Towards Exploratory Search of Scientific Information

Dorota Głowacka[1], Ksenia Konyushkova[1], Tuukka Ruotsalo[2], and Samuel Kaski[12]

[1] Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki

[2] Helsinki Institute for Information Technology HIIT, Aalto University

Abstract. The paper introduces design and algorithms behind an information retrieval system developed to support scientists in finding specific scientific literature. The main goal of the system is to assist scientists in situations where their ability to express their information needs is difficult or their knowledge of a given research area is limited. The system provides an interface which allows the user to explore a given database of scientific articles through giving feedback on keywords associated with the displayed articles. We incorporate a number of exploration-exploitation strategies into the system in order to support the users' exploratory search behavior. Initial user study indicates that the system makes users' perception of the article space and finding relevant articles easier.

Keywords: exploration-exploitation, exploratory search, information retrieval, LinRel

1 Introduction

The amount of new scientific literature is estimated to be millions of publications worldwide per year; the growth rate of PubMed alone is now 1.8 papers per minute[3], and Google Scholar indexes 2.93 million papers for the year 2011[4].

The main enablers of access to scientific information are information retrieval engines developed specifically for the scientific literature. These engines are often built using traditional information retrieval methods that rely on keyword search. Information needs of searchers, however, are often exploratory [7]. Researchers are often uncertain of the goal of their query, unfamiliar with the topic they are researching, rely on vague topical conception of the information they need, and often need to learn more about a given topic in order to better formulate their search query.

While the state-of-the-art re-ranking approaches of information retrieval provide effective performance on average, the exploratory nature of the search is

[3] http://www.ncbi.nlm.nih.gov/pubmed/
[4] http://scholar.google.fi/

weakly supported and the users are provided with a narrow set of documents. This is manifested as two major problems that the current scientific information retrieval engines suffer from:

1. *Query drift* – the users change their focus of search during a search session. As the user learns more about a given topic during a search session, she may wish to dynamically alter her original query as the search session progresses. The search engine should actively support the user to achieve this aim.
2. *Diversity* of the search results is required to support the navigational information access. Most existing approaches to diversity try to ensure a diverse set of results in the top-s ranked documents but lack support for exploratory search.

Therefore, we need a system that allows us to address the above-mentioned shortcomings. We are in the process of developing a new information access method which couples advanced machine learning techniques with information visualization and interaction to boost exploratory search. The users actively engage in an exploratory search loop where they manipulate article features and ranking using exploration-exploitation paradigm. We expect the search to become significantly faster and add user flexibility by allowing the exploration and query manipulation.

The paper is structured as follows. In the next section, we provide a short overview of related literature. Section 3 provides a general overview of the system including the visual search interface and use examples. In sections 4, 5 and 6 we concentrate on the main components of the system back-end: article retrieval, keyword exploration and article exploration. Finally, we conclude in section 7 discussing the main results, limitations of the system and directions for future work.

2 Related Work

The importance of exploratory search has of significant interests in the information retrieval community as well as in the human computer interaction community. Marchionini [8] discusses in details modern information needs and the growing importance of the exploratory search for tasks of learning and investigating. From the algorithmic perspective, the exploratory search problem has been approached from many angles. For instance, Radlinski et al. [11] and Yue et al. [14] tried to solve the learning to rank problem using Bandits algorithms. Radlinski et al. [11] were the first to introduce a setting where the system learns the user preferences in on line manner without labeled dataset, while taking into account similarities between documents and feedback from the users. Dueling Bandits Algorithm introduced in [14] is designed for settings where the feedback is given in ordinal form. Exploration/exploitation techniques are often used in task involving information retrieval or recommender systems, such as filtering [3], recommendation [16] and ads placing [10]. However, most of the literature on exploratory search systems is in the field of Human-Computer Interactions and concentrates on the design of graphical user interfaces [9]. Similarly, the majority of the existing exploratory systems concentrate on data visualisation and facilitating user manipulations during the search procedure [5], [1], [13].

