Negative Relevance Feedback for Exploratory Search with Visual Interactive Intent Modeling

Jaakko Peltonen[1,2] **Jonathan Strahl**[2] **Patrik Floréen**[2,3]
[1]Faculty of Natural Sciences, University of Tampere, Tampere, Finland
[2]Department of Computer Science, Aalto University, Espoo, Finland
[3]Department of Computer Science, University of Helsinki, Helsinki, Finland
{jaakko.peltonen, jonathan.strahl}@aalto.fi, patrik.floreen@hiit.fi

ABSTRACT
In difficult information seeking tasks, the majority of top-ranked documents for an initial query may be non-relevant, and negative relevance feedback may then help find relevant documents. Traditional negative relevance feedback has been studied on document results; we introduce a system and interface for negative feedback in a novel exploratory search setting, where continuous-valued feedback is directly given to keyword features of an inferred probabilistic user intent model. The introduced system allows both positive and negative feedback directly on an interactive visual interface, by letting the user manipulate keywords on an optimized visualization of modeled user intent. Feedback on the interactive intent model lets the user direct the search: Relevance of keywords is estimated from feedback by Bayesian inference, influence of feedback is increased by a novel propagation step, documents are retrieved by likelihoods of relevant versus non-relevant intents, and the most relevant keywords (having the highest upper confidence bounds of relevance) and the most non-relevant ones (having the smallest lower confidence bounds of relevance) are shown as options for further feedback. We carry out task-based information seeking experiments with real users on difficult real tasks; we compare the system to the nearest state of the art baseline allowing positive feedback only, and show negative feedback significantly improves the quality of retrieved information and user satisfaction for difficult tasks.

ACM Classification Keywords
H.3.3 Information Storage and Retrieval: Information Search and Retrieval—Relevance Feedback; Retrieval Models

Author Keywords
Negative Relevance Feedback; Difficult Queries; Query Intent; Query reformulation; Search Interfaces; Language Model; Novelty in Information Retrieval; Presentation of Retrieval Results; User Intent Model; Interactive Exploratory Search

IUI 2017 March 13-16, 2017, Limassol, Cyprus

ACM ISBN 978-1-4503-4348-0/17/03.
DOI: http://dx.doi.org/10.1145/3025171.3025222

INTRODUCTION
Information seeking is a crucial and everyday knowledge work task. While some information needs can be answered by simple look-up queries from easily found and identified sources, other information seeking can be more difficult. It has been estimated that up to 50% of searching is informational and exploratory, involving multiple information needs and queries [11].

In this paper we propose an exploratory search system that allows both positive and negative feedback to be given directly to modeled user intents represented as an interactive visualization. Interaction with the intent model lets users direct their search. The system extends a previous system that allowed positive feedback only [25]. The system is based on machine learning solutions for user intent modeling from feedback, for retrieval of documents based on probabilistic scoring, and for dimensionality reduction based interactive visualization of an intent model as an organized radial display of keywords.

In difficult information seeking tasks, search systems can fail to support information seeking and can fail to yield relevant results for search queries; these queries are known as difficult queries. A search system can fail to return relevant results for a number of reasons, including poorly formulated search phrases, or because the content being sought is hard to describe with any simple search phrase, or because a broad range of documents are relevant and cannot be easily covered with a simple query [5, 6, 7, 15]. Testing a number of these failures on the interactive exploratory search system with only positive feedback we found that it is possible to correct most of the examples tried. We did however find a particularly difficult type of search task that was very challenging even with an interactive exploratory search system; searching for interesting subtopics when the initial parent task has an overwhelming popular set of unwanted subtopics, e.g., searching for interesting methods (subtopics) for machine learning for time series (the main topic) while not interested in popular and common methods like support vector machines (a popular subtopic) or artificial neural networks (another popular subtopic). Using feedback on relevant keywords failed to improve the top-ten relevant results in some of these cases. We use this structure of a task to initialize the search with a difficult query, where it is difficult to direct the search to relevant results with only positive feedback, to test the benefit of negative relevance feedback for exploratory search with user intent modeling.

Our aim is to show that even with keyword feedback on a user model negative relevance feedback can still improve the search efficiency.

Previous studies have shown with simulations and often artificial queries that negative relevance feedback (NRF) on documents, in the form of binary feedback, can help to improve the search results efficiently when the query is difficult [15, 16, 19, 29, 30]. We continue research on NRF using real search tasks with continuous-valued feedback on document keywords. Furthermore, studies [16, 21] showed that NRF can be made more effective by generalization and aiming for a generalization that is close to the relevant documents. Our system generalizes the negative relevance feedback using documents that are estimated to be relevant by the system and shares these keywords with the user through the visualisation allowing the user to correct the model. We believe this helps to improve the effectiveness of the NRF. In the past it has been shown in [10] if the results are not bad, then negative relevance feedback is not beneficial, not even for a standard text-only retrieval system. Therefore we focus our studies on the most difficult real queries we can produce to test the system.

In addition to improving search efficiency we noticed that when using NRF the results were more varied and novel than when using only positive relevance feedback (PRF). There is evidence of this in our results on the system and in the feedback questionnaire from the user. The intuition is that if a search includes an unwanted subtopic, for example support vector machine, with PRF the user may spot a relevant alternative subtopic and give positive feedback to that one topic; the downside is that this will focus the search on that subtopic and possibly move away from other interesting subtopics. In contrast, when giving NRF directly to the unwanted term, documents with that term are penalized and new documents can rise in the ranks that would not increase in relevance with only positive feedback on one particular subtopic. These new documents will be relevant to the broader positive feedback of the main topic, but will now exclude the unwanted subtopic, allowing the results to be more varied. Lastly, our system allows continuous-valued feedback and we noticed users made use of it; a large percentage of feedback was discrete (fully positive, negative, or indifferent) but a substantial amount was continuous-valued.

RELATED WORK

Negative relevance feedback, difficult queries and document feature feedback. In information retrieval, relevance feedback from the user is used to iteratively improve query results. Typically, results of an initial query are presented to the user, the user gives relevance feedback on the documents that were relevant (or irrelevant), and the system improves the query based on the feedback and returns updated results. This procedure is repeated as long as the user is willing to give feedback. There are several types of relevance feedback: implicit, explicit and pseudo-relevance feedback. We focus on explicit relevance feedback, where the user gives direct feedback to the item, and pseudo-relevance feedback, where feedback is given automatically making some assumptions about the current results [22, Chapter 9].

