROI have a weight set to 1, pixels outside a weight of 0 and are ignored when distortion is evaluated. The same extension can also be used to limit the effect on the computed similarity to pixels that are inside the ROI rather than the entire query image. This is therefore relevant even when a single region is selected. In Fig. 4, all white pixels would be assigned a weight of 0. Notice that the transformed images do not perfectly fit since SIFT and the derived affine transformation cannot compensate for viewpoint changes in 3D and movement and deformation of the objects in itself, like the cat not holding exactly the same pose. However, IDM allows for small deformations which will also reduce such effects. In comparison to Fig. 1(b), many background pixels are ignored which otherwise would impact similarity score, e.g., the mirror just behind the right ear of the cat.

For using several regions without strict spatial constraints, e.g., both regions must be present but no particular spatial arrangement is enforced, regions have to be evaluated separately. Similar to [12], distance combining functions based on Fuzzy-And, Fuzzy-Or, and Weighted Average can be used whenever the retrieved image should contain at least one region similar to each query region (And semantics), to at least one query region (Or semantics) or a weighted combination in case of different importance. In case of even more sophisticated semantics, e.g., if two query regions may not be matched to the same region in the image of the database as it is enforced in [13], these constraints would need to be evaluated already in the matching phase, in order to avoid overlapping regions.

By the means of relevance feedback, the parameters for similarity and allowed transformations as well as weights for keypoints and regions can be refined. Since our approach expects feedback on regions, differences in background or other unrelated parts of the image which otherwise could affect query refinement negatively, will not have an impact.

5 Summary and Conclusion

This paper presents an approach that allows a user in CBIR to specify his information need in a user-friendly, yet powerful way: i.) the user can select (almost) free-form regions of interest inside the query; ii.) the user can add new regions to the query and refine previously selected ones; iii.) the user can give a detailed feedback by selecting such regions, which also gives the opportunity to re-adjust weights. The approach will provide users the expressiveness for queries, similar to what he is used to from text retrieval. In addition, this will enable far more expressive relevance feedback.

Novel devices for interacting with the system reduce the burden of defining regions in images significantly. In the context of the ongoing QbS project, we have identified and implemented the basic components for making use of a digital pen for region selection. In particular, this includes a component for matching regions of interest to other images based on keypoints. Defining such regions and therefore excluding unrelated parts of the image also reduce the computation costs during matching and can enhance the quality of similarity computation.

Subsequent clustering and filtering make SIFT applicable to user-specified query images, i.e., where no pre-set threshold can be used. First experiments have shown that matching of two images takes several milliseconds. If applied naïvely to searching within large collections thousands of images, this clearly can sum up to overall retrieval times exceeding the criterion for interactive response significantly. Yet, the approach can be seamlessly parallelized, either on a multi-core machine or in a cluster or Grid environment. In addition, sophisticated pre-selection (e.g., combination with high-level features or limitations on transformations) will further speed up the retrieval process. This makes the approach highly appropriate for use in an interactive mode where the query can already be refined based on the first images retrieved. Detailed evaluations and user studies will have to show how much regions of interest can improve image retrieval quality and how well the interaction modalities and patterns are accepted by users.

References

1. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Blobworld: image segmentation using expectation-maximization and its application to image querying. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(8) (Aug 2002) 1026–1038
2. Rui, Y., Huang, T., Mehrotra, S.: Content-based image retrieval with relevance feedback in MARS. Proc. of Int. Conf. on Image Processing (ICIP 1997) **2** (Oct 1997) 815–818
3. Springmann, M., Ispas, A., Schuldt, H., Norrie, M.C., Signer, B.: Towards query by sketch. In: Proc. of 2nd Int. DELOS Conference. (Dec 2007)
4. Norrie, M.C., Palinginis, A., Signer, B.: Content publishing framework for interactive paper documents. In: DocEng '05: Proceedings of the 2005 ACM symposium on Document engineering, New York, NY, USA, ACM (2005) 187–196
5. Gevers, T., Smeulders, A.W.M.: Interactive query formulation for object search. In: Proc. of 3rd Int. Conf. on Visual Information and Information Systems (VISUAL '99). (Jun 1999) 593–600
6. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. **23**(3) (2004) 309–314
7. Deselaers, T., Müller, H., Deserno, T.M.: Automatic medical image annotation in ImageCLEF 2007. Pattern Recognition Letters (2008)
8. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. of 7th Int. Conf. on Computer Vision (ICCV'99). Volume 02. (1999) 1150–1158
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2) (2004) 91–110
10. Springmann, M., Dander, A., Schuldt, H.: Improving efficiency and effectiveness of the image distortion model. Pattern Recognition Letters (2008)
11. Studer, M.: Fast scale and orientation matching of images using scale-invariant features (2008) external MSc Thesis at University of Fribourg, Switzerland, performed at University of Basel.
12. Böhm, K., Mlivoncic, M., Schek, H.J., Weber, R.: Fast evaluation techniques for complex similarity queries. In: VLDB. (2001) 211–220
13. Weber, R., Mlivoncic, M.: Efficient region-based image retrieval. In: CIKM '03: Proceedings of the twelfth International Conference on Information and Knowledge Management, ACM Press (2003) 69–76

An Experimental System for Adaptive Services in Information Retrieval

Claus-Peter Klas[1], Sascha Kriewel[2], and Matthias Hemmje[1]

[1] FernUniversität in Hagen,
Claus-Peter.Klas|Matthias.Hemmje@FernUni-Hagen.de
[2] Universität Duisburg-Essen, sk@is.inf.uni-due.de

Abstract. Information retrieval and digital library systems of today offer only minimal personalisation and customisation functionality for individual users and user groups. DAFFODIL acts as virtual digital library with strategic user support and allows gathering of rich information about users – implicitly and explicitly through special tools and an integrated event logging framework. Based on the collected data DAFFODIL is used to experiment with innovative adaptive services for information retrieval. This paper presents extensions of the DAFFODIL architecture used for building such adaptive services, and possible personalisation and adaptive usage scenarios.

1 Introduction

The digital libraries of today – in addition to preventing a satisfying search experience by presenting users with too many access points, different query forms and limited functionalities – rarely offer users and groups personalisation and task-based customisation. Only very limited use is made of the user's context, e. g. the task, search situation, or knowledge domain of the user. Even adaption to environmental factors, which could include adaption of services to handle service breakdown, low network bandwidth or other user-independent problems or situations, is rare.

We will describe the adaptive architecture and extensions of the DAFFODIL framework, which serves as an experimental system for innovative information retrieval (IR) services. The adaptive service architecture provides researchers and developers of digital libraries comprehensive implicit and explicit data about users and the system. The DAFFODIL framework can be used to enable high-level personalisation functionality to satisfy individual needs and preferences.

In the following section we present a model and an architecture for adaptive services. In the context of some current and possible future adaption scenarios the implicit and explicit data that is currently gathered by the DAFFODIL framework is described. We close with a short summary and outlook.

2 Adaptivity in Information Retrieval

A classical IR system consists of three major "components": **users** (in **context**), who interact with the IR **system** to search for **content**. Adaptivity should focus on each of these components. We can identify:

Adaptive system services gather knowledge about the whole computer system, consisting of all running services. The information can be used to optimise processes, enhance quality of service or system security.
Focussing just on the data sources, the gathering of knowledge about technical and content aspects, such as access parameters and quality or features of the content, can be used to enhance response time or answer quality.

Adaptive content services focus on the transferred information given by user queries and result documents from a semantic viewpoint. Adaptive knowledge gathered by classical IR functionality can be used to enhance the results for the user.

Adaptive user services allow for adaptivity and personalisation based on a user model. The graphical user interface, the presented information as well as other services can be adapted to individual user or groups.

IR systems that want to offer such services should be context-aware and self-adaptive. Context-aware services adapt according to the current context of the user, which could be the current role of the user, the work or search task, an organisational context, or physical context like the device used to retrieve information or the location from which the user searches. Being self-adaptive, the services are able to evaluate their own behaviour and learn from the interaction with users, changing their behaviour accordingly.

2.1 Daffodil Framework

We would like to introduce the Daffodil framework[1] as an experimental system for the evaluation of adaptive services. Daffodil is a virtual digital library system providing access to many sources from the domain of computer science, and targeted at strategic support of users during the information seeking and retrieval process. It provides basic and high-level search functions for exploring and managing information objects including annotations over a federation of heterogeneous digital libraries (DLs) based on a service-oriented architecture. For structuring the functionality, we employ the concept of high-level search activities for strategic support. A comprehensive evaluation in [Klas et al. 04] showed that the system supported most of the information seeking and retrieval aspects needed for a computer scientist's daily routine.

In order to enable adaptivity Daffodil collects implicit and explicit user interactions and system actions as described in previous publications [Klas et al. 06]. This interaction can be examined and captured at various levels of abstraction, starting at the system/hardware level and covering the complete spectrum of user-system interaction.

[1] `http://www.daffodil.de`

3 Adaptation and Personalisation Scenarios

3.1 Information Retrieval

Personalisation in the area of information retrieval usually implies either re-ranking of the result list or dynamically enhancing the query based on context and preferences of the user [Koutrika & Ioannidis 04]. It is generally used to cope with the problem of information overload.

To overcome this problem, Landwich et al. [Landwich et al. 08] introduce a conceptual model to capture the context of information dialogs. The current context is described by four general sets, named the *content*, *interest*, *relevance* and *result* set. As the user starts to search, four dialog subsets are captured during each query step, the *explored recall*, the *focused recall*, the *verified recall* and the *stored recall* set. Through the combination of several queries, the current context narrows, and focuses more and more on the relevant documents.

Based on the information derived from these sets, corresponding visualisation and interaction tools are used at each step in the information dialog. Thus users are enabled to strategically optimise their information behaviour and more quickly satisfy their information need.

3.2 User Interface

Depending on the work and search task, as well as other aspects of the user's context, different parts of an information system's user interface gain more or less importance, or benefit from adaption to the current situation. Within Daffodil we explored this in the search domain of movies and songs, dynamically adapting the user interface not only to specific steps during a media search task, but also to the varying experience of users. Users of different levels of search experience will benefit from different selections of IR tools and result presentation. For experienced users, some tools that beginners would use as scaffolds can be removed, while other tools can offer more options that would confuse search novices.

3.3 Adaptive Suggestions

Supporting end users in finding and deriving good strategies for satisfying their information search is a well known problem [Kriewel 06]. Depending on the level of system involvement a search system can suggest useful strategies for completing a specific work – either upon user request or pro-actively – or might offer to automatically execute strategies for very common search problems with no or with reduced user intervention.

A scenario under research within the Daffodil framework is finding search strategies by analysing search logs, and using these findings for suggesting situationally appropriate strategies to the user [Kriewel & Fuhr 07]. Based on the current user situation after a query the system provides suggestions for the next steps to the user. The suggestions are ranked according to collaborative judgement of users in previous situations, and then adapted to the specific current situation. The suggester system learns from interacting with users, and can improve the ranking for future situations.

4 Summary and Outlook

We presented a generic classification of adaptivity for information retrieval systems and digital libraries. We also gave a brief description of the DAFFODIL framework, focussing on the possibility to gather explicit and implicit information about users and system. This allows the use of DAFFODIL as a highly versatile experimental system in adaptive information retrieval and digital library research. In three scenarios – information retrieval, user interface and strategic suggestions – we look into existing and upcoming adaptive services within DAFFODIL.

In current research we focus on the information retrieval scenario, investigating implicit and explicit relevance feedback in combination with interactive information retrieval and information visualisation. Additional investigations are necessary to identify further possible adaptive services. Furthermore we need to capture and analyse the user-system interaction to complete the picture of users and their tasks and context.

References

[Klas et al. 04] **Klas, C.-P.; Fuhr, N.; Schaefer, A.** (2004). Evaluating Strategic Support for Information Access in the DAFFODIL System. In: Heery, R.; Lyon, L. (Hrsg.): *Research and Advanced Technology for Digital Libraries. Proc. European Conference on Digital Libraries (ECDL 2004)*, Lecture Notes in Computer Science. Springer, Heidelberg et al.

[Klas et al. 06] **Klas, C.-P.; Albrechtsen, H.; Fuhr, N.; Hansen, P.; Kapidakis, S.; Kovács, L.; Kriewel, S.; Micsik, A.; Papatheodorou, C.; Tsakonas, G.; Jacob, E.** (2006). A Logging Scheme for Comparative Digital Library Evaluation. In: Gonzalo, J.; Thanos, C.; Verdejo, M. F.; Carrasco, R. C. (Hrsg.): *Research and Advanced Technology for Digital Libraries. Proc. of the 10th European Conference on Digital Libraries (ECDL 2006)*, Lecture Notes in Computer Science, S. 267–278. Springer, Heidelberg et al.

[Koutrika & Ioannidis 04] **Koutrika, G.; Ioannidis, Y. E.** (2004). Personalization of Queries based on User Preferences. In: *Preferences.*

[Kriewel & Fuhr 07] **Kriewel, S.; Fuhr, N.** (2007). Adaptive Search Suggestions for Digital Libraries. In: *Asian Digital Libraries: Looking Back 10 Years and Forging New Frontiers (ICADL 2007)*, S. 220–229.

[Kriewel 06] **Kriewel, S.** (2006). Finding and Using Strategies for Search Situations in Digital Libraries. *Bulletin of the IEEE Technical Committee on Digital Libraries 2(2).* http://www.ieee-tcdl.org/Bulletin/v2n2/kriewel/kriewel.html.

[Landwich et al. 08] **Landwich, P.; Vogel, T.; Klas, C.-P.; Hemmje, M.** (2008). Supporting Patent Retrieval in the Context of Innovation-Processes by Means of Information Visualisation. In: *Proc. of ECKM 2008.*

Poster Presentations

Automatically Creating and Comparing Features and Contexts

Pablo Bermejo[1], Frank Hopfgartner[2], and Joemon M. Jose[2]

[1] University of Castilla-La Mancha, Albacete, Spain,
pbermejo@dsi.uclm.es
[2] University of Glasgow, Glasgow G12 8RZ, United Kingdom,
{hopfgarf,jj}@dcs.gla.ac.uk

Abstract. Representation of multimedia documents is a major problem in multimedia information retrieval due to the semantic gap. In this work, we present a tool to automatically create and evaluate three different kind of features for a database to decide what kind of representation is better suited for representing multimedia documents. Our tool provides the option of splitting interaction log files into different contexts to study their influence on classifiers' performance.

1 Introduction

A common approach to predict the value of the class a document belongs to is to select representative document features. In the text domain, predestined features are vocabulary features, as they can express the semantics of the documents. In the multimedia domain however, selecting representative features becomes a challenging task. Many videos for instance are not enriched with textual annotations [6], relying on vocabulary features is hence not adequate. Moreover, widely used low-level features such as colour histograms or textures are not sufficient enough to describe the content of the document they are meant to represent. The reason for this is the semantic gap [9]: the difference between (computable) low-level representations of a multimedia document and the actual semantic meaning of the document. Identifying representative features and feature combinations can be a key breakthrough in the challenge of bridging the semantic gap, as the quality of the used set of features is of great importance for a classifier to achieve a good performance [2].