3 System overview

The primary goal of the system is to assist scientists in finding and exploring the relevant literature on a given research topic quickly and effectively. A tool for manipulation of article features on the concept space does not only make the search faster, but also helps the user to control the degree of the query drift and by that define the desired level of diversity in the interactive search process. The system currently indexes over 50 million resources from the following data sources: the Web of Science prepared by THOMSON REUTERS, Inc., the Digital Library of the Association of Computing Machinery (ACM), the Digital Library of Institute of Electrical and Electronics Engineers (IEEE), and the Digital Library of Springer. In general, the algorithms and the interface described in this paper can be adapted for other databases and does not need to be limited only to scientific literature search.

The main idea behind the interactive interface is that instead of typing queries at each iteration, the user navigates through the contents by manipulating the keywords on the display, which can result in new keywords appearing on the screen as well as a new set of documents being presented to the user. The visual search interface and an example search session are presented in Figures 1-3.

The search starts with the user typing in a query, which results in a set of terms being displayed in the circle on the left hand-side of the screen and a set of articles being displayed on the right hand-side of the screen. The displayed articles contain the title, author(s), data of publication, abstract and keywords associated with them. One of the current limitations of the system is that it does not include citation scores for the articles, which makes its usage more complicated for users without a prior knowledge of the field as they cannot judge how significant each article is. The user can access the full contents of a paper by clicking on it. The user can manipulate the keywords on the left hand-side to indicate how relevant they are to her search: the closer to the circle center a given keyword is moved, the more relevant it is to the user. The user can manipulate as many keywords as she likes. The user can also drag keywords from underneath the displayed articles to the circle to indicate that these keywords are highly relevant. After the initial keyword manipulation, new keywords and new articles are displayed. The search continues until the user is satisfied with the results.

The data flow described above from the system perspective is illustrated in Figure 4. Three main blocks in the system tackle initial retrieval and ranking, explorations in keywords and articles spaces.

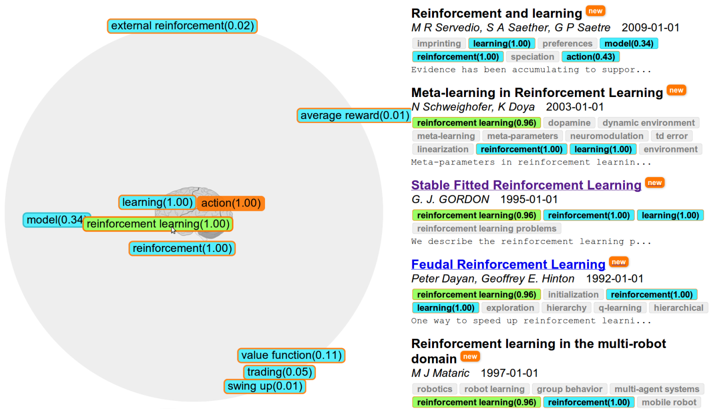

Fig. 1. First iteration: the user is broadly interested in reinforcement learning; having got an initial set of documents and keywords, she moves the keywords *reinforcement learning* and *action* to the center of the circle.

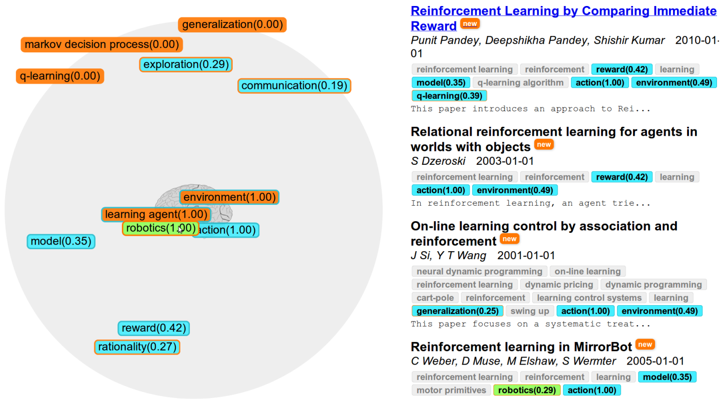

Fig. 2. Second iteration: a new set of keywords is presented and the articles are re-ranked; the user selects the following keywords: *robotics*, *learning agent* and *environment*, and removes the following terms: *generalization*, *markov decision process* and *q-learning*

