uRank: Exploring Document Recommendations through an Interactive User-Driven Approach

Cecilia di Sciascio
Know-Center GmbH
Graz, Austria
cdisciascio@know-center.at

Vedran Sabol
Know-Center GmbH
Graz, Austria
vsabol@know-center.at

Eduardo Veas
Know-Center GmbH
Graz, Austria
eveas@know-center.at

ABSTRACT

Whenever we gather or organize knowledge, the task of searching inevitably takes precedence. As exploration unfolds, it becomes cumbersome to reorganize resources along new interests, as any new search brings new results. Despite huge advances in retrieval and recommender systems from the algorithmic point of view, many real-world interfaces have remained largely unchanged: results appear in an infinite list ordered by relevance with respect to the current query. We introduce *uRank*, a user-driven visual tool for exploration and discovery of textual document recommendations. It includes a view summarizing the content of the recommendation set, combined with interactive methods for understanding, refining and reorganizing documents on-the-fly as information needs evolve. We provide a formal experiment showing that *uRank* users can browse the document collection and efficiently gather items relevant to particular topics of interest with significantly lower cognitive load compared to traditional list-based representations.

General Terms

Theory

Keywords

recommending interface, exploratory search, visual analytics, sensemaking

1. INTRODUCTION

With the advent of electronic archival, seeking for information occupies a large portion of our daily productive time. Thus, the skill to find and organize the right information has become paramount. Exploratory search is part of a discovery process in which the user often becomes familiar with new terminology in order to filter out irrelevant content and spot potentially interesting items. For example, after inspecting a few documents related to robots, sub-topics like human-robot interaction or virtual environments could attract the user's attention. Exploration requires careful inspection of at least a few titles and abstracts, when not full documents, before becoming familiar with the underlying topic. Advanced search engines and recommender systems (RS) have grown as the preferred solution for contextualized search by narrowing down the number of entries that need to be explored at a time.

Traditional information retrieval (IR) systems strongly depend on precise user-generated queries that should be iteratively reformulated in order to express evolving information needs. However, formulating queries has proven to be more complicated for humans than plainly recognizing information visually [6]. Hence, the combination of IR with machine learning and HCI techniques led to a shift towards – mostly Web-based – browsing search strategies that rely on on-the-fly selections, navigation and trial-and-error [15]. As users manipulate data through visual elements, they are able to drill down and find patterns, relations or different levels of detail that would otherwise remain invisible to the bare eye [32]. Moreover, well-designed interactive interfaces can effectively address information overload issues that may arise due to limited attention span and human capacity to absorb information at once.

Sometimes RS can be more limited than IR systems if they do not tackle trust factors that may hinder user engagement in exploration. As Swearingen et al. [27] pointed out in their seminal work, the RS has to persuade the user to try the recommended items. To fulfill such challenge not only the recommendation algorithm has to fetch items effectively, but also the user interfaces must deliver recommendations in a way that they can be compared and explained [22]. The willingness to provide feedback is directly related to the overall perception and satisfaction the user has of the RS [13]. Explanatory interfaces increase confidence in the system (trust) by explaining how the system works (transparency) [28] and allowing users to tell the system when it is wrong (scrutability) [11]. Hence, to warrant increased user involvement the RS has to justify recommendations and let the user customize their generation.

In this work we focus mainly on transparency and controllability aspects and, to some extent, on predictability as well. *uRank* [1] is and interactive user-driven tool that supports exploration of textual document recommendations through:

i) an automatically generated overview of the document collection depicted as augmented keyword tags,

ii) a drag-and-drop-based mechanism for refining search interests, and

iii) a transparent stacked-bar representation to convey document ranking and scores, plus query term contribution. A user study revealed that *uRank* incurs in lower workload compared to a traditional list representation.

[1] http://eexcessvideos.know-center.tugraz.at/urank-demo.mp4

2. RELATED WORK

2.1 Search Result Visualization

Modern search interfaces assist user exploration in a variety of ways. For example, query expansion techniques like *Insyder*'s Visual Query [21] address the query formulation problem by leveraging stored related concepts to help the user extend the initial query. Tile-based visualizations like *TileBars* [7] and *HotMap* [9] make an efficient use of space to convey relative frequency of query terms through – gray or color – shaded squares, and in the case of the former, also their distribution within documents and relative document length. This paradigm aims to foster analytical understanding of Boolean-type queries, hence they do not yield any rank or relevance score. All these approaches rely on the user being able to express precise information needs and do not support browsing-based discovery within the already available results.

Faceted search interfaces allow for organizing or filtering items throughout orthogonal categories. Despite being particularly useful for inspecting enriched multimedia catalogs [33, 23], they require metadata categories and hardly support topic-wise exploration.

Rankings conveying document relevance have been discouraged as opaque an under-informative [7]. However, the advantage of ranked lists is that users know where to start their search for potentially relevant documents and that they employ a familiar format of presentation. A study [24] suggests that: *i)* users prefer bars over numbers or the absence of graphical explanations of relevance scores, and *ii)* relevance scores encourage users to explore beyond the first two results. As a tradeoff, lists imply a sequential search through consecutive items and only a small subset is visible at a given time, thus they are mostly apt for sets no larger than a few tens of documents. Focus+Context and Overview+Detail techniques [20, 9] sometimes help overcome this limitation while alternative layouts like *RankSpiral*'s [25] rolled list can scale up to hundreds and maybe thousands of documents. Other approaches such as *WebSearchViz* [16] and *ProjSnippet* [3] propose complementary visualizations to ordered lists, yet unintuitive context switching is a potential problem when analyzing different aspects of the same document.