During a search session, most of the time, the initial query will contain many relevant results and the user can give positive relevance feedback to documents that are relevant; the system can then improve the query based on this feedback. In some cases, most or all of the top-ten results of the query are irrelevant to the search; these cases can be referred to as *difficult queries*. Information retrieval systems still perform badly for some difficult queries [5, 6, 7, 20]. When a query is difficult, users can reformulate the query or look for articles further down the list to give positive relevance feedback (PRF) to try to improve the search results [9, 15, 16], or users could give negative relevance feedback (NRF), if such possibility is available, typically to the documents that are wrongly ranked highly. Previous studies on NRF have shown that it is generally not as useful as PRF [10], so most work on relevance feedback is on PRF only. However, studies have shown that NRF can be more effective than PRF in the case of difficult queries [15, 16, 19, 29, 30]. We show this is also true for interactive exploratory search systems with user modeling using continuous-valued NRF on estimated relevance of keywords.

To improve the effect of NRF, the feedback should be generalized and the generalization should aim to be close to relevant documents, but not too close [16]. The idea is that NRF should have an impact on documents that are estimated to be most relevant, but should not generalize so much as to impact too many highly ranked relevant documents; the goal is to reduce the rank of documents that are ranked highly, but are not relevant [10].

Relevance feedback has been incorporated into information retrieval models, but there has been difficulty incorporating relevance feedback in a principled way into language modeling (LM) using a query likelihood for information retrieval [36]. However, NRF for LM is generally more effective than for vector space models [30]. Heuristic approaches to incorporate relevance feedback into LM have been presented, including approaches to incorporate negative feedback into LM [15].

In [32] users can give relevance feedback on keywords by editing them in a result list, allowing negative feedback through deletion or positive through highlighting; simple generalization of the feedback and visualization through tag clouds is provided. Our approach provides several advantages over this: Firstly, instead of edits on documents, our feedback is given to *features of an inferred user intent model*, directly curating the system's understanding of search intent. Secondly, we *visualize the intent model in a 2D way* with keywords organized by relevance and by influence on the search; users can give feedback to most important features of the intent model efficiently even for large result sets. In contrast, [32] use an essentially one-dimensional visualization by tag clouds of top keywords at every 100 documents of the ranking, which may contain redundancies, does not show similarities of keyword influence, and can make it hard to find weak features to emphasize. Lastly, in our system, feedback influences results through *inference of user intent*, impact of feedback is increased by features letting feedback influence non-directly co-occurring keywords, and the new intent model is found by Bayesian inference from all feedback. In contrast, [32] do not build a user

intent model, but only penalize/reward, for each feedback term, documents having "extended" terms, directly co-occurring top terms or top terms with differing weight in a query expansion. The penalty is large and fixed-valued for each penalty/reward term, whereas we allow continuous-valued feedback and infer weights for each keyword.

A concern with NRF is the user's perception of it. User experience studies with NRF have shown bad feedback from users, due to fear of using NRF on documents having relevant information [3, 26]. In contrast, in our setting NRF is given to topics (keywords) instead of entire documents, allowing more precise targeting of feedback to only the undesired information content within documents.

Exploratory search, interactive intent modeling and the intent radar. Support for exploratory search has involved term or query suggestions [18], facets [14] and (cluster-based) result visualization [12], time-consuming feedback mechanisms, or further focus within the initial scope [14]. We instead concentrate on a recent novel framework for interactive information retrieval with intent modeling, which uses a visualization to display estimates of the user intent and allows the user to adjust them through manipulation of keywords ([25]; see also, e.g., [1, 23, 31] for further developments of the framework). The system provides a radar-like interactive visualization to the user, where estimated relevances and future predicted relevances of document keywords are displayed. The system uses a multinomial language model and a likelihood based document scoring to retrieve the initial set of articles. The system takes feedback into account by a linear prediction model for keyword relevances, where upper confidence bounds (UCBs) are then used to balance exploitation of current relevance estimates and exploration to break away from the initial context. The intuition is to score keywords highly if their mean relevance estimate and its uncertainty are high. The system was reported to perform well in information seeking experiments, yielding better task performance and more relevant documents and keywords than a traditional system, and we therefore pick it as a basis for extension and compare to it as a baseline system in experiments. Our experiments show we outperform this already state of the art baseline, a strong indication of the goodness of our negative feedback solution.

NEGATIVE RELEVANCE FEEDBACK FOR EXPLORATORY SEARCH WITH INTENT MODELING

We now present our system for exploratory search with positive and negative keyword feedback; we first give a walk through and then discuss the machine learning solutions.

Walk-through of the System

A user may initiate a new search session by typing a query into the search box as usual, for example "machine learning for time series". Top-ranked search results are retrieved and listed on the right-hand side, showing the title, date, venue, authors, list of keywords and abstract for each article. The predicted user intent is visualized by arranging keywords in a radial visualization denoted an "intent radar": the center represents the user, the innermost gray area shows the top-10 predicted most relevant keywords, and the middle area shows an organization of currently low-scoring keywords shown as dots that form rough clusters (colors show clusters); they represent directions in the information space, keywords that would become relevant with similar feedback, and would thus direct the search in similar directions. Keywords corresponding to dots can be inspected with a fisheye lens functionality. Compared to the system of [25] the user will see an additional light-red colored outer ring (red ring) on the Intent Radar, see Fig. 1, which is where the user can give NRF to the system. Initially the red ring is empty, and as long as the user gives no NRF there will be no keywords in the red ring. The user may drag keywords on the radar to desired positions, which yields feedback (relevance scores) which the system uses to infer an updated intent model. Positive feedback means dragging a keyword into the inner gray area; disinterest may be indicated by dragging a keyword into the middle area; and active dislike (NRF) by dragging a keyword into the red ring. As soon as the user gives NRF to a keyword, the system will build a negative language model to penalize documents and display the negative keywords with the lowest lower-confidence bounds, see Fig. 2.

Machine Learning for Interactive Intent Modeling from Positive and Negative Feedback

The search system has four components that must respond to user feedback. In order, 1) *intent modeling* based on feedback received so far, 2) *retrieval of documents* based on the intent model, 3) *optimized visualization* of the intent model, and 4) *presentation of results in the interactive frontend* which then gathers feedback for the next iteration. Steps 1-3 are done by machine learning, and must incorporate both positive and negative feedback; we present their solutions next.

Intent modeling. Suppose user feedback on a subset of keywords has been received. The task of intent modeling is to generalize the feedback to estimate interest over all keywords, and quantify uncertainty of the estimates. We also choose which keywords to show to the user as options for further feedback: We balance exploration and exploitation using both *upper* and *lower* confidence bounds to pick keywords having a chance to get high positive or low negative feedback as discussed below. Lastly, we quantify changes of the predictions in response to potential additional feedback.