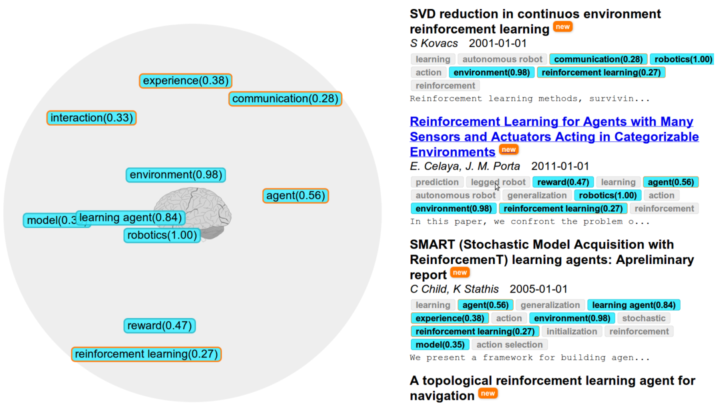

Fig. 3. Third iteration: the final set of keywords and articles are predicted by the system after user's manipulations. The user reached her aim - she explored the domain and found articles about reinforcement learning methods combined with applications to robotics.

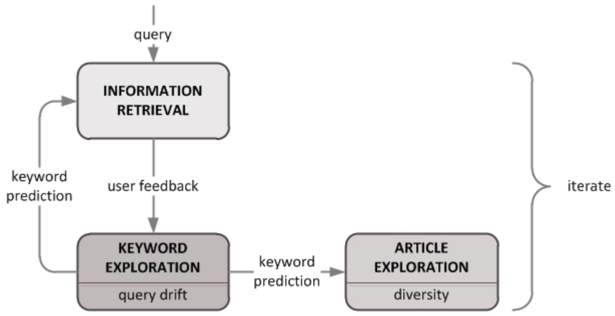

Fig. 4. Overview of data flow in the exploratory search system

4 Initial Document Retrieval

First, we employ a language modeling approach of information retrieval to estimate the initial result set. The language model ranking is chosen to be simple as we need a fast approach for initial ranking which is explored further relying on user feedback.

As we are dealing with a large number of documents, the initial retrieval is performed by utilizing a probabilistic language model. The language model is a unigram model with a Bayesian Dirichlet smoothing [15]. We rank the top-s documents based on the probability that they generate the query. The top-s documents are then used in the exploration-exploitation phase. The value of s can be any number between a few hundred and a few million, in the current version of the system it is set to be 300.

This allows us a fast ranking method that takes into account the user's query, but ensures that the initially retrieved document set consists of a variety of results. In this way, we can reduce the number of documents that our exploration-exploitation algorithms have to deal with throughout a given search session. The keywords prior vector language model is predicted by LinRel algorithm that will be discussed further together with keyword exploration in section 4.

4.1 User Feedback

The novelty of the system is in user giving the feedback on the currently displayed properties (keywords) of the documents that is further translated into effective information retrieval using reinforcement learning methods for exploring documents and features spaces. We assume that the user is looking for a set of keywords related to the interest. The system is informed about the relevance of displayed keywords by the relevance scores. The user gives feedback to the system by moving a keyword closer to or further from the center of the circle, whereby keywords in the center are given value 1 with the value getting smaller the further away from the center a keyword is moved. Keywords placed on the edge of the circle or beyond are assigned value 0. Thus, the feedback of the user is given by a relevance score $r \in [0, 1]$, where 1 indicates that the keyword is of high interest to the user and 0 indicates to being of no interest at all. Keywords with feedback 0 are excluded from appearing again on the screen for the remainder of a given search session. The formal protocol for this model is the following:

- In each iteration, the system selects n keywords and presents them to the user.
- The user provides relevance scores $r_i \in [0, 1], i = 1, \ldots, n$ for the displayed keywords.

