Kristoffer Berntsen

What is in a recommendation?

The case of the bX Article Recommender

Master thesis 2021
Master in Library and Information Science

OsloMet – Oslo Metropolitan University,
Department of Archivistics, Library and Information Sciences

Abstract

This thesis explores the bX Article Recommender, a recommender system for research papers, developed by Ex Libris. It views the system as embedded in a historical and social context, and studies it from both a technical and epistemological perspective. Its technical components are studied by asking how it harvests data and generates recommendations on the basis of this data. Furthermore it explores implicit epistemological statements contained within the system. For example the system reflects and instrumentalises a certain interpretation of the phenomena 'relatedness'. The thesis demonstrates how the bX Article Recommender's output is influenced by human impact and decisions. When designing recommender systems many choices have to be made, choices with great implications for the produced recommendations.

Samandrag

Denne masteroppgåva undersøkjer bX Article Recommender, eit tilrådingssystem for forskingsartiklar utvikla av Ex Libris. Oppgåva ser systemet som forankra i ein historisk og sosial kontekst, og studerer systemet både frå eit teknisk og eit epistemologisk perspektiv. Dei tekniske komponentane blir studert ved å spørje korleis systemet samlar data og bygger tilrådingar på bakgrunn av desse. Vidare blir implisitte epistemologiske utsegn i systemet utforska. Til dømes reflekterer og instrumentaliserer systemet ei viss tolking av fenomenet «likskap». Oppgåva viser korleis tilrådingane frå bX Article Recommender er påverka av menneskelege val og avgjersler. Når ein utviklar tilrådingssystem må ein ta mange val, og desse vala har store implikasjonar for dei produserte tilrådingane.

OsloMet – Oslo Metropolitan University,
Department of Archivistics, Library and Information Sciences

Acknowledgement

To my family, thank you for your support and encouragement throughout my years as a part-time master student, and in the work with this thesis.

Thank you to my employer for the opportunity to study and providing a part-time study leave. Thank you to all teachers and co-students at the master program for interesting and stimulating courses and discussions.

Thank you to my supervisor, Pia Borlund, for believing in my project, whilst also asking tough questions and providing valuable feedback.

Table of contents

1 Introduction

1.1 Background

The object of study in this thesis is the bX Article Recommender, developed by Ex Libris[1]. This is an instance of what is sometimes called 'research paper recommender systems'. The goal of such systems are, simply put, to recommend relevant research papers to its users. This particular system analyses logs of usage data and builds recommendations from them. In my thesis the bX Article Recommender functions both as an example of a specific recommender system, and as a study object that helps showcase a particular way of approaching and studying recommender systems. That is, I am interested in what I can learn about bX[2] in particular, but I am just as interested in the act of studying a recommender system as a system embedded in a social context.

That is not to say that the bX Article Recommender is an uninteresting case in itself, or that it is chosen at random. This particular recommender system is interesting because it is so tightly integrated into the context of Norwegian academic libraries. In Norway ca. 80 academic, institutional and research libraries make up the members of the BIBSYS-consortium[3]. These libraries all use a range of cloud-based services by Ex Libris, among others the discovery system/online library catalogue called Primo – which in Norway is labelled as Oria. The bX Article Recommender is embedded in Oria, and libraries in the consortium have the choice to active bX in their own, local instances of Oria. It is thus a recommender system that many librarians, researchers and students will meet.

1.2 Motivation

The connection between the world of research and education and the world of library and information science (LIS) has always been strong; the act of helping users find relevant (scientific) literature has traditionally been an important problem for LIS. Therefore, it should be of interest to develop the knowledge about services relating to these problems – such as research paper recommender systems. Since the bX Article Recommender is an object in the 'lifeworld' of many

1 https://exlibrisgroup.com/
2 When I in the thesis refer to bX, I mean the bX Article Recommender.
3 https://www.unit.no/oversikt-over-deltakere-i-bibsys-konsortiet

librarians, not only in Norway, but all over the world, it should be particularly interesting to look at it more closely.

I also want to improve my understanding and thinking on recommender systems. As a larger, more generic phenomenon, it is my view that the ubiquity of recommender systems in everyday life is strong. Perhaps just as strong as some argue that search and search engines have become (see for example Haider & Sundin, 2019).

In this thesis I want to approach the bX Article Recommender not only from a technical side, but also from an epistemological side. By taking the ontological and epistemological view that recommender systems reflects human actions it also means that things could have been otherwise, and thus

> if software is seen as a genuine form of cultural expression and intervention, the conceptual horizon mobilized in technical artifacts is a relevant object of study for the humanities and social sciences as well as a site for political assertion and struggle. (Rieder, 2017, p. 104)

I am approaching the wider topic of research paper recommender systems from a somewhat unusual angle. Another underlying motivation is to 'connect' the technical side of library and information science, with approaches from the humanities side of it. The motivations mentioned here are related to my epistemological and ontological stance, which I will outline in the next section.

1.3 Ontological and epistemological stance

The idea to formulate my thesis' epistemological and ontological stance, and to hopefully make them clear to the reader, is greatly inspired by Jennifer Mason's (2018) thought-provoking book on qualitative research. So is my approach to answering them. In her book, Mason encourages an active, creative and investigative approach when doing qualitative research (p. x). One way to nourish this can be to formulate the ontological and epistemological stance behind ones research. This, in my experience, also helps establish a greater sense of ownership to ones research questions; it helps with pinpointing what ones research is *about.* Mason's line of thought also influences my research questions and research design, as will be seen. My epistemic and ontological view of recommender systems is also very much inspired by two articles by Bernhard Rieder (2012, 2017) on algorithms and software.

Ontological perspective

I understand recommender systems (RS) as human-made constructs, and therefore as social and cultural products – reflecting attitudes, motives, ideas and interpretations of the world. RS are meaningful components of the social world – meaningful here implying that something can be known about them, and that they are meaningful objects to study. This includes the view that one can and should study the inner workings of such systems, and that these are influenced by both a social and a technical context.

My ontological perspective is thus opposed to a view that to a lesser degree, or not at all, take in the social context of such systems. It is also opposed to a view that theorise about social and political effects of the systems, but does not, in any deeper sense, take the technical side into consideration. The lack of technical considerations is something Rieder criticizes many scholars in the social sciences and humanities of (Rieder, 2017, p. 101). My ontological perspective allows for an approach that shares a common goal with Rieder (2017): "My goal, however, was to find a level of description where an encounter between technical and larger cultural principles becomes possible, a level where neither 'side' is reduced to a caricature" (p. 114).

