Directing Exploratory Search with Interactive Intent Modeling

Tuukka Ruotsalo[1,*], Jaakko Peltonen[1,*], Manuel J.A. Eugster[1,*], Dorota Głowacka[2],
Ksenia Konyushkova[2], Kumaripaba Athukorala[2], Ilkka Kosunen[2], Aki Reijonen[2],
Petri Myllymäki[2], Giulio Jacucci[2], Samuel Kaski[1,2]
Helsinki Institute for Information Technology HIIT
[1]Aalto University PO Box 15600, 00076 Aalto, Finland
[2]University of Helsinki, Department of Computer Science, Gustaf Hällströmin katu 2b, Helsinki, Finland
first.last@hiit.fi

ABSTRACT

We introduce interactive intent modeling, where the user directs exploratory search by providing feedback for estimates of search intents. The estimated intents are visualized for interaction on an *Intent Radar*, a novel visual interface that organizes intents onto a radial layout where relevant intents are close and similar intents have similar angles. The user can give rapid feedback on the visualized intents, from which the system learns and visualizes improved intent estimates. We systematically evaluated the effect of the interactive intent modeling in a mixed-method task-based information seeking setting with 30 users, where we compared two interface variants for interactive intent modeling, intent radar and a simpler list-based interface, to a conventional search system. The results show that interactive intent modeling significantly improves users' task performance and the quality of retrieved information.

Keywords

Exploratory Search, Intent Modeling, Search User Interfaces

Categories and Subject Descriptors

H.3.m. [**Information Search and Retrieval**]: Miscellaneous

General Terms

Search User Interfaces, Search Intent Prediction

1. INTRODUCTION

Studies have estimated that up to 50% of searching is informational and the corresponding search behavior is exploratory and spreads across individual queries and information needs [5]. One of the main problems in exploratory search is that it can be hard, if not impossible, for users to formulate queries precisely, since information needs evolve throughout the search session as users gain more information. In a commonly observed exploratory search strategy, the information seeker issues a quick, imprecise query, hoping to get into approximately the right part of the information space, and then directs the search to obtain the information of interest around the initial entry-point in the information space [2, 10]. Despite existing evidence on such behavior of the users [5], current methods to support users to explore are either based on typed queries, suggesting terms or rephrased queries [8], facets [12], result visualization and navigation through clusters [6], or they rely on relevance feedback mechanisms proven to be tedious to use [7]; or emphasize narrowing down the search within the initial query scope [12].

Existing techniques are effective for tasks where the user's goal is well defined and success is measured based on system response to well formed queries [6, 12]. In exploratory search the user's information needs evolve throughout the course of the search and her ability to direct the search to solve her task is critical [4, 9].

We introduce interactive intent modeling that lets users direct exploration via rapid relevance feedback in an interactive model-based loop where the user's search intents are estimated and visualized for interaction. The user iteratively adjusts the model by relevance feedback on keywords representing the current search intent. In the interface, keywords representing estimated search intents are arranged onto an *Intent Radar*, as a radial layout where relevant intents are close and similar intents have similar angles.

To evaluate the effect of interactive intent modeling on exploratory search we conducted a mixed–method task–based user experiment with 30 users performing a scientific information seeking task. Two interface variants, Intent Radar and a simpler list-based interface, were compared to a conventional typed-query system that did not support interactive intent modeling. The results show that interactive intent modeling improves the quality of retrieved information, the ability of users to target interactions to direct exploratory search, and the task performance of the users.