We assume that the relevance score z_k of a keyword x_k is a random variable with expected value $\mathbb{E}[z_k] = x_k \cdot w, w \in \mathbb{R}^d$, such that the expected relevance score is a linear function of the keywords features. The unknown weight vector w is essentially the representation of the user's query and determines the relevance of keywords.

5 Keyword Exploration

As mentioned earlier, the primary goal of the system is to support query drift and diversity of displayed results. We use two exploration-exploitation strategies in our system: one strategy for keywords exploration and a separate strategy for document exploration. Exploration in the keywords space not only reflects the query drift but also helps in finding a set of useful keywords and gives an alternative way for giving a feedback through interactive interface.

5.1 Keyword Representation

In order to preserve the dependency between the keywords and the articles, we use a very simple keyword representation. Each keyword k is represented as a binary vector of length $\mid D \mid$ indicating its presence or absence in an article, where D is the whole set of documents, $\mid D \mid$ is the number of documents and d is an individual document. Next, we calculate the tf-idf measure [6] for every keyword. The tf-idf representation of every keyword in terms of documents is:

$$tf\text{-}idf(k, d) = tf(d) \cdot idf(k)$$

where idf is the inverse document frequency for a given keyword and $tf(d)$ is keyword frequency in particular document d. As a keyword can only appear at most once in every article, the keyword frequency $tf(d)$ for keywords that are present in a document is calculated as:

$$tf(d) = \frac{1}{\mid k \in K : k \in d \mid}$$

where $\mid k \in K : k \in d \mid$ is a number of keywords in article d.

Inverse document frequency is calculated as:

$$idf(k) = \log \frac{\mid D \mid}{\mid d \in D : k \in d \mid}$$

where $\mid d \in D : k \in d \mid$ is the number of documents where keyword k is present. Inverse document frequency is a measure of prevalence of the keyword in the whole set of documents. The more common the keyword is, the smaller this measure is. For example, general concepts would get smaller weight than names of specific algorithms. By combining these two measures we assign higher values to keywords that are less frequent in the database.

5.2 Exploration Algorithm

In order to help the user to explore the keyword space, we use the LinRel algorithm [2]. At each iteration, LinRel suggest new keywords to be presented based on the feedback from the user obtained in previous iterations. In each iteration, the LinRel algorithm obtains an estimate $\hat{w}$ by solving a linear regression problem. Suppose we have matrix X, where each row x_i is a feature vector of keywords presented so far. Let $r = (r_1, r_2 ... r_p)^\top$ be the column vector of relevance scores received so far from the user, where p is a number of iterations. Thus, LinRel tries to estimate $\hat{w}$ by solving the linear regression problem:

$$r = X \cdot w \ .$$

Based on the estimated weight vector $\hat{w}$, LinRel calculates an estimated relevance score $\hat{r_i} = x_i \cdot \hat{w}$ for each keyword i that has not already been presented to the user. As mentioned earlier, in order to deal with the exploration-exploitation trade-off, we choose keywords to present not with the highest score, but with the largest upper confidence bound for the relevance score. If σ_i is an upper bound on standard deviation of relevance estimate $\hat{r}_i$, the upper confidence bound of keyword i is calculated as:

$$r_i + \gamma \sigma_i \ ,$$

where $\gamma > 0$ is a constant used to adjust the confidence level of the upper confidence bound.

In regularized version of the algorithm we add regularization factor λ and the relevance score m_i of keyword x_i is calculated as:

$$m_i = x_i \cdot (X^\top \cdot X + \lambda I)^{-1} X^\top \ .$$

and the keywords that maximize

$$m_i \cdot r + \frac{\alpha}{2} \|m_i\|$$

are selected for presentation.