Epistemological perspective

I believe that meaningful things can be known about social and cultural products – including research paper recommender systems. I think that such systems can be interpreted and read, and that the systems themselves (through their construction by human actors) interpret the world. The same applies for texts about such systems. This also implies that concepts such as ideas, attitudes etc. exists and are knowable. I believe that knowledge about products like recommender systems can be demonstrated through clear and carefully constructed arguments. My goal is not to say or prove anything about objective qualities of such systems. I follow Mason (2018, p. 219) in that I do not seek a *solution* to my research questions, but arguments – and arguments in the sense of presenting an interpretation, a perspective, analysis or line of reasoning.

1.4 Research questions

Mason introduces the concept of an 'intellectual puzzle', which reflects the essence of what the researcher(s) wants to investigate. She argues that all research should be formed around such a puzzle and its related research questions (Mason, 2018, p. 3). How the puzzle is formulated is connected to the ontological and epistemological stances of the research. The table below (Table 1) shows different examples of 'types' of intellectual puzzles. These are not meant to cover all possibilities, and are first and foremost a tool to help think with.

Type of puzzle	**Description**
Developmental	How/why did *x* or *y* develop?
Mechanical	How does *x/y* work? Why does it work this way?
Comparative	What knowledge can be gained by comparing *x* to *y?*
Causal/predicative	How does *x* influence *y*? Or, what causes *x/y*?

Table 1: Types of intellectual puzzles, based on Mason (2018, pp. 11–13).

An intellectual puzzle might be a combination of different types. In my thesis, the overarching puzzle can perhaps best be formulated as a mechanical puzzle: How does the bX Article Recommender work? Why does it work this way? At the same time I'm also interested in developmental aspects.

Mason views a qualitative research approach as especially apt for mechanical arguments: "Qualitative research is in my view particularly good at supporting 'mechanical' and 'processual' arguments that focus on how social phenomena and processes operate or are constituted, or how they change" (Mason, 2018, p. 221). Such arguments are also not pure 'descriptions'; implying some objective social reality out there ready to be disclosed (p. 221). My intellectual puzzle is thus perhaps more wide reaching than first meets the eye. It cannot fully be 'answered' by pointing to some technical description, or by observations such as 'by clicking here this happens'. The 'why' part of the puzzle entails a wider, social context. Also, Rieder (2012) remarks that if you view software as an object in-the-world (as I do), then the question is not "what is an algorithm?" but "what is *in* an algorithm?". In line with this my overarching puzzle could also be formulated as: What is *in* the bX Article Recommender?

To help approach my intellectual puzzle, I am asking the following three research questions:

1. How does the bX Article Recommender harvest and represent usage data?
2. How does the bX Article Recommender exploit usage data to generate recommendations?
3. What epistemological ideas and interpretations of the world operate in the bX Article Recommender?

Research question one and two focus more on the technical side, and relate to the system's use of usage data as basis for its recommendations. But these questions will also touch upon the wider social and contextual side of the bX Article Recommender. Question number three will look at findings related to the previous two questions, and view these from an 'epistemological' perspective. There is a dynamic at play where the epistemological forms the technical, and vice versa.

1.5 Limitations

In the writing of this thesis it has been a goal to use sources that in principle are readily available to anyone. By doing so I am implicitly trying to demonstrate how much an interested user can realistically find out about how bX works from the outside alone. This relates to the common notion of algorithms as 'black boxes'. I have for example not had access to the bX API (more on the API later). This also mimics the information available for libraries potentially interested in bX, short of contacting Ex Libris themselves. This thesis is a theoretical one, where all my data consist of published texts of different kinds.

1.6 Research design

This thesis does not follow a clear-cut research methodology with an established way of doing things. My most important data sources are previous research articles and available documentation from Ex Libris. My approach can perhaps be said to have some resemblance to the act of close reading. I am trying to read and interpret my sources as precise as I can, but from a particular viewpoint. It is my research questions, and my epistemological and ontological stance, that is forming what I look for, and how I interpret smaller details in the light of a larger whole. I have not felt a particular need to identify a 'standardised' research methodology and adapt the thesis accordingly. I mentioned that Mason in her book argued for a nourishing a creative and investigative energy to do qualitative research. This can not be achieved, in her view, by merely adapting readily available instructions or blueprints from research methodology books (p. x). Instead she calls for a fluid approach to qualitative research[4]:

> Thinking qualitatively means rejecting the idea of a research design as a single document which is an entire advance blueprint for a piece of research. It also means rejecting the idea of a priori strategic and design decisions, or that such decisions can and should be made only at the beginning of the research process. This is because qualitative research is characteristically exploratory, fluid and flexible, data-driven and context-sensitive. Given that, it would be both inimical and impossible to write an entire advance blueprint. (Mason, 2002, p. 24)

This is *not* to say that anything goes. I think the validity of my approach depends on its ability to provide arguments for answering my intellectual puzzle and research questions; and that these arguments can be demonstrated to be appropriate and not mere, misguided speculation. According to Mason, the 'end product' is central for judging the validity of a qualitative interpretation, and this includes showing the reader why one has come to see an argument as appropriate or persuasive (Mason, 2018, p. 219).

1.7 Structure of the text

The rest of the thesis is structured in the following way. The next chapter will sketch an outline of previous research on (research paper) recommender systems and bX in particular. Thereafter comes a theoretical chapter, introducing some concepts for analysing, writing and thinking about algorithms as social objects (a recommender system consists of algorithms). It will inform later chapters with a set of descriptory and analytical tools, but it is my hope that the chapter also is interesting on its own, and that the theoretical concepts therein are perceived as potentially useful outside the scope of this thesis.

4 The following quotation is from the second edition of her book.

Following the theory chapter are three chapters that each handle one of the research questions. After that I will showcase how my findings can help inform other discussions and studies of the bX Article Recommender. I will do this by looking at the concept of serendipity. This also ties into the following chapter on further research. Then I provide some concluding remarks, and to the very last comes the bibliography and the list of figures and tables.

2 Previous research

In this chapter I will at first briefly introduce the world of recommender system research, before turning to the sub-genre of *research paper* recommender systems. Lastly I will look at research in which the bX Article Recommender is an object of study, or research which mentions bX. The former category, research in which bX is studied more in-depth, is very scarce.

It is not my goal to give a complete review of the field of (research paper) recommender system research, but to provide a reasonable backdrop for my own work, and to give it some context.

2.1 Recommender system research

The following can function as a very general and simple definition of a recommender system. It is taken from the *Recommender Systems Handbook*, a large compilation of works on different areas of recommender system research:

> Recommender Systems (Rss) are software tools and techniques that provide suggestions for items that are most likely of interest to a particular user. The suggestions relate to various decision-making processes, such as what items to buy, what music to listen to, or what online news to read. (Ricci et al., 2015b, p. 1)

A recommender system, then, recommends something, to someone, and does so in a particular setting. From this simple notion springs forth a large research area, dealing with questions ranging from evaluation methods and different technical approaches, to user privacy, user studies and more.