Although ranked list are not a novelty, our approach attempts to leverage the advantages provided by lists; i.e. user familiarity, and augment them with stacked-bar charts to convey document relevance and query term contribution in a transparent manner. *Insyder*'s bar graph [21] is an example of augmented ranked lists that displays document an keyword relevance relevance with disjoint horizontal bars aligned to separate baselines. Although layered bar dispositions are appropriate for visualizing distribution of values in each category across items, comparison of overall quantities and the contribution of each category to the totals is better supported by stacked-bar configurations [26]. Additionally, we rely on interaction as the key to provide controllability over the ranking criteria and hence support browsing-based exploratory search.

LineUp [4] has proven the simplicity and usefulness of stacked bars to represent multi-attribute rankings. Despite targeting data of different nature – *uRanks*'s domain is rather unstructured with no measurable attributes –, the visual technique itself served as inspiration for our work.

2.2 Recommending Interfaces

In recent years, considerable efforts have been invested into leveraging the power of social RS through visual interfaces [17, 12]. As for textual content, *TalkExplorer* [29] and *SetFusion* [18] are examples of interfaces for exploration of conference talk recommendations. The former is mostly focused in depicting relationships among recommendations, users and tags in a transparent manner, while *SetFusion* emphasizes controllability over a hybrid RS. Rankings are not transparent though, as there is no explanation as to how they were obtained. Kangasraasio et al. [10] highlighted that not only allowing the user to influence the RS is important, but also adding predictability features that produce an effect of causality for user actions.

With *uRank* we intend to enhance predictability through document hint previews (section 3.1.1), allow the user to control the ranking by choosing keywords as parameters, and support understanding by means of a transparent graphic representation for scores (section 3.2).

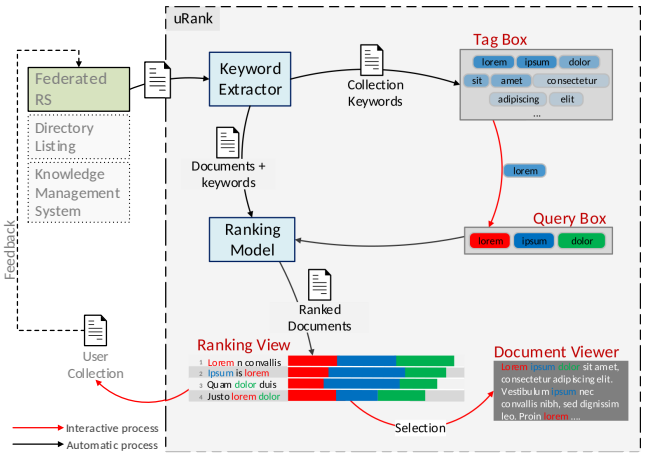

Figure 1: *uRank* visual analytics workflow showing automatic (black arrows) and interactive mechanisms (red arrows)

3. URANK VISUAL ANALYTICS

uRank is a visual analytics approach that combines lightweight text analytics and an augmented ranked list to assist in exploratory search of textual recommendations. The Web-based implementation is fed with textual document surrogates by a federated RS (F-RS) connected to several sources. A keyword extraction module analyzes all titles and abstracts and outputs a set of representative terms for the whole collection and for each document. The UI allows users to explore the collection content and refine information needs in terms of topic keywords. As the user selects terms of interest, the ranking is updated, bringing related documents closer to the top and pushing down the less relevant ones. Figure 1 outlines the workflow between automatic and interactive components.

uRank's layout is arranged in a multiview fashion that displays different perspectives of the document recommendations. Following Baldonados's guidelines [30], we decided to limit the number of views to keep display space requirements relatively low. Therefore, instead of multiple overlapping views, we favor a reduced number of perspectives fitting in any laptop or desktop screen. The GUI dynamically scales to the window size, remaining undistorted up to a screen width of approximately 770 px.

The GUI presents the data in juxtaposed views that add to a semantic Overview+Detail scheme [2] with three levels of granularity: *Collection overview.* The *Tag Box* (Figure 2.A) summarizes the entire collection through by representing keywords as augmented tags. *Documents overview.* The *Document List* shows titles augmented with ranking information and the *Ranking View* displays stacked bar charts depicting document relevance scores (Figure 2.C and D, respectively). Together they represent mini-