Feedback is given as relevance scores r_i for a subset of keywords i. Unlike [25], scores must allow both positive and negative feedback: We use the range $[-1,1]$ where 1 denotes maximal positive interest, 0 denotes disinterest and -1 denotes active dislike (negative interest). We use probabilistic linear regression to model user interest and its uncertainty. Given a document set with n documents, suppose each keyword i has an $n \times 1$ feature representation $\mathbf{k}_i$ where each element represents association of the keyword to one of the documents. Unlike [25], the $\mathbf{k}_i$ are not simple TF-IDF features, we describe them later in this section. We model relevances r_i so that their expectation is a linear function $E[r_i] = \mathbf{k}_i^\top \mathbf{w}$. The model (parameter $\mathbf{w}$) is estimated with a Gaussian noise model from an observed set of S feedback scores $\mathbf{r}^{feedback}$ with an $S \times n$ matrix $\mathbf{K}$ containing the feature representations of those keywords as its rows; it can be shown [2] the upper

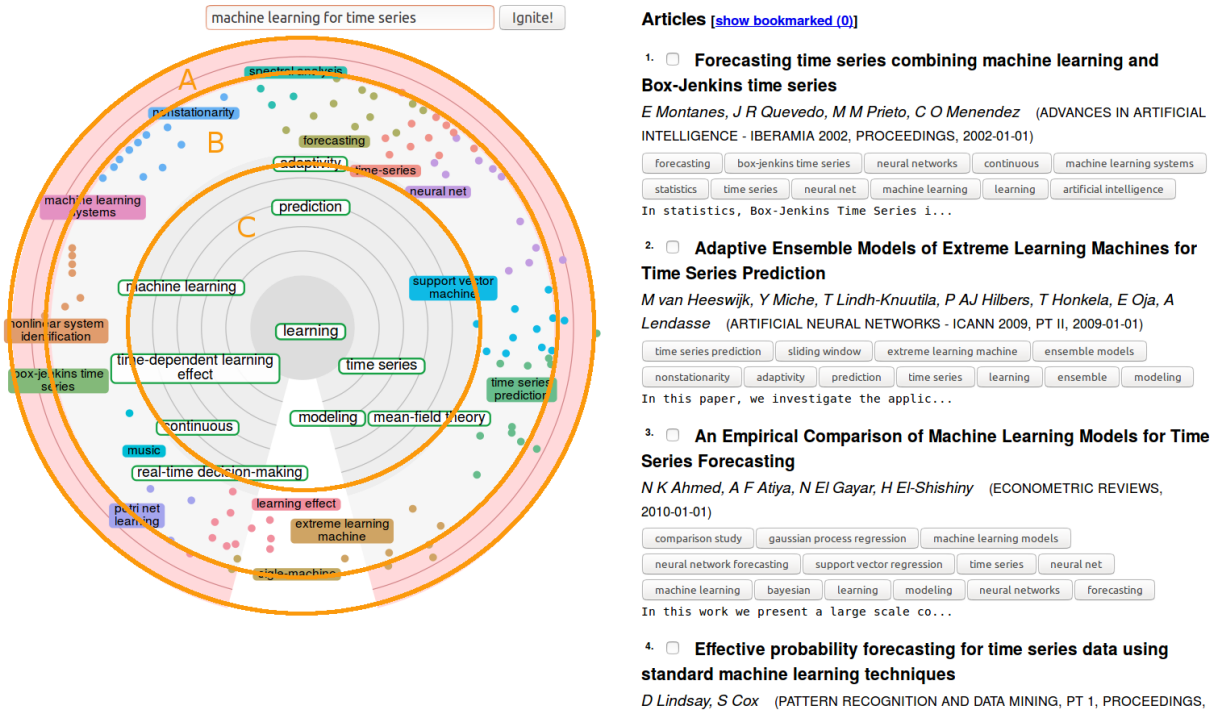

Figure 1. The Intent Radar interface with negative feedback: The search intent radar comprises the inner circle (C), where keyword relevance estimates represent the distance to the center, high relevance at the center with linear decay to zero on the outer border, the future-predictions ring (B), where predicted future keywords are presented, and the negative-feedback ring (A) where a user can drag words that will penalize the documents. The angle of keywords relates to the similarity of neighboring keywords; keywords on the same angle in the inner circle and outer rings are similar.

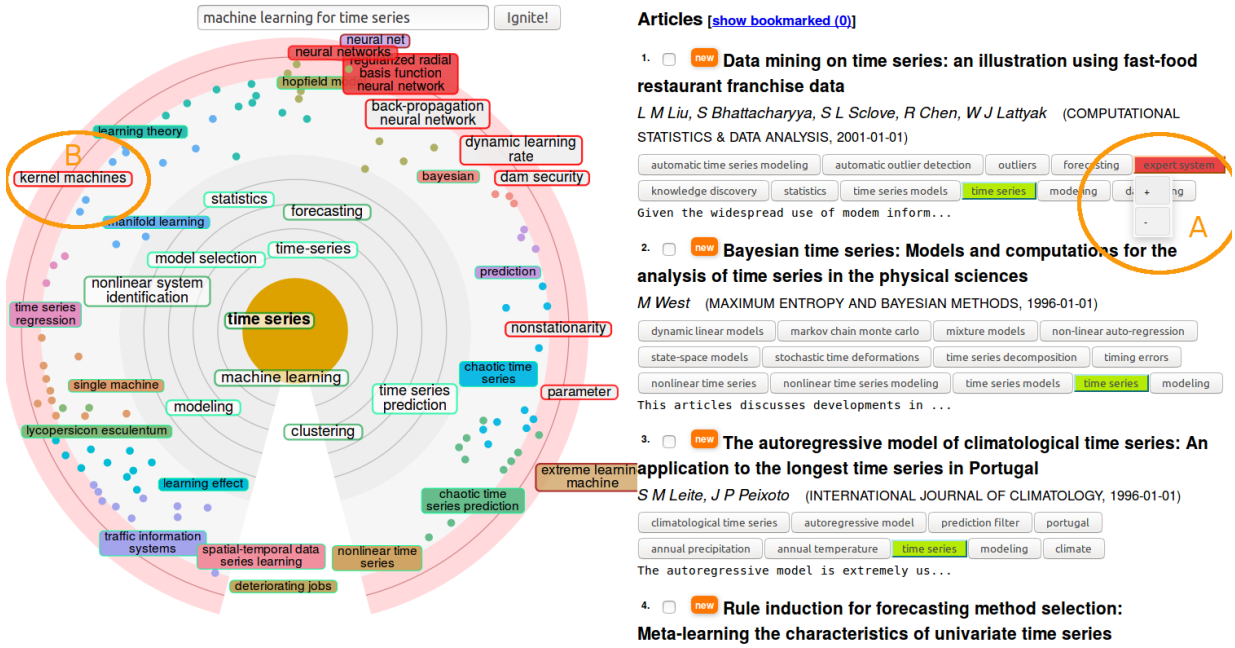

Figure 2. The Intent Radar after receiving negative feedback: On receiving negative feedback by dragging the keyword into the light-red outer ring or by giving negative feedback directly to keywords in the article list (A) the next update of the system will estimate relevance of keywords. Keywords with the lowest lower-confidence bounds of estimated relevance will be displayed in the red outer ring, e.g., the keyword "kernel machines" (B) is one of the ten negative intent keywords.