In this paper, we present our work on automatically transforming the representation of a set of video documents into one of three different feature types: behavior, object and vocabulary features. The software is based on Weka [12] and can be used to evaluate a dataset based on extracted features.

Our tool allows to split the features into different contexts that can be found in log files of user experiments, where users test a video retrieval system by searching for different topics and under different conditions. Thus it can be investigated which contexts improve the classifier's performance and which ones make it perform worse. Some example contexts our tool can handle are:

1. *Topic.* Splitting the database with respect to the topic a shot was accessed by the user.
2. *Condition.* User might perform the search under different conditions, for example different interfaces.
3. *User Experience.* Database is split according to the experience of users accessing shots.
4. *Topic Difficulty.* Database is split according to the difficult (as perceived by user) of each search task.
5. *Mix of contexts.* Contexts can be mixed in pairs.

This paper is structured as follows: In the next section, we introduce the three types of features which can be analysed by our tool. In Section 3 we provide an overview how to use our tool. Finally, in Section 4 we present our main conclusions for this work.

2 Feature Types

So far, research on feature construction and document classification has been concentrated on three different categories: behaviour features, object features and vocabulary features. In the remainder of this section, we will introduce these features.

2.1 Behaviour Features

User Behavior Features [5, 1] give information about how the user interacts with a document. In retrieval systems, this information is related to the actions the user performs on retrieved documents. Hence, behaviour features can be extracted from user log files. We consider the behavior features shown in Table 1. They can be split into three groups: *Click-Through features*, which represent information about clicks the user performed on retrieved documents; *Browsing features*, which show different metrics about time spent with a result and *Query-Text features*, which count words in the current text query and make comparisons with other text queries.

2.2 Object Features

Object Features are extracted from the data collection. Typical features are *low-level features* and additional *metadata*. Common visual features used in the video domain are for example colour layout, dominant colour, textures or edge histograms which have been extracted from representative keyframes. Audio features can be rhythm histograms or spectrum descriptors of a video. In the video domain, *metadata* keeps information such as the length of retrieved results or information related to the textual transcript of the retrieved videos.

Feature name	**Description**
ClickFreq	Number of mouse clicks on shot
ClickProb	*ClickFreq* divided by total number of clicks
ClickDev	Deviation of *ClickProb*
TimeOnShot	Time the user has been performing any action on shot
CumulativeTimeOnShots	*TimeOnShot* added to time on previous shots
TimeOnAllShots	Sum of time on all shots
CumulativeTimeOnTopic	Time spent under current topic
MeanTimePerShotForThisQuery	Mean of all values for *TimeOnShot*
DevAvgTimePerShotForThisQuery	Deviation of *MeanTimePerShotForThisQuery*
DevAvgCumulativeTimeOnShots	Deviation of *CumulativeTimeOnShots*
DevAvgCumulativeTimeOnTopic	Deviation of *CumulativeTimeOnTopic*
QueryLength	Number of words in current text query
WordsSharedWithLastQuery	Number of equal words in current query and last query

Table 1. Behavior Features used to predict shots relevance

2.3 Vocabulary Features

Vocabulary Features are a bag of words created from the transcript of multimedia data. It is expected that video relevance classification based on textual transcripts is promising due to the fact that text has more descriptive power than, for example, low-level visual features. Nevertheless, transcripts do not always relate to the content of a video [13] and shots do not have a long transcript to base an analysis on. Therefore, we consider not only the transcript of one single video element, but also from the neighboured n shots with respect to the time of the video. In our case, text is not used to compute statistics, but to create a vocabulary of words to perform the typical task of classification based on text. It is expected that n-Windowed Vocabulary features perform better than creating a bag of words from the transcript of a single shot only. Our tool filters text through a stop-word list and performs Porter stemming [8] for further analysis. When we use text to create a bag of words and evaluate using a bayesian classifier, we do not use Naive Bayes but the Naive Bayes Multinomial, which is recommended for text classification [7].

3 Tool Usage

A first condition for using the tool is to format the user log files in the Attribute-Relation File Format (.arff) as introduced for the Weka machine learning software [12]. After tuning all parameters in the configuration files, the tool can be executed by running the following Java classes:

1. *CreateFeatures*. This class converts the input file to another .arff file where instances are represented by the created features. The output file contains a projection of the previous log file where the only features are the ones created, besides the tuple descriptor features (userID, shot name,...). Parameters to tune are: type of features to create, kind of relevance (official or user-defined) to use, order of the output features.

2. *SplitContexts*. This program splits the constructed database in as many parts as the chosen context can be split. Several contexts can be chosen from the configuration file. For example, if in the experiment there were four search topics and the tool's user wants to split the .arff file in contexts based on search topic, then four .arff files will be created. Each of these files will contain instances with the same topic attribute value in the original file. If no context differentiation is desired, then no split is done. The descriptor tuple features are removed in any context case as the aim is to evaluate using just the created features (behaviour, object or vocabulary) without any knowledge of userID, topicID, or shot name.
3. *Evaluate*. The evaluation class performs a cross validation of the .arff files created by SplitContexts. Weka is used for the actual classification task. The following parameters can be defined in the configuration file: number of folders in cross validation, times to perform cross validation, classifiers to use (Naïve Bayes, Naïve Bayes Multinomial, Support Vectors Machine, k-NN, and AODE available) and balance level of training sets (from 0 for no change, to 100 to become training sets fully balanced). The class outputs several metrics computed for each classifier and possible class value: Accuracy, Precision, Recall, F1-measure.
4. *IncrementalSelection*. Finally, an incremental wrapper-based feature subset selection [3] is performed over the chosen databases and then a projection of the data using the selected features is evaluated. Evaluation results are displayed showing the chosen feature subsets and the same metrics mentioned above.

3.1 Example of work done using this took

To learn how different kinds of constructed features affect relevance prediction, Bermejo et al. [4] used data logs from two users experiments [11, 10]. Each log file contained verbose data of the actions each user performed while interacting with different video retrieval interfaces. Bermejo et al. used the presented tool to construct Behavior, Object and Vocabulary Features from these log files. They conclude that windowed Vocabulary features seem to be the best when there is text associated with the video. Behavior features were only useful if stored behavior is used in a collaborative system to influence the retrieved results. Finally, Object features return lesser relevant shots and are more consistent.

4 Conclusions and Future Work

Our tool makes it easy to create three different kind of features and thus to compare which representation is better suited for the representation of multimedia documents. Contexts splits can be created and their influence on classifiers studied. Finally, feature selection can be easily performed over databases or context splits created. As future work, we propose to improve the command line interface into a GUI and make the configuration files simpler to tune for the user.

5 Acknowledgments

This work has been partially supported by the JCCM under project PCI08-0048-8577, MEC under project TIN2007-67418-C03-01 and FEDER funds. The second author was supported by the European Commission under contract K-Space (FP6-027026).

References

1. E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26, New York, NY, USA, 2006. ACM Press.
2. R. Bekkerman, A. McCallum, and G. Huang. Automatic categorization of email into folders: Bechmark experiments on enron and sri corpora. Technical report, Department of Computer Science. University of Massachusetts, Amherst., 2005.
3. P. Bermejo, J. Gámez, and J. Puerta. On incremental wrapper-based attribute selection: experimental analysis of the relevance criteria. In *IPMU'08: Proceedings of the 12th Intl. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2008.
4. P. Bermejo, H. Joho, R. Villa, and J. M. Jose. Comparison of feature construction methods for video relevance prediction. In *MMM '09: Proceedings of the 15th International MultiMedia Modeling Conference*, 1 2009. to appear.
5. S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Syst.*, 23(2):147–168, 2005.
6. M. J. Halvey and M. T. Keane. Analysis of online video search and sharing. In *HT '07: Proceedings of the eighteenth conference on Hypertext and hypermedia*, pages 217–226, New York, NY, USA, 2007. ACM.
7. A. McCallum and N. K. A comparison of event models for naive bayes text classification. In *AAAI/ICML-98 Workshop on Learning for Text Categorization*, pages 41–48, 1998.
8. M. F. Porter. An algorithm for suffix stripping. pages 313–316, 1980.
9. A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000.
10. R. Villa, N. Gildea, and J. M. Jose. FacetBrowser: a user interface for complex search tasks. In *ACM Multimedia 2008*, 10 2008. to appear.
11. R. Villa, N. Gildea, and J. M. Jose. A study of awareness in multimedia search. In *Joint Conference on Digital Libraries*, pages 221–230, June 2008.
12. I. H. Witten and F. Eibe. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, San Francisco, 2 edition, 2005.
13. R. Yan and A. G. Hauptmann. Co-retrieval: A boosted reranking approach for video retrieval. In *CIVR*, pages 60–69, 2004.

A CBR framework for implementing community-aware web search engine

Rushed Kanawati

LIPN - UMR 7030
Université Paris 13 - CNRS
99 AV. J-B. Célément 93430 Villetaneuse, France
rushed.kanawati@lipn.univ-paris13.fr

Abstract. We qualify a search engine to be *community-aware* if results it returns to answer a query Q_u submitted by a user u depends, not only on the query itself, but also on the community profile to which the user u belongs. In this paper we present a general case-based reasoning approach for mining past queries submitted by a community of like-minded people in order to re-rank future result-lists in function of implied preferences of the community.

1 Introduction

Query-based web search engines, such as GOOGLE and YAHOO have become unavoidable tools for information searching. The simplicity of the query-based searching is probably the main reason behind their wide popularity. However, general purpose web search engines operate in a *one-size fits all* mode [1]: They do not take into account the context in which queries they process have been issued. Community-aware web search engines follow a new approach that aims to cope with shortcomings of existing web search engines [1, 6, 5, 19, 16, 15].

A community is simply defined as a group of like-minded people that share some information needs. A community can be composed of members of an organization (ex. a R&D research team), or simply users that submit queries using a search box embedded in a specific web page [6]. The basic idea of a community-aware search engine is to mine user's evaluations of past queries in order to *customize* results of current similar queries. The usefulness of community-awareness is bound to two basic hypothesis:

– H1. Like-minded people submit (a lot of) similar queries,
– H2. It is possible to compute user's evaluations of returned results in an easy and an *implicit* way.

Hypothesis $H1$ seems to be valid for web search tasks, as shown in some user's querying behaviour analyzed in [20, 17]. As for hypothesis H2, it is now common to use result's selection (or click through data) to induce user's evaluations of obtained results[19, 8, 2, 14].

Mining past community queries can be applied to perform different tasks: query expansion [12, 10], query reformulation [16], query clustering [21], result list modification [6] or result list re-ranking [9, 15, 4, 18]. In this work we present CASE: a framework for implementing a community-aware web search result re-ranking systems. The framework provides components that allow to capture the web querying behaviour of a community of users and to mine the query log in order to re-rank result list of current queries. The focus, in this work, is put on the re-ranking module (or component). A case-based reasoner is used to implement the re-ranking module. Three main hot spots[1] are identified in this module. The paper is organized as follows: next we present a quick review of community-aware web search result re-ranking approaches. The CASE framework architecture is described in section 3.1. The re-ranking module outlines are given in 3.2. Finally we conclude in section 4.

2 Related work

One early work describing a community-aware result re-ranking approach is presented in [9]. In this work authors propose to couple an explicit document recommendation system with a search engine. Documents that are recommended to the community through the recommendation systems are used to re-rank results obtained from a web search engine. Naturally, result documents that are similar to highly recommended documents are ranked first.

In the I-SPY system [4, 13] the query log is organized in a hit matrix H where rows are indexed by past submitted queries and columns are documents retrieved in response to these queries. An entry $H[i, j]$ gives how many times document d_j has been selected when presented as an answer to query q_i. Let q_T be a target query and $R_{qT} = \{d_1 \ldots, d_n\}$ be the associated ordered result list. The system retrieves from the hit matrix rows indexed by queries similar to q_T. Let q_S be a query similar to q_T, the relevancy of a document d_j is given by $rel(d_j, q_S) = \frac{H_{SJ}}{\sum_{\forall j} H_{sj}}$. This approach requires an overlap between results lists of currents queries and past ones. The high dynamic nature end great evolving rate of the web make this condition hard to hold.

In [1] authors propose to compute the community of a user submitting a query based on the current observed interactions. The community-based information is used to provides context for the queries. The final rank of the result list is a combination of the the classical query-document relevancy and relevancy of each document to the community context.

In [18] authors propose to user collaborative filtering technique in order to re-rank results according to the user profile and the community profile. This approach requires identifying users in order to learn their profiles.

[1] In a component framework a hot spot is a place where adaptations can occur.

3 The Case framework

3.1 General architecture

As illustrated on figure 1, the Case framework has a distributed, modular and an extensible architecture. The framework is composed of a set of inter-communicating modules. Each module can produce a set of events and can register to get events generated by others. An *event broker* module provides both event registration and subscription services. This allows to tailor the system architecture as needed (ex. adding new wrappers to get make a build a meta-search engine).

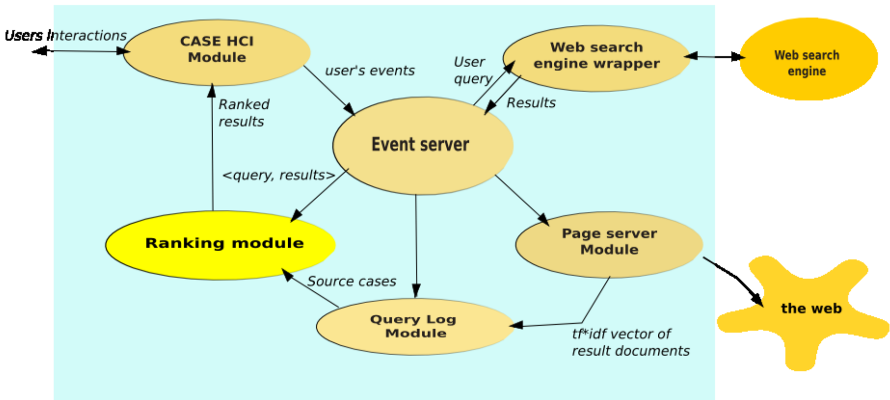


Fig. 1. General architecture of the CASE system

In addition to the *event broker* module, the following modules are actually provided:

1. *User Interface module (HCI module):* This module is implemented as a simple web server that provides users with the usual web searching interface. It acts also as a *web proxy* that allow to fetch result documents selected by a user.
2. *Web search engine wrapper:* This module forwards user's queries to the wrapped search engine. For each query q it retrieves the top m documents returned by the search engine.
3. *Page server module:* Given a document URL, the role of this module is to provide a keyword vector representing the document. Currently, this module treats only HTML pages. The top k *frequent* terms (after removing stop keywords), are returned.