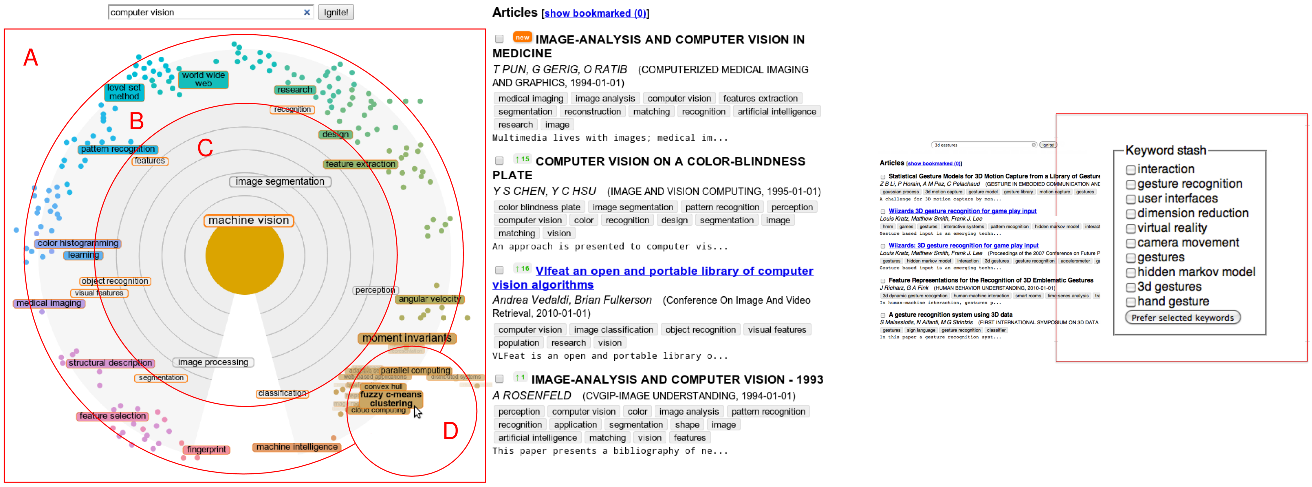
2. INTERACTIVE INTENT MODELING

We illustrate interactive intent modeling and the novel Intent Radar visualization by a walk-through example of an information seeking task (Figure 1). Imagine the user issues a query "machine vision"; the system responds with the predicted user intent and projected potential future intents along with a list of documents.

User interface. Besides a typical query box and article list, the interface uses a novel *Intent Radar* visualization, which represents search intents as relevant keywords corresponding to the predicted intents. The center of the Intent Radar represents the user. The inner gray circle represents the current search intent. The outer grey area represents future intent projections: potential directions

[*] Equal contributions

CIKM '13 San Francisco, USA
ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://–enter the whole DOI string from rightsreview form confirmation.



Articles [show bookmarked (0)]

□ new **IMAGE-ANALYSIS AND COMPUTER VISION IN MEDICINE**
T PUN, G GERIG, O RATIB (COMPUTERIZED MEDICAL IMAGING AND GRAPHICS, 1994-01-01)
medical imaging | image analysis | computer vision | features extraction | segmentation | reconstruction | matching | recognition | artificial intelligence | research | image
Multimedia lives with images; medical im...

□ ↑15 **COMPUTER VISION ON A COLOR-BLINDNESS PLATE**
Y S CHEN, Y C HSU (IMAGE AND VISION COMPUTING, 1995-01-01)
color blindness plate | image segmentation | pattern recognition | perception | computer vision | color | recognition | design | segmentation | image | matching | vision
An approach is presented to computer vis...

□ ↑16 **Vlfeat an open and portable library of computer vision algorithms**
Andrea Vedaldi, Brian Fulkerson (Conference On Image And Video Retrieval, 2010-01-01)
computer vision | image classification | object recognition | visual features | population | research | vision
VLFeat is an open and portable library o...

□ ↑1 **IMAGE-ANALYSIS AND COMPUTER VISION - 1993**
A ROSENFELD (CVGIP-IMAGE UNDERSTANDING, 1994-01-01)
perception | computer vision | color | image analysis | pattern recognition | recognition | application | segmentation | shape | image | artificial intelligence | matching | vision | features
This paper presents a bibliography of ne...

3d gestures
Ignite!

Articles [show bookmarked (0)]

□ **Statistical Gesture Models for 3D Motion Capture from a Library of Gesture**
Z B Li, P Horain, A M Pez, C Pelachaud (GESTURE IN EMBODIED COMMUNICATION AND
gaussian process | 3d motion capture | gesture model | gesture library | motion capture | gestures
A challenge for 3D motion capture by mon...

□ **Wiizards 3D gesture recognition for game play input**
Louis Kratz, Matthew Smith, Frank J. Lee
hmm | games | gestures | interactive systems | pattern recognition | hidden markov model | interact
Gesture based input is an emerging techn...

□ **Wiizards: 3D gesture recognition for game play input**
Louis Kratz, Matthew Smith, Frank J. Lee (Proceedings of the 2007 Conference on Future P
gestures | hidden markov model | interaction | 3d gestures | gesture recognition | accelerometer | ge
Gesture based input is an emerging techn...

□ **Feature Representations for the Recognition of 3D Emblematic Gestures**
J Richarz, G A Fink (HUMAN BEHAVIOR UNDERSTANDING, 2010-01-01)
3d dynamic gesture recogmtion | human-machine interaction | smart rooms | time-senes analysis | tra
In human-machine interaction, gestures p...

□ **A gesture recognition system using 3D data**
S Malassiotis, N Aifanti, M G Strintzis (FIRST INTERNATIONAL SYMPOSIUM ON 3D DATA
gestures | sign language | gesture recognition | classifier
In this paper a gesture recognition syst...

Keyword stash
□ interaction
□ gesture recognition
□ user interfaces
□ dimension reduction
□ virtual reality
□ camera movement
□ gestures
□ hidden markov model
□ 3d gestures
□ hand gesture
Prefer selected keywords

Figure 1: *Left:* The Intent Radar interface. Besides a query box and article list, search intents are visualized through keywords on a radial layout (A). The orange center area represents the user: the closer a keyword is to the center the more relevant it is to the estimated intent. The intent model used for retrieval is visualized as keywords in the inner circle (C); projected future intents are visualized as keywords in the outer circle (B). Keywords can be inspected with a fisheye lens (D). *Right:* The Intent List interface. The current estimated intent is shown as a list of keywords (zoomed and marked with a red rectangle). Users can type queries and give relevance feedback by clicking keywords in the list or underneath articles, and get a new set of documents and keywords by clicking the "Prefer selected keywords" button. The conventional Typed Query system has only the query typing functionality.

the user may like to follow given the current search intent estimate. The *radius* of keywords represents their relevance: the closer a keyword is to the center the more relevant it is for the current estimated search intent. *Angles* of keywords represent their similarity: similar angles indicate similar intents. The interface colors keywords based on a clustering to distinguish topically different search intents from each other. Keywords with highest relevance in each cluster are shown with labels to characterize the cluster, other keywords are shown as dots that can be enlarged with a fisheye lens.