Traditionally there has been a close co-operation between the academical and commercial world when it comes to the development of recommender systems. A quick glance at the list of contributors to a work like *Recommender Systems Handbook* (Ricci et al., 2015a) shows that they are either from the industry or from technological/computer science departments of various universities and other higher education facilities. A further example is the most important conference for the field, the *ACM Conference on Recommender Systems* (RecSys), that has been arranged fourteen times by the *Association for Computing Machinery* (ACM). The 2019-edition gathered more than 900 participants. This included researchers and research groups from both academia and the industry[5]. The sponsors included, among others, Spotify, Booking.com, Netflix, Amazon and Google. This is also in line with the observation by Ricci et al. (2015b, p. 15) that practical and commercial aspects are important guiding factors in recommender system research. It is worth to have this point in mind while reading this thesis. The bX Article Recommender is itself a commercial product, institutions pay for it via a subscription.

The field of recommender system research is large and multifaceted. In the next part I will zoom in on a specific sub-genre, that of *research paper* recommender systems.

2.2 Research paper recommender system research

Although the amount of research on research paper recommender systems (RPRS) is quite small compared to other branches of recommender system research, it still has been an area of interest. In the following I lean on a fairly recent and comprehensive literature survey of RPRS-research published by Beel et al. in 2016. The study gives an interesting sketch of the research area, but not necessarily a flattering one.

In all the authors identify 217 articles[6]published between 1998-2013, of which they review 127 more closely (Beel et al., 2016, p. 307). The authors detail what recommendation approaches are used and presented in the different papers, and how different approaches and systems are evaluated. It does not seem to be many works that explores RPRS as epistemic objects within a larger, social context. It is the pragmatic goal of developing the 'best possible' systems that is at the forefront. Beel et al. (2016) also find that the research area is small and has few authorities; 73 % of the authors published no more than one paper on RPRS. This might be a sign that there is not enough prestige in the research area, and/or that researchers do not find it worthwhile to stick around.

5 https://recsys.acm.org/recsys19/

6 "Articles" here consisting of the following: Peer-review conference papers (59 %), journal articles (16 %), pre-prints (5 %), others (20 %, including web pages and presentations). The number of conference papers is perhaps an indicator of the practical orientation within the field.

Furthermore the authors find that several articles have unsatisfying foundations for evaluation, lack information about the algorithms involved, and even conclude that most of the works that include user studies have too few participants to yield any meaningful results. All the reviewed user studies were also lab studies, and gathering of qualitative user feedback were rarely used (Beel et al., 2016, p. 311). In addition to this, lack of access to live 'real-world' systems for researchers is also identified as a shortcoming (p. 312).

The particular challenges facing the field notwithstanding, research paper recommender system research also does not seem to play any major role within the larger area of recommender system research. In the previously mentioned *Recommender System Handbook*, a book over 1000 pages long, works on research paper recommender systems are barely alluded to. A full-text search of the content reveals only a handful of instances that cite articles on the subject, and then often with a focus on the approach used, not on the problem of recommending research articles *per se*. I also searched the citation and abstract database Scopus for works from to the RecSys conference. This yielded 1456 documents, but of these, only a handful (~10) could be said to be about the act of recommending research papers to users[7].

Meanwhile, the review by Beel et al. (2016) has 291 citations in Scopus. The majority of these citations does not seem to stem from works about *research paper* recommender systems, though they are not non-existing. Among the 291 citations, there are several recent articles that handle the problem of recommending research paper to users, thus the field is producing new works. But all in all, it is fair to say that the research area is a minor one, at least within the larger context of recommender system research. Even so, the reviewed literature also speaks for the bX Article Recommender as an interesting case to study; it is one of the rare cases where conducted research led to the development of a commercial research paper recommender system that is still in use. In the next part I will comment on some instances where bX has been mentioned in the literature, or actually been an object of study in itself.

7 Based on the following search: (research AND ((paper OR article OR citation) AND (recommender OR recommendation) AND (system OR systems))), yielding 158 results, and then a manual screening of title and abstracts.

2.3 The bX Article Recommender

The bX Article Recommender is briefly mentioned in several peer-reviewed articles, but has seldom been the subject or case being studied. I have not been able to find previous research on bX that poses research questions similar to mine.

Mentions of bX can be found in different types of works, from articles to doctoral theses. A common trait is that they do not shed much light on the workings of bX. Some merely mention bX as a recommender system based on usage data (Aravind, 2018) or on cloud computing (Chen et al., 2012), or compare bX to commercial web site recommender systems that recommends items that other users found interesting (Barner & Tal, 2012; Greenberg & Bar-Ilan, 2017).

The work by Greenberg & Bar-Ilan (2017) has some more interest to it, since it does include a look at the use of bX recommendations. Though this is limited to a study of monthly average full-texts requests stemming from bX recommendations. The authors find the number to be low (p. 460). They speculate that the relatively low usage might be because the recommendations are presented under other relevant links, and that the users do not notice the recommendations (p. 464). Their work studied an earlier version of bX, which presented the recommendations in a slightly different way. Nonetheless, it is interesting that it takes the user interface into consideration.

In addition to the work by Greenberg & Bar-Ilan (2017), there are only three works I have found that actually can be said to do research *on* bX. The first I will mention is a study done by Andre Vellino published as “Recommending research articles using citation data” (Vellino, 2015). In this work he compares recommendations generated by the bX Article Recommender, with those generated by a system that uses citation data as basis for its recommendations. It is a work in which bX, or, the recommendations generated by bX, is an object of study and comparison. The article also gives a fine overview of some of the systems and approaches for recommending research papers that were developed at the time (Vellino, 2015, pp. 598–600).

Vellino randomly picks 9453 articles in a digital library and uses these as a starting point for generating recommendations with both systems. Then he compares the generated recommendations according to various criteria, among others for coverage (how many articles produces recommendations) and serendipity. To measure the latter he introduces the notion of 'semantic distance': If the produced recommendations comes from semantically similar journals, then the semantic distance is low. If the produced recommendations comes from more semantic disperse journals, they are said to be more serendipitous (Vellino, 2015, p. 600). I will discuss the concept of serendipity and bX further in chapter 7.

Vellino finds that bX generates recommendations for a larger number of articles compared to the citation-based system, and that bX produces recommendations that are semantically close to the seed article. He also finds that the results from bX are biased towards newer material, with a publication date of -0.6 years, which means articles published later then the seed article (Vellino, 2015, pp. 605–606).