4. *Query log module:* This module saves the history of users querying activities. An entry in the query log file is composed of the following fields:
   - *qid*: this is the query unique identifier.
   - *q*: this is the query keyword list.
   - $R_q = \{r_q^1, r_q^2, \ldots, r_q^n\}$: is the result list. r_q^i is the i^{th} document in the result list as displayed to the user (after the applying the re-ranking approach). Each document $r_q^i \in R_q$ is defined by two attributes: 1) the document URL (denoted $R_q^i.url$ and 2) a weighted keyword vector denoted by $r_q^i.v$. This vector is simply the classical $tf * idf$ weights of terms in r_q^i defined over the set of terms obtained from all documents in R_q [3].
   - S_q is an ordered list that enumerates documents selected by the user from the list R_q.
5. *Ranking module:* This is the main module in the system. The goal of this module is to compute for each couple $< q, R_q >$ a permutation of R_q where highly relevant documents, form the community point of view, are ranked at the top. The ranking module is the main focus of this paper and will be detailed in the next sections.

3.2 The Ranking approach: an overview

A case based-based reasoner is used to implement the ranking approach. Each entry in the query log file is used to edit a source case. A case is defined by the following attributes:

- a query q, which is represented by an ordered list of keywords,
- the query result list R_q. Each document $r_i \in R_q$ is represented by an identifier (i.e. URL), and a keyword vector,
- a voting function V_q. This is a function that allows to rank result lists retrieved to answer a query q' similar to q. An example of such a voting function is given in [15].

The ranking approach functions as follows: first, the system searches the case base for cases that have a *similar query*. It then applies the voting function provided by each retrieved case to compute a rank of the result lists returned by the search engine. Finally a ranking merging function is applied in order to obtain the final ranking which is presented to the user. Following this approach, three main hot spots are provided in the framework:

- *The query similarity function*: various query similarity function can be used. We can classify query similarities into fours main classes: a) term-based measures, b) result-based measures, c) selection-based and d) hybrid measures. While most works apply simple term-based measures, there still be a need to evaluate the performances of other types of similarities on the overall re-ranking effectiveness.
- *The voting function*: this is a central issue in our approach. In [15] one first voting function has been proposed. However the proposed vote scheme

has a complexity of $O(n^2)$ where n is the number of returned results. A function whose complexity is $O(n)$ is now proposed but we believe that the effectiveness of the approach depends highly on the quality of the voting function.
- *The rank aggregation scheme*: again, different rank aggregation schemes can be proposed. Algorithms provided by the voting theory are first candidates [7]. More recent algorithms developed for meta-search engines can also be considered [11].

The different concrete adaption of these hot spots result in different re-ranking performances. The goal of the framework is to help in comparing performances of different concrete approaches. Currently, a first concrete application has been developed.

4 Conclusion

In this work, an original framework for building a community-aware web search results re-ranking systems is presented. Three main hot spots are proposed. The goal is to ease the comparison among different possible adaptations of these hot spots. Currently, one first concrete implementation has been developed. We claim that, the voting function is the most important hot spot to consider. The idea is to learn from past research session (queries) a ranking function that can rank documents that have never been rated before by users. We believe that this is an important feature to have in the context of the ever evolving web.

References

1. Rodrigo B. Almeida and Virgilio A. F. Almeida. A community-aware search engine. In *WWW'2004*, pages 413–421, New Yoprk, May 2004. ACM.
2. Ricardo A. Baeza-Yates. Applications of web query mining. In *ECIR*, pages 7–22, 2005.
3. Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press / Addison-Wesley, 1999.
4. Evelyn Balfe and Barry Smyth. Query mining for community based web search. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI04)*, pages 594 – 598, 2004.
5. Evelyn Balfe and Barry Smyth. An analysis of query similarity in collaborative web search. In *ECIR*, pages 330–344, 2005.
6. Evelyn Balfe and Barry Smyth. A comparative analysis of query similarity metrics for community-based web search. In Héctor Muñoz-Avila and Francesco Ricci, editors, *ICCBR*, volume 3620 of *Lecture Notes in Computer Science*, pages 63–77. Springer, 2005.
7. J.C. Borda. Mémoire sur les élections au scrutin. *Comptes rendus de l'Académie des sciences, traduit par Alfred de Grazia comme Mathematical Derivation of a election system , Isis, vol 44, pp 42-51*, 1781.

8. O. Boydell, B. Smyth, C. Gurrin, and F. Smeaton. A study of selection noise in collaborative web search. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 1595–1597, Edimbourg, Ecosse, 2005.
9. Boris Chidlovski. Collaborative re-ranking of search results. In *Proceedings of the AAAI workshop on artificial intelligence for Web search*, pages 18–22, 2000.
10. Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. Probabilistic query expansion using query logs. In *WWW*, pages 325–332, 2002.
11. Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW*, pages 613–622, 2001.
12. Larry Fitzpatrick and Mei Dent. Automatic feedback using past queries: Social searching? In *SIGIR*, pages 306–313. ACM, 1997.
13. Jill Freyne, Barry Smyth, Maurice Coyle, Evelyn Balfe, and Peter Briggs. Further experiments on collaborative ranking in community-based web search. *Artif. Intell. Rev.*, 21(3-4):229–252, 2004.
14. Thorsten Joachims, Laura A. Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, 25(2), 2007.
15. Rushed Kanawati. A Multi-agent System for Improving Result Ranking Service of Web Search Engines. In *10th ISPE International conference on concurrent engineering (CE 2003)*, volume 1, pages 91–99. Balkema, June 2003.
16. Rushed Kanawati, Michel Jaczynski, Brigitte Trousse, and Jean-Marc Anderloi. Applying the Broadway recommendation computation approach for implementing a query refinement service in the CBKB meta-search engine. In Brigitte Trousse and Alain Mille, editors, *Conférence français sur le raisonnement à partir de cas (RàPC'99)*, pages 17–26, Palaiseau, France, june 1999. AFIA.
17. Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In Xiaohua Jia, editor, *Infoscale*, volume 152 of *ACM International Conference Proceeding Series*, page 1. ACM, 2006.
18. U. Rohini and Vasudeva Varma. A novel approach for re-ranking of search results using collaborative filtering. In *ICCTA*, pages 491–496. IEEE Computer Society, 2007.
19. Xuehua Shen, Bin Tan, and ChengXiang Zhai. Implicit user modeling for personalized search. In Otthein Herzog, Hans-Jörg Schek, Norbert Fuhr, Abdur Chowdhury, and Wilfried Teiken, editors, *CIKM*, pages 824–831. ACM, 2005.
20. Barry Smyth, Evelyn Balfe, Jill Freyne, Peter Briggs, Maurice Coyle, and Oisín Boydell. Exploiting query repetition and regularity in an adaptive community-based web search engine. *User Model. User-Adapt. Interact.*, 14(5):383–423, 2004.
21. Ji-Rong Wen, Jian-Yun Nie, and HongJiang Zhang. Query clustering using user logs. *ACM Trans. Inf. Syst.*, 20(1):59–81, 2002.

An Adaptive Method to Associate Pictures with Indexing Terms

Marco A. Palomino, Michael P. Oakes, and Yan Xu

University of Sunderland, David Goldman Informatics Centre
Sunderland, SR6 0DD, United Kingdom
{marco.palomino,michael.oakes,yan.xu-1}@sunderland.ac.uk

Abstract. We present a method to automatically acquire a set of keywords that characterise a picture collection. Our method compares captions associated with pictures with a model of general English language. The words that deviate from the model are very specific of the collection and thus make appropriate indexing terms. The discovery of additional indexing terms for each picture is performed by means of a similarity matrix, which records the degree of association between pairs of indexing terms. Our method offers a solution to the construction of indices for multimedia search engines.

1 Introduction

Indexing is the process by which a vocabulary of keywords is assigned to all the documents of a corpus. Since the index relation is the primary connection between user queries and the documents that can satisfy them, choosing a suitable vocabulary of keywords is at the core of the information retrieval problem.

As part of the EU-funded *VITALAS project* [17], which aims to provide a solution to the intelligent access to professional multimedia archives, we have experimented with novel techniques for the automatic extraction of keywords. Such keywords must typify the content of large multimedia collections that comprise text, pictures, audio and video. As a proof of concept, this paper focuses, exclusively, on the extraction of keywords from a set of 100,000 image captions provided by the *Belga Press Agency* [2].

An important source of information about indexing terms that can be exploited in information retrieval is *co-occurrence*: the fact that two or more terms occur in the same documents more often than chance [12]. Consider the example in Table 1: *Caption 1* is likely to be relevant to the query since it contains all the terms in the query. However, *Caption 2* is also a good candidate for retrieval: its terms *soccer* and *football* co-occur with *champions* and *league*, which can be evidence for semantic relatedness.

An adaptive retrieval system has previously been created for the domain of tourism by Berger *et al.* [3]. This system was based on an associative network in which semantic relations between terms were stored, and spreading activation was used to find new, related terms not in the original problem specification.

	Term 1	Term 2	Term 3	Term 4
Query	soccer	football		
Caption 1	soccer	football	champions	league
Caption 2			champions	league

Table 1. Exploiting co-occurrence in computing content similarity

The discovery of new terms was a form of *implicit query expansion*. Wang *et al.* [20] learned concept relations from short natural language texts, and stored them in a structure called a *fuzzy associated concept mapping*. New concepts, not explicitly present in the original texts, are recommended to the user based on this mapping, to "stimulate individual reflection and generate new knowledge", and the method could also be used to generate additional terms for a search engine query. In this paper, rather than focusing on the query, we annotate our searchable documents with additional indexing terms not present in the original texts, to enable the user to find images using keywords which were not present in the original image captions. Rather than using an associative network or concept mapping in which only selected relations between concepts are stored, we use a similarity matrix in which every concept has a relation with every other concept, namely a real value in the range 0 to 1. Although the current version of our system does not allow the user to filter the additional suggested terms, our system is nevertheless adaptive because the learned concept relations are specific to the domain of the documents, here captions press agency photographs.

The remainder of this document is organised as follows: Section 2 introduces our method for the automatic extraction of keywords, which is based on the *chi-square test*—a useful statistical technique to determine if observed data deviates from expected data under a particular hypothesis [9,14]. Section 3 describes the *service-oriented architecture* that we have employed to implement a prototype indexing system. Section 4 explains our use of a similarity matrix to capture semantic relatedness, and, lastly, Section 5 offers our conclusions.

2 The Chi-Square Test

Previous research has shown the effectiveness of the chi-square test to find words that are truly characteristic of a corpus [11,13,15]. As proposed by Rayson *et al.* [15], we have employed the chi-square test to compare captions associated with pictures posted on the Belga website with a model of general English language. The words that deviated from the model were very specific of Belga pictures and thus made an appropriate set of keywords. Indeed, the words yielded by the chi-square test were evaluated by professional annotators, who concluded that they were descriptive of the collection and likely to make suitable indexing terms.

2.1 Comparison Corpus

It should be observed that the chi-square test requires a comparison between the corpus for which we want to find out its characteristic keywords and a *comparison corpus*. Since we wanted to identify words characteristic of Belga captions, we required a comparison corpus that did not belong to the news-agency domain. After considering our choices, we opted for *Europarl* [10].

Readers interested in our results should look at the full list of indexing terms derived using the chi-square test, which is available from the authors' website (`http://osiris.sunderland.ac.uk/cs0mpl/Belga/`).

3 Service-Oriented Architecture

One of the goals of the VITALAS project is developing a fully-functional prototype engine to search for multimedia content. The components comprised by the engine have been developed by different research partners, based in separate locations. Updates and improvements are integrated as seamlessly as possible, and the engine scales to increasingly larger collections.

We have decided to use a *service-oriented architecture*, where components exchange data with one another, as they participate in the indexing and retrieval processes. In order to be combined and reused, the components are accessible over the Internet. To do this work, we are using the *WebLab platform* [18].

Due to space limitations, we will describe solely the implementation of a Web service in charge of associating indexing terms with each image caption. Such a service *tokenises*, *stems* and removes *stop-words* from a caption. Afterwards, it determines its particular set of indexing terms. The full set of indexing terms is computed previously using the method described in the former section. Each term is assigned a TF-IDF weight, and the whole list of terms is sorted by weight in descending order.

4 Similarity Matrix

In this section, we describe the discovery of additional indexing terms for each image, which are hopefully relevant even though they do not appear in the original image captions. We begin by adding additional terms which consistently co-occur with those terms that do appear in the image captions. Thus, an image whose caption contains the index term *ac_milan_soccer* would also be indexed by *soccer*, *soccer_world_club*, *champions_league* and *goal*, all of which repeatedly co-occur with *ac_milan_soccer* in the collection as a whole. Users making queries containing any of these terms might well be interested in seeing images with *ac_milan_soccer* in their captions, and thus we widen the net of potentially successful query terms. Our method consists of creating a term-term similarity matrix showing the degree of association between every pair of indexing terms. To derive each entry in this matrix, we first obtain the *document vectors* for each pair of terms.

Each entry in a document vector shows the number of times a term was found in each caption in the collection—a vector of $[1, 0, 3]$ would show that a given term was found once in caption number 1, did not occur in caption number 2, and was found 3 times in caption number 3. For multi-word keywords, we sum the number of times each of the constituent single words are found in each document. The similarity of a pair of terms is given by the cosine similarity of their document vectors [19]—each term has a similarity of 1 with itself. For example, the most similar terms to *academy_award* were *academy_award_oscar* (0.99), *award* (0.99), *ceremony* (0.46), *nature* (0.31), *singers* (0.27), *video* (0.25), *classic* (0.24), *hand* (0.20), *film_festival_cannes* (0.15) and *filming* (0.15).

To reduce storage space, only the similarities between each term and the 40 most similar terms were stored. If the similarity between term I and term J is the matrix entry $Sim(I, J)$, then each term I in the controlled vocabulary is given a weight according to the formula:

$$Weight(I) = \sum_{J \in doc} Sim(I, J).$$

In our current system, we use only *first order associations*—namely direct similarities between the original indexing terms in the captions and the terms in the controlled vocabulary, as opposed to *second order similarities*, where not only the direct similarity of the original terms to the other terms were considered, but also indirect similarities, such as if *soccer* is related to *team*, and *team* is related to *player*, then *soccer* is related to *player*. This was due to the difficulty of picking up exact synonyms in the first order process, as such alternative forms rarely occur together in a single document. The related process of spreading activation can be stopped at the n^{th} order of association [16]. Finally, the term-term associations found by Ferber *et al.* associative model tend to stabilise after an infinite number of iterations [7]. While we consider only term-term associations, other architectures such as Ferber *et al.* [7], and spreading activation, work in alternating cycles, where similarities between terms and documents are found at one cycle, and then similarities between documents and terms are found at the next cycle. Our model takes advantage of the interdependencies of terms: instead of each query term being expanded in isolation and the resulting set of query terms being pooled, we take into account the similarity between all the original index terms and each potential new term at the same time.

Salton and Buckley [16] found that associative retrieval methods work only if locally valid relations between terms are used—i.e., relations which are valid for a specific document collection in a given domain. By generating a similarity matrix automatically from domain specific documents, we are able to produce an indexing system adapted to that particular domain much better than if we had used the term relations available from a thesaurus or library classifications.

Latent semantic indexing (LSI) is another matrix-based technique that projects queries and documents into a space with "latent" semantic dimensions [4,5]. Co-occurring terms are projected onto the same dimensions, non-co-occurring terms are projected onto different dimensions.