We use a polar coordinate system and radial layout. This lets the visualization focus on the relation between the intents which is more important than their exact weights. It also allows users to select directions through a non-intrusive relevance feedback mechanism, where the user pulls keywords closer to the center of the radar. The radial layout has a good tradeoff between the amount of shown information and comprehensibility: a simple list of keywords only uses one degree of freedom and does not show keyword relationships, whereas higher than two-dimensional visualizations could make interaction with the visualization more difficult [3].

Interaction and feedback. The user can provide relevance feedback for the intents by dragging a keyword on the Intent Radar (closer to center means higher relevance) or by clicking a keyword under a document (assigns full relevance). Negative relevance feedback is possible by dragging a keyword outside the radar.

In the first iteration no user feedback is available, and documents and keywords are selected based on pseudo-feedback acquired from the top-ranked documents and visualized for the user. The user browses the visualization, notices keywords "infrared" and "cameras", drags them towards the center of the radar, and clicks the center to retrieve new estimates of intent and documents. Then the system computes new estimates for the user's current and potential future intents, and visualizes them on the Intent Radar.

2.1 Document Retrieval Model

We use the language modeling approach of information retrieval to estimate the relevance ranking of documents d_j given the estimate of the user's search intent. The intent model yields a keyword weight vector $\hat{\mathbf{v}}$ having a weight $\hat{v}_i$ for each keyword k_i. As feedback is not available on the first iteration, we use the typed query with weight 1 as the intent model. Documents are ranked by their probability given the intent model. We use a probabilistic multinomial unigram language model. The $\hat{\mathbf{v}}$ is treated as a (small) sample of a desired document, and documents d_j are ranked by the probability that $\hat{\mathbf{v}}$ would be observed as a random sample from the language model M_{d_j} for the document; with maximum likelihood estimation we get $\hat{P}(\hat{\mathbf{v}}|M_{d_j}) = \prod_{i=1}^{|\hat{\mathbf{v}}|} \hat{v}_i \hat{P}_{mle}(k_i|M_{d_j})$, and to avoid zero probabilities and improve the estimation we then compute a smoothed estimate by Bayesian Dirichlet smoothing so that $\hat{P}_{mle}(k_i|M_{d_j}) = \frac{c(k_i|d_j)+\mu p(k_i|C)}{\sum_k c(k|d_j)+\mu}$ where $c(k|d_j)$ is the count of keyword k in document d_j, $p(k_i|C)$ is the occurrence probability (proportion) of keyword k_i in the whole document collection, and the parameter μ is set to 2000 as suggested in the literature [13].

The documents d_j are ranked by $\alpha_j = \hat{P}(\hat{\mathbf{v}}|M_{d_j})$. We could just show the top ranked documents, but to expose the user to more novel documents, we sample a set of documents from the list and display them in ranked order. This favors documents whose keywords often received positive user feedback. We use Dirichlet Sampling, where a value $f_j \sim Gamma(\alpha_j, 1) = f_j^{\alpha_j - 1} e^{-f_j} / \Gamma(\alpha_j)$ is sampled for each document d_j, and the documents with highest f_j are shown to the user. At each iteration, the weight α_j is increased by 1 for documents d_j where at least one keyword got positive user feedback, and the weights are then renormalized.

2.2 Learning the Search Intent

Our model uses two main representations: the current estimate of *search intent*, and the *alternative future intents* that could occur in response to future feedback of the user; they are visualized in the inner and outer circle in Figure 1.We represent the current estimated search intent as a *relevance vector* $\hat{\mathbf{r}}^{current}$ over keywords, and the alternative future intents as a set of the same kind of relevance vectors $\hat{\mathbf{r}}^{future,l}$ predicted into the future, called the *future* *relevance vectors*. Each vector $\hat{\mathbf{r}}^{future,l}$, $l = 1, \ldots, L$, is a projection of the current search intent into the future in response to a set of L feedback operations the user could potentially use.

The user provides relevance feedback to search intents by giving relevance scores $r_i \in [0, 1]$ to a subset of J keywords $k_i, i = 1, \ldots, J$. Here $r_i = 1$ denotes keyword k_i is highly relevant to the user and she would like to direct her search in that direction, and $r_i = 0$ denotes the keyword is of no interest to the user.

Estimating keyword relevances. Let each keyword k_i be represented as a binary $n \times 1$ vector $\mathbf{k}_i$ telling which of the n documents the keyword appeared in. To boost significance of documents with rare keywords, we convert the $\mathbf{k}_i$ into *tf-idf* representation.