6 Document Exploration

The diversity of the displayed documents is ensured by employing another exploration-exploitation strategy. Thus, the aim of the system is not only to display articles with high relevance scores but also to show articles which might be potentially of interest to the user. In the current version of search, there is a clear connection between the documents and the displayed keywords as all the documents displayed on the right hand-side of the screen are always accompanied by the keywords associated with them. Also, the user expectation is that they will be shown documents containing the keywords to which they gave positive feedback. Thus, we assume that by indicating the relevance of a given keyword, the user implicitly indicates the relevance of all the documents containing that particular keyword. However, if at each iteration we only display documents containing the keywords that the user manipulated at the last iteration, then we risk losing the search context that we have learnt so far from previous key manipulations in a given session. This issue becomes obvious if we consider a situation where the user repeatedly specifies his query by narrowing down the domain and following specific application or aspect. In order to tackle this problem, we apply Thompson sampling bandit [12], [4] to the document selection process.

An intuitive idea of multi-armed bandit is coming from an example of a gambler (agent) playing a row of slot machines (arms of a bandit) with some unknown prizes (rewards). At every iteration, agent faces a decision-making problem which arm to play and receives some reward for it. The agent is aiming to maximize the reward, but as he doesn't know how rewards are generated, he has not only to exploit the most promising arms, but also to explore deferent actions. The challenge in multi-armed problems is to balance between exploration and exploitation.

Thompson sampling is one of oldest heuristics to address the exploration - exploitation trade-off. The idea of Thompson sampling is to randomly draw each arm according to its probability of being optimal. Thompson sampling is best understood in a Bayesian setting as follows. The set of past observations O is made of triplets (c_i, a_i, r_i), where c_i is a contest, a_i is an action and r_i is reward received. Observations are modeled using a parametric likelihood function $P(r \mid a, c, \theta)$ depending on some parameters θ. Given some prior distribution $P(\theta)$ on these parameters, the posterior distribution of these parameters is given by the Bayes rule, $P(\theta \mid D) \propto \prod P(r_i \mid a_i, c_i, \theta)P(\theta)$. Ideally, we would like to choose the action maximizing the expected reward, $\max_a \mathbb{E}(r \mid a, c, \theta^\star)$. If we are just interested in maximizing the immediate reward (exploitation), then we should choose the action that maximizes $\mathbb{E}(r \mid a, c)$. But in an exploration - exploitation setting, the probability matching heuristic consists in randomly selecting an action a according to its probability of being optimal. The Thompson sampling algorithm is briefly summarized in Algorithm 1 [4].

In the standard multi-armed bandit, each action corresponds to the choice of an arm. The reward of the i-th arm follows a Beta distribution with mean θ. When applied to the document selection problem in our system, we assume that each document is bandit arm with a Beta distribution. After giving a positive

Algorithm 1 Thompson sampling

$O = \{\}$
for $t = 1, \ldots, end$ **do**
  Receive context c_t
  Draw θ^t according to $P(\theta \mid O)$
  Select $\max_a \mathbb{E}(r \mid a, c_t, \theta^t)$
  Observe reward r_t
  $O = O \cup (c_t, r_t, a_t)$
end for

feedback to a given keyword, we assume that the user implicitly selected all the documents containing it. If the documents (arms) are with a Beta(α, β) distribution, we associate α with the success measure of the document and β with its failure measure judging by the user manipulations with document features. Then we increase the α parameter of all the documents containing a keyword with a positive relevance feedback by 1, while all the remaining documents have their β parameter increased by 1. After updating the parameters, we sample n documents to present to the user. The procedure is summarized in Algorithm 2 [4].

Algorithm 2 Thompson sampling for the Bernoulli bandit

Require: α and β parameters for a Beta distribution
$S_i = 0, F_i = 0, \forall i$, {Success and failure counters}
for $t = 1, \ldots, end$ **do**
  for $j = 1, \ldots, n$ **do**
    Draw θ_j according to Beta$(S_j + \alpha, F_j + \beta)$
  end for
  Draw arm $j^\star = \max_j \theta_j$ and observe reward r
  if $r == 1$, i.e. the document contains a keyword with a positive relevance feedback **then**
    $S_{j^\star} = S_{j^\star} + 1$
  else
    $F_{j^\star} = F_{j^\star} + 1$
  end if
end for

7 Discussion

We presented the interface and algorithms used in our prototype search system designed to help scientist to explore scientific articles within a given topic as well as to help them to get familiarised with a new research area.