The article by Vellino thus yields some interesting observations of output generated by the bX Article Recommender, but it does not give any details into how the algorithms operate, or the technical structure behind bX. The description of bX does not go much in depth, besides briefly referring to the most central research behind its design. Nevertheless, it is perhaps the most interesting research done on bX that I have found.

Another peer-reviewed study that I want to mention is Ponsford et al. (2011). This study reports on a usability test of some aspects of the online library catalogue of Texas A&M University libraries, including the bX Article Recommender. The 21 participants discussed their perceived quality of the recommendations generated by bX. One of the research questions is: "Do users understand how bX recommendations work without further explanation and do they find the recommendations relevant to their research needs?" (Ponsford et al., 2011, p. 163). 16 of 19 said the recommendations were relevant and were inclined to follow up on the recommendations (p. 166). Even though a small sample size, the question of how the bX Article Recommender works illustrates how hard it is to know:

> When asked how they thought bX worked, seven assumed the list of recommendations was based on something to do with keywords; only two suggested that it worked like Amazon; and one thought it was like Google Scholar. One suggested they were based on "similarity," and three assumed they were citations from the original article. Only one volunteer said it was based on the articles on which people clicked. (Ponsford et al., 2011, p. 166)

They conclude that "while users did not necessarily understand how the bX recommendations were generated, they were mostly satisfied with the quality of the recommendations and would both use the recommendations themselves and thought their students would find them useful as well" (Ponsford et al., 2011, p. 168). Even though this study sheds some light on how a small sample of people experienced recommendations generated by bX, it does not really give any more detail into *how* bX works.

The final work I will highlight is a master's thesis from HiOA[8], submitted in 2012. The thesis explores the topic of user personalisation in academic libraries, and one of the studied cases is the bX Article Recommender. Unstructured questions about bX were sent via e-mail to personnel at two university libraries in Norway to get an indication of usage. It was found that none of the libraries had promoted the services and that usage was limited (Beyene, 2012, p. 49). The thesis describes bX in similar ways to other works that mention bX, namely as similar to commercial systems' 'customers who bought this also bought'. The description is also mainly based on Ex Libris own product presentation (Beyene, 2012, p. 47).

To conclude, there exists research that in different ways explores output generated by bX – but I have not been able to find research that tries to go more in depth into *how* the output is generated, i.e., how bX is built. Although I am in some degree using the same sources as the above mentioned works when it comes details about bX, my approach seems to represent something new in the research on bX. This is also related to the theoretical outlook, which I will introduce in the next chapter.

8 Høgskolen i Oslo og Akershus (Oslo and Akershus university college of applied sciences), which today is OsloMet.

3 Theory

This thesis is a small contribution to the body of work that studies software, programs and algorithms as cultural products. The idea and inspiration to do this came when I read the special issue on algorithms and (social) power in the scholarly journal *Information, Communication & Society.* In the introductory article to the issue it is claimed that rather than being visible and clear, "it is far more common for algorithmic processes to pass us by without being noticed" (Beer, 2017, p. 2). This simple remark in my view says something true about the omnipresence of algorithms, and, at the same time, also showcase the following paradox: Even though algorithms are everywhere, they are almost always invisible to the user – for the most of us they only manifest themselves indirectly by their results or *output*.

Even the term algorithm can by itself be vague and not particularly precise nor clarifying. That is one of the reasons I am especially inspired by the media scholar Bernhard Rieder's contribution to the aforementioned special issue. His article both outlines and exemplifies a way of researching algorithms by studying them at a *concrete* level and not as some generic phenomenon. He does so by presenting and interpreting M.E. Maron's work on the Bayes' classifier in the 1960s. In another, earlier work, which I will also refer to, Rieder studies one particular algorithm, namely *PageRank.* These articles have recently been slightly reworked and collected as part of a larger work (Rieder, 2020), but I will stick to them as they were first published – they are both, in my view, still relevant and worth reading in their original form. The article from 2017 especially introduces several useful and illuminating theoretical concepts, which I will introduce one by one. These are algorithmic language, algorithmic literature, algorithmic techniques, 'interested readings of reality' and mechanical reasoning.

3.1 Algorithmic language, algorithmic literature and algorithmic technique

Rieder draws a distinction between *algorithmic language* and *algorithmic literature. Algorithmic language* is the computer code – the medium in which an algorithm is expressed. *Algorithmic literature* are the programs themselves, small or large, designed in a setting and with a concrete purpose in mind (Rieder, 2017, p. 101). This can be seen as a parallel to the relationship between a given human language and its literature.

Thus the mere existence of a coding language does not create a specific program, the same way that the existence of a language does not automatically produce a given literature. Rieder (2017, p. 101) explains it as follows: "Software, like language, allow for the expression and mechanization of a wide range of ideas and objectives, even if basic principles and historically accumulated knowledge and convention structure possibilities and actual outcomes". Algorithmic literature is highly contextual and concrete, it is the meeting of a medium of expression (code) and the world. It is Rieder's (2017) view that research on algorithms traditionally have focused on algorithmic language, rather than algorithmic literature – something he criticizes.

To illustrate the different levels of language and literature, I have constructed the following example:

```
print("On a scale from 1-10, how do you rate the book?")
bookRating = input()
ratingUpdate(bookId,bookRating)
```

Imagine that this is an excerpt from the code behind a book discussion app. It takes an input from the user (a rating) and stores the result as a variable (bookRating). Lastly, it calls on a function to update the rating of the given book. It is written in one particular algorithmic language (Python), but could easily have been written in others. The main concern could be on how the app best could be written to make it as 'pythonic' as possible, for example to make it less resource demanding to run. Then the primary focus would be on the algorithmic language and the use of it.

In a different part of the code this rating input is manipulated and used, for example to calculate the average rating for the book, to categorize the user (is he strict or does he give many high ratings), to provide the user with recommendations for new books and so on. Then the focus is on algorithmic literature, that is, the fully flexed app, designed with a concrete purpose, and which is abound with underlying choices: How should the ratings be used? What are the minimum rating threshold for recommending a book, how many ratings are necessary etc. Of course, such decisions, questions, and possible solutions are what recommender system research and development is all about. It should also be noted that such questions are often not tied to the algorithmic language (programming language) per se. This leads to a third theoretical distinction, that of *algorithmic techniques.*

Rieder (2017, p. 102) states that: “Algorithmic techniques are ... units of knowledge and expertise in the domain of software making”. These have grown forth over time and become part of a ‘computational’ way of thinking; being able to abstract from a concrete situation so that one can apply a known algorithmic technique (p. 102), and the same algorithmic technique can often be applied for quite different purposes. In this way, Rieder states, algorithmic techniques are both general and diverse at the same time (Rieder, 2017, p. 103). They are in a way a readily available tool-box which can help programmers when approaching a given problem. One such problem can be: How can we recommend relevant research papers to our users? This idea of a common reservoir of knowledge can also been seen from the definition of a recommender system given earlier: “Recommender Systems (Rss) are software tools and techniques that provide suggestions for items that are most likely of interest to a particular user” (Ricci et al., 2015b, p. 1).