We assume the relevance score r_i of a keyword k_i is a random variable with expected value $\mathbb{E}[r_i] = \mathbf{k}_i^\top \mathbf{w}$, which is a linear function of the keyword representations $\mathbf{k}_i$. The unknown weight vector $\mathbf{w}$ determines the relevance of keywords. The vector $\mathbf{w}$ is estimated based on the relevance feedback given so far in the search session.

Estimating the weight vector. The algorithm maintains an estimate $\hat{\mathbf{w}}$ of the vector $\mathbf{w}$ which maps keyword features to relevance scores. To estimate $\mathbf{w}$ for a given search iteration, we use the LinRel algorithm [1]. In each search iteration, LinRel yields an estimate $\hat{\mathbf{w}}$. Let $\mathbf{K}$ be a matrix where each row $\mathbf{k}_i^\top$ is a feature representation of one of the keywords k_i shown so far, and let the column vector $\mathbf{r}^{feedback} = [r_1, r_2, \ldots, r_p]^\top$ contain the p relevance scores received so far from the user. LinRel estimates $\hat{\mathbf{w}}$ by solving the linear regression $\mathbf{r}^{feedback} = \mathbf{K}\mathbf{w}$, and calculates an estimated relevance score $\hat{r}_i = \mathbf{k}_i^\top \hat{\mathbf{w}}$ for each keyword k_i.

Selecting keywords for presentation to the user. At each iteration the system might simply pick the keywords with highest estimated relevance scores, but if $\hat{\mathbf{w}}$ is based on a small set of feedback, this exploitative choice might be suboptimal; or the system might exploratively pick keywords where feedback would improve accuracy of $\hat{\mathbf{w}}$. To deal with the exploration-exploitation tradeoff we select keywords not with the highest relevance score, but with the largest upper confidence bound for the score. If σ_i is an upper bound on standard deviation of the relevance estimate $\hat{r}_i$, the upper confidence bound of keyword k_i is computed as $\hat{r}_i + \alpha\sigma_i$, where $\alpha > 0$ is a constant used to adjust the confidence level of the bound. Let $\mathbf{r}^{feedback}$ again denote the vector of all relevance scores received from the user. In each iteration, LinRel computes $\mathbf{s}_i = \mathbf{K}(\mathbf{K}^\top\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{k}_i$ where λ is a regularization parameter, and the keywords k_i that maximize $\mathbf{s}_i^\top \mathbf{r}^{feedback} + \frac{\alpha}{2}\|\mathbf{s}_i\|$ are selected for presentation; they represent the estimated current search intent and are visualised in the inner grey circle of the Intent Radar visualization (Figure 1). We use LinRel since it allows, at the same time, to maximize relevance of intent estimates based on user interactions and reduce system uncertainty about the relevant intents that occurs because of limited and possibly suboptimal feedback.

Estimating alternative future intents. Our approach not only estimates user's current intents, but also suggests potential search directions to the user. At each iteration, based on the current estimated search intent (relevance vector $\hat{\mathbf{r}}^{current}$ over keywords), the system estimates a set of *alternative future search intents* (future estimates of the relevance vector). The future search intent is estimated for each of L alternative feedbacks $l = 1, \ldots, L$; in each feedback l, a pseudo-relevance feedback of 1 is given to the lth keyword in the search intent visualization, the feedback is added to the feedback from previous search iterations, and LinRel is used to estimate the future relevance vector $\hat{\mathbf{r}}^{future,l}$ for keywords.

Each $\hat{\mathbf{r}}^{future,l}$ provides the user a set of keywords she would most likely be shown, if she decided to give positive feedback to the lth currently shown keyword. Thus the user gets a view of L potential search directions which can be explored in more detail.

Denote the current estimated search intent as $\hat{\mathbf{r}}^{current} = [\hat{r}_1^{current}, \ldots, \hat{r}_{N_{keywords}}^{current}]^\top$, where $\hat{r}_l^{current}$ is the estimated relevance of the lth keyword. Future intents are estimated as the $N_{keywords} \times L$ matrix $\hat{\mathbf{R}}^{future}$, where the element in row i, column l, is $\hat{r}_i^{future,l} \in [0, 1]$, predicted relevance of the ith keyword in the next search iteration according to the lth future intent.

2.3 Layout Optimization

We optimize a data-driven layout for the search intent and alternative future intents on the Intent Radar interface. We optimize locations of keywords in the inner circle (representing current intent) and keywords in the outer circle (representing future intents) by probabilistic modeling-based nonlinear dimensionality reduction.

Representation of the outer keywords. We lay out the future potentially relevant keywords into the outer circle, based on their potential future relevances. Consider the matrix $\hat{\mathbf{R}}^{future}$ of predicted future keyword relevances across a set of future search intents as discussed in Section 2.2. Each keyword k_i in the outer circle can be characterized by row i of $\hat{\mathbf{R}}^{future}$, that is, by the row vector $\tilde{\mathbf{r}}_i = [\hat{r}_i^{future,l}, \ldots, \hat{r}_i^{future,L}]$ where $\hat{r}_i^{future,l} \in [0, 1]$ is the estimated relevance of k_i in the lth future search intent.