In the latent semantic space, a query and a document can have a high cosine similarity even if they do not share any terms—as long as their terms are semantically similar according to the co-occurrence analysis. We can think of LSI as an alternative similarity metric to word overlap measures such as TF-IDF.

The application of LSI to information retrieval was originally proposed by Deerwester *et al.* at Bellcore [6]. It has been compared to standard vector space search on various references and has recently been employed in statistical data mining and knowledge discovery [8]. LSI's strength in high recall searches reported in a number of papers is not surprising, since a method that takes co-occurrence into account is expected to achieve higher recall. LSI has recently been used by Bader and Chew [1] to associate terms in one language with related terms in other languages.

5 Conclusions

We have described our method for extracting terms from image captions, which will act as indexing terms and phrases for these images. Also, we have described a method based on an associative matrix for the discovery of additional indexing terms that do not appear literally in the image captions. Since this is work in progress, we still need to compare the retrieval effectiveness of our original set of keywords with the combined set of original and associated keywords.

Acknowledgments. This work was supported under the EU-funded VITALAS project (FP6-045389).

References

1. Bader, B.W., and Chew, P.A. Enhancing Multilingual Latent Semantic Analysis with Term Alignment Information. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 49–56, Manchester, UK, August 2008.
2. Belga. *Belga Press Agency*. http://www.belga.be/.
3. Berger, H., Dittenbach, M., and Merkl, D. An Adaptive Information Retrieval System based on Associative Networks. In *Proceedings of the 1st Asian-Pacific Conference on Conceptual Modelling - Volume 31*, pages 27–36, Dunedian, New Zealand, 2004.
4. Berry, M.W., and Browne, M. *Understanding Search Engines: Mathematical Modeling and Text Retrieval.* Society for Industrial and Applied Mathematics, April 2005.
5. Berry, M., Drmavc, Z., and Jessup, E. Matrices, Vector Spaces, and Information Retrieval. *SIAM Review*, 41(2):335–362, June 1999.
6. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., and Harshman, R. Indexing by Latent Semantic Analysis. *Journal of the Society for Information Science*, 41(6):391–407, 1990.
7. Ferber, R., Wettler, M., and Rapp, R. An Associative Model of Word Selection in the Generation of Search Queries. *Journal of the American Society for Information Science*, 46(9):685–699, October 1995.

8. Giles, J., Wo, L., and Berry, M. *Statistical Data Mining and Knowledge Discovery*, GTP (General Text Parser) Software for Text Mining, pages 455–471. CRC Press, 2003.
9. Greenwood, P.E., and Nikulin, M.S. *A Guide to Chi-Square Testing*. Wiley-InterScience, New York, 1996.
10. Koehn, P. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Machine Translation Summit*, Phuket, Thailand, September 2005. Asia-Pacific Association for Machine Translation (AAMT).
11. Leech, G., and Fallon, R. Computer Corpora - What do They Tell Us about Culture? *International Computer Archive of Modern and Medieval English*, (16):85–100, 1992.
12. Manning, C., and Schutze, H. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, May 1999.
13. Oakes, M.P., and Farrow, M. Use of the Chi-Squared Test to Examine Vocabulary Differences in English Language Corpora Representing Seven Different Countries. *Literary and Linguistic Computing*, 22(1):85–99, 2007.
14. Plackett, R.L. Karl Pearson and the Chi-Squared Test. *International Statistical Review/Revue Internationale de Statistique*, 51(1):59–72, April 1983.
15. Rayson, P., Leech, G., and Hodges, M. Social Differentiation in the Use of English Vocabulary: Some Analyses of the Conversational Component of the British National Corpus. *International Journal of Corpus Linguistics*, 2(1):133–152, 1997.
16. Salton, G., and Buckley, C. On the Use of Spreading Activation Methods in Automatic Information Retrieval. In *Proceedings of the 11th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 147–160, Grenoble, France, June 1988.
17. VITALAS. *Video and Image Indexing and Retrieval in the Large Scale*. `http://www.vitalas.org/`.
18. WebLab. *An Open Platform for Processing Multimedia Documents*. `http://weblab-project.org/`.
19. Widdows, D. Measuring Similarity and Distance. In *Geometry and Meaning*. CSLI Publications, November 2004.
20. W.M.Wang, C.F.Cheung, W.B.Lee, and S.K.Kwok. Mining Knowledge from Natural Language Texts Using Fuzzy Associated Concept Mapping. *Information Processing and Management*, 44(5):1707–1719, September 2008.

Searching Without Text in an Interactive TV Environment

Paul Ferguson, Hyowon Lee, Cathal Gurrin, Sorin Sav, Thomas Foures, Stephane Lacote, Alan F. Smeaton, and Noel O'Connor

Centre for Digital Video Processing, Dublin City University, Ireland.

Abstract. Compared with a traditional desktop application, the interactive TV (iTV) environment provides additional challenges when providing advanced search facilities. In order to allow effective searching within this domain we propose utilising the user's viewing context, in order to adapt both the underlying search that is performed, as well as the way in which the results are presented to the user. We describe how we have integrated this type of search into our own prototype iTV system.

Key words: adaptive search, multimedia, interactive TV

1 Introduction

While watching TV on an iTV device, the ability to search for related content, both in the recorded content on the device, as well as externally (on websites such as YouTube[1]), can be extremely useful. Naturally the user may do this themselves while watching TV: for example, using an additional computer to search the WWW for related content. Alternatively in a multimedia information retrieval system, such as those used annually in TRECVid [Smeaton et al., 2004] to search for video content relevant to provided topics – if a similar type of system is provided to search a user's recorded content, then this can provide an additional (much more advanced) way to find related and relevant content for the user. Both these approaches for finding related content require complex interactions from the user, typically through the use of a keyboard and mouse, however, this type of interaction is not suitable for the "lean-back" type of interaction that is associated with TV viewing [Jensen, 2005]. Instead we propose to utilise the viewing context of the user in order to dynamically search for related context, so that the user may only require a simple remote control with which to search, and without the need of formulating queries and selecting the types of search that they wish to perform.

Essentially there are a number of different sources that the user can search in order to find related content, the question is how best to automatically search these so that the user is presented with the most appropriate content, with only minimal input required from the user. In Section 2 we describe the sources

[1] http://www.youtube.com

Second International Workshop on Adaptive Information Retrieval, London, UK

that are available to be searched; Section 3 describes the contexts in which the search may be performed, mentioning the sources which are be utilised from each context, as well as showing how the system tailors the search results to the user depending on the context in which the search is performed; Section 4 draws conclusions as well as discusses future work.

2 Search Sources

There are a number of different sources of information associated with different types of video content, it is these sources that can be used to find similar content, and here we discuss these different sources of information:

Text: Text-based similarity provides a simple and straightforward method of finding similar content, allowing matching between TV programs based on any text associated with the program. For each TV program we have a number of different text sources: title and description (from the Electronic Program Guide (EPG)); genre classification; actor and director information for movies; team and competition information for sports content; news story description. Apart from the text provided by the EPG, the remaining text is supplied using user annotations, however it is envisioned that most (if not all) of these user annotations could eventually be replaced with automatic tagging, for example by using existing Internet resources such as IMDB [2] for movie annotation.

Visual Similarity: For each program that a user chooses to record, the system can do further visual analysis of its video content – allowing matching between programs, based on their visual similarity. Currently this visual matching is done based on the colour similarity, using colour histograms.

External Sources: There are a number of different Internet resources containing large amounts of video content, most notably is the "video community" YouTube, which contains millions of videos – many of which are related to TV and movie content. Therefore it makes sense to provide a mechanism to view these videos, as they are likely to be of interest to TV viewers. Our system uses the YouTube search API to search this resource in order find matching content.

3 Search Contexts

As described in Section 2 there are a number of different sources that are available to be searched, the main issue is then how best to search these, so that the user can receive meaningful results with minimal interaction effort. The way in which we aim to do this is by using the user's viewing context. Our system defines four different viewing contexts (General, Movie, Sport and News), and depending on which category the user's current program belongs to, their search for similar content will use a specific search and result presentation mechanism defined for that type of context. For each of these contexts we will discuss how the system deals with each search source, and how the results are presented to the user.

[2] http://www.imdb.com

The “Find Similar” content function is invoked by the user pressing a dedicated button on a remote control – this causes a Find Similar panel to slide in from the left of the screen. The contents of this Find Similar panel are then changed, depending on the context in which the viewer is currently watching TV, as we now discuss.

General: The general Find Similar panel displays two different categories of results: 1. similar “Shows” and 2. similar “Clips” – as illustrated in Fig. 1, where it does not contain any of the optional categories (i.e. Movies, Games, Stories). The shows category matches videos based on the entire video. Firstly this displays a list of similar shows that are similar to recorded content on the iTV device, this matching is done using text similarity between the program’s title and description. Secondly a list of the most similar YouTube videos is displayed, this uses the YouTube search API to match videos (again using the program’s title and description text). The clips section provides matching video clips from other video content that has been recorded on the iTV device, with the matching being done based on the visual similarity of the current segment of video that the viewer is watching on screen at that moment.

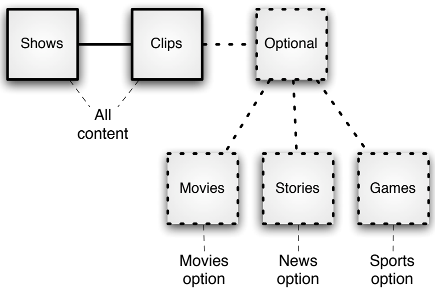


Fig. 1. Find Similar *categories* that are displayed, depending on the viewing context.

Movie: If the viewer is watching a movie when they press the Find Similar button they are again presented with the same “Shows” and “Clips” elements, as previously discussed, although in addition to these, there is an additional “Movies” element presented to the user (as shown in Fig. 1) – which provides additional matching that is relevant for movie content. Firstly this provides a matching of recorded content, limited only to movie content, secondly it allows filtering the results based on the director of the movie, as well as the actors that are present in the movie. Also the results that are returned within the “Clips” element of the panel are filtered to only contain clips of other movies, rather than searching through all types of content.

News: If the viewer is watching a news broadcast when they press the Find Similar button they are presented with an alternative panel, again containing the

"Shows" and "Clips" elements, with the addition of a "News" element, as shown in Fig. 1. This news element contains specific news stories that are matching the currently playing news story – potentially allowing matching of news stories from different news broadcasters. These results are calculated using text similarity between each news story's description. In addition to this the YouTube videos are also matched based on the news story description, rather than the title of the entire broadcast, allowing for the retrieval of much more relevant results based on the currently viewed news story.

Sport: Finally if the viewer is watching a sports show when they press the Find Similar button they are presented with a modified panel. Again this panel contains the default "Shows" and "Clips" elements, although these are complemented with an additional "Games" element (shown in Fig. 1). This games element provides a list of recorded games that are matched based on their similarity with the currently playing sport's *team* and *competition* information (as mentioned in Section 2) as well as the sport's type (as defined by the video's genre type). Also the results presented in the "Clips" element of the panel are filtered to only contain clips from the same sports type and again these clips are matched based on the currently playing video clip i.e. if the currently playing soccer game shows some exciting play around the goal area the aim is to find other similar shots of action in this area of the field.

4 Conclusions

In this paper we have presented a mechanism for finding related video on an iTV, using a number of different sources of information. We have shown how our system adapts both its sources which it searches, as well as the way in which its results are presented, all based on the TV content the viewer is currently watching. In our future work we aim to evaluate both the relevance of the results presented to the user, as well as a user evaluation of the navigation between the different categories of results that are presented to them.

Acknowledgement

We wish to acknowledge the support of Samsung Electronics, Korea, for the research presented in this paper.

References

[Jensen, 2005] Jensen, J. F. (2005). Interactive television: new genres, new format, new content. In *IE2005: Proceedings of the second Australasian conference on Interactive entertainment*, pages 89–96, Sydney, Australia, Australia. Creativity & Cognition Studios Press.

[Smeaton et al., 2004] Smeaton, A., Over, P., and Kraaij, W. (2004). TRECVID: evaluating the effectiveness of information retrieval tasks on digital video. *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 652–655.

Metadata Visualisation Techniques for Emotional Speech Corpora

Charlie Cullen[1], Brian Vaughan[1], Spyros Kousidis[1], John McAuley[1]

[1] Digital Media Centre, Dublin Institute of Technology, Aungier Street, Dublin, Ireland
{charlie.cullen, brian.vaughan, spyros.kousidis, john.mcauley}@dmc.dit.ie

Abstract. Our research in emotional speech analysis has led to the construction of dedicated high quality, online corpora of natural emotional speech assets. Once obtained, the annotation and analysis of these assets was necessary in order to develop a database of both analysis data and metadata relating to each speech act. With annotation complete, the means by which this data may be presented to the user online for analysis, retrieval and organization is the current focus of our investigations. Building on a web interface developed in Ruby on Rails, we are now working towards a visually driven GUI built on Adobe Flex. This paper details our initial work towards this goal, defining the rationale behind development and also demonstrating work achieved to date.

1 Introduction

Work undertaken as part of the SALERO [1] project uses mood induction procedures (MIP) [2, 3] to develop corpora of high quality emotional speech assets obtained under laboratory conditions [4, 5]. The resulting emotional assets can be claimed to be natural and spontaneous, arising out of the manipulation of the task and the interaction of the participants as opposed to voluntary or knowingly coerced attempts to generate emotional states. Once recorded, assets are analysed to define the acoustic parameters of interest [6] and then annotated to include descriptive metadata. All assets, metadata and acoustic analysis data are then stored in a database for query and retrieval, forming the initial speech corpora defined in this paper. In the following sections, a brief description of the corpus, metadata and analysis data parameters are given to illustrate the data available for visualization.

2 Emotional Speech Corpora Design and Construction

The emotional speech corpus used [3] is built on MySQL, with automated annotation tools being implemented in Ajax and Ruby on Rails to allow batch upload and metadata annotation of assets to be performed. The corpus currently contains over 750 assets, with further experiments planned to increase the data set. For initial implementation purposes, the corpus database provides editors with the ability to insert assets, in the form of wav files, and related acoustic analysis data, in the form of

SMIL files. The database parses the SMIL files and populates the corresponding database tables as part of a persistent corpus database back-end. Secondly, editors require remote access to corpus assets. Following initial trials, it was decided to provide the ability for batch uploads, allowing an editor to upload several related assets with the same metadata at the any one time. Each asset is annotated using the IMDI metadata [7] schema, which defines groups of recording sessions within an overall project. Each session relates to a specific recording containing speech acts involving various actors (participants) who deliver them. Each speech act is also defined in terms of its content (e.g. genre/planning type/social context e.t.c) using a combination of open a closed vocabularies for various descriptors. Each asset is also defined in terms of acoustic and emotional analysis data [6] relating to vowels in a speech act. This data is obtained by batch analysis, and is held in standard SMIL format for querying. Analysis data includes acoustic parameters such as duration, pitch and intensity curves, vowel identification, prominence, jitter, shimmer and other voice quality attributes.