3.2 ‘Interested reading of reality’

Another theoretical idea from Rieder that I want to introduce is that of an ‘interested reading’ of reality. This, in my view, embodies the core epistemological stance for Rieder’s article, and also for my thesis. That it is an important notion for Rieder as well can be see from the subtitle of his article, “Scrutinizing an algorithmic technique: the Bayes classifier as interested reading of reality” (Rieder, 2017).

What Rieder means by interested reading is that algorithms, or algorithmic techniques, are not neutral, i.e., they are interested. For an algorithmic technique to be applied, things must be broken down into records that can be manipulated by a computer. That means reducing phenomena to bits and bytes, which Rieder (2017, p. 103) labels a ‘datafied reality’. This involves purpose; the ‘datafication’ has a goal, for example to provide input data for a recommendation algorithm. Recommendations from recommender systems can then been seen as the result of calculations applied to a problem statement (What is a relevant recommendation in this situation?). In another article Rieder calls this the shadow of computation – where ideas must be expressed in a form that can be made computable (Rieder, 2012, 1. Introduction section). Many-faceted phenomena like relevance, content etc., have to be reduced to numbers before any calculation, any algorithmic technique, can be applied in the first place. The notion of 'interested reading' incorporates all of this quite nicely.

This quantification of reality is of course not something that first came about with computers. Long before computers, statistics were accepted as a valid tool for explaining and representing reality. "The application of calculation to practical matters … culminates in the emergence of statistics as the predominant way to look at and act in a world seen as dynamic and opaque" (Rieder, 2017, p. 104). By the end of the 19th-century mathematical methods to find patterns and dependencies between variables were developed. But such methods for identifying and using variables are not a purely objective undertaking, according to Rieder (2017, p. 105), "it is a cognitive operation that generates an *interpretation* of the relationship between numbers and, by extension, the world they purport to describe".

This act of interpretation is something unavoidable also when creating software, and also includes a reduction of meaning:

> Meaning is thus conceived in a highly reductive manner … any running system requires and relies in some way on selection, formalization, and reduction. Datafication thereby translates fundamental assumptions about the application domain into data structures and reifies them. (Rieder, 2017, p. 109)

In my view these 'fundamental assumptions', if not synonymous with 'epistemological assumptions', at least contain them. Again, this act of interpretation and reduction is what constitutes an interested reading of reality. I think that this theoretical concept shows that exploring how a program formalizes and represents data and how it defines and uses its variables, is a fruitful starting point for an analysis. This is what I try to do with my research questions. An overall focus is then on how output is produced and made possible, which I will now elaborate a little further, by looking at the production of output.

3.3 The production of output

Say that a librarian after talking to a patron recommends the patron a list of five books, and compare this with a list of five recommendations made by a recommender system. The output – the end product – is essentially the same, a list of five books, but the way they were made is different.

In the first case it easier to see that the particular outcome is decided by a human; the librarian choosing to recommend exactly those works. How the librarian ended up with those five books and not others, is nevertheless a highly complex process to describe, but in this case there is at least a possibility to ask the librarian some questions and start a dialogue to try to gain insight.
I think a recommender system is complex in a different way, and that this is often tied to scale. It is often the sheer amount of (input) data that can make it hard to predict a given outcome, which are often based on calculations done at a speed and scale that no human can match, though the mathematics behind them do not have to be very complex. If someone had all the data available to bX, and were asked to generate recommendations based on a given article, it would be an impossible task because of the amount of data that potentially could affect the outcome, and the amount of calculations that would have to be done.

Another difference is that in the case of the librarian, the librarian himself fully decides the outcome, but in the case of the recommender system, a human actor does not decide the exact outcome[9]. Looking at calculations from a perspective of power, Rieder (2017, p. 113) writes: "As previously noted, mechanical reasoning does not eliminate power, but reconfigures it and shifts human discretion from the definition of outcomes to the definition of procedures, mechanisms, or techniques that *produce* outcomes". The bX Article Recommender is an example of such 'mechanical reasoning'. Here the human actors do not decide the particular outcome (the contents of a list of recommendations), but rather defines that which produces the outcome (algorithms).

9 This is not to say that librarian's recommendations are 'neutral'. The librarian also chooses works *not* to recommend, probably don't know of all works that could be useful and so on.

The produced output can only be observed, and then if deemed unsatisfying according to some criteria, the parameters can be tweaked and the output observed again. In complex, large-scale algorithms, this lack of control over definitions of outcome can have unfortunate consequences; systems can produce output that is unwanted or biased. This is something that has gained increased attention also in more main stream media, on the background of books such as *Algorithms of oppression* (Noble, 2018).

The concept of mechanical reasoning also shows that there is no straightforward dichotomy between (computer) calculation and (human) judgement (Rieder, 2017, p. 113), in that, human judgement and choices affects the calculation and output. In this way, more calculation means that more, not less, judgement is required (p. 113). In my view this is also one of the reasons why humanly curated recommendations and critics are needed – lists curated by algorithms does not 'escape' human judgement anyway nor are they objective. To Rieder, the question of the objectivity of algorithms is a false trail, "this change in perspective should prompt us to trade the stale question whether algorithms are 'objective' or not for an investigation into the interpretative commitments, purposes, and benchmarks specific calculative assemblages subscribe to" (Rieder, 2017, p. 113). Before a recommender system can do its thing, lots of interpretive, epistemological questions have already been asked and answered, whether consciously or not.

So far I have presented the theoretical concepts of algorithmic language, literature and technique, and the idea that algorithms encompass interested readings of reality. I have also described how it can be fruitful to turn the attention to the production of output and 'datafication', a process where a phenomena is reduced to something an algorithm can act on. It is not certain that I will be able to point to one particular algorithmic technique that the bX Article Recommender applies, or discover a bunch of different parameters that it uses. In this case I think that such a 'negative' finding is also of interest. In studies of actual running recommender system, what is not possible to find out will be a natural part of the description. So far I have introduced my project and tried to place it in both a theoretical landscape, and in a landscape of previous research. It is now time to start approaching my research questions.

4 The harvesting of usage data

In this chapter I will try to describe the technical foundations for the bX Article Recommender, and how these relate to the capture of usage data. In other words this chapter approaches research question 1:

> How does the bX Article Recommender harvest and represent usage data?