The norm $||\tilde{\mathbf{r}}_i||$ represents overall predicted relevance of keyword k_i across future search intents; we use it as the radius of k_i on the radar. The vector $\bar{\mathbf{r}}_i = \tilde{\mathbf{r}}_i / ||\tilde{\mathbf{r}}_i||$ then tells which future search intents make k_i most relevant, that is, which direction of future intent k_i is associated with. We use a radial layout in which keywords associated with similar future intents have similar angles.

Layout of keywords in the outer circle. Keywords k_i and k_j in the outer circle can be called *neighbors* if their characterizations $\bar{\mathbf{r}}_i, \bar{\mathbf{r}}_j$ are similar: the keywords most similar to k_i can be described as a probabilistic neighbor distribution $p_i = \{p(j|i)\}$ where

$$p(j|i) = \exp(-||\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_j||^2/\sigma_i^2) \cdot (\sum_{j'} \exp(-||\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_{j'}||^2/\sigma_i^2))^{-1}$$

and the σ_i are set as in [11]. On the display k_i and k_j appear similar in the outer circle if they have close-by directions (angles) a_i and a_j; the keywords that appear most similar to k_i in the outer circle can then be described by neighbor distribution $q_i = \{q(j|i)\}$ where

$$q(j|i) = \exp(-|a_i - a_j|^2/\sigma_i^2) \cdot \sum_{j'} \exp(-|a_i - a_{j'}||^2/\sigma_i^2))^{-1} .$$

The task of the layout algorithm is to place keywords so that neighboring keywords on the display have neighboring characterizations. To do so, we measure the total Kullback-Leibler divergence D_{KL} between the neighborhoods of display locations versus characterizations, as $(\sum_s D_{KL}(p_i, q_i) + \sum_s D_{KL}(q_i, p_i))/2$. The total divergence is a function of the angles a_i of the keywords in the outer circle; we optimize the a_i by gradient descent to minimize the total divergence. A similar approach was used to visualize fixed data sets in [11], see [11] for optimization details. Analogously to [11], it can be shown this layout approach corresponds to *optimizing information retrieval of neighboring keywords from the display layout* (minimizing misses and false positives of such retrieval).

Highlighting of keywords in the outer circle. To highlight the structure in the outer circle layout, we apply a simple agglomerative clustering to angles a_i of keywords in the outer circle. In detail, start a cluster from the keyword with the smallest angle, and iteratively add the keyword with the next largest angle into the cluster as long as the angle difference is below a treshold and the size of the cluster is smaller than a specified percentage of all keywords in the outer circle; when either condition fails start the next cluster. We show clusters with different colors, and show for each cluster the label of the predicted most relevant keyword (having largest $||\tilde{\mathbf{r}}_i||$).

Layout of the keywords in the inner circle. The keywords in the inner circle represent the current search intent; for each such keyword k_l, its radius naturally represents its current estimated relevance $\hat{r}_l \in [0, 1]$. The angles a_l of keywords in the inner circle must be placed consistently with the layout of the outer circle (the keywords of future search intents): since we estimate the alternative future search intents in response to an interaction with an inner keyword k_l, a_l should represent which future keywords become most relevant in the lth future search intent. We thus set a_l to the highest weighted mode of angles a_i of future keywords k_i, where the angle of each future keyword is weighted by the predicted future relevance $\hat{r}_i^{future,l}$. The resulting angle a_l of each keyword k_l in the inner circle indicates which keywords would become relevant by interacting with k_l: thus the angles of keywords in the inner circle indicate directions of future search intent.

3. USER EXPERIMENTS

A task-based user experiment was designed to investigate the effects of interactive intent modeling on exploratory search. The advantage of a task-based setting is that it allows us to measure natural user interaction and task performance, but still retain the advantages of a controlled experiment. We setup the experiments to answer the following research questions:

1. User task performance: Does the interaction paradigm lead to better user responses in the given tasks? **2. Quality of displayed information:** Does the paradigm help users reach high quality information in response to interactions? **3. Interaction support for directing exploration:** Does the paradigm elicit more interaction from the user? Is the elicited interaction targeted to relevant interaction options? Does the paradigm let the user explore novel information more than a conventional system where users might be constrained by limited interaction capabilities?

3.1 Experimental Design

We chose a $2 \times 3 \times 5$ between subjects design with two search tasks, three system setups and five users for each task/system combination. We chose the design to avoid learning effects of users as each user only used one of the systems and performed a single task.

Three systems were created: two versions of our interactive intent modeling with different extents of intent prediction and visualization, denoted as "Intent Radar" and "Intent List", and a conventional typed-query based system "Typed Query".