3 Speech Corpus Visualisation Interface

Subsequent to populating the corpus database, a suitable user interface to query the assets for the purposes of analysis and retrieval was needed. Although an initial browser had been constructed using Ruby on Rails, it was decided to leverage the potential of new technologies such as Adobe Flex to provide a rich and interactive user interface for the corpus.

3.1 Query Builder

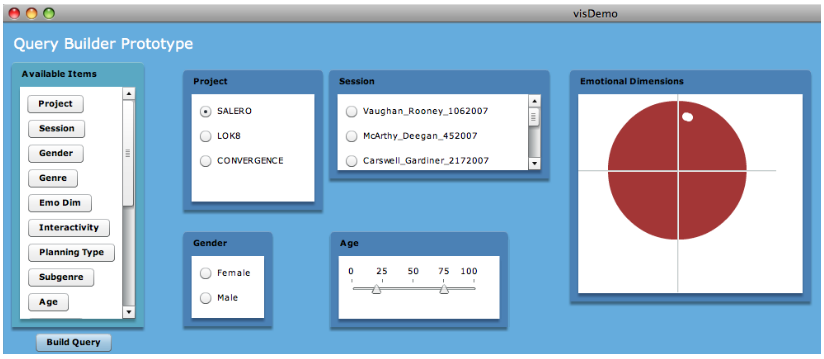


Fig. 1. Prototype of the corpus query builder. The user has chosen assets from the Salero project with an age range of 25 to 75 and a high level of emotional activation.

The query builder interface (Figure 1 above) aims to provide the user with an intuitive means of building queries based on the corpus data. Each parameter can be introduced by clicking a button, and then specific values for that parameter can be set in the relevant window. Usability trials are being performed to determine the most useful groupings of query elements, so that common queries are easily performed. In addition, saving and reloading existing queries is also under development.

3.2 Query Viewer

The query viewer aims to take advantage of the Flex mechanism of drilling down into chart data, so that initial views can be examined in greater detail without losing the context of the results provided (Figure 2):

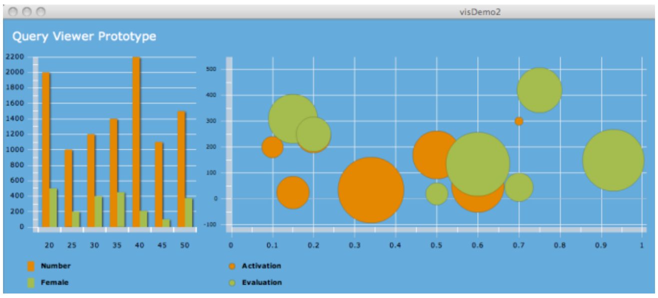


Fig. 2. Prototype of the corpus query viewer. Total assets for the Salero project are displayed in the left chart, with female assets being defined separately. The bubble chart displays emotional dimensions, with the y-axis being number of assets and radius being the number of user ratings.

From initial views built on the query results, users can drill down into specific results with as single mouse click to examine a specific asset in terms of its acoustic analysis data (Figure 2):

Fig. 2. Prototype of the acoustic analysis viewer. Pitch and intensity curves are displayed at the top, with all vowels present displayed relative to the their prominence in the middle chart. The pitch and intensity curves for the current vowel are displayed in the right, as are the emotional dimensions for the current asset.

4 Ongoing and Future Work

This development seeks to leverage web 2.0 technologies as often as possible in order to provide an intuitive, online GUI tool for researchers working with emotional speech corpora. Current work aims to provide the user with tools that are both fast and effective, while resisting the temptation to provide visually entertaining elements that often prove cumbersome on repeated use (e.g. intro movies and slow animated transitions). At time of writing, the development of the interface is ongoing, with frequent user trials being used to focus the development of the final interface. It is intended to develop a fully online prototype by the end of 2008, so that more extensive testing may be performed in a live environment.

Acknowledgments: The research leading to this paper was partially supported by the European Commission under contract IST-FP6-027122 “SALERO”.

References

1. Haas, W., Thallinger, G., Cano, P., Cullen, C., Bürger, T.: SALERO - Semantic Audiovisual Entertainment Reusable Objects. International Conference on Semantic and Digital Media Technologies (SAMT), Athens, Greece (2006)
2. Gerrards-Hesse, A., Spies, K., Hesse, F.W.: Experimental inductions of emotional states and their effectiveness: A review. British Journal of Psychology **85** (1994) 55-78
3. Cullen, C., Vaughan, B., Kousidis, S., Wang, Y.: Generation of High Quality Audio Natural Emotional Speech Corpus using Task Based Mood Induction.: InSciT, Merida (2006)
4. Cullen, C., Vaughan, B., Kousidis, S.: Emotional speech corpus construction, annotation and distribution. The sixth international conference on Language Resources and Evaluation, LREC 2008, Marrakech, Morocco (2008)
5. Johnstone, T., Reekum, C.M.v., Hird, K., Kirsner, K., Scherer, K.R.: Affective speech elicited with a computer game. Emotion (2005) 513-518
6. Cullen, C., Vaughan, B., Kousidis, S., Reilly, F.: A vowel-stress emotional speech analysis method. The 5th International Conference on Cybernetics and Information Technologies, Systems and Applications, CITSA 2008, Genoa, Italy (2008)
7. ISLE: IMDI (ISLE Metadata Initiative), Metadata Elements for Session Descriptions. (2003)

A readability measure for an information retrieval process adapted to dyslexics

Laurianne Sitbon[1] and Patrice Bellot[2]

[1] NICTA, Queensland University of Technogy, Brisbane, Australia, laurianne.sitbon@nicta.com.au
[2] LIA, University of Avignon, France, patrice.bellot@univ-avignon.fr

Abstract. This paper provides a framework for an estimation of relevance that takes readability into account. The experiments focus on dyslexic users and are realized on TREC and CLEF ad-hoc task data.

1 Introduction

Most studies intending to take user needs into account focus on specific informational needs that can be expressed (or not) through the context in terms of topical interest. However, early thoughts about the nature of relevance in information retrieval that were reviewed in [8] incorporated a dimension relative to the information need, sometimes partly not expressed by a query. Text reading is a hard task for impaired readers. When seeking for information their reading time can be lowered by selecting the most emphreadable documents containing the relevant information. Among all possible reading impairments, cognitive or visual, we focus on dyslexia.

Retrieving readable documents. The most widespread formula of readability is *Reading Ease Score* [3]. It is based on the average sentence length ASL, and the average number of syllables per word ASW[3]. It varies on a 0-100 scale, where 30 means difficult and 70 means suitable for adult audiences. For example, MS Word provides this measure as statistical tool and recommend a score around 60-70. The parameters have been estimated on english language for which the measure known as *Flesch score* is computed by Equation 1, with d a document.

$$L_F(d) = 206.835 - 1.015 \times ASL - 84.6 \times ASW \quad (1)$$

Linear function for relevance. Taking readability into account in any information retrieval system leads to redefine the notion of relevance by making it more subdivided than topical relevance. The final score of relevance Sri can be splitted into two components, which are topical relevance — that can be expressed according to the rank of document d such as given by following a classical IR model — and readability relevance (Equation 8).

[3] In a web context, visual cues such as contrasts, fonts, size, spacing or sentence alignments should be taken into account.

According to the linear discriminant function framework described in [12] [4] [2], the relevance score Sri of a document can be expressed by several components (Equation 2)[4]. Equation 3 is the projection of these components on topical relevance and readability relevance.

$$Sri = w_0 + \sum_{i=1}^{n} w_i Rel_i \quad (2)$$
$$Sri = w_0 + w_1 Rel_{topical} + w_2 Rel_{readability} \quad (3)$$

The values of weights w_0, w_1 and w_2 could be estimated by means of some interactive experiments and one can choose $w_0 = 0$. Currently, both readability and topical relevance are normalized scores. If stating an integration factor λ with $0 < \lambda < 1$, $w_1 = \lambda$ and $w_2 = 1 - \lambda$, the global relevance score can be calculated with Equation 4.

$$Sri = (1 - \lambda) \cdot Rel_{topical} + \lambda \cdot Rel_{readability} \quad (4)$$

The topical relevance may be either estimated with the relevance score provided by the information retrieval system or with the rank of document d in the output of the system.

$$Rel_{topical}(d) = Sim(d, q) \quad (5)$$
$$Rel_{topical}(d) = 1 - \tfrac{Rk(d)}{N} \quad (6)$$

Topical and readability relevance scores are given by Equation 5 and 6, with $Sim(d, q)$ the similarity score between document d and query q, $Rk(d)$ the rank of the document in the original output, and N the size of the ranked list. According to the previous equations, the two formula for computing the global relevance Sri of a document d related to a query q are respectively Equation 7 and 8. These remain applicable to any IR system as well they provide some similarity scores or ranking.

$$Sri(d) = (1 - \lambda) \times Sim(d, q) + \lambda \times \frac{L_F(d)}{100} \quad (7)$$
$$Sri(d) = (1 - \lambda) \cdot \left(1 - \frac{Rank(d)}{N}\right) + \lambda \cdot \frac{L_F(d)}{100} \quad (8)$$

with q the query, N the number of retrieved documents.

Experiments. TREC 8 ad-hoc task (`http://trec.nist.gov`) provides 50 queries concerning almost 530 000 documents. The experiments are based on document retrieval results obtained with Lucene search engine by using the title field of the TREC topics.

The impact of the rescoring Formula 8 is studied by evaluating precision variation for 10 and 20 first retrieved documents by the evolution of average

[4] Derived probabilistic approaches such as *probFuse* [7] that models relevant documents could be an another way to combine scores.

readability score of these documents, when λ varies between 0 and 1. The scores obtained with $\lambda = 0$ are only based on relevance. The scores obtained with $\lambda = 1$ only involve readability. The logarithmic evolution of curves (Figure 1) for re-scoring shows that readability can be improved with a necessary precision lost. The results of re-ranking are optimal for $\lambda = 0.2$ as the readability is increased of 10 points without decreasing precision values.

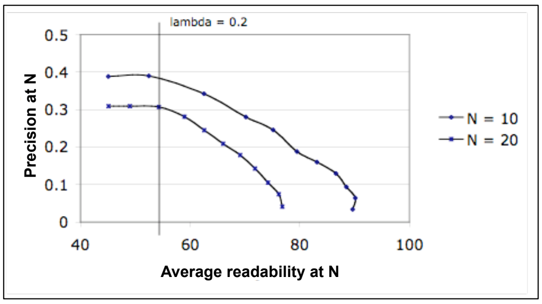


Fig. 1. Precision at N (10 or 20) by average readability score of N first documents obtained by re-ranking (Equation 8) with various λ values, on TREC data.

2 Sentence readability score for dyslexics

In this section, we will investigate how a specific measure can be established in the particular focus of dyslexic readers. Reading implies multiple cognitive processes, and among models that have been proposed, the most referred to is the dual route model [1], partially illustrated in the simulator architecture of Figure 2.

The first way of retrieving pronunciation of words is usually named "direct route" or "lexicosemantic route". It is a visual way of retrieving high frequency words, that are directly associated to their meaning. The second route is called "graphophonological route", "indirect route" or "sublexical route". Is is used to decode pseudo-words or low frequency words, by assembling phonemes following grapheme-to-phoneme correspondences. This second ability is often badly learned by dyslexics as dyslexia involves a reduced ability to discriminate speech-sounds in spoken words [9]. A difficult word can either be a low frequency word or a low grapho-phonemic-consistency (GPC) word. As examples, *school* is a high frequency word but a low GPC word, whereas *anagram* is a rare word but a high GPC word. While previous proposals for estimating readability with classification algorithms uses text graded by teachers for training, we prefer to base the learning directly on readers' performance. The readability measure we will obtain is an estimation of the reading time which we assume to be representative of the readability. We used data initially collected for a psycholinguistics

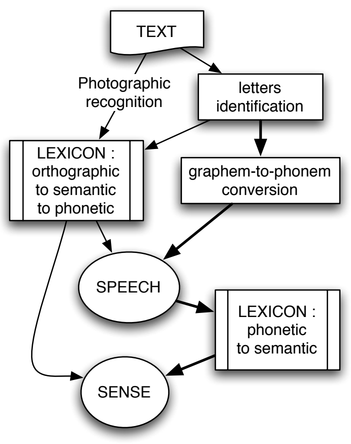


Fig. 2. Dual Route Model representation of human reading [1].

experiment by Speech and Language laboratory of the University of Provence (LPL) as reference corpus.

Nine French dyslexic children had to read 20 twelve-word long sentences screen printed (word after word on keystroke). This allows to record each word reading time. *My grandmother's dog loves playing with my slippers* is a translation of a tested sentences. The effective reading of sentences is checked by a semantic and visual validation task where children have to choose the picture (among two) corresponding to the previously read sentence. This leads to a solution through regression algorithms based on classifiers. We have tested two approaches. Linear regression, might lead to a new formula with a coefficient for each parameter, which is suitable for psychology practitioners. The regressive SVM (*Support Vector Machines*) method has a known ability to manage small data sets, like reading times of sentences at user level (20 samples).

An extended set of parameters dedicated to dyslexics. The features we used come from previous work on language modelling for readability as much as psycholinguistical work on the reading process of dyslexic readers. The set of parameters of a sentence is based on its composing words properties. For each word we use : its length in number of letters, its frequency (according to Manulex, a grade-level lexical database from French elementary-school readers [6] [11]), its part-of-speech tag [10], and its grapheme to phoneme cohesion that is estimated by the number of phonemes divided by the number of letters. A word with a cohesion of 1 will be transparent for the reader as a word with a lower cohesion score will contain complex graphemes (group of characters producing a unique phoneme) or silent characters (frequent in French). This is only an estimation since we did not consider the letters producing several phonemes (such as x). For sentence, each parameter value is either the average, or the number of occurrences of the parameters values of its composing words.

The classifiers are first tested on small datasets consisting of the data of one reader only, *local models*, evaluated by 10-fold cross validation (we used the WEKA toolkit [13] : the classifiers tested are SMOreg with a polynomial linear kernel and LinearRegression with a Greedy attribute selection method). Classifiers are also evaluated on *global models* trained on a dataset including data from all readers. Support vector machines performed better on user specific models than linear regression. The mean absolute error rates obtained with a random assignment indicates a good accuracy of all classifiers, both performing well compared to a standard obtained using a random assignment. Particularly in the use of global models on sentences, a model with linear regression offers the best results in terms of accuracy combining average error rate and correlation coefficient. The features selected by the linear regression algorithm are the number of adverbs, the number of conjunctions and the cohesion estimated by the number of letters divided by the number of phonemes (Formula 10).

Application to information retrieval. Since the data available for training algorithms on reading time are in French only, we applied the previous framework with with ad-hoc French data of CLEF[5]. It provides an evaluation framework of information retrieval systems on some European languages. Specifically, there is a monolingual *ad-hoc* document retrieval task in French that consists in retrieving relevant documents to 60 queries, each one expressed within three levels of details. We focus on the first level which is a set of keywords. The size of the CLEF *ad-hoc* collection is about 130,000 documents.