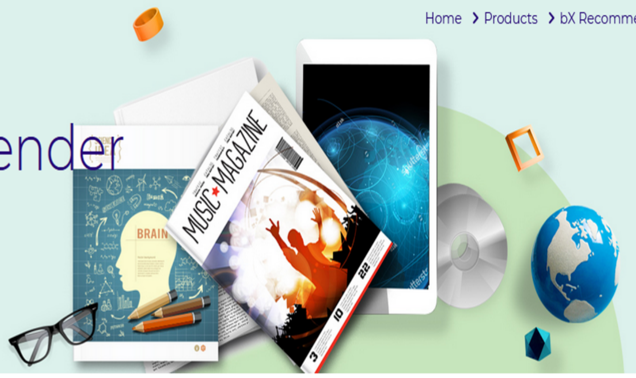
In light of the theoretical chapter, this is the same as asking: How does bX produce data in a form on which an algorithm can act. The ambition is to provide the reader with a notion of how technologies behind bX work. Again, the overall goal is to incorporate a solid technical understanding as part of the analysis, echoing Rieder when he writes that “my goal, however, was to find a level of description where an encounter between technical and larger cultural principles becomes possible, a level where neither ‘side’ is reduced to a caricature” (Rieder, 2017, p. 117).

To begin I will look at the description provided by Ex Libris at their bX product page[10]. Following that, the underlying technical architecture for bX and its components will be studied. Lastly the content of the chapter is exemplified via the description of a real-life scenario. The next couple of pages shows a screenshot of the bX product page, as it was captured in August 2021 (Figure 1).

10 https://www.exlibrisgroup.com/products/bX-recommender/

bX Article Recommender

Enrich your discovery experience



bX captures anonymous usage information from millions of scholars around the world, then leverages this data to enrich and expand the user discovery experience with relevant recommendations for articles and ebooks. Starting from an article of interest, bX provides users with other relevant articles for the same topic. While the initial article serves as an entry point, the recommended material can provide new inspiration for learning and broaden the scope of research, going beyond the initial search query. Depending on their nature, recommendations can help narrow or widen learning topics, provide new keywords to describe the topic, and allow the user to find items by chance through serendipitous discovery.

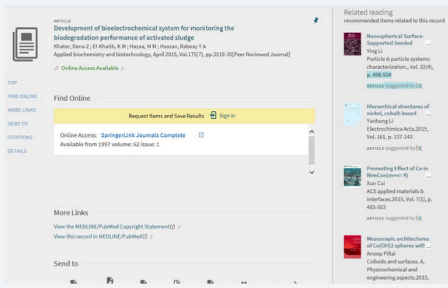
Meet the Expert: Christine Stohn on Exploration and Big Data

How does bX work?

bX harvests link resolver usage data from many academic institutions around the world. If two articles are used in the same session, the system analyzes the connection between them and stores the items in a co-retrieval network. Because bX recommendations are based on link resolver usage, they are truly platform- and content-neutral. The usage is generated through discovery systems, A&I databases, publisher platforms, and any other source that links users to full text via a link resolver. The articles may be from different journals, publishers, and platforms.

Where can I find bX Recommendations?

bX is a subscription service that is embedded into Primo and Summon discovery services, the SFX and Alma link resolver interfaces, and the Leganto reading list solution. APIs are available to easily embed the service and its article recommendations into other interfaces.



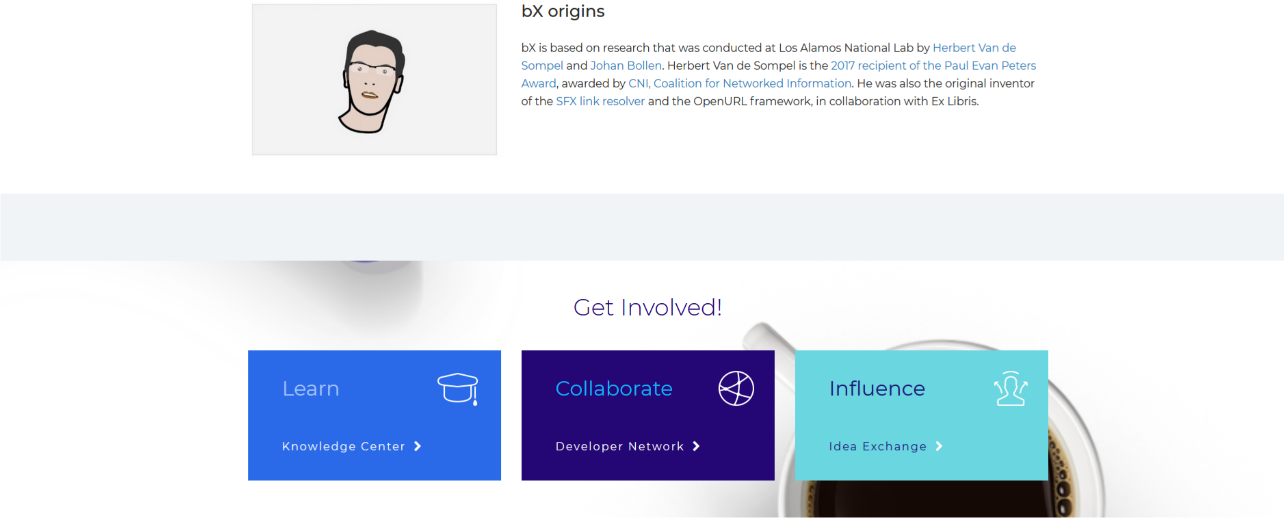


Figure 1: The bX Article Recommender product page (screenshot).

4.1 How bX Article Recommender works, as presented by Ex Libris

The above page is the most general presentation of bX. Its intended audience is probably existing or potential future users/costumers. As can be seen, the page is relatively sparse in content, but it does give some pointers towards how bX works. From the beginning it is made clear that usage information, more commonly labelled 'usage data', is the bread and butter of the system: "bX captures anonymous usage information from millions of scholars around the world, then leverages this data to enrich and expand the user discovery experience with relevant recommendations for articles and ebooks" (Ex Libris, n.d.-e). This sentence expresses the core idea behind bX: The exploitation of usage data – harvested from a global scholarly world – to generate recommendations.

Further on follow clues of a wider technological landscape: "bX harvests link resolver usage data from many academic institutions around the world. If two articles are used in the same session, the system analyzes the connection between them and stores the items in a co-retrieval network" (Ex Libris, n.d.-e). This can be seen as a very general description about what bX does and how it does it. The term 'co-retrieval network', for example, points to a broader, network theoretical/computer scientific landscape of ideas and techniques, but no further explanation is given. Later I will look closer at ideas and concepts from network theory, and how these are embedded into bX.