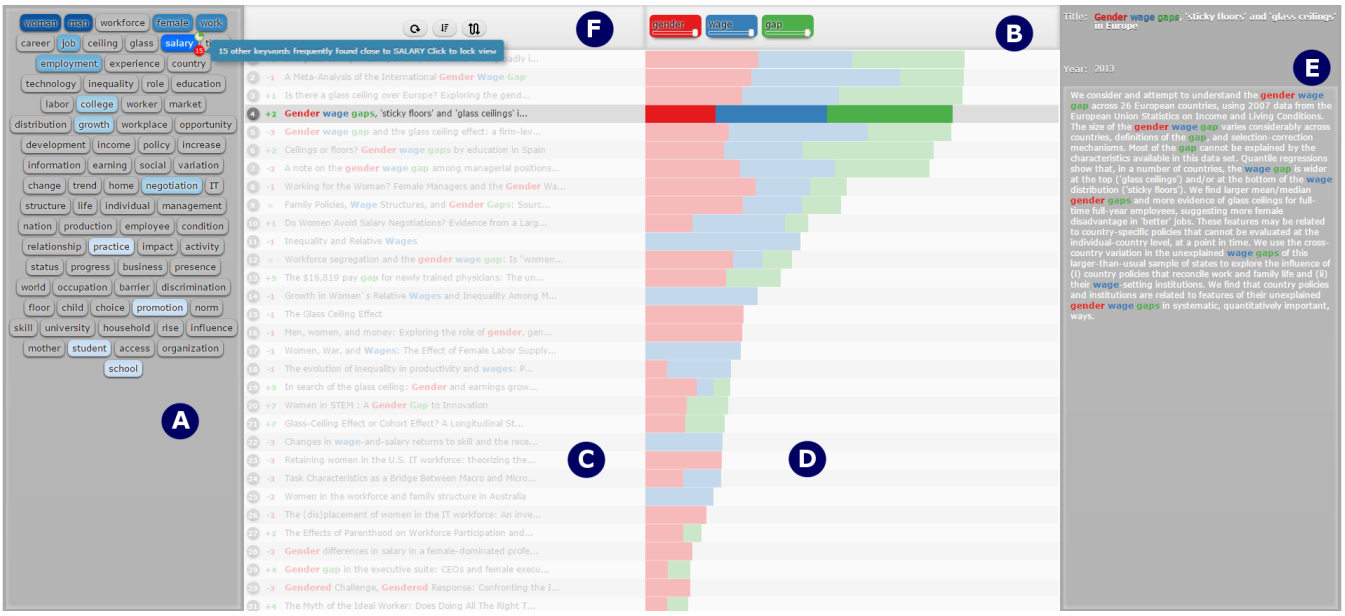

Figure 2: *uRank* User Interface displaying a ranking of documents for the keywords "gender", "wage" and "gap". The user has selected the third item in the list. A. The *Tag Box* presents a keyword-based summary, B. the *Query Box* contains the selected keywords that originated the current ranking state, C. the *Document List* presents a list with augmented document titles. D. the *Ranking View* renders stacked bars indicating relevance scores, E. the *Document Viewer* shows the title, year and snippet of the selected document with augmented keywords, and F. the *Ranking Controls* wrap buttons for ranking settings.

mal views of documents where they can be differentiated by title or position in the ranking and compared at a glance basing on the presence of certain keywords of interest. *Document detailed view.* For a document selected in the list, the *Document Viewer* (Figure 2.E) displays the title and snippet with color-augmented keywords.

These views can be modified through interaction with the *Ranking Controls* (Figure 2.F) and the *Query Box* (Figure 2.B). The former provides controls to reset the ranking or switch ranking modes between overall and maximum score. The latter is the container where the user drops keywords tags to trigger changes in the ranking visualization.

3.1 Collection Overview

uRank automatically extracts keywords from the recommended documents with a twofold purpose: i) give an overview of the collection, and ii) provide manipulable elements that serve as input for an on-the-fly ranking mechanism (see section 3.2).

Summarizing the collection in a few representative terms allows the user to scan the recommendations and grasp the general topic at a glance, before even reading any of them. This is particularly important in the context of collections brought by RS, where the user is normally not directly generating the queries that feed the search engine.

3.1.1 Inspecting the Collection

The *Tag Box* provides a summary of the recommended texts as a whole by presenting extracted keywords as tags. Keywords tags are arranged in a bag-of-words fashion, encoding relative frequencies through position and intensity (Figure 2.A). The descending ordering conveys document frequency (DF) while five levels of blue shading help the user identify groups of keywords in the same frequency range. Redundant coding is intentional and aims at maximizing distinctiveness among items in the keyword set [32].

At first glance, the *Tag Box* gives an outline of the covered topic in terms of keywords and their relative frequencies. Nevertheless, a bag-of-words representation per se does not supply further details about how a keyword relates to other keywords or documents. We bridge this gap by augmenting tags with two compact visual hints – visible on mouse over – that reveal additional information: *i)* co-occurence respect to other keywords, and *i)* a preview of the effect of selecting the keyword.

The document hint (Figure 3) consists in a pie chart that conveys the proportion of documents in which the keyword appears. A tooltip indicates the exact quantity and percentage. Upon clicking on the document hint, unrelated documents are dimmed so that documents containing the keyword remain in focus Even unranked documents become discretely visible at the bottom of the *Document List*. This hint provides certain predictability regarding the effect of selecting a keyword, in terms of which ranked items will change their scores and which documents will be added to the ranking.

The co-occurrence hint (Figure 2.A) shows the number of frequently co-occurring keywords in a red circle. Moving the mouse pointer over it brings co-occurring terms to focus by dimming the others in the background. Clicking on the visual hint locks the view so that the user can hover over co-occurring keywords, which shows a tooltip stating the amount of co-occurrences between the hovered and the selected keyword. This hint supports the user in finding possible key phrases and sub-topics within the collection.

3.1.2 Mining a collection of documents

The aforementioned interactive features are supported by a combination of well-known text-mining techniques that extend the recommended documents with document vectors and provide meaningful terms to populate the *Tag Box*.