The readability measure we propose combines our machine learning based estimation of reading time ($Time$) with the French version L_{French} of the *Reading Ease Score* introduced by [5]:

$$L'(d) = \frac{Time(d) + (100 - L_{French}(d))}{2} \tag{9}$$

with:

$$Time(d) = 1.12 \times ADV - 0.69 \times CON + 6.48 \times COH + 15.58 \tag{10}$$

$$L_{French}(d) = 207 - 1.015 \times ASL - 73.6 \times ASW \tag{11}$$

where ADV is the number of adverbs in the sentence, CON is the number of conjunctions, COH is the cohesion, SL is the sentence length, and ASW is the average number of syllables per word.

We performed some experiments on CLEF data (similar to the previously ones on TREC data) with this new readabiliy measure L'_F by reranking the set of retrieved documents (Formula 8). The integration of this readability factor in the scoring shows the readability gain is about 20% with a low precision lost ($< 10\%$) at 10 or at 20 documents. The values of precision are globally lower on French data due to a lower amount of relevant documents per query, but the influence of the integration of the readability factor in the estimation of relevance is very similar.

[5] `http://www.clef-campaign.org`

3 Conclusion and perspectives

We have proposed a new way of estimating relevance that takes some non-informational needs of users into account. This is achieved using a linear function which has the advantages of being simple, efficient, and controllable by the user directly. However, this has to be tested in real-life conditions. The readability measure proposed for dyslexic users could take advantage of additional phonologic parameters such as the frequency of reading rules involved in the reading of a sentence. There is every reason to hope that this work will lead towards an integration of non-informational user needs in the information retrieval process.

References

1. M. Coltheart and K. Rastle. Serial processing in reading aloud : Evidence for dual-route models in reading. *Journal of experimental psychology*, 20:1197–1211, 1994.
2. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification.* John Wiley & Sons, Inc, 2001.
3. R. Flesch. A new readability yardstick. *Journal of applied psychology*, 32:221–233, 1948.
4. J. Gao, H. Qi, X. Xia, and J.-Y. Nie. Linear discriminant model for information retrieval. In *Proceedings of SIGIR'05*, pages 290–297, 2005.
5. L. Kandel and A. Moles. Application de l'indice de flesch à la langue française. *The journal of educationnal research*, 21:283–287, 1958.
6. B. Lété, L. Sprenger-Charolles, and P. Colé. Manulex : A grade-level lexical database from french elementary-school readers. *Behavior Research Methods, Instruments, and Computers*, 36:156–166, 2004.
7. D. Lillis, F. Toolan, R. Collier, and J. Dunnion. Probfuse: a probabilistic approach to data fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, Seattle, Washington, USA, 2006. ACM Press.
8. S. Mizzaro. Relevance : the whole history. *Journal of the American Society for Information Science*, 48(9):810–832, 1997.
9. F. Ramus, S. Rosen, S. Dakin, B. Day, C. J.M., S. White, and F. U. Theories of developmental dyslexia : insights from a multiple case study of dyslexics adults. *Brain*, 126:841–865, 2003.
10. H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK, 1994.
11. M. Stuart, M. Dixon, J. Masterson, and B. Gray. Children's early reading vocabulary: Description and word frequency lists. *British Journal of Educational Psychology*, 74(3):585–598, 2003.
12. C. C. Vogt and G. W. Cottrell. Fusion via a linear combination of scores. *Information Retrieval*, 1(3):151–173, 1999.
13. I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, San Francisco, 1999.

Improving Information Access by Relevance and Topical Feedback

Rianne Kaptein[1] and Jaap Kamps[1,2]

[1] Archives and Information Studies, Faculty of Humanities, University of Amsterdam
[2] ISLA, Faculty of Science, University of Amsterdam

Abstract. One of the main bottle-necks in providing more effective information access is the poverty of the query end. With an average query length of about two terms, users provide only a highly ambiguous statement of the, often complex, underlying information need. Implicit and explicit feedback can provide us with additional information that can help disambiguate the query and provide more focused search results. We investigate the effects of using different types of feedback. Retrieval results of pseudo-relevance, explicit relevance and topical feedback are compared. Although on average explicit relevance feedback in combination with pseudo relevance feedback works best, for individual queries results are unpredictable. There is a large potential for improvement if we can predict which type of feedback will perform best for a query. Since we are dealing with feedback potentially provided by users standard evaluation measures are not sufficient to evaluate feedback techniques, and the quality of user interaction should also be taken into account.

1 Introduction

Relevance feedback is a commonly used feedback technique to improve search results [5, 6]. Documents that are considered relevant, either because the documents are top-ranked, or because the user marked them as relevant, are exploited in a second iteration of the retrieval process. Another, not so common, feedback technique we discuss in this paper is topical feedback. Instead of using the (presumed) relevant documents, topical feedback uses topic categories considered relevant to the query.

A known drawback of using feedback methods is that while the results on average improve, there are large differences between individual queries. Pseudo-relevance feedback works best for queries where the initial retrieval results are already good. Explicit feedback and topical feedback produce more unpredictable results. While the risks of most feedback techniques can be mitigated by putting more weight on the original query (run) and less on the feedback, this also decreases the potential positive effects of the feedback. If we could predict the effect of using the different feedback techniques on individual queries we can improve retrieval performance and minimize the effort needed from the user. In this paper we analyze the effects of using pseudo-relevance feedback, explicit feedback and topical context by looking at different statistics and at the individual queries.

While explicit relevance feedback does produce the best results when looking at standard evaluation measures, for practical applications it might not be the most attractive option. Explicit relevance feedback can consist of marking one or more documents relevant, or by providing example relevant documents. Users can judge either top-ranked documents or some documents predicted to be the most informative for feedback purposes. Only when the supposed most informative documents are presented for judging to the user, the system is involved.

There are some other forms of feedback that are less static, i.e. the required input from the user depends on the query and the system supports the user by providing intelligent suggestions. For example, Google's spelling suggestions detect possible spelling mistakes; when your query is "relevence", on top of the result list Google asks: "Did you mean *relevance*". Or, when we want to use topical feedback, questions like "Do you want to focus on sports?" or "Are you looking for a person's home page?" can be asked. When these follow-up questions are relevant to the query and easy to answer these kinds of interaction might be more appealing to users than simply marking relevant documents.

Evaluation of feedback approaches is complicated because the interaction with the system is dynamic, and performance depends on the feedback of users. Standard TREC evaluation measures are static and do not have a natural way to incorporate feedback [4]. Instead, feedback documents can be removed from the result ranking, creating a so-called residual ranking, or the feedback documents can be frozen on their position in the initial ranking [1]. Since the standard evaluation measures on their own are not satisfactory, in this paper also we look at some other factors that influence the user's experience.

2 Feedback

2.1 Relevance Feedback

A widely used relevance feedback model was introduced by Lavrenko and Croft [3]. This so-called relevance model provides a formal method to determine the probability $P(w|R)$ of observing a word w in the documents relevant to a particular query. They are using the top-ranked documents retrieved by the query as implicit feedback, but the same model can be used when explicit relevance judgments are available. The method is a massive query expansion technique where the original query is completely replaced with a distribution over the entire vocabulary of the feedback documents. Their results show significant improvements in performance with increases in MAP from 10 to 30% on TREC datasets. However, the gain in performance tends to be foremost the improvement of topics that already did well in the initial run. From the user's point of view, returning worse results on weak queries might not weigh up to the benefit of returning better results on already well performing queries.

Explicit feedback does require an effort from the user. One or more documents have to be marked either relevant or non-relevant. A known relevant document is far from a panacea. The search results after feedback will be biased towards documents similar to the documents marked as relevant. If the relevant document does not cover all aspects of the topic, documents covering other aspects of the topic will be ranked too low. Explicit relevance feedback can also exploit non-relevant documents. While the combination of relevant and non-relevant documents can be beneficial, if only non-relevant documents are available performance will marginally improve at best [7], and the user's effort of judging the documents may be ineffective.

2.2 Topical Feedback

Another method for feedback that we take into consideration is topical feedback. Topical feedback uses topic categories that are considered relevant to the query. Topic categories can be chosen from specialized or general topic directories such as DMOZ, Yahoo! Directory or Wikipedia. The categories can be assigned explicitly by the user, or derived implicitly by applying text categorization techniques to either the query or the top-ranked documents. Another possibility is to show some suggested categories that depend on the query, or to show questions like "Do you want to focus on the twentieth century?" We will use DMOZ as our topic directory, and look at feedback in the form of a DMOZ category that is relevant to the query. We assume that all web sites in the chosen DMOZ category, and all of its direct subcategories are relevant to the query. The feedback model is built from the text on these web sites. From a user study we conducted, we can conclude DMOZ categories are suitable to categorize query topics, and users think it is easy to categorize query topics.

Advantages of topical feedback are that the sites in the DMOZ directory are of high quality and selected by human editors, thus providing us with potentially good feedback documents. A disadvantage of using a topic directory is that not for every query there is an applicable topic category. The DMOZ directory is very general however, and if there is no topic category that applies to the query, there is usually a higher level category under which the query can be placed. Effectively communicating the category to the user is essential, and the topical feedback will by design generate clear intelligible labels (in constrast with, for example, clustering techniques [2]).

Pseudo-relevance feedback can easily be combined with explicit relevance feedback and topical feedback. In our case, after the initial run we first apply explicit relevance or topical feedback, and then apply pseudo-relevance feedback in a third iteration.

3 Experiments

In order to explore the effects and performance of relevance and topical feedback we have conducted experiments on the TREC 2008 Relevance Feedback Track data. First, to compare pseudo-relevance feedback with explicit relevance feedback, we applied both of them and their combination. As explicit relevance feedback, we use the first document of the initial retrieval results that is judged

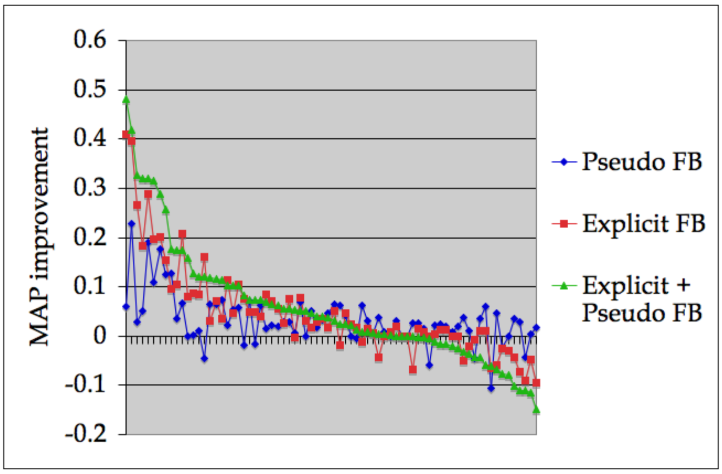


Fig. 1. Absolute difference in Map per query.

as relevant. For evaluation we remove the known relevant document from the results.

Figure 1 gives absolute improvement in MAP per query ordered by MAP improvement of explicit relevance feedback combined with pseudo-relevance feedback. On average, explicit feedback combined with pseudo-relevance feedback performs best with a MAP of 0.3300. Pseudo-relevance feedback achieves a MAP of 0.3044, explicit feedback a MAP of 0.3198. Looking at precision at 10, similar performance improvements can be seen. Comparing relevance feedback to the combination with pseudo-feedback the effects of the combination are larger, on the positive and the negative side.

Another factor in explicit feedback is the user's relevance judgment. The TREC style relevance judgments are on a three-way scale of "not relevant," "relevant," and "highly relevant." However, when looking at our TREC Terabyte topics average MAP improvement of feedback based on highly relevant documents is not higher than MAP improvement of feedback based on relevant documents. So, a document that is judged as highly relevant is not automatically a better document for feedback.

Secondly, we apply topical feedback, obtained by a user study, to 25 of the topics allowing us to explore the effects using the different kinds of feedback. In the user study test persons assign topical categories to query topics by selecting categories from a list and by searching in the DMOZ directory. The list contains categories that are produced by automatic topic categorization using the query, top-ranked documents, and a category title match with the query. Some additional questions about confidence and fit of the category also have to be answered. Each query is categorized by at least two test persons. We select one category that applies best according to the test persons for each query, and use this category as topical feedback.

Table 1. Number of queries for which a feedback method gives the best results.

Model	Baseline		Relevance FB		Topical FB	
Additional blind FB	no	yes	no	yes	no	yes
# Topics with best MAP	1	5	3	8	6	2
# Topics with best P10	4	7	9	12	4	10

We can now compare the results of implicit and explicit relevance feedback and topical feedback. One of the first things to be noticed is that there is a lot of variation in what kind of feedback works. As can be seen in Table 1, each of the retrieval techniques works best for some of the queries. In case multiple retrieval techniques have the same best P10, they are all counted as best. It is hard to predict however which kind of feedback will work best on a particular query. If we would be able to perfectly predict which feedback should be used, MAP would be 0.3917—an improvement of 42.3% over the baseline! This almost doubles the improvement that is achieved with the best single feedback technique.

We do find indicators to predict whether topical feedback technique will improve over the baseline results or not. It turns out the user provided factors "confidence" and the "fit of the category" (based on the user study) do not have a strong correlation to performance improvement. The factors "fraction of query terms in category title" and "fraction of query terms in top ranked terms" do have a strong correlation with performance improvements. When the weight of the feedback is adjusted according to the query terms in the category title or the top-ranked terms, we see an improvement in the results. For pseudo-relevance feedback and explicit feedback there is no such correlation between the fraction of query terms in top ranked terms of the feedback model and the performance improvement. Since the feedback is based on top ranked documents, the query terms always occur frequently in these documents.

There is also a positive side to the fact that the fit of the category does not correlate much to performance improvement. Sometimes categories that are clearly broader than the query, do lead to improvements. The queries "handwriting recognition" and "Hidden Markov Model HMM" both improve considerably when the topical model of category "Computers-Artificial Intelligence-Machine Learning" is applied. So it seems categories on more general levels than the specific queries are useful and one topical model can be beneficial to multiple queries.

4 Conclusion

In this paper we have analyzed the effects of different types of feedback. We found there is not one type of feedback that works best for all or the majority of queries. There are not only large differences between the performance of one type of feedback on different queries, but also between the performance of different types of feedback on the same query. If we would be able to use the full potential of feedback by correctly choosing the most beneficial type of feedback for each query, large performance improvements can be achieved. While user studies are needed to see what types of feedback users prefer, we do not expect users to be able to choose the best type of feedback considering standard evaluation measures, this should be a task for the system.

Acknowledgments Rianne Kaptein was supported by the Netherlands Organization for Scientific Research (NWO) grant # 612.066.513. Jaap Kamps was supported by NWO under grants # 612.066.302, 612.066.513, 639.072.601, and 640.001.501), and by the E.U.'s 6th FP for RTD (project MultiMATCH contract IST-033104).

Bibliography

[1] Y. K. Chang, C. Cirillo, and J. Razon. Evaluation of feedback retrieval using modified freezing, residual collection and test and control groups. In G. Salton, editor, *The SMART retrieval system - experiments in automatic document processing*, pages 355–370, 1971.

[2] M. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1996.

[3] V. Lavrenko and W. B. Croft. Relevance-based language models. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2001.