confidence bound (UCB) for relevance of a new keyword i is then $\mathbf{s}_i^\top \mathbf{r}^{feedback} + \frac{\alpha}{2}||s_i||$, where α adjusts the confidence level (exploration) and $\mathbf{s}_i = \mathbf{K}(\mathbf{K}^\top\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{k}_i$ with regularization parameter λ. High UCB identifies keywords having a chance to get high positive feedback; we also evaluate the lower confidence bound (LCB) as $\mathbf{s}_i^\top \mathbf{r}^{feedback} - \frac{\alpha}{2}||s_i||$, highly negative LCB identifies keywords having a chance to get high negative feedback. The top-10 keywords (with highest UCBs) and the bottom-10 (with most negative LCBs) will be used to score retrieved documents, and will be shown to the user on the interface, in the innermost and outermost rings of the radar display, respectively.

In addition to the current intent model described above, we evaluate alternative future intents: predicted keyword relevances following alternative choices of additional feedback. Future intents are used for well-organized visual presentation: keywords behaving similarly across future intents will be shown grouped together, as described later in this section. We evaluate 10 alternative intents k: In each, we add a further positive feedback (relevance score 1) to the feedback set for the kth of the top-10 keywords having highest UCB, and rerun the relevance estimation with the added feedback.

Multi-step keyword features. Feedback on a keyword should impact predictions for related keywords, but this may not happen with sparse features and few samples. The method of [25] suffers from this problem, which arises both in our negative feedback context and in general; we now describe the problem and introduce a solution. When relevance of a new keyword is predicted by Bayesian linear regression from its document occurrences, the prediction is affected only by those documents having keywords that have received feedback; many of them might not co-occur with the new keyword. Since TF-IDF matrices are sparse, this overreliance on direct co-occurrences ignores much of the information available in keyword feedback. For example, keywords receiving positive feedback may not directly co-occur in the same documents with those having negative feedback; with sparse TF-IDF features, the two keyword groups would not influence one another, which is mathematically visible as a diagonal block structure in $\mathbf{K}$. This effect may occur within subgroups of positive or negative feedback keywords as well, and is common when the number of feedbacks or documents is small; it harms ability to direct search by feedback and ability to present keyword relationships in a visualization.

We solve the problem by improved features based on graph transitions that spread keyword influence to "friends of friends of friends", as follows. Starting from an initial matrix $\mathbf{K}$ with each row $\mathbf{k}_i^\top$ as a TF-IDF vector over documents, we form a Markov transition matrix $\mathbf{P}$ between documents, so that $P_{ij} = \frac{[\mathbf{K}^\top\mathbf{K}]_{ij}}{\sum_j [\mathbf{K}^\top\mathbf{K}]_{ij}}$ for documents i and j, which is proportional to a sum of TF-IDF strengths over keywords co-occurring in the documents. We then form $\mathbf{P}_{multi} = 0.5\mathbf{I} + 0.25\mathbf{P} + 0.1875\mathbf{P}^2 + 0.0625\mathbf{P}^3$ representing a mixture of random walks up to 3 steps between documents; the weights are simply one possible gradual decay. We then replace sparse TF-IDF keyword features by $\mathbf{k}_i^{new} = \mathbf{P}_{multi}\mathbf{k}_i$ and $\mathbf{K}^{new} = \mathbf{K}\mathbf{P}_{multi}$ in the estimation equations. In the new representation, a keyword is associated to a document, if occurrences of the keyword are reachable from the document by several steps (here at most 3) through the document graph. This solves the influence problem, and essentially yields stronger effect for the user's feedback by allowing it to influence more of the keywords. Moreover, since we spread the influence through the current data-driven co-occurrence graph, the spread of influence is computed in a search-task specific way, instead of e.g. using naive dictionary relationships of keywords to spread their influence.

Document retrieval. The intent model yields two models: The set K_P of top-10 positive scoring keywords forms a positive model, with UCB scores v_i^P collected into a vector $\mathbf{v}_P$; and the set K_N of bottom-10 most negative scoring keywords forms a negative model, with LCB scores v_i^N (times -1 so that scores are positive) collected as $\mathbf{v}_N$. For document retrieval, they are considered as small samples of an idealized desired and undesired document respectively. We then use a language modeling approach to retrieve documents. For a document d, we obtain a unigram language model M_d as a normalized count from TF-IDF weights in the document, smoothed towards the population mean by Bayesian Dirichlet smoothing as in [25]. The document is then rewarded for being able to generate positive keywords and penalized for being able to generate negative ones; in other words, given a unigram language model M_d for the document, the document is scored by a likelihood ratio (or Bayes factor) between a hypothesis that the positive intent is generated from the document and a hypothesis that the negative intent is generated from the document, thus

$$score(d) = \log \frac{P(\mathbf{v}_P|M_d)}{P(\mathbf{v}_N|M_d)} \tag{1}$$

$$= \sum_{i\in K_P} v_i^P \log P(v_i^P|M_d) - \sum_{i\in K_N} v_i^N \log P(v_i^N|M_d)\,. \tag{2}$$

In a language model with positive feedback only such as in [25], low keyword feedback has relatively little effect: It affects which keywords end up in the top-10, and a document concentrating on other keywords would be penalized slightly since M_d then has less probability mass allocated to the top-10. In comparison, in the model above negative-scored keywords can much more directly and strongly penalize documents directly from user feedback.

Visualization of the intent model. A strength of the previous system by [25] is efficient comprehension of the intent model and browsing of feedback options through interactive visualization and we extend that for negative feedback. As described in *Intent modeling* earlier in this section, we predict relevances for keywords in 10 alternative future intents, each corresponding to an additional positive feedback for one of the top-scoring positive keywords. Each alternative yields a relevance score for all keywords: For each keyword i we collect the alternative scores in a 10×1 vector $\mathbf{r}_i^{future}$; we pick the ten top-scoring UCB keywords, ten most negative LCB keywords, and up to 300 keywords with highest norms of $\mathbf{r}_i^{future}$ for presentation on the radar, and normalize their vectors. We then apply a neighborhood-preserving dimensionality reduction based on the Neighbor Retrieval Visualizer

method (NeRV [28]; see also related methods in [17, 33, 34, 35]) on these high-dimensional keyword features as described in [25], reducing them to 1-dimensional coordinates shown as angles of the keywords in the radar display. The top-ten scoring positive keywords are shown in the central area and radial positions directly show their UCBs (closer to center is higher); the bottom-ten most negative keywords are shown in the outer red ring and radial positions correspond to their LCBs (towards the outer edge is more negative); and for the 300 keywords in the middle area we show the original norm of $\mathbf{r}_i^{future}$ as the radial position, indicating their average potential for future relevance. The difference to [25] is that we separately pick out negative keywords and dedicate a new outer ring of the radar to show them; by using an outermost ring we allow large angular resolution to show different types of potential negative intents, with a trade off of using less space for radial resolution.