Document vectors ideally include only content-bearing terms like nouns and frequent adjectives – appearing in at least 50% of the collection –, hence it is not enough to just rely on a list of stop words

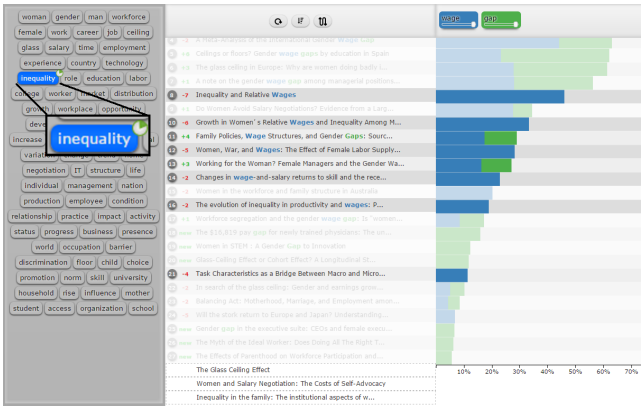

Figure 3: Document hints show a preview of documents containing the hovered keyword, even if they are currently unranked

to remove meaningless terms. Firstly, we perform a part-of-speech tagging (POS tagging) [1] step to identify words that meet our criteria, i.e. common and proper nouns and adjectives. Filtering out non-frequent adjectives requires an extra step. Then, plural nouns are singularized, proper nouns are kept capitalized and terms in upper case, e.g. "IT", remain unchanged. We apply the Porter Stemmer method [19] over the resulting terms, in order to increase the probability of matching for similar words, e.g. "robot", "robots" and "robotics" all match the stem "robot". A document vector is thus conformed by stemmed versions of content-bearing terms.

Next, we generate a weighing scheme by computing TF-IDF (term frequency inverse document frequency) for each term in a document vector. The score is a statistical measure of how important the term is to a document in a collection. Therefore, the more frequent a term is in a document and the fewer times it appears in the corpora, the higher its score will be. Documents' metadata are extended with these weighted document vectors.

To fill the *Tag Box* with representative keywords for the collection set, all document keywords are collected in a global keyword set. Global keywords are sorted by document frequency (DF), i.e. the number of documents in which they appear, regardless of the frequency within documents. To avoid overpopulating the *Tag Box*, only terms with DF above certain threshold (by default 5) are taken into account. Note that terms used to label keyword tags are actual words and not plain stems. Scanning a summary of stemmed words would turn unintuitive for users. Thus, we keep a record of all term variations matching each stem, in order to allow for reverse stemming and pick one representative word as follows:
1. if there is only one term for a stem, use it to label the tag,
2. if a stem has two variants, one in lower case and the other in upper case or capitalized, use it in lower case,
3. use a term that ends in 'ion', 'ment', 'ism' or 'ty',
4. use a term matching the stem,
5. use the shortest term.

To feed the document hint (Figure 3), *uRank* attaches a list of bearing documents to each global keyword. For the case of co-occurrence hints (Figure 2.A), keyword co-occurrences with a maximum word distance of 5 and a minimum of 4 repetitions are recorded.

3.2 Ranking Documents On The Fly

In theory, recommendations returned by a RS are already ranked by relevance. However, in practice the lack of control thereof could hinder user engagement if the GUI does not provide enough ratio-

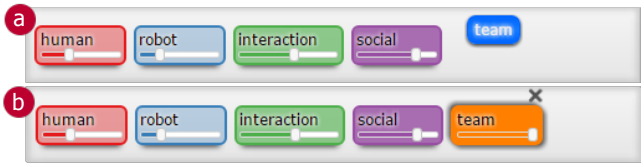

Figure 4: a) Keyword tag before being dropped in Tag Box. b) Keyword tag after dropped: weight slider and delete button added, background color changed according to a categorical color scale. Weight sliders have been tuned.

nale for the recommendations and features for shaping the recommendation criteria. Hence, one of *uRank*'s major features is the user-driven mechanism for re-organizing documents as information needs evolve, along with its visually transparent logic.

3.2.1 Ranking Visualization

The ranking-based visualization consists of a list of document titles (Figure 2.C) and stacked bar charts (Figure 2.D) depicting rank and relevance scores for documents and keywords within them. Document titles are initially listed following the order in which they were supplied by the F-RS.

Interactions with the view are the means for users to directly or indirectly manipulate the data [31]. In *uRank*, changes in the ranking visualization originate from keyword tag manipulation inside the *Query Box* (Figure 2.B). As the user manipulates tags, selected keywords are immediately forwarded to the *Ranking Model* as ranking parameters. Selected tags are re-rendered by adding a weight slider, a delete button on the right-upper corner – visible on hover – and a specific background color determined by a qualitative palette (Figure 4). We chose Color Brewer's [5] *9-class Set 1* palette for background color encoding, as it allows the user to clearly distinguish tags from one another. When the user adjusts a weight slider, the intensity of the tag's background color changes accordingly (see Figure 4). We provide three possibilities for keyword tag manipulation:

- **Addition**: keyword tags in the *Tag Box* can be manually unpinned (Figure 4a), dragged with the mouse pointer and dropped into the *Query Box* (Figure 4b).
- **Weight change**: tags in the *Query Box* contain weight sliders that can be tuned to assign a keyword a higher or lower priority in the ranking.
- **Deletion**: tags can be removed from the *Query Box* and returned to their initial position in the *Tag Box* by clicking on the delete button.