[4] S. Robertson, S. Walker, and M. Beaulieu. Experimentation as a way of life: Okapi at TREC. *Information Processing & Management*, 36:95–108, 2000.

[5] J. Rocchio, Jr. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice-Hall Series in Automatic Computation, chapter 14, pages 313–323. Prentice-Hall, Englewood Cliffs NJ, 1971.

[6] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41:288–297, 1990.

[7] X. Wang, H. Fang, and C. Zhai. A study of methods for negative relevance feedback. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2008.

Combining Cognitive and System-Oriented Approaches for Designing IR User Interfaces

Norbert Fuhr, Matthias Jordan, and Ingo Frommholz

University of Duisburg-Essen, Duisburg, Germany
{norbert.fuhr|matthias.jordan|ingo.frommholz}@uni-due.de

Abstract. In this paper, we present a new generic framework for designing user interfaces for interactive IR. Based on the distinction between the logical, layout and content view on documents, we regard IR as a combination of selection, projection, organization and visualization on these aspects. These transformations should be configured according to the information seeking strategy currently chosen by the user.

1 Introduction

Although interactive information retrieval (IIR) is a commodity nowadays, there has been little research on the systematic design of appropriate user interfaces. The current 'standard' is the single search box as offered by most Web search engines, in combination with a linear result list. On the other hand, research in the area of cognitive approaches has led to the formulation of a number of cognitive IR models, which explicitly state certain user needs with regard to the IR system involved. However, the problem of mapping these needs onto system features has been addressed by only a few researchers so far.

In this paper, after briefly discussing the notion of information seeking strategies, we present a new system-oriented framework for designing IIR interfaces, and then we show how this framework can be combined with the cognitive approach.

2 Information Seeking Strategies

We follow the notion of information-seeking strategies (ISS) as proposed by Belkin et al. (see, e.g., [4] for an overview); they propose four dimensions for classifying ISSs: the method, mode and goal of seeking, and the resource used for the process. Regarding the method of seeking, *scanning* refers to the process of looking sequentially at each item from a certain set, whereas *searching* stands for a more goal-directed search. If the user can specify the wanted items, retrieval mode is *specification*, otherwise the mode is *recognition*. The goal of seeking can be either *learning* or *selecting*. Finally, users can interact either with the *information* objects themselves or with *meta-information*. An important distinction of this scheme is that between method of interaction and mode of retrieval:

There are cases where a user gives a good specification of a wanted item, but the system does not support searching according to these criteria, and so she has to scan the object in question. (e.g. the user remembers a melody from her music collection, but the system does not support music retrieval). Conversely, the user may search for a known item, but cannot give a good specification, and thus can only recognize the object when she sees it (e.g. the user remembers a book on text mining with a remarkable cover which she wants to retrieve, but she will only recognize it when she looks at each of the books on 'text mining').

3 The LACOSTIR Model

In our current project LACOSTIR, we focus on layout, content and structure of documents and transformations thereof as part of the interactive IR process. The goal is the development of a framework for designing user interfaces for IIR. Here we briefly describe the major concepts of our approach.

In contrast to the distinction between information and meta-information as proposed by Belkin et al., we think that content, structure and layout are more important document facets when we are dealing with information seeking [2]. Classical IR methods usually focus on the *content view*. The *structure view* reflects the (logical) document structure and the data in specific elements thereof — this view lies within the focus of recent research, for example in the context of XML retrieval. A third view is the *layout view* on documents which is concerned with the display of documents on a medium — which is e.g. important for recognizing an item seen before.

Starting from this distinction, we have developed a system-oriented model of interactive retrieval, which is depicted in Figure 1.

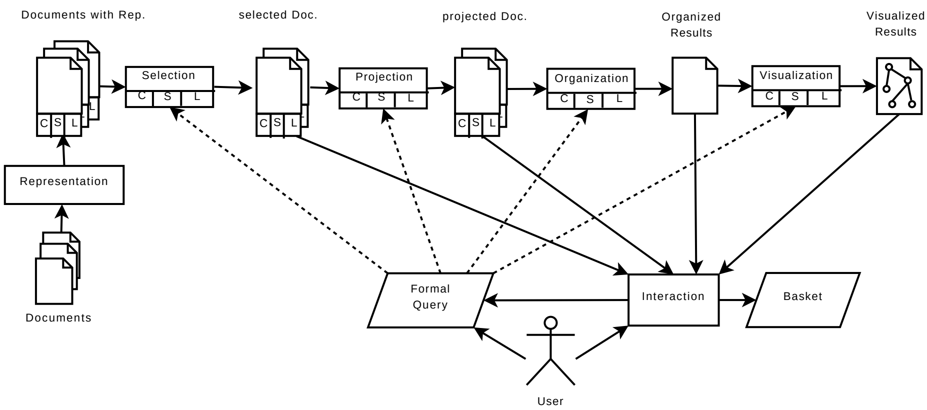


Fig. 1. System-oriented model on information visualization

(Automatic) indexing methods transfer the collection into suitable document representations, which may contain structure and layout elements besides the content. A *selection* operator being applied to these representations selects the documents matching the query. A *projection* operator extracts features from resulting documents matching the query in order to create suitable surrogates; these can contain, for instance, a summary (content), title and author (structure), or a thumbnail of the title page (layout). In the next step, an *organization* operator structures the projected result, e.g. a linear list, a tree or graph, or a list of sets (in the case of clustering). Finally, *visualization* displays this organized result in a certain way (e.g., there are many alternative methods for tree visualization). These four operators (selection, projection, organization, visualization) might be modified by the user during the seeking process; new queries might be posed (resulting in a new selection), but also the projection and organization operators might be changed, or different visualizations applied to the same result.

In most of today's IR systems, the user can only modify the selection operator, whereas the three other operators are fixed. Although this approach might ease the handling of such the systems (especially for beginners), for many ISSs, this results in a very poor system support.

4 Supporting Information Seeking Strategies

Information seeking strategies, as they are discussed in Section 2, can be regarded as parts of an information seeking episode; during an episode, users move from one ISS to another. Different ISSs should be supported by different techniques [1]. Furthermore, search must be viewed as an interactive process. Additionally, user's actions give some feedback about her needs, which should be considered by the system.

This raises the main question of our research: how can we support different ISSs using the document aspects and transformations as proposed in Section 3?

As an example, let us return to the text mining book with the remarkable cover. Assuming that the user remembers exactly that the title contained the words 'text mining', this would be mapped onto a *selection* referring to the *logical* structure; alternatively, if she only knew that it was about this topic, this would lead to a *content selection*. In order to recognize the 'remarkable cover', *projection* should include the *layout* (mapping it to a thumbnail of the cover) — possibly along with the essential bibliographic data. *Organization* of the result set could produce a single list (ordered by some logical criterion, like e.g. publication year or author name); alternatively, the system could cluster the cover thumbnails. Finally, these organized results should be *visualized* in a meaningful way.

In order to aid the user in finding relevant information, an IR system should be able to flexibly react on different tasks, situations and contexts by supporting different ISSs. The challenge is to relate system features to search strategies, i.e. which support technique should be used for a given strategy, which projection, which organization? We think that most of today's systems are restricted to content searches, with specification as seeking mode and searching as retrieval mode. For this case, the projections (typically a combination of some logical elements and a query-biased summary) applied and the standard organization in form of a ranked list are appropriate. However, when other ISSs have to be involved, especially with scanning instead of searching and/or recognition instead of specification, then current systems are not flexible enough to offer appropriate support.

Given a system that implements the major features of our framework, we have the problem of choosing the right transformations for the ISS currently applied by the user. Thus, the system needs more information from the user, in order to provide the appropriate transformations. Especially for recognition and scanning, the distinction between layout, content and structure of documents is important, depending on which aspect the user is currently focusing on. Thus, the typical 'search box' interface of Web search engines will not be sufficient for providing appropriate support. Possible alternatives include complex forms, wizards and query by example.

Such complex interfaces bear the risk of exposing the user to a high cognitive load. IR systems should thus provide one single interface with multiple interaction functionalities that give maximum support for every ISS but still adhere strictly to usability standards. The selection of the interaction functionalities offered may also vary between different devices and available input and output modalities.

5 Conclusion and Outlook

In this paper, we have described a new approach for designing interactive IR systems. By differentiating between logical, layout and content aspects of documents, and by introducing the notions of selection, projection, organization and visualization, we have laid out the design space for such systems. We have shown that different information search strategies require different parameter settings for these transformations. Currently, we are working at the development of an IIR system implementing the major concepts of our framework. Furthermore, we are investigating methods for assigning configuration of these transformations to ISSs. In addition, we want to combine this approach with an interactive information visualization cycle [3] and continue the evaluation of adaptive systems (which has started with considering only content in [4]) by adding the aspects of structure and layout.

References

1. N.J Belkin. Intelligent information retrieval: Whose intelligence? In *Proceedings of the ISI 1996*, pages 25–31. Universitätsverlag Konstanz, 1996.
2. N. Fuhr. Konzepte zur Gestaltung zukünftiger Information-Retrieval-Systeme. In *Experimentelles und praktisches Information Retrieval*, pages 59–75. Universitätsverlag Konstanz, 1992.

3. P. Landwich, M. Hemmje, and N. Fuhr. Ansatz zu einem konzeptionellen Modell für interaktive Information-Retrieval-Systeme mit Unterstützung von Informationsvisualisierung. In *Proceedings of the ISI 2007*, pages 327–332, 2007.
4. Xiaojun Yuan and Nicholas J. Belkin. Supporting multiple information-seeking strategies in a single system framework. In *Proceedings of the SIGIR 2007*, pages 247–254, New York, NY, USA, 2007. ACM.

Information Search Stage Based Representations

Abdigani Diriye*[1], Ann Blandford [1] and Anastasios Tombros [2]

[1]UCL Interaction Centre, UCL, London, United Kingdom, WC1E 6BT
[2]Department of Computer Science Queen Mary, University of London, London, United Kingdom E1 4NS
{a.diriye, a.blandford}@ucl.ac.uk
{tassos}@dcs.qmul.ac.uk

Abstract. In this position paper we discuss the relationship between user search behaviours and information retrieval. Our preliminary findings suggest a relationship between users' search stage, search strategies, and the information sought. We conclude by proposing work that tries to provide user interaction and document representations that are associated with information search stages

Keywords: User interface, document representation, search strategies, search tasks.

1 Introduction

Information has become a readily available and ubiquitous resource in our daily lives. Developments within information retrieval (IR) have spawned more effective and efficient systems that allow precise retrieval of documents pertaining to our requests. Considerable efforts have been invested in improving the system-centred aspects of IR, and it is only in the last decade that the focus has shifted to issues concerning user interaction and information use. IR practitioners have come to the conclusion that presenting a set of search results relating to a user query is not an entirely adequate solution to enable the user to resolve their information need. Given the inherent inadequacies in the design of IR systems [2], users have developed coping mechanisms, such as multiple queries and browsing strategies, to enable them to overcome the system's shortcomings [13]. These strategies are however taxing, cognitively demanding and can result in poor user experience.

Seminal work by Ingwersen [7], Kuhlthau [8], Dervin [6] and Blandford et *al.* [4], [5] have contributed to this gradual paradigmatic shift in the focus of the use and evaluation of IR systems. Their work has helped create awareness of the importance of user behaviour, and usability issues in the use of IR systems.

* Contact author.

2 Information Search Behaviours

Despite the theoretical progress made in IR, the classic IR model is still prominent within the design of modern systems. This restrictive and oversimplified representation of the IR process places too much emphasis on the retrieval of documents and implicitly assumes a static information need [2]. More *'real-life'* models of searching have been proposed; of interest are the models proposed by Bates [2] and Pirolli and Card [10]. Bates's berry-picking model entails that the user satisfies their information need by selecting information from several documents, rather than from a single source. The model implies that users' information needs are non-static and evolve throughout the search [2].

Pirolli and Card's Information Foraging Theory complements Bates's model by trying to explain users' adaptation of searching behaviour to problems and constraints in their environment while searching for information. The process of information foraging involves the navigation through information patches depending on information scents. Information scents inform the user of the value and accessibility of an information patch, and based on this the user formulates their next move during their search [10]. When users are foraging within a specific document, certain document criteria and features are used to assess their relevance. Work by Barry [1] has identified document features affecting user relevance judgement. Tombros et *al.* [11] conducted further work that analysed the variability of document features and tasks when users were assessing relevance. The chief finding from their study concerned the features used to assess relevance, and the dependence on the type of task and search stage [11]. These findings support postulations by Vakkari [14], [15], Belkin [3] and Navarro-Prieto et *al.* [9]. Vakkari [14] argues that the information needed and user search tactics and strategies are linked to the task stage. Vakkari also notes that Wersig and Windel [12], and Belkin [3] have all speculated that different problem stages should be addressed with different retrieval strategies and support [15]. Vakkari summarizes his review of this phenomenon and states:

"…various aspects of information searching are deeply rooted in the process of task performance. It seems that the information needed, and search tactics used (including the choice of terms and operators, as well as relevance judgments) are systematically linked to the stage of task performance." – [15, p452].

This apparent dependency between users' search stage and document features should be harnessed by the IR system to provide more effective system feedback and document representations appropriate to the user's information search stage.

3 Preliminary Observational Findings

To understand the factors affecting user search behaviours, a study was conducted involving 18 participants undertaking three different search tasks (one exploratory, and two known-item tasks with varying degrees of difficulty) on an experimental IR system. The participants were asked to note down their understanding of the search task before and after the search, and were then asked to complete a post-hoc questionnaire. During the search, participants were asked to perform a think-aloud protocol, and their interactions with the system were recorded using screen-capturing software. In analyzing the data, we have recorded the observations and performed 'open coding' using the methods in Grounded Theory.

3.1 Search Behaviour

Based on our observational data, users typically began their search with an overview of the topics and concepts involved in the search task. This strategy was employed by users to acquire some knowledge of the information available before they *'dived in'* and carried out further, more detailed searches, for example:

"I'm just gonna [sic] start off with a quick and dirty search on racial profiling just to familiarize myself" – P11

"I'm trying to look for a general article or document." – P9

During the initial stages of the search, users' relevance criteria were largely based on document properties (e.g. date, source, and level of detail), as the search progressed, the focus changed to specific lower level document aspects (e.g. presence/absence of a topic within the document). An example of this was participant 18; at the start of the search the participant comments:

"The top [document] seems good according to Google, and it's titled definition which is a good place to start"

Further along in their search, the participant remarked:

"This is the 3rd in the list, and this seems to give me both sides of the story, but the other two only gave me one side."

Whilst users were engaged in obtaining an overview of the information space, a scanning search strategy was used to obtain relevant information. When users' understanding of their information need had developed, their search strategy changed from basic scanning motion to analytical exploration, and more thorough browsing of specific documents. Based on the observational data, the search behaviours manifested by the users, and the relevance criteria employed, are specific to the information search stages. One such incident involved participant 16:

"I'm gonna [sic] click on their site to see whether I can get some kind of overview [on the topic], I'm just scanning to see whether I can find anything."

Later on in the search, when the participant selects a relevant document he comments:

"I'm just reading down [the document] to see whether there is something."