The two systems with interactive intent modeling are as follows. Intent Radar implements the full versions of interactive intent modeling with future intent prediction and Intent Radar visualization as described in previous sections. The implemented system updated search results and the interface in response to interactions under three seconds. Intent List implements only intent estimation and has a simpler interface that visualizes the intent model for the user as a list. Figure 1 (bottom right) shows a screen shot of this interface. The users interact with the system by typing queries and providing binary relevance feedback on keywords shown under each document, as well as on keywords in the list.

The Typed Query system is a query-based system, where neither intent modeling nor visualization are used. Users express their information needs only by typing queries. Keywords are visualized underneath the articles; users can use them as cues for new typed queries, but cannot directly interact with them. Essentially, the interface has the same features as the Intent List shown in Figure 1 (bottom right) but without the keyword list visualization.

3.2 Search Tasks

We chose a task type that is complex enough to ensure that some interaction is necessary for users to acquire the information to accomplish the task; is complex enough to allow users to choose the kind of interaction that best supports solving the task; and is complex enough to reveal exploratory search behavior. The tasks were defined as scientific writing scenarios, i.e., participants were asked to prepare materials to write an essay on a given topic. The assignments were (1) to search for relevant articles that they would be likely to use as reference source in their essay and (2) to answer a set of predefined questions related to the task topic.

We recruited two post-doctoral researchers to define two information seeking tasks. The task fields chosen by the experts were "semantic search" and "robotics". The experts wrote task descriptions using this template: "Imagine that you are writing a scientific essay on the topic. Search scientific documents that you find useful for this essay". To provide clear goals for exploration, the experts provided questions about specific aspects of the topic. The questions defined by the experts for the robotics tasks were: "What are the sub-fields, application areas and algorithms commonly used in the field of robotics"; for the semantic search task the questions were: "What are the techniques used to acquire semantics, methods used in practical implementation, organization of results, and the role of Semantic Web technologies in semantic search".

3.3 Procedure

We recruited 30 students from two universities to participate in the study. All the participants were graduate students with a background in computer science or a related field. In a prior background survey we ensured that every participant had conducted literature search before and was neither an expert nor a novice in the topic of the assigned search task (self-assessment on a scale from 1 to 5; we selected people who rate themselves between 2 and 4).

The basic protocol for each experiment scenario was the following: demonstration of the system (10 min) and performing of the search task by the participant (30 min). The experiments were performed in an office like environment using standard equipment The demonstration of the system was done by the instructor using a separate computer. All user interactions were logged with timestamps: typed queries, the documents and keywords presented by the system in response to interactions, the keywords the user interacted with, and the articles the user bookmarked.

3.4 Data

We used a dataset of over 50 million scientific documents from the Web of Science prepared by THOMSON REUTERS, Inc., and from the Digital Libraries of the Association of Computing Machinery (ACM), the Institute of Electrical and Electronics Engineers (IEEE), and Springer. The dataset contains the following information about each document: title, abstract, keywords, author names, publication year and publication forum.

3.5 Relevance Assessments

Experts conducted two types of double-blind relevance assessments. For the *quality of information displayed*, all documents and keywords that were presented to the participants by any of the three systems were pooled resulting in a collection of 5612 documents and 4097 keywords. The experts assessed the articles on binary scale on three levels: (1) relevance—is this article relevant to the search topic; (2) obviousness—is this a well-known overview article in a given research area; and (3) novelty—is this article an uncommon yet relevant to a given topic or specific subtopic in a given research area. These assessments constituted the ground truth for evaluating retrieval performance of the systems. The ground truth consisted of 3384 relevant documents (731 were obvious and 2653 were novel). Experts also assessed the keywords on three levels: (1) relevance—is this keyword relevant for the topic; (2) general—does this keyword describe a relevant subfield, (3) specific—does this keyword describe a relevant specifier for the subfield? The Cohen Kappa test indicated substantial agreement between experts, Kappa = 0.71, $p < 0.001$. For the *quality of responses of the users to the tasks*, for each question answers of all participants were pooled and assessed by experts on a 5-point Likert scale.

3.6 Evaluation Aspects and Measures

User task performance was the main measure of success. It was measured using an averaged score of expert assessments of the participants' written answers in response to the tasks. The given written answers were evaluated by the same experts who wrote the task descriptions and conducted the article assessments. The experts scored each answer between 0 (no answer) and 5 (perfect answer). In addition, we measured the number of bookmarked relevant, obvious, and novel documents the users were displayed in response to their interactions while completing the tasks.

Quality of displayed information was measured by precision, recall, and F-measure. The measures were computed both for the documents displayed for the user, and for the keywords the user interacted with. These characterize the quality of document users were able to reach and the quality of keywords users chose to manipulate. The measures were computed with respect to the different assessment categories, so that for the documents we considered in turn either the relevant, or the obvious, or the novel documents as the ground truth; for the keywords we similarly took the relevant, general, and specific keywords in turn as the ground truth.

Interaction support for directing exploration was measured using two separate types of measures. First, we measured the number and type of interactions (typed query or interaction with the intent model). Second, we measured the type of information (novel or obvious) received in response to different types of interactions. These measures characterize how well a particular type of interaction was able to support each user to direct the search to relevant information, and in particular characterize the differences of the interaction types in finding obvious and novel information.