As the document ranking is generated, the *Document List* is resorted in descending order by overall score and stacked bars appear in the *Ranking View*, horizontally aligned to each list item. Items with null score are hidden, shrinking the list size to fit only ranked items. The total width of stacked bars indicates the overall score of a document and bar fragments represent the individual contribution of keywords to the overall score. Bar colors match the color encoding for selected keywords in the *Query Box*, enabling the user to make an immediate association between keyword tags and bars. Missing colored bars in a stack denote the absence of certain words in the document surrogate. Additionally, each item in the Document List contains two types of numeric indicators: the first one - in a dark circle - shows the position of a document in the ranking while the adjacent colored number reveals how many positions

Figure 5: Ranking visualization in maximum score mode: documents are ranked basing on the keyword with highest score

the document has shifted, encoding upward and downward shifts in green and red, respectively. This graphic representation attempts to help the user concentrate only on useful items and ignore the rest by bringing likely relevant items to the top, pushing less relevant ones to the bottom and hiding those that seem completely irrelevant.

uRank allows for choosing between two ranking modes: overall score (selected by default) and maximum score (Figure 5). In maximum score mode, the Ranking View renders a single color-coded bar per document in order to emphasize its most influential keyword. Finally, resetting the visualization clears the *Query Box* and the *Ranking View*, relocating all selected keywords in the *Tag Box* and restoring the *Document List* to its initial state.

3.2.2 *Document Ranking Computation*

Quick content exploration in *uRank* depends on its ability to readily re-sort documents according to changing information needs. As the user manipulates keyword tags and builds queries from a subset of the global keyword collection, *uRank* computes documents scores to arrange them accordingly in a document ranking. We assume that some keywords are more important to the topic model than others and allow the user to assign weights to them.

Document scores are relevance measures for documents respect to a query. As titles and snippets are the only content available for retrieved document surrogates, these scores are computed with a term-frequency scheme. Term distribution schemes are rather adequate for long or full texts and are hence out of our scope. Boolean models have the disadvantages that they not only consider every term equally important but also produce absolute values that preclude document ranking.

The *Ranking Model* implements a vector space model to compute document-query similarity using the document vectors previously generated during keyword extraction (section 3.1.2). Nonetheless, a single relevance measure like cosine similarity alone is not enough to convey query-term contribution, given that the best overall matches are not necessarily the ones in which most query terms are found [7, 14]. The contribution that each query term adds to the document score should be clear in the visual representation, in order to give the user a transparent explanation as to why a document ranks in a higher position than another. Therefore, we break down the cosine similarity computation and obtain individual scores for each query term, which are then added up as an overall relevance score.

Given a document collection D and a set of weighted query terms T , such that $\forall t \in T : 0 \leq w_t \leq 1$; the relevance score for term t in document vector $d \in D$ respect to query terms T is calculated as follows:

$$s(t_d) = \frac{tfidf(t_d) \times w_t}{|d| \times \sqrt{|T|}},$$

where $tfidf(t_d)$ is the tf-idf score for term t in document d and $|d|$ is the norm for vector d.

The overall score of a document $S(d)$ is then computed as the sum of each individual term score $s(t_d)$. The collection D is next sorted in descending order by overall score with the quicksort algorithm and ranking positions are assigned in such way that documents with equivalent overall score share the same place.

Alternatively, users can rank documents by maximum score, in which case $S(d) = \max(s(t_d))$.

3.3 Details on Demand

Once the user identifies documents that seem worth further inspecting, the next logical step is to drill down one by one to determine whether the initial assumption holds. The *Document Viewer* (Figure 2.D) gives access to textual content - title and snippet - and available metadata for a particular document. Query terms are highlighted in the text following the same color coding for tags in the *Query Box* and stacked bars in the *Ranking View*. These simple visual cues pop out from their surroundings, enabling the user to preattentively recognize keywords in the text and perceive their general context prior to conscious reading.

3.4 Change-Awareness Cues and Attention Guidance

We favor the use of animation to convey ranking-state transitions rather than abrupt static changes. Animated transitions are inherently intuitive and engaging, giving a perception of causality and intentionality [8]. As the user manipulates a keyword tag in the *Query Box*, *uRank* raises change awareness in the following way:

- Keyword tags are re-styled as explained in section 3.2.1. If the tag is removed from the *Query Box*, animation is used to shift the tag to its original position in the *Tag Box* at a perceivable pace.
- Depending on the type of ranking transition, the *Document List* shows a specific effect:
  - If the ranking is generated for the first time, an accordion-like upward animation shows that its nature has changed from a plain list to a ranked one.
  - If the ranking is updated, list items shift to their new positions at a perceptible pace.
  - If ranking positions remain unchanged, the list stays static as a soft top-down shadow crosses it.
- Green or red shading effects are applied on the left side of list items moving up or down, respectively, disappearing after a few seconds.
- Stacked bars grow from left to right revealing new overall and keyword scores.

The user can closely follow how particular documents shift positions by clicking on the watch - eye-shaped - icon. The item is brought to focus as it is surrounded with a slightly darker shadow and the title is underlined. Also, watched documents remain on top of the z-index during list animations, avoiding being overlaid by other list items.

The same principle of softening changes is applied to re-direct user attention when a document is selected in the *Ranking View*. The selected row is highlighted and the snippet appears in the *Document Viewer* in a fade-in fashion. Animated transitions for ranking-state changes and document selection help the user intuitively switch contexts, either from the *Tag Box* to the *Document List* and *Ranking View*, or from the latter to the Document Viewer. As Baldonado [30] states in the rule of attention management, perceptual techniques lead the users attention to the right view at the right time.

4. EVALUATION

The goal of this study was to find out how people responded when working with our tool. In the current scenario, recommendations were delivered in a sorted list with no relevance information. Since we aim at supporting exploratory search, we hypothesized that participants using *uRank* would be able to gather items faster and with less difficulty, compared to a typical list-based UI.