EXPERIMENTS

The aim of the experiments was to compare the search efficiency and performance of two interactive exploratory search systems, the only difference between them being the addition of NRF as described in this paper. In order to test the benefits of negative relevance feedback it was necessary to construct difficult queries (where the majority of the top results were not relevant).

Task design

As we have pointed out, we do not claim negative feedback yields a notable advantage in all situations: For the simplest queries the initial results may already be good, and even if they are not, for many queries positive feedback suffices; moreover, in an exploratory interactive system where a large set of potential positive feedback is available, even more tasks may be solvable by well-chosen positive feedback than with a more traditional typed-query only system. However, we will show there are nevertheless situations where positive feedback still does not suffice and negative feedback is beneficial.

In order to see the benefits of negative relevance feedback it is necessary to find a difficult query, i.e., one where most or all of the top-ten documents are irrelevant to the search [30]. A study on real-world difficult queries has found some causes that may make a query difficult, however understanding the causes of difficult queries is challenging [5, 6, 7, 27]. Furthermore, predicting query difficulty is an active area of research for improving information retrieval performance [8, 13].

Notably, in an exploratory search system like the two compared systems, it is not enough for merely the initial state to contain few relevant documents: It must also be nontrivial to direct the search towards relevant documents. Thus, many of the above categories of causes for difficult queries were tried and failed to be difficult for either of the systems being tested using only positive feedback; the powerful interactive visualization available in both systems made some supposedly difficult queries easy to correct. For example, for synonyms, the keyword radar enabled the user to choose a keyword that would add the correct context to the ambiguous meaning of the word; for confounding terms finding related terms in the radar that were relevant could quickly correct the search; and many other initially difficult queries could quickly be corrected.

It was observed during trials of different types of difficult query that some topics shared many keywords and document terms with other topics and disjoining these topics was challenging. For example the topic "machine learning for time series" shared many keywords with the topic "machine learning for time series with support vector machines", it is likely that the use of support vector machines is very popular for applying machine learning to time series data so the majority of the papers are about both topics. Using the interactive exploratory search system to separate these topics was very challenging, as so many keywords appeared in the same documents for the main topic and the main topic with the sub topic. We took advantage of this in designing our tasks. After a lot of testing various tasks we used the following tasks (pre-defined initial queries in quotation marks): A. “machine learning for time series" not support vector machine (SVM) or artificial neural network, B. “metaheuristics for optimization" not biologically inspired algorithms and C. “machine learning for automatic time series" not SVM, hidden Markov models or Gaussian mixture models. These type of queries occur in real world tasks when searching for interesting and novel documents on a main topic and the results are mostly populated with common results, but the user is looking for a less common sub-topic rendering the popular results non-relevant.

Experiment Setup

Fifteen participants were given three scientific information seeking tasks to be performed on both systems.

Data set. Both systems used the same data source of approximately 50 million scientific articles from Web of Science, ACM, IEEE, and Springer.

Task definition. As discussed in the previous section, each task involved searching for a main topic, e.g., machine learning for time series, but not including uninteresting subtopics, e.g., support vector machines or artificial neural networks. The objective of each task was to make as many as possible of the top-ten articles relevant to the task, preferably all ten: That is, the articles should be relevant to the main topic but should not include any unwanted subtopics. We focused the study on how effective the user feedback was for improving the search: Users were thus told not to write new queries, but to find the relevant articles by starting from the given query and by manipulating the keywords in the visualization. Each task had a time limit of fifteen minutes, but users could stop earlier if they believed that the objective was met before the time limit.

Ordering of tasks. In total, each user performed six tasks (three task definitions on two systems). Our aim was to study differences of user performance between the two systems, while minimizing the effect of other variables such as learning a task or becoming fatigued. We created a different ordering of the six tasks for each of the 15 users, in order to have a balanced amount of occurrences for each combination of three variables that could cause undesirable learning or fatigue effects: Order of the task definitions (e.g., where task A precedes B and B precedes C), order in which each system is used for each task

(system N preceding system P and vice versa for each task) and the overall order of each task and system combination (Task B on system N is performed as the first, second, third etc. position).

To ensure fair comparison of systems we created a standard process for the experiment, described below, and ensured the hardware for both systems was equally powerful. Both systems ran on the same back-end hardware and software. The front-end hardware through which users accessed the interactive systems was also standardized: All users performed the experiments on equivalent laptops of the same specification (Apple MacBook Air, OS X El Capitan, 1.8 GHz Intel i7 processor, 4GB RAM) using the same laptop mouse pad. The experiments were conducted in a quiet room with good lighting and users sat at a table. For both systems, users received the same instructions, demonstration and time to familiarize themselves with the system as described below.

Stages of the experiment. We performed the experiments in sessions of up to three simultaneous participants. To ensure uniform instruction across sessions, each experiment was conducted following a written procedure and users were provided with the same detailed instructions, as follows.

1. Pre-experiment evaluation. At the beginning of each experiment session, each user first completed a pre-experiment evaluation to collect data on their pre-existing knowledge of computer science, information retrieval and each of the task topics.

2. System demonstration. Participants were then shown a short instructional video that demonstrated the functionality of the systems. Each user was then given a demonstration of the system, including the addition of the negative feedback area, as this is an addition to the other system. During the demo the user was given an explanation of the visualization with or without the negative feedback that included the placement of the keywords on the radar, the difference between the inner current user intent estimates and the future prediction estimates in the next outermost ring, the red outer ring for giving negative relevance feedback and the ability to give plus or minus full feedback to keywords in the article list.

3. Hands-on learning. Following the video, the users had five minutes to experiment with the system and ask any questions to clarify the workings of the system.

4. Task descriptions. First, the users opened a task feedback sheet for each task variant. For each task the user was then given approximately one-and-a-half pages of background information on the main topic and the subtopics to aid them in identifying keywords that were relevant and irrelevant. It was recommended to read the information prior to starting the task.