4. RESULTS

The results are summarized in Figure 2 and discussed in detail in the following sections corresponding to the evaluation aspects.

4.1 Task Performance

The main result of the experiments is that the users of the Intent Radar system achieve significantly better task performance than the users in the Intent List and the Typed Query systems. For Intent Radar users' responses to the tasks are graded to be significantly better by experts than the responses of the users of the other systems as shown in Figure 2 (Task performance). The results are statistically significant (Friedman test with post-hoc analysis, $p < 0.05$ for Intent Radar vs. Typed Query, $p < 0.05$ for Intent Radar vs. Intent List). Note that, all participants were able to accomplish the tasks and completed the task in the given timeframe (no significant time differences between the systems or tasks).

4.2 Quality of Displayed Information

Figure 2 (Quality of displayed information) shows the quality of displayed articles and the quality of keywords users interacted with. The two versions of the interactive intent modeling achieve substantially better performance than the Typed Query comparison system. The differences are statistically significant using the non-parametric McNemar's test for categorical data with Bonferroni correction to correct for the multiple comparisons ($p < 0.001$).

The Intent List shows slightly better performance for obvious documents. A possible explanation is that the less advanced interaction capabilities in the Intent List interface, and even more limited in the Typed Query comparison system, make it more difficult to move away from the initial query context, thus failing to increase recall but preserving slightly better precision.

The quality of the keywords the users interacted with is significantly better (higher F-measure) for the Intent Radar interface than for the Intent List interface, for all relevant keywords and for both subcategories (general and specific keywords). This indicates that the Intent Radar interface has made it easier to target interactions to more relevant keywords. Moreover, the significantly higher quality of the displayed keywords themselves can add to the users' understanding of the information seeking task and is an explanation for the increased task performance for users of Intent Radar.

4.3 Interaction Support for Exploration

Figure 2 (Interaction support for exploration) shows that users adopt and make use of interactive intent modeling when offered to them. In particular, users interacted with the Intent Radar interface twice as much as with the Intent List and nearly four times more than the Typed QueryṪyped queries were used equally in each interface, and the intent models were interacted with in cycles in which typed keywords were first issued and then intent models were used to direct the search. This indicates that users did not replace the typed queries with interaction with the intent models, but rather directed their search from the initially issued imprecise query. The users of the Typed Query system had trouble reaching novel information. A possible explanation is that coming up with queries was difficult for users of the Typed Query system as intent models were not available. This was the case even though they could see the keywords under each document returned by the system and could use them as cues for typed queries. As noted in Figure 2 (Quality of displayed information), the keywords users interacted with were highly relevant (high precision in the Relevant category), for both Intent List and Intent Radar; thus the elicited interaction with the intent models and the further increased interaction in Intent Radar were targeted to relevant interaction options.

Interestingly, the interactive intent modeling engages users to move more rapidly in the information space. Users in the Intent Radar and in the Intent List conditions chose to use typed queries as a shortcut to a previous view; this is seen in the fact that users repeat typed queries more with the Intent Radar interface (14% queries were repeats) and the Intent List interface (20%) than with Typed Query (4%). Users of the Intent Radarcondition repeated fewer queries than the users of the simpler version, perhaps because the full interface already allows efficient movement through the visualized current and future search intents.

An important aspect of the interaction support is also whether the interaction with the predicted intents made it possible for the users to direct the search and to reach more novel information. The results in Figure 2 (Interaction support for exploration) show that users were successful in directing their search with interactive intent modeling. After directing the search using interaction with the intents, users were displayed a significantly larger portion of novel documents than after typing queries. Conversely, the users were displayed a larger portion of obvious documents in response to typed queries. This suggests that the interaction with the intent model enables users to direct their search and find novel documents that are not found using the typed queries, but at the same time achieve more relevant information than conventional search