We were also interested in observing the effect of exposing users to different sizes of recommendation lists. We expected that without this relevance information, a slight growth in the number of displayed items would frustrate the user at the moment of deciding which items should be inspected in detail in the first place. For example, finding the 5 most relevant items in a list of ten appears as an easy task, whereas accomplishing the same task but searching a list of forty or sixty items would be more time consuming and entail a heavier cognitive load.

4.1 Method

We conducted an offline evaluation where participants performed four iterations of the same task with either *uRank* (U) or a baseline list-based UI (L) with usual Web browser tools, e.g. *Control+F* keyword search. Furthermore, we introduced two variations in the number of items to which participants were exposed, namely 30 or 60 items. Therefore, the study was structured in a 2 x 2 repeated measures design with *tool* and *#items* as independent variables, each with 2 levels (*tool* = U/L, *#items* = 30/60).

The general task goal was to "find 5 relevant items" for the given topic and all participants had to perform one task for each combination of the independent variables, i.e. **U-30**, **U-60**, **L-30** and **L-60**.

To counterbalance learning effects, we chose four different topics covering a spectrum of cultural, technical and scientific content: *Women in workforce* (WW), *Robots* (Ro), *Augmented Reality* (AR) and *Circular economy* (CE). Thus, *topic* was treated as a random variable within constraints. We corroborated that participants were not knowledgeable in any of the topics. All variable combinations were randomized and assigned with balanced Latin Square.

Wikipedia provides a well-defined article for each topic mentioned above. We considered them as fictional initial exploration scenarios but participants were not exposed to them. Instead, we simulated a situation in which the user has already received a list of recommendations while exploring certain Wikipedia page. Therefore, we prepared static recommendation lists of 60 and 30 items for each topic and used them as inputs for *uRank* throughout the different participants and tasks. To create each list, portions of texts from the original Wikipedia articles were fed to the F-RS, which preprocessed the text and created queries that were forwarded to a number of content providers. The result was a sorted merged list of items from each provider with no scoring information.

Each task comprised three sub-tasks (Q1, Q2 and Q3) that consisted in finding the 5 most relevant items for a given piece of text. In Q1 and Q2 we targeted a specific search and the supplied text was limited to two or three words. Q3 was designed as a broad-search sub-task where we provided an entire paragraph extracted from the Wikipedia page and the users had to decide themselves which keywords described the topic better. The motivation to ask for the "most relevant" documents was to avoid careless selection.

We recorded completion time for every individual sub-task and for the overall task. To measure workload, we leveraged a 7-likert scale NASA TLX questionnaire covering six workload dimensions.

4.1.1 Participants

Twenty four (24) participants took part in the study (*11* female, *13* male, between 22 and 37 years old). We recruited mainly graduate and post-graduate students from the medical and computer science domains. None of them is majoring in the topic areas selected for the study.

Table 1: Participants found *uRank* reduces workload in all dimensions

Dimension	$F(1,23)$	p	ε
Mental Demand	19.70	$p < .05$	.10
Physical Demand	14.52	$p < .01$	.07
Temporal Demand	7.72	$p < .05$	.05
Performance	11.80	$p < .01$	.10
Effort	48.60	$p < .001$	.22
Frustration	15.12	$p < .01$	.07
Workload	35.25	$p < .01$	.20

4.1.2 Procedure

A session started with an introductory video explaining the functionality of *uRank*. Each participant got exactly the same instructions. Then came a short training session with a different topic (Renaissance) to let participants familiarize with *uRank* and the baseline the tool. At the beginning of the first task, the system showed a short text describing the topic and the task to be fulfilled. After reading the text, the participant pressed "Start" to redirect the browser to the corresponding UI. At this point, the first sub-task began and the internal timer initiated the count, without disturbing the user. The goal of the task and the reference text were shown in the upper part of the UI. Participants were able to select items by clicking on the star-shaped icon and inspect them later on a drop-down list. In a pilot study, we realized that asking for the "most" relevant items made the experiment overly long, as participants tried to carefully inspect their selections (particularly in the L condition). Then we decided to limit the duration of the three tasks to 3m, 3m and 6m respectively. The time constraint was not a hard deadline. During the study the experimenter reminded the participants when the allotted time was almost over, but did not force them to abandon. The sub-task concluded when the participant clicked on the "Finished" button. The UI alerted participants when attempting to finish without collecting 5 items, but allowed them to continue if desired. The second sub-task started immediately afterward and once the whole task was completed they had to fill the NASA TLX questionnaire. The procedure for the remaining tasks was repeated following the same steps. Finally, participants were asked about comments and preferences.

4.2 Results

Workload: A two-way repeated measures ANOVA with *tool* and *#items* as independent variables revealed a significant effect of *tool* on perceived workload F(1,23)=35.254, $p < .01, \varepsilon = .18$. Bonferroni post-hoc tests showed significantly lower workload when using *uRank* ($p < .001$). We also assessed the effect for each workload dimension. Again, ANOVA showed a significant effect of *tool* in all of them, as shown in Table 1. (*#items*) did not have a major effect in any case.