5. Performing the task. On starting a task variant, a fifteen minute countdown timer started, and a predefined query was run to initialize the search. All randomization in the system, e.g., for sampling initial keywords, were fixed with the same seed so that each user had the same starting point. Users were then free to use the interactive visualization to achieve the objective of the task.

6. Post-task evaluation of relevance. On completion of each task variant, users were asked for their belief of which documents were not relevant and what keywords helped to indicate relevance and irrelevance during the search. This was done to ensure that users' perception of which articles and keywords were relevant matched that of the experts.

7. Post-session questionnaire. When a user had completed the whole experiment (all tasks), the user was asked to fill a user experience questionnaire to get feedback on their experience using both systems. The questionnaire featured questions based on two well-established frameworks, for user-centric evaluation of recommender systems (ResQue) by [24] and SUS for usability [4]. The chosen questions are listed in Tab. 2.

Experiment evaluation

We hypothesize that for the chosen tasks, 1) despite the rich exploratory options for positive feedback available in the systems, the proposed system having negative feedback capability will still yield more relevant results than the baseline positive-feedback-only system; 2) the availability of negative feedback in the proposed system will allow users to reach good results earlier than in the baseline system; 3) the proposed system will yield more diverse relevant results than the baseline; and 4) users will have a better user experience with the proposed system.

When performing a task, each user starts the system with an initial set of document results that the user can iteratively improve; in each iteration the user can manipulate one or more keywords to send feedback to the system, and press the update button to trigger the system to retrieve another set of results based on the initial query and all the user feedback given up to that point. This process is repeated until the user wishes to finish the search session, or the fifteen-minute time limit is reached. We log all user interactions and in particular the set of documents seen in each iteration.

Expert rating of documents. For each of the three tasks, all documents seen by all users were rated by three experts, using the following scoring system, "certainly relevant"=4, "possibly relevant"=3, "possibly irrelevant"=2, "certainly irrelevant"=1. This four-grade system was used in place of a binary relevance since the search tasks were complex scientific search tasks and in some documents it was not possible to be certain about relevance based on the information visible to users and experts (title, abstract, and keywords). Agreement of the experts in a small subset of documents was checked. For any set of top-ten ranked documents returned for a user at any point during the search, we may then give an overall score for the document set simply as the mean expert rating over the ten documents, yielding a number between 1 (all documents are certainly irrelevant) and 4 (all documents are certainly relevant).

User experience ratings. On completion of the tasks the user is asked to give user-experience feedback to rate the two systems, see Tab. 2 for the list of questions. Each question used a Likert scale where responses were scored as "strongly disagree" = -2, "disagree" = -1, "neither agree nor disagree" = 0, "agree" = 1 and "strongly agree" = 2, except for a few questions that used

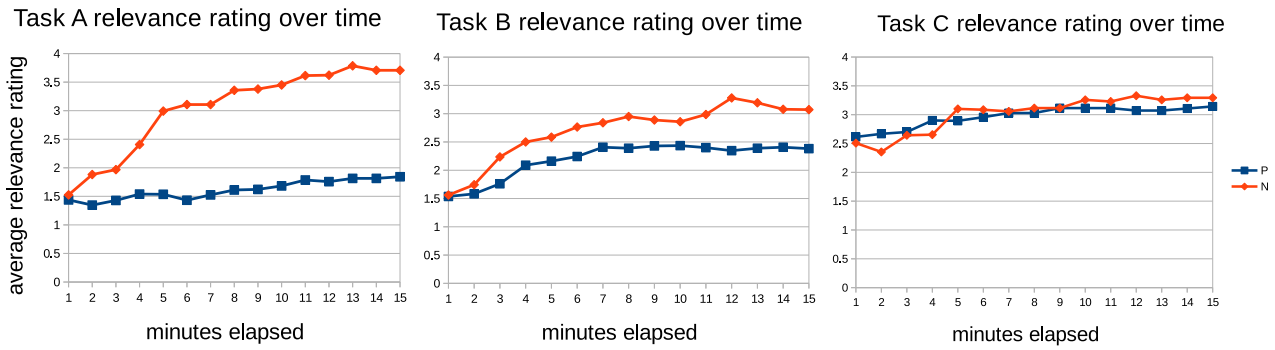

Figure 3. Efficiency between system P (positive only feedback) and system N (positive and negative feedback) as an average relevance rating of the top-ten articles over all users at each minute of the task. ANOVA p-value results for the final set of document relevance scores: task A = $1.5x10^{-8}$**, B=** 0.04**, C=**0.47**.**

the negation of this as agreeing to the question was a negative response.

To evaluate each system we use several measures:

1. Overall search performance. For each task, the total expert-given score of the final top-ten results, averaged over users, is used to evaluate the overall search performance of each system; see Tab. 1.

2. Efficiency. In addition to the final state of the system, we evaluate the change of the total expert-given score of the top-ten results over time (on each update throughout the search session) in order to quantify the efficiency of the search system, in the sense of how fast users are able to reach good results; see Fig. 3.

3. Diversity. We wish to evaluate whether negative feedback could yield more diverse relevant documents than positive feedback only. Given the set of relevant documents (those rated 3 or 4 by experts) found by all users together, we may perform a content-based clustering of the documents; here we use simple *k*-means on unigram feature vectors derived from either the title, keywords, or abstract, and cluster documents into 5 clusters. For any individual user, the relevant documents found at the final state then form a histogram over the content-based clusters which can be normalized to a multinomial distribution, and the diversity of the user's relevant documents can be evaluated by entropy of that distribution; see Tab. 1. Additionally we look at a subset of the questions from the user experience questionnaire that relate to the diversity and novelty of the documents to measure the user's opinion of the diversity of the results, see Tab. 1 for the results and see Tab. 2 for the subset of questions selected to measure user's opinion on diversity of each system.

Result Analysis

We see from the results of task A in Fig. 3 and Tab. 1 that NRF can indeed improve overall results and be more efficient than using positive feedback alone when the query is sufficiently difficult. For task B the evidence is less strong and for task C the evidence is weak. Finding situations when the query is difficult and positive feedback is not beneficial has been a challenge as these cases are not very common, but they do occur and we have certainly shown that with task A.

Our attempts to use *k*-means clustering on the title, keywords and abstract text in order to identify five clusters, group the relevant results into those clusters and count the entropy of those cluster counts did not show strong evidence of more diversity from the results of the system using positive and negative feedback, see Tab. 1. However, the feedback from the users, both in comments afterwards and in the feedback questionnaires point towards greater perceived diversity when using NRF. See Tab. 1 for the results and Tab. 2 for the questions on diversity.