3.2 Search Task and Document Interaction

The data gathered supports our claim that user search behaviours are not only associated with specific information search stages, but also influenced by the search task. For example, search task 1 (racial profiling) involved users collating and analysing relevant documents to formulate an answer. Their information search behaviour was driven by a need to develop knowledge of the topic, and search task specific behaviours were exhibited that resulted in the users filtering the documents by search topics, level of detail and document properties. Unlike task 1, tasks 2 (human smuggling) and 3 (wrongful conviction) were known-item searches, and involved identifying documents that discussed certain topics. The search tactics for tasks 2 and 3 were focused primarily around query formulation, and the focus of these searches were conceptual, whereas in task 1, the focus was more low-level, and concerned specific document details.

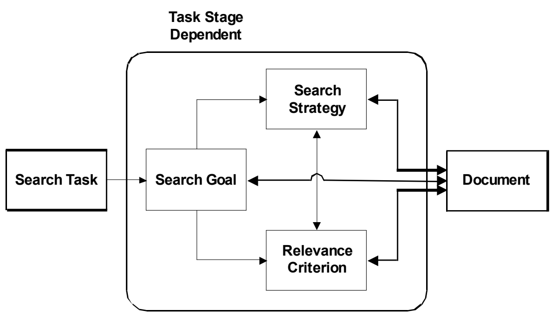


Figure 1: Augmented model of interactivity.

The preliminary findings from our study corroborate work by Vakkari [14] and Navarro-Prieto et *al*. [9] on task-based searching. In Figure 1, we illustrate our findings based on Navarro-Prieto et *al.*'s original model of interactivity [9]. Our model shows the origins of the search goal, and the affective links between the search strategy, search goal and relevance criterion.

In the context of a search task, when the user is undertaking a task, a search goal would be derived that would affect the search strategy and tactics (e.g. broadening/refinement of query, scanning, browsing) employed, and relevance criteria (e.g. document type, source, topic, length, date, etc.) used. These entities would dictate the documents selected and the method of interaction. Based on the information the user extracts from the document, this would be used to inform the formulation and selection of subsequent search strategies, relevance criteria and search goals. With the progression of the search task, the search goal, relevance criteria and search strategy are dynamic entities in this model, and so evolve with the task and information search stage.

4 Future Work

This position paper sets out to instigate further work and discussions into information search stage based representations. Existing work supports the position that users can benefit from information dependent on their information search stage. The preliminary findings reported, and the literature reviewed suggests that search tasks, search strategies and relevance criteria are good indicators of the user's search stage. By designing IR systems to incorporate these factors, we can better tailor the interaction between the user and system; enabling appropriate search results, document representations and interaction to be provided. However, further work into understanding how and when to use adaptive techniques within representations and user interaction is needed.

5 Acknowledgement

The authors are indebted to the participants who took part in the experiments. This research is supported by an ESPRC DTA studentship.

References

[1] Barry, C. L. (1998). Document representations and clues to document relevance. Journal of the American Society for Information Science, 49 (14), 1293 1303.

[2] Bates, M. (1989). The design of browsing and berry-picking techniques for the online search interface. Online Review, 13 (5), 407-424.

[3] Belkin, N. J. (1980). Anomalous states of knowledge as a basis for information retrieval. Canadian Journal of Information Science, 5, 133-143.

[4] Blandford, A., Keith, S., Connell, I., Edwards, H. (2004). Analytical usability evaluation for digital libraries: a case study, Digital Libraries. Proceedings of the 2004 Joint ACM/2004, ieeexplore.ieee.org

[5] Blandford, A., Adams, A., Attfields, S., Buchanan G., Gow, J., Makri, S., Rimmer, J., and Warwick, C. (2008). Pret a rapporter: evaluating digital libraries alone and in context. In Information Processing Management, volume 44, pages 4–21.

[6] Dervin, B. (1983). An overview of Sense-Making research: Concepts, methods, and results to date. Paper presented at the annual meeting of the International Communication Association, Dallas, TX.

[7] Ingwersen, P. (1992) Information retrieval interaction. London: Taylor Graham.

[8] Kuhlthau, C. (1993). Principle for uncertainty for information seeking. Journal of Documentation, 49 (4), 339-355.

[9] Navarro-Prieto, R., Scaife, M., & Rogers, Y. (1999). Cognitive strategies in Web searching. Retrieved August 29, 2008 from: http://zing.ncsl.nist.gov/hfweb/proceedings/navarro-prieto/index.html.

[10] Pirolli, P. and Card, S.K. (1999) Information foraging. Psychological Review 106(4) (pp. 643-675).

[11] Tombros, A., Ruthven, I., Jose, J.M. (2005). How Users Assess Web Pages for Information Seeking. Journal of the American Society for Information Science and Technology. 56(4): p. 327-344.

[12] Wersig, G., & Windel, G. (1985). Information science needs a theory of information action. Social Science Information Studies, 5, 11-23.

[13] Wilson, M., schraefel, m.c., White, R. (2007) Evaluating advanced interfaces using established information-seeking models. Technical Report, School of Electronics and Computer Science, University of Southampton. http://eprints.ecs.soton.ac.uk/13

[14] Vakkari, P. (2001). A theory of the task-based information retrieval process: A summary and generalisation of a longitudinal study. Journal of Documentation, 57, 44-60.

[15] Vakkari, P. (2003). Task-based information searching. Annual Review of Information Science and Technology, 37, 413-464.

Putting Digital Government in Context

Luanne Freund

School of Library, Archival and Information Studies
University of British Columbia
luanne.freund@ubc.ca

Abstract. This position paper outlines a framework for a contextual model of the digital government domain to support more effective information retrieval..

Keywords: contextual information retrieval, digital government, tasks, genres

1 Introduction

Digital government – the delivery of government information and services via the Internet - has become a commonplace and widespread phenomenon. In countries such as Canada, which have implemented digital government initiatives, a large and rapidly growing quantity of information is now available online [1]. Much of this information is of high quality, in the sense that it is authoritative and current, and covers a wide range of information needs. Public use of government information on the Web is growing. Approximately 98 million (about 32%) Americans used government websites in 2004, an increase from 66 million in 2002 [2, 3]. Similarly, about one third of Canadians (8.2 million) used the Internet to access government information from their homes in 2005. However, in the face of digital information overload the average citizen may have great difficulty finding what they need [4]. E-government portals have been implemented to reduce this complexity and facilitate access to information and services, but research shows that more people use general search engines (37%) than portal sites (8%) to reach government information [2]. Given the social and political importance of public access to government information, more sophisticated search tools and approaches are sorely needed.

This position paper will outline a framework for a contextual model of the digital government domain to support more effective information retrieval. The framework builds upon previous work on contextual search in a corporate domain [5], and focuses upon the case of Canadian digital government. The paper proposes some directions for future research.

2 Modeling Context for Digital Government

Simple contextual model

Interactive information retrieval can be represented as a set of relationships between human actors, information objects and systems. Putting this interaction in context, it is possible to identify a great number of variables and factors that influence the quality of the process and of the outcomes. The approach taken in this work builds upon a study of workplace searching that identified and characterized a small subset of these variables that can be used to build a simple contextual model: domain, task and genre [6]. This approach suggests that within domains representing some shared body of knowledge and practice, patterns emerge with respect to common activities (tasks), common forms of communication (genres), and the relationships between them. (Figure 1). These task-genre relationships are based on alignment of purpose or function, as illustrated by the relationship between procedural tasks and step-by-step instructions or the task of purchasing something and a catalogue. This relationship has little to with the topical relevance of an information object and much to do with its usefulness: the quality of being well-suited and directly applicable to the task at hand. This simple contextual model suggests that within domains, the usefulness of an information object can be assessed based on patterns of relationships between sets of common tasks and common genres, and this usefulness measure can be combined with traditional measures of topical relevance for purposes of ranking [7].

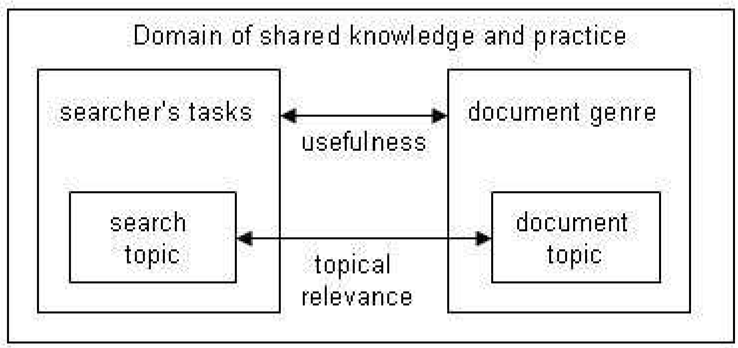


Figure 1: Simple contextual model for interactive IR

The case of digital government

The domain of digital government offers a potentially valuable test arena for this contextual approach to information retrieval. In Canada, as in many other countries, digital government information exists in a relatively well-defined and structured web-based information environment, which is easily identifiable through domain name conventions. It consists primarily of public, rather than proprietary or access-controlled information, and the user community is a broad and easily accessible target population for research. Finding government information is necessary for citizens in the course of key life events, such as paying taxes and getting married, so searching is likely to be a serious, goal-driven activity, similar to that observed in workplace and professional domains. However, since most people only search for government information occasionally, for personal matters, they are unlikely to be expert searchers in this domain, which means that the potential benefit of tools to support more effective searching is high. Finally, like other national governments, the Canadian government has a clear mandate to manage and provide public access to information, which means that certain formatting and meta-data standards are in place that can be exploited by search systems.

Some preliminary work has been done on the relationship between tasks and genres in e-government. Haraldsen et al.[8] conducted a use-case study of how genre can contribute to the definition of systems requirements in the design of a life-event-based e-government portal. The life-events approach is analogous to task-based approaches in its emphasis on the events and activities in peoples' lives as drivers for government information seeking and use: birth, marriage, graduation, death, etc. Life-events portals are user-centered access points to e-government information that reach across government departments, cutting through bureaucracy in order to meet the needs of the citizen-user [9, 10]. Haraldsen studied the genres of communication used in exchanges between the government and the public in the context of the life event of "building a house" and then used the genres to derive system requirements for a life-events portal [8]. The study concluded that genre analysis can help bridge the gap between technological and social aspects of designing e-government systems and create greater understanding of the information system on both sides.

Current and Future Research

We are currently engaged in research to examine the relationship between task and genre in the digital government domain. The following broad research questions guide the research

- What tools and strategies are used by members of the public when searching for online government information?
- Which genres are recognizable by information producers (government employees) and which by consumers (the public)?
- To what extent does the public use genre as a means of assessing and selecting government information?
- What patterns of association exist among tasks and genres in this domain?

We have developed a set of 20 simulated work task scenarios to represent a range of everyday information seeking needs. The scenarios are balanced by government sub-domain (10 each related to health and environment) and five types of information tasks: learning, fact-finding, deciding, doing, and problem-solving. Two sample scenarios are presented in Table 1.

We are currently examining how the tasks represented in these scenarios relate to Genre metadata in use as part of the Canadian government Metadata standard. The “Type” element, defined as the “nature or genre” of a resource, is an optional element, and is used together with a Type taxonomy consisting of 50 terms [11]. The taxonomy contains a broad range of Types, including common Web and print genres (FAQ, report), government document genres (statistics, standards) and many more specific genres, related to particular disciplines and government services (licences, agreements). Based on some preliminary analysis, we estimate that about 1/5 of Canadian government documents in HTML format contain Type metadata [12]. While this genre tagging is neither comprehensive nor highly consistent, it does represent a large, publicly available, manually tagged genre corpus, which is quite rare.

Table 1: Sample Search Scenarios

1. You have a friend with a 2-month-old baby boy who suffers from severe colic. Your friend has started giving him gripe water, which she buys across the border in the United States where it is available. Thinking that there may be some reason it is not available in Canada, you decide to do some checking to see if gripe water is considered to be safe for infants. Search for official health and safety information on gripe water. (Sub-Domain = Health; Information Task = Fact-Finding)
2. You have just started a new job, which you enjoy, but the level of noise in the workplace is much higher than you are used to. You have a hard time concentrating and often have a headache at the end of the day. You have spoken with your supervisor, who thinks it is a personal rather than a workplace issue. You need to find a solution to the problem one way or the other for your own health and well-being. Search for information that will help you better understand and resolve this problem. (Sub-Domain = Health; Information Task = Problem Solving)

Later this year, we will conduct two experimental user studies to better understand how people search for online government information, the extent to which genre is recognized and used as a means of assessing and selecting government information, and the nature of the task-genre relationship in this domain. The first study will involve 24 participants using the Internet to search for government information in response to the task scenarios. Participants will save relevant documents and following each session will explain why each document was selected. All search sessions will be recorded using logging and screen capture software. In the second study, 40 participants will assess the relevance and identify the genre of a set of 20 documents from the Government of Canada domain, based on the documents selected in the first study, for each of 5 scenarios. The genre classes assigned by participants will be compared with genre classes assigned from the Government of Canada metadata taxonomy [11].

While this work is still in the exploratory stage, the long-term goal is to develop a model of the task-genre relationships in this domain that can be used to inform the design of a contextual search engine for government information.

References

1. Nilsen, K., E-government in Canada, in Comparative perspectives on e-government, P. Hernon, R. Cullen, and H.C. Releya, Editors. (2006) Scarecrow Press: Lanham, MD.
2. Horrigan, J. B. How Americans get in touch with government: Pew Internet & American Life Project (2004)
3. Larsen, E., & Rainie, L. The rise of the e-citizen: how people use government agencies' Web sites. Washington, D.C.: Pew Internet & American Life Project (2002).
4. Fountain, J. E.. Building the virtual state: information technology and institutional change. (2001) Washington, DC: Brookings Institution Press.
5. Freund, L., E.G. Toms, and C.L.A. Clarke. Modeling task-genre relationships for IR in the workplace. in ACM SIGIR Conference (2005) Salvador Brazil.
6. Freund, L. and E.G. Toms. Using contextual factors to match intent. in Workshop on Information Retrieval in Context (IRiX), SIGIR (2005).
7. Freund, L. Situating relevance through task-genre relationships. Bulletin for the American Society for Information Science and Technology, 34 (5), (2008) 23-26.
8. Haraldsen, M., Päivärinta, T., Sein, M. K., & Stray, T. D. Developing e-government portals: from life-events through genres to requirements. Paper presented at the 11th Norwegian Conference on Information Systems (2004)
9. Kavaidas, G., & Tambouris, E. GovML: a markup language for describing public services and life events. Paper presented at the Knowledge Management in Electronic Government: 4th IFIP International Working Conference (2003).
10. Vintnar, M., & Leben, A. The concepts of an active life-event public portal. Paper presented at the EGOV (2002).
11. Treasury Board of Canada. Government of Canada type scheme (2003) http://www.tbs-sct.gc.ca/im-gi/mwg-gtm/typ-typ/docs/2003/schem/schem_e. asp
12. Freund, L., Nilsen, C. Assessing a genre-based approach to online government information. Proceedings of the 36th annual conference of the Canadian Association for Information, Science (CAIS), Vancouver, Canada (2008).