Completion Time: We analyzed the task overall completion time, as well as completion times for each sub-task. A two-way repeated measures ANOVA revealed a significant effect of *tool* on overall completion time F(1,23)=4.94, $p < .05, \varepsilon = .02$. This effect disappeared in a Bonferroni post-hoc comparison. For Q1 and Q2 ANOVA reported no significant effect, but it showed a significant effect of *tool* on completion time for Q3, F(1,23)=6.2,

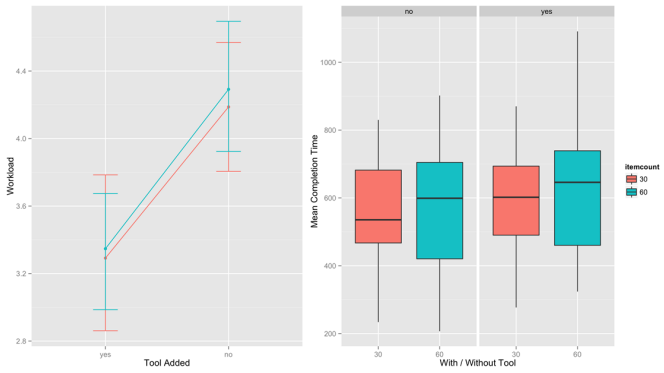

Figure 6: Results. (Left) Workload interaction lines show that *uRank* is significantly less demanding. (Right) Boxplots of time completion for each condition show a regularity towards using all available time.

Table 2: Similarities in collections gathered during evaluation

Sub-task	Comparison	WW	Ro	AR	CE	All topics
Q1	U vs L	.55	.79	.58	.74	.66
	U-30 vs U-60	.71	.83	.94	.67	.79
	L-30 vs L-60	.58	.83	.56	.56	.63
Q2	U vs L	.70	.86	.84	.86	.81
	U-30 vs U-60	.84	.89	.90	.93	.89
	L-30 vs L-60	.82	.74	.81	.87	.81
Q3	U vs L	.75	.72	.75	.63	.72
	U-30 vs U-60	.64	.88	.75	.62	.72
	L-30 vs U-60	.59	.66	.63	.33	.55

$p < .05, \varepsilon = .05$. As a surprise, post-hoc comparison showed that using *uRank* took significantly longer.

Performance: Relevance is a rather subjective measure. Hence, instead of contrasting item selections to some ground truth, we analyzed "consensus" in item selection.

We aggregated the collections gathered under the manipulated conditions and computed cosine similarity across UI (*tool*), data set size (*#items*), topic (WW, Ro, AR, and CE) and sub-task (Q1, Q2 and Q3).

Overall, there was a high similarity between collections produced with *uRank* and those obtained with the list-based UI across all sub-tasks. Choices regarding relevant documents matched three out of four times ($M = .73$, $SD = .1$).

Table 2 shows that collections produced with our tool (U) for the two variations of *#items* (U-30 vs U-60) turned highly similar regardless of topic and sub-task ($M = .8$, $SD = .12$, with a minimum of .62). Comparisons for a typical list-based UI (L) displaying 30 and 60 items (L-30 vs L-60) denote greater diversity ($M = .67$, $SD = .16$, with a minimum of .33) in item selection.

Interestingly, similarity values tend to decrease for broad search task (Q3) ($M = .66$, $SD = .13$) respect to targeted search (Q1 and Q2) ($M = .77$, $SD = .13$).

4.3 Discussion

The study results shed a light on how people interact with a tool like *uRank*. For each hypotheses we contrasted the results with the subjective feedback acquired after evaluation.

Workload: The results support our hypothesis that *uRank* incurs in lower workload during exploratory search, both in specific and broad search tasks. Participants commented feeling alleviated when they could browse the ranking and instantly discard document that did not contain any word of interest. As a remark, the majority claimed that a few tasks were too hard to solve, especially without the *uRank*, because sometimes the terms of interest barely appeared in the titles or were perceived as too ambiguous, e.g. "participation of women in the workforce". Also dealing with technical texts about unfamiliar topics was posed some strain. For example, two participants had to momentarily interrupt exploration to look up a word they did not understand. In spite of that, workload was significantly lower with *uRank* across all dimensions.

Completion Time: We expected people would be faster performing with *uRank* than using a browser-based keyword filter, but completion times were not significantly different. The closing interview revealed that participants who had collected five items before the due time exploited the remainder to refine their selections. In general, participants understood that they were not expected to perform perfectly but to do their best in the given time. However, we noticed that a small group that behaved in the opposite way reported feeling more pressed by time and not satisfied with their performance. The general tendency is reflected in the significant result on temporal demand: participants felt significantly less pressed to finish while performing with *uRank*. The lower subjective time pressure suggests that participants indeed had more time to analyze their choices with *uRank*.

Performance: The results suggest that our tool produces more uniform results as the number of items to which users are exposed grows. Nevertheless, the proportion of matching documents in list-generated collections – two out of three – still conveys a moderate consensus.

The decrease in consensus for broad search task respect to targeted search could be explained by the inherent variability across participants at the moment of chosing the terms of interest for a given text larger than a couple of words.

5. CONCLUSION

We introduced a visual tool for exploration, discovery and analysis of recommendations of textual documents. *uRank* aims to help the user: *i)* quickly overview the most important topics in a collection of documents, *ii)* interact with content to describe a topic in terms of keywords, and *iii)* on-the-fly reorganize the documents along keywords describing a topic.

This paper presented the reasoning line for the visual and interactive design and a comparative user study where we evaluated the experience of collecting relevant items to topics of interest. Participants found it significantly more *relaxing* to work with *uRank*, and most of them wanted to start actively using it in their scientific endeavors (e.g., report or paper writing). Yet, selecting the right keywords to describe a topic is not a trivial task, as it showed on the performance results of the evaluation. We will continue to explore different techniques, e.g. topic modeling, in the near future. As for the GUI, we will work further on solving scaling problems, for example when the amount of tags in the *Tag Box* or the length of the result list becomes unmanageable. Moreover, we will leverage the document selections collected during the evaluation as feedback to improve recommendations, closing the interactive loop with the RS as depicted in Figure 1.