Our system allows continuous-valued feedback, with discrete values available as extremes, and we found users made use of both. In detail, discrete-valued feedback results from selecting feedback as plus or minus from the article-list keywords (as shown in part A in Fig. 2) or by dragging a keyword in the radar (as shown in Fig. 1) either to the centre (positive, +1), to the future-predictions ring (indifferent, 0) or to the outside edge and beyond (negative, -1); continuous-valued feedback results from dragging keywords to intermediate positions. From the user logs we found that 65% of feedback was discrete-valued and a substantial 35% of feedback was continuous-valued.

The users were overall positive towards the system with NRF. In the questionnaire, the ratings were better for the variant using also negative feedback for *all* questions, except for the question on how quickly the user became familiar with the system (# 13). Interestingly, the questions that were most statistically significant related to the user's perception of their control of the system, so users seem to feel more in control of what the system does with NRF.

CONCLUSIONS AND DISCUSSION

We presented a novel system for exploratory search with both positive and negative continuous-valued feedback, based on machine learning solutions for modeling of user intent from the feedback, retrieval based on the modeled intent, and interactive visualization of the modeled intent. The system extends a previously proposed well-performing exploratory search system that provided positive feedback capability only. In user experiments on scientific information seeking we showed that

	Avg. rel.			Avg. entropy over topic-cluster counts									Questionnaire	
				Title			Keywords			Abstract				
Sys.	A	B	C	A	B	C	A	B	C	A	B	C	All	Div. rel.
P	2.0	2.5	3.2	0.9	0.8	1.0	0.9	0.8	0.5	0.9	0.7	0.6	0.0135	0.2922
N	3.8	2.9	3.4	1.1	0.8	0.8	0.9	0.70	0.3	0.8	0.7	0.4	0.1486	0.7987

Table 1. Sys.: Positive-only P and positive-and-negative N systems; Avg. rel.: Average rated relevance scores of all users for final top-ten documents for tasks A, B & C; Avg. entropy over topic-cluster counts: Average entropy over cluster membership counts of all relevant documents in the final top-ten for each user to measure diversity; Questionnaire: Average score of user-centric system evaluation for each system, results shown for all (All) responses and the subset of diversity-related questions (Div. rel.); for details on the grouping of the questions see Table 2

#	ext.	question	P	N
1	d	The keywords recommended to me matched my interests.	0.79	1
2	d	The articles recommended to me matched my interests.	0.43	1.07
3	d	**The system showed me new and interesting articles from which I was able to learn new and relevant information.***	0.5	1.29
4	dr	The articles recommended to me are similar to each other.	-0.43	-0.07
5	dr	I was only provided with articles on general topics.	0.64	0.71
6		**The system provides an adequate way for me to revise my preferences.****	-0.43	1.21
7		The system explains why the articles are recommended to me.	0.64	0.86
8		The information provided for the articles is sufficient for me.	0.64	0.86
9		The information provided for the keywords is sufficient for me.	0.29	0.43
10		The labels of the system interface are clear and adequate.	0.57	0.57
11		The design of the system interface (how to give feedback, how to move keywords and scroll through the articles) is clear and adequate.	0.93	0.79
12		The layout of the system interface (the positioning of the keywords, radar and article list) is attractive and adequate.	0.5	0.57
13		I became familiar with the system very quickly.	1.29	1.21
14		**I easily found the articles that were relevant to my search.***	0	0.79
15		**It is easy to learn to tell the system what I like.****	-0.29	1.14
16	d	I found it easy to make the system recommend different articles to me.	0.07	0.86
17	d	I found it easy to make the system recommend different keywords to me.	-0.21	0.36
18		**It is easy to train the system to update my preferences.***	-0.14	0.71
19		**It is easy for me to inform the system if I dislike the recommended articles.****	-0.64	1.07
20		**Using the system to find what I like is easy.***	0	0.79
21		I quickly became productive with the system.	0.14	0.79
22	dr	Finding a relevant article, even with the help of the system, consumes too much time.	0	0.64
23		**The recommended keywords effectively helped me find relevant articles.***	0.64	1
24		**I feel supported to find what I like with the help of the system.****	0.07	0.79
25		**I feel in control of telling the system what I want.****	-0.5	0.79
26		I understood why the articles were recommended to me.	0.14	0.714
27	r	**The system seems to control my decision process rather than me.***	-0.43	0.57
28	d	**Overall, I am satisfied with the recommender.***	0.29	0.86
29	d	I am convinced the articles recommended to me are relevant to the topic.	0.64	1.14
30	d	I am confident I will benefit from the articles recommended to me.	0.5	0.93
31	r	The recommended articles made me confused about my choice.	0.2	0.64
32	r	The recommended keywords made me confused about my choice.	-0.074	0.43
33		**The system can be trusted.***	0.36	0.93
34		**If a system such as this exists, I will use it to find scientific articles.***	0.43	1.14
35		**I will use this system again if given the opportunity.****	0.36	1.14
36		**I will use this type of system frequently if given the opportunity***.	-0.07	0.57
37		I will tell my friends about this system.	0.43	0.79

Table 2. Post-experiment user experience questions: The extra information (*ext.*) identifies questions that are used to measure diversity 'd' and questions for which the answer score is negated 'r'; thus larger numbers correspond to better agreement with our hypotheses. One-way ANOVA was run for each question between the 14 user responses for each system, highlighted above are responses with p-value $< 0.05 = *$ and $< 0.01 = **$. The average score over all users for each system is in the last two columns, P as positive-feedback only system and N as positive and negative feedback.

for some types of difficult information seeking tasks, negative feedback is beneficial even in an exploratory system where a wide range of positive feedback is available: The proposed system outperformed the previous state of the art system on performance, evolution of the performance over time, and in user experience. The results show that negative relevance feedback is beneficial for some difficult queries, even in interactive exploratory search systems whose user model allows continuous-valued relevance feedback on document features, and hence allows more specific feedback than feedback on a document only.

It is an encouraging sign that even after relatively little exposure to the features of the exploratory search system, users were able to benefit from negative feedback; analysis of a learning curve could be done as a follow-up study.

Our multi-step keyword features used document-keyword structure to enrich the feature space for relevance estimation to increase the relevance influence of a small amount of feedback with a sparse feature matrix. This brings about the question if other feature spaces may further improve relevance estimation in this domain.

Our system handles continuous valued relevance feedback, that was shown to be used by users, although to a lesser extent than discrete-valued feedback. An interesting follow-up study would be to analyse the use of the continuous valued feedback, e.g., is discrete value used initially and then continuous valued corrections made to refine the search?

Interestingly, a clear winner in diversity of relevant documents was not found, indicating potential directions of further improvements.

ACKNOWLEDGMENTS

Authors belong to Helsinki Insitute for Information Technology HIIT and the COIN centre of excellence. The work was funded in part by TEKES (Re:Know2 project) and in part by Academy of Finland decisions 252845, 256233, and 295694.