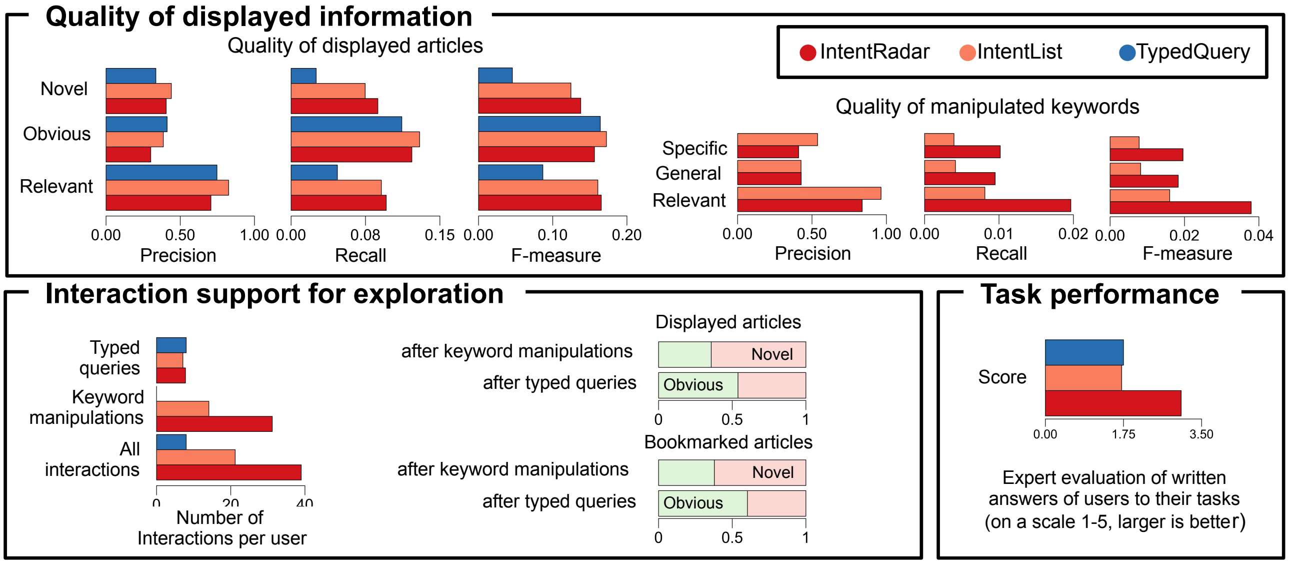


Figure 2: Results of the user experiments divided according to the evaluation aspects: Task performance (scores of user answers; see Section 4.1), Quality of displayed information (information retrieval measures for documents and keywords; see Section 4.2), and Interaction support for exploration (amounts, types, and proportion of novel and obvious information displayed in response to interaction; see Section 4.3). Interactive intent modeling significantly improves users' task performance, interaction support for directing exploration, user experience, and quality of displayed information compared to a conventional typed-query system. The best overall results are achieved with the Intent Radar interface.

systems. A similar effect is also present in the documents users bookmarked. Users bookmarked more novel documents from the results that they received in response to interactions with the intent models, while users bookmarked more obvious documents from the results they obtained using typed queries.

Overall the results suggest that interactive intent modeling, in particular the Intent Radar interface, which complements future intent prediction with appropriate visualization, allowed users to reach the novel documents that were harder to find with the Typed Query system.

5. CONCLUSIONS

In this paper we introduced interactive intent modeling for directing exploratory search and demonstrated its usefulness in task-based user experiments. Our results show that interactive intent modeling can significantly improve users' performance in exploratory search tasks. The improvements can be attributed to improved quality of displayed information in response to user interactions, better targeted interaction between the user and the system, and improved support for directing search to achieve novel information. Interaction with intent visualization does not replace the query-typing interaction, but offer an additional complementary way to express more specific intents to direct search to explore novel information. In particular, the improved quality of information, when displayed on the Intent Radar interface, also transfers to improved task performance.

6. ACKNOWLEDGMENTS

Certain data included herein are derived from the Web of Science prepared by THOMSON REUTERS, Inc., Philadelphia, Pennsylvania, USA: Copyright THOMSON REUTERS, 2011. All rights reserved. Data is also included from the Digital Libraries of the ACM, IEEE, and Springer.

7. REFERENCES

[1] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3:397 – 422, 2002.

[2] M. Bates. Where should the person stop and the information search interfaces start? *IEEE Data Engineering Bulletin*, 26(5), 1990.

[3] G. Draper, Y. Livnat, and R. Riesenfeld. A survey of radial methods for information visualization. *IEEE T. Vis. Comput. Gr.*, 15(5):759 –776, sept.-oct. 2009.

[4] D. Glowacka, T. Ruotsalo, K. Konyushkova, K. Athukorala, G. Jacucci, and S. Kaski. Directing exploratory search: Reinforcement learning from user interactions with keywords. In *Proc. IUI'13*, pages 117–128. ACM, 2013.

[5] M. A. Hearst. *Search User Interfaces*. Cambridge University Press, 1 edition, 2009.

[6] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *Proc. SIGIR'96*, pages 76–84. ACM, 1996.

[7] D. Kelly and X. Fu. Elicitation of term relevance feedback: an investigation of term source and context. In *Proc. SIGIR'06*, pages 453–460. ACM, 2006.

[8] D. Kelly, K. Gyllstrom, and E. W. Bailey. A comparison of query and term suggestion features for interactive searching. In *Proc. SIGIR'09*, pages 371–378. ACM, 2009.

[9] T. Ruotsalo, K. Athukorala, D. Głowacka, K. Konyushkova, A. Oulasvirta, S. Kaipiainen, S. Kaski, and G. Jacucci. Supporting exploratory search tasks with interactive user modeling. In *Proc. ASIS&T' 13*, pages nn–nn. ASIS&T, 2013.

[10] J. Teevan, C. Alvarado, M. S. Ackerman, and D. R. Karger. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *Proc. of SIGCHI*, pages 415–422, 2004.

[11] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *J. Mach. Learn. Res.*, 11:451–490, 2010.

[12] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In *Proc. CHI'03*, pages 401–408. ACM, 2003.

[13] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, Apr. 2004.