6. REFERENCES

[1] E. Brill. A simple rule-based part of speech tagger. In *Proceedings of the workshop on Speech and Natural Language - HLT '91*, page 112, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

[2] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys*, 41(1):1–31, 2008.

[3] E. Gomez-Nieto, F. San Roman, P. Pagliosa, W. Casaca, E. S. Helou, M. C. F. de Oliveira, and L. G. Nonato. Similarity preserving snippet-based visualization of web search results. *IEEE transactions on visualization and computer graphics*, 20(3):457–70, Mar. 2014.

[4] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. LineUp: visual analysis of multi-attribute rankings. *IEEE transactions on visualization and computer graphics*, 19(12):2277–86, Dec. 2013.

[5] M. Harrower and C. A. Brewer. ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal*, 40(1):27–37, June 2003.

[6] M. Hearst. User interfaces for search. *Modern Information Retrieval*, 2011.

[7] M. A. Hearst. TileBars: Visualization of Term Distribution Information in Full Text Information Access. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '95*, pages 59–66. ACM Press, 1995.

[8] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE transactions on visualization and computer graphics*, 13(6):1240–7, 2007.

[9] O. Hoeber and X. D. Yang. The Visual Exploration of Web Search Results Using HotMap. In *Proceedings of the Information Visualization (IV06)*, 2006.

[10] A. Kangasrääsiö, D. Gowacka, and S. Kaski. Improving Controllability and Predictability of Interactive Recommendation Interfaces for Exploratory Search. In *IUI*, pages 247–251, 2015.

[11] J. Kay. Scrutable adaptation: Because we can and must. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4018 LNCS, pages 11–19, 2006.

[12] B. P. Knijnenburg, S. Bostandjiev, J. O'Donovan, and A. Kobsa. Inspectability and control in social recommenders. *Proceedings of the 6th ACM conference on Recommender systems - RecSys '12*, page 43, 2012.

[13] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell. Explaining the user experience of recommender systems. *User Modelling and User-Adapted Interaction*, 22(4-5):441–504, 2012.

[14] C. D. Manning. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[15] G. Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41, 2006.

[16] T. N. Nguyen and J. Zhang. A novel visualization model for web search results. *IEEE transactions on visualization and computer graphics*, 12(5):981–8, 2006.

[17] J. O'Donovan, B. Smyth, B. Gretarsson, S. Bostandjiev, and T. Höllerer. PeerChooser: Visual Interactive Recommendation. *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1085–1088, 2008.

[18] D. Parra, P. Brusilovsky, and C. Trattner. See what you want to see: Visual User-Driven Approach for Hybrid Recommendation. *Proceedings of the 19th international conference on Intelligent User Interfaces - IUI '14*, pages 235–240, 2014.

[19] M. Porter. An algorithm for suffix stripping. *Program: electronic library and information systems*, 40(3):211–218, 1980.

[20] R. Rao and S. K. Card. The table lens. In *Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI '94*, number April, pages 318–322, New York, New York, USA, 1994. ACM Press.

[21] H. Reiterer, G. Tullius, and T. Mann. Insyder: a content-based visual-information-seeking system for the web. *International Journal on Digital Libraries*, pages 25–41, 2005.

[22] F. Ricci, L. Rokach, and B. Shapira. Introduction to recommender systems handbook. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 1–35. Springer, 2011.

[23] C. Seifert, J. Jurgovsky, and M. Granitzer. FacetScape : A Visualization for Exploring the Search Space. In *Proceedings 18th International Conference on Information Visualzation*, pages 94–101, 2014.

[24] G. Shani and N. Tractinsky. Displaying relevance scores for search results. *Proceedings of the 36th international ACM SIGIR13*, pages 901–904, 2013.

[25] A. Spoerri. Coordinated Views and Tight Coupling to Support Meta Searching. In *Proceedings of Second International Conference on Coordinated and Multiple Views in Exploratory Visualization*, pages 39–48, 2004.

[26] M. Streit and N. Gehlenborg. Bar charts and box plots. *Nature methods*, 11(2):117, Feb. 2014.

[27] K. Swearingen and R. Sinha. Beyond Algorithms Beyond Algorithms : An HCI Perspective on Recommender Systems. *ACM SIGIR 2001 Workshop on Recommender Systems (2001)*, pages 1–11, 2001.

[28] N. Tintarev and J. Masthoff. Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction*, 22(4-5):399–439, Oct. 2012.

[29] K. Verbert, D. Parra, P. Brusilovsky, and E. Duval. Visualizing recommendations to support exploration, transparency and controllability. *Proceedings of the 2013 international conference on Intelligent user interfaces - IUI '13*, page 351, 2013.

[30] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. *Proceedings of the working conference on Advanced visual interfaces (AVI)*, pages 110–119, 2000.

[31] M. O. Ward, G. Grinstein, and D. A. Keim. *Interactive Data Visualization: Foundations, Techniques, and Application*. A. K. Peters, Ltd, May 2010.

[32] C. Ware. *Information visualization: perception for design*. Elsevier, 3rd edition, 2013.

[33] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. *Proceedings of the conference on Human factors in computing systems - CHI '03*, pages 401–408, 2003.