REFERENCES

1. S. Andolina, K. Klouche, J. Peltonen, M. Hoque, T. Ruotsalo, D. Cabral, A. Klami, D. Glowacka, P. Floreen, and G. Jacucci. 2015. IntentStreams: Smart Parallel Search Streams for Branching Exploratory Search. In *Proc. IUI'15*. 300–305.
2. P. Auer. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* 3 (2002), 397–422.
3. N.J. Belkin, J. Perez Carballo, C. Cool, S. Lin, S. Y. Park, S.Y. Rieh, P. Savage, C. Sikora, H. Xie, and J. Allan. 1998. Rutgers' TREC-6 Interactive Track Experience. In *Proc. TREC-6*. NIST, 597–610.
4. J. Brooke and others. 1996. SUS-A quick and dirty usability scale. *Usability Evaluation in Industry* 189, 194 (1996), 4–7.
5. C. Buckley. 2004. Why Current IR Engines Fail. In *Proc. SIGIR'04*. ACM, 584–585.
6. C. Buckley. 2009. Why current IR engines fail. *Information Retrieval* 12, 6 (2009), 652–665.
7. D. Carmel, E. Yom-Tov, A. Darlow, and D. Pelleg. 2006. What Makes a Query Difficult?. In *Proc. SIGIR'06*. ACM, 390–397.
8. S. Cronen-Townsend, Y. Zhou, and W. B. Croft. 2002. Predicting Query Performance. In *Proc. SIGIR'02*. ACM, 299–306.
9. A. Diriye, G. Kumaran, and J. Huang. 2012. *Interactive Search Support for Difficult Web Queries*. Springer, 37–49.
10. M. D. Dunlop. 1997. The Effect of Accessing Nonmatching Documents on Relevance Feedback. *ACM Transactions on Information Systems* 15, 2 (April 1997), 137–153.
11. M. A. Hearst. 2009. *Search User Interfaces* (1st edition ed.). Cambridge University Press.
12. M. A. Hearst and J. O. Pedersen. 1996. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *Proc. SIGIR'96*. ACM, 76–84.
13. Y. Huang, T. Luo, X. Wang, K. Hui, W.-J. Wang, and B. He. 2014. *On Evaluating Query Performance Predictors*. Springer, 184–194.
14. K.-P-Yee, K. Swearingen, K. Li, and M. Hearst. 2003. Faceted metadata for image search and browsing. In *Proc. CHI'03*. ACM, 401–408.
15. M. Karimzadehgan. 2012. *Systematic Optimization of Search Engines for Difficult Queries*. Ph.D. Dissertation. Champaign, IL, USA. Advisor(s) Zhai, Chengxiang. AAI3600414.
16. M. Karimzadehgan and C. Zhai. 2011. Improving Retrieval Accuracy of Difficult Queries Through Generalizing Negative Document Language Models. In *Proc. CIKM 2011*. ACM, 27–36.
17. S. Kaski and J. Peltonen. 2011. Dimensionality Reduction for Data Visualization. *IEEE Signal Processing Magazine* 28, 2 (2011), 100–104.
18. D. Kelly, K. Gyllstrom, and E. W. Bailey. 2009. A comparison of query and term suggestion features for interactive searching. In *Proc. SIGIR'09*. ACM, 371–378.
19. M.L. Kherfi, D. Ziou, and A. Bernardi. 2003. Combining positive and negative examples in relevance feedback for content-based image retrieval. *Journal of Visual Communication and Image Representation* 14, 4 (2003), 428 – 457.
20. J. Liu, C. S. Kim, and C. Creel. 2015. Exploring Search Task Difficulty Reasons in Different Task Types and User Knowledge Groups. *Information Processing & Management* 51, 3 (2015), 273–285.
21. Y. Ma and H. Lin. 2014. A Multiple Relevance Feedback Strategy with Positive and Negative Models. *PLoS ONE* 9, 8 (08 2014), 1–10.

22. C. D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.
23. J. Peltonen, K. Belorustceva, and T. Ruotsalo. 2017. Topic-Relevance Map: Visualization for Improving Search Result Comprehension. In *Proc. IUI'17*.
24. P. Pu, L. Chen, and R. Hu. 2011. A User-centric Evaluation Framework for Recommender Systems. In *Proc. RecSys'11*. ACM, 157–164.
25. T. Ruotsalo, J. Peltonen, M. Eugster, D. Glowacka, K. Konyushkova, K. Athukorala, I. Kosunen, A. Reijonen, P. Myllymäki, G. Jacucci, and S. Kaski. 2013. Directing exploratory search with interactive intent modeling. In *Proc. CIKM 2013*. ACM, 1759–1764.
26. I. Ruthven and M. Lalmas. 2003. A Survey on the Use of Relevance Feedback for Information Access Systems. *Knowl. Eng. Rev.* 18, 2 (June 2003), 95–145.
27. J. Savoy. 2007. Why Do Successful Search Systems Fail for Some Topics. In *Proc. SAC'07*. ACM, 872–877.
28. J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski. 2010. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research* 11 (2010), 451–490.
29. X. Wang, H. Fang, and C. Zhai. 2007. Improve Retrieval Accuracy for Difficult Queries Using Negative Feedback. In *Proc. CIKM 2007*. ACM, 991–994.
30. X. Wang, H. Fang, and C. Zhai. 2008. A Study of Methods for Negative Relevance Feedback. In *Proc. SIGIR'08*. ACM, 219–226.
31. C. Wongchokprasitti, J. Peltonen, T. Ruotsalo, P. Bandyopadhyay, G. Jacucci, and P. Brusilovsky. 2015. User Model In a Box: Cross-System User Model Transfer for Resolving Cold Start Problems. In *Proc. UMAP'15*. 289–301.
32. T. Yamamoto, S. Nakamura, and K. Tanaka. 2007. Rerank-by-Example: Efficient Browsing of Web Search Results. In *Proc. DEXA 2007*. 801–810.
33. Z. Yang, J. Peltonen, and S. Kaski. 2013. Scalable Optimization of Neighbor Embedding for Visualization. In *Proc. ICML'13*. 127–135.
34. Z. Yang, J. Peltonen, and S. Kaski. 2014. Optimization Equivalence of Divergences Improves Neighbor Embedding. In *Proc. ICML'14*. 460–468.
35. Z. Yang, J. Peltonen, and S. Kaski. 2015. Majorization-Minimization for Manifold Embedding. In *Proc. AISTATS'15*. 1088–1097.
36. C. Zhai. 2008. Statistical Language Models for Information Retrieval. *Synthesis Lectures on Human Language Technologies* 1, 1 (2008), 1–141.