We have conducted initial pilot user study with 6 users. The study revealed that the system has positive impact in exploratory visual search and in many tasks users find it more appropriate than traditional search engines that are based on only document ranking and query typing. The pilot users appreciate flexible and adapting nature and intuitive way of navigation in the complex concept space. The respondents learnt how to use the system easily and claimed that they discovered new new articles matching their interests in the topic area. We are currently in the process of running a more extensive user study with a target of assessing which aspects of the system help users to best navigate in complex information spaces, both from the HCI and the Machine Learning perspective.

The system is currently in the prototype phase and there are several remaining research challenges. We are still looking for more elegant ways of combining exploratory and retrieval models, we are investigating other ways of representing the documents using models based on semantic representations of the documents and based on citation graphs. Another possibility is to utilise the user profile as well as user's previous searchers in order to improve the initial search results.

Acknowledgements

The system currently indexes over 50 million resources from the following data sources. Certain data included herein are derived from the Web of Science prepared by THOMSON REUTERS, Inc., Philadelphia, Pennsylvania, USA: Copyright THOMSON REUTERS, 2011. All rights reserved. Data is also included from the Digital Library of the Association of Computing Machinery (ACM), the Digital Library of Institute of Electrical and Electronics Engineers (IEEE), and the Digital Library of Springer.

We want to thank Kumaripaba Atukorala and Giulio Jacucci: for HCI side, Samuli Kaipiainen: for visual interface, Murad Kamalov: for system integrating and troubleshooting.

References

1. Ahlberg, C. Spotfire: an information exploration environment. *SIGMOD Rec. 25*, 4 (Dec. 1996), 25–29.
2. Auer, P. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research 3* (2002), 397 – 422.
3. Balabanović, M. Exploring versus exploiting when learning user models for text recommendation. *User Modeling and User-Adapted Interaction 8* (1998), 71–102. 10.1023/A:1008205606173.
4. Chapelle, O., and Li, L. An empirical evaluation of thompson sampling. In *Proceedings of Neural Information Processing Systems* (2011), vol. 25.
5. Hearst, M. A. Tilebars: visualization of term distribution information in full text information access. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1995), CHI '95, ACM Press/Addison-Wesley Publishing Co., pp. 59–66.
6. Jones, K. S. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* (1972).

7. Marchionini, G. Exploratory search: from finding to understanding. *Commun. ACM 49*, 4 (Apr. 2006), 41–46.
8. Marchionini, G. Exploratory search: from finding to understanding. *Commun. ACM 49*, 4 (Apr. 2006), 41–46.
9. Marchionini, G., and White, R. Find what you need, understand what you find. *International Journal of Human-Computer Interaction 23*, 3 (2007), 205–237.
10. Pandey, S., Agarwal, D., Chakrabarti, D., and Josifovski, V. Bandits for taxonomies: A modelbased approach. In *In In Proc. of the SIAM International Conference on Data Mining* (2007), SDM.
11. Radlinski, F., Kleinberg, R., and Joachims, T. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning* (New York, NY, USA, 2008), ICML '08, ACM, pp. 784–791.
12. Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* (1933).
13. Williamson, C., and Shneiderman, B. The dynamic homefinder: evaluating dynamic queries in a real-estate information exploration system. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 1992), SIGIR '92, ACM, pp. 338–346.
14. Yue, Y., and Joachims, T. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning* (New York, NY, USA, 2009), ICML '09, ACM, pp. 1201–1208.
15. Zhai, C., and Lafferty, J. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst. 22*, 2 (Apr. 2004), 179–214.
16. Zhang, Y., Xu, W., and Callan, J. Exploration and exploitation in adaptive filtering based on bayesian active learning. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003* (2003), pp. 896–903.