Another concept mentioned in the above quotation is that of a 'link resolver'. This technology is probably well known among more system-oriented librarians, but maybe not to most readers, so when stating that "because bX recommendations are based on link resolver usage, they are truly platform- and content-neutral" (Ex Libris, n.d.-e), it is not immediately clear what this means. What is clear, is that the link resolver somehow gathers usage data from across different platforms in the online academic world: "The usage is generated through discovery systems, A&I [abstracting and indexing] databases, publisher platforms, and any other source that links users to full text via a link resolver. The articles may be from different journals, publishers, and platforms" (Ex Libris, n.d.-e).

It is not directly stated *why* one should be interested in subscribing to bX, and the description of *how* recommendations are generated is in my view firmly operating on a surface level. Though you could argue that the page at least gives customers and/or end-users some idea of how the product works.

Nevertheless, towards the end of the page, almost as a pointer to the interested reader, the research that bX is based on is mentioned. It does not provide a direct hyperlink to it, but a search will eventually locate the article entitled "An architecture for the aggregation and analysis of scholarly usage data", written by Johan Bollen and Herbert Van de Sompel and published in the conference proceedings of the IEEE/ACM Joint Conference on Digital Libraries in 2006. In the acknowledgement of this work, Ex Libris are thanked: "We thank Ex Libris who played an enabling role in our research and development efforts" (Bollen & Van de Sompel, 2006, p. 307).

This work discusses and develops a prototype research paper recommender system, based on harvested usage data. The system resembles bX of today, though it is not named as bX in the article. Since research by Bollen and Van de Sompel is alluded to as the origin of bX, I will assume that the bX Article Recommender at least shares the core ideas and approach with the prototype developed by Bollen and Van de Sompel. Still, this work does *no*t give direct access to the inner details of the bX algorithm(s).

4.2 The research and underlying technologies of bX

When bX is put under scrutiny as a technical object that is both historical and social situated, it becomes clear that it does not stem from neither a historical nor a technological vacuum. The history of technologies that bX utilises started years before the public announcement of bX in 2009.

In 2006 scholarly information systems already collected large amounts of usage data, but seldom made use of it. Among others because of concerns of user privacy and lack of a standard framework for representing such usage data (Bollen & Van de Sompel, 2006, p. 298). These were obstacles that Bollen and Van de Sompel wanted to overcome with their architecture: "This section outlines a technical, standards-based architecture for recording, representing, sharing and mining usage information of scholarly information services" (Bollen & Van de Sompel, 2006, p. 299). And this 'usage information' (usage data) is the fuel for bX – it consists of aggregation of traces that individual users leave behind; mouse clicks, IP addresses, time and dates. On the basis of captured usage data, the authors stress that their framework can inform not only recommender systems but also help libraries with collection development, with measuring the quality and impact of scholarly works, and doing trend analysis of the scholarly world (p. 301).

The three most important technologies that help bX capture and exchange usage data is OpenURL, link resolvers and the OAI-PMH protocol. None of these were specifically created for bX, but they all make the existence of bX possible. They will be presented in more detail soon, but first I will provide an outline of the different stages in the architecture developed by Bollen and Van de Sompel.

4.3 The architecture for capturing usage data

The architecture can be divided into four steps or operations. All the steps revolve around usage data. Steps 1-3 are about the collection, representation and exchange of usage data. These in turn enable step four, service provision, which are services created on the basis of the data collected in step 1-3; such as the bX Article Recommender. These are summarised in the following table (Table 2):

1	Intra-institutional aggregation of usage data	Harvesting of usage data generated by users connected to an institution like a university.
2	Exposure of institutional usage data	Encode and expose this usage data in a standardized way to a log repository.
3	Inter-institutional aggregation of usage data	Harvesting of data from many different institutional log repositories.
4	Service provision	So called 'value-added' services, bX recommendations being one example.

Table 2: Standards-based architecture for representing, sharing and mining usage information of scholarly information services (based on Bollen & Van de Sompel, 2006).

In step one the starting point is the individual user at an academic institution and the usage data which he generates. This user has access to a certain *information environment* (Bollen & Van de Sompel, 2006). Limited to online resources, an information environment can for example consist of the online library catalogue from the institutional library, academic databases that the library subscribes to, and freely available scholarly search engines like Google Scholar. A simplified illustration of an information environment is shown on the next page (Figure 2). At the heart of this environment is the so called 'link resolver' (sometimes called 'linking server').

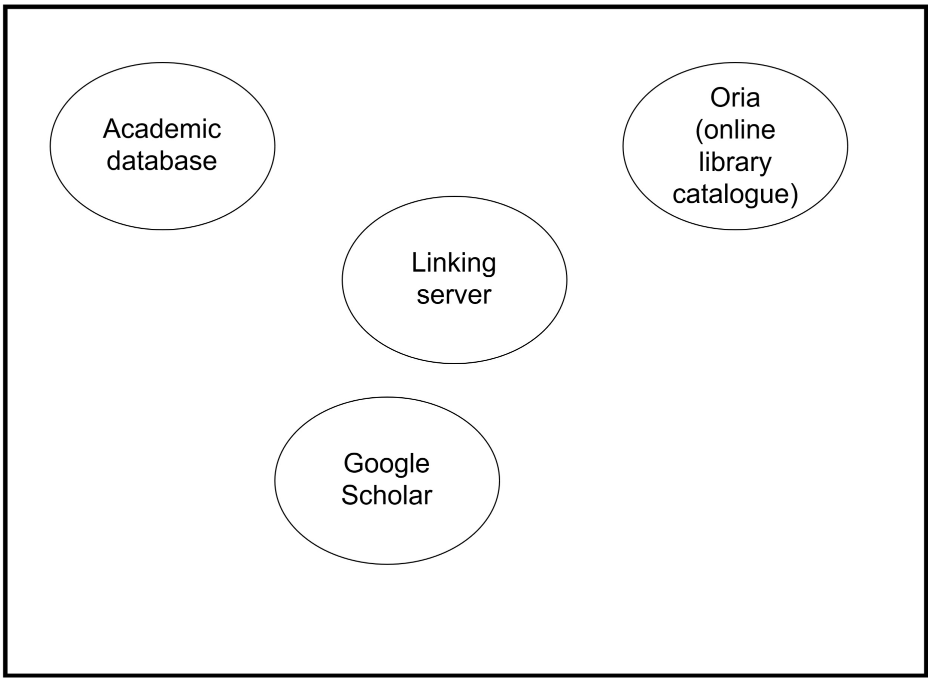


Figure 2: The information environment

The link resolver logs OpenURL requests generated by users in the information environment (Bollen & Van de Sompel, 2006, p. 299). OpenURLs are a special type of hyperlink, and when a user clicks on one, it is called a 'OpenURL request'. The link resolver can also be seen as a roundabout that 'traffic' in the information environment pass through. The resolver receives OpenURL requests and redirects them in the appropriate direction. Indeed, link resolvers were originally developed to ease user access to electronic collections:

> When a library subscribes to multiple databases and full-text resources, it can be a challenge to determine if the full-text of an article cited in one database is available in another resource. Link resolving software makes this task easier by acting as a bridge between databases; patrons easily go from a journal citation in one database to the full-text of the journal in another database. (Munson, 2006, p. 17)

It is the link resolver, along with the OpenURL framework, that allows for context-sensitive information environments, where the main goal is to provide seamless access for end-users. Links to full-text version can for example be inserted as OpenURLs into a result page in the library catalogue:

> This OpenURL is an HTTP GET request carrying metadata that are essential to identify the referenced work. It points to the linking server of the users' [sic] institution which contains

> a rule engine powered by a knowledge database that is typically maintained by the user's institutional library. (Bollen & Van de Sompel, 2006, p. 299)

In the prototype system developed by Bollen and Van de Sompel, a single usage event is defined as an OpenURL request targeted at a link resolver. These usage events are encoded and represented as a so-called ContextObject (Bollen & Van de Sompel, 2006, p. 300). More details on this will follow, but the most important feature of ContextObjects is that they contain metadata about an item (typically a journal article), the user, usage environment, and the time; *what*, *who* and *when.* This makes it possible, by analysing usage logs, to reconstruct a user session and follow a user's movement across the information environment. For example that the user clicked on an OpenURL in the online library catalogue, taking him to the full text version in an academic database, and then clicked on a reference in the literature list of the article.

By using the OpenURL framework the usage data is also stored in a standardized way at a local level. This paves the way for step 3, *inter-institutional* harvesting. When different institutions store their usage data following the same metadata scheme, and this usage data is exchanged in a standardized way, a global collection of usage data is possible. The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) is used for harvesting data from individual log repositories and combine them into one large database. Van de Sompel is one of the creators of the OAI-PMH, and the use of OAI-PMH to exchange library usage data was developed by him and others in an earlier work (Van de Sompel et al., 2003).

The figure on the following page (Figure 3) illustrates the flow of usage data from the local to the global.

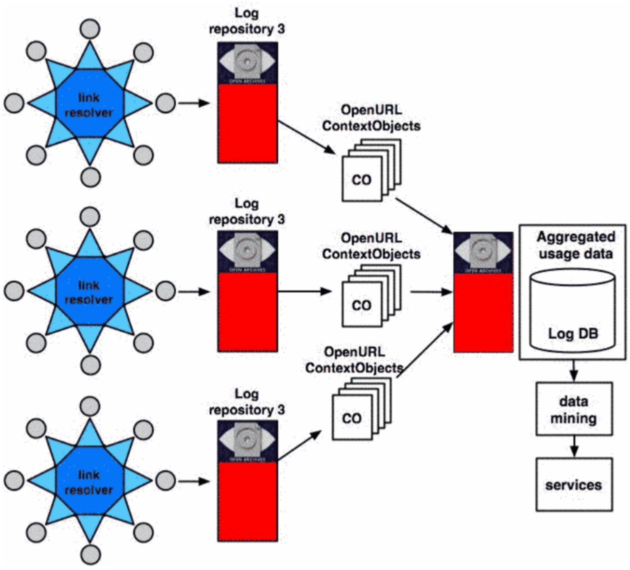


Figure 3: Architecture for usage data harvesting (reproduced from Bollen & Van de Sompel, 2006, p. 303)

Usage data from institutional link resolvers (blue) are exposed to local log repositories, which in turn are harvested by a large log database (the rightmost red-repository/the Log DB). To make use of data stemming from different information environments, it is essential that the data is represented in a standardized way. If not, a whole new layer of data cleaning would be added to the process and make it much more of a herculean effort. OpenURL helps secure the interoperability of the data: "The OpenURL provides a standardized format for transporting bibliographic metadata about objects between information services" (Van de Sompel & Beit-Arie, 2001, Introduction section). I will now look more close at OpenURL since it plays an essential role in how bX both captures and represents usage data.

4.4 OpenURL

An OpenURL is a type of Uniform Resource Locator (URL). URLs specifies the location of a resource and how it can be retrieved, and are commonly talked about as a link or hyperlink in everyday language. Typically a URL is static, it redirects users to a given resource in the same way. Basically, an OpenURL differs in that it considers the 'context' of the user and can be dynamically created based on this context. For example an OpenURL to a full-text version of an article can be inserted into a search result page only if the user have access to it via an institutional subscription. A static

URL, on the other hand, would redirect every user to the article no matter if they can access it or not. It was precisely from these kind of problems that the OpenURL framework evolved: Digital library users should not be presented with a link to a full text version if they do not have access to it.

The history of OpenURL started in the late 1990s-early 2000s, and since 2005 it has been a standard maintained by National Information Standards Organization (NISO)[11]. The growing availability and distribution of academic content online around the millennium, made it problematic both to maintain an overview of, and to link between, scholarly content. The 'appropriate-copy problem' was identified by people in library, publishing and information service communities. This problem arise when the same (online) resource have copies that exists in multiple places, and each copy has a different access policy (NISO, 2010, p. V). The best case scenario is that any given user is directed to the copy with the right access policy for him, which is often based on the institution the user belongs to.

A series of works on link resolvers and reference linking in online, library environments (Van de Sompel & Hochstenbach, 1999a, 1999b, 1999c), tackled the following problem statement: "Given bibliographic metadata, how does one present relevant extended services for it?" (Van de Sompel & Hochstenbach, 1999c, The problem statement section). This is related to the appropriate-copy problem presented above, and indeed, a couple of years later the OpenURL framework was presented as a part of the answer by Van de Sompel, this time with a co-author from Ex Libris (Van de Sompel & Beit-Arie, 2001). In this article Van de Sompel and Beit-Arie demonstrate how information environments that apply OpenURLs can include them with item metadata. As described earlier, the OpenURL manifest itself as a hyperlink:

> By clicking an OpenURL for a work, the user requests that the service component [link resolvers] deliver extended services for that work. The service component takes the OpenURL as input and collects metadata and identifiers for the work. It can do this by directly parsing such information from the OpenURL and/or by fetching it using the

11 NISO is a "not-for-profit membership organization that identifies, develops, maintains, and publishes technical standards to manage information" and is accredited by ANSI (American National Standards Institute), see more at https://www.niso.org/welcome-to-niso

> metadata pointer that was provided in the OpenURL. This pointer can lead into the original resource or into another one. Once identifiers and metadata are collected, the service component will evaluate them and provide extended service links to the user. (Van de Sompel & Beit-Arie, 2001, 'The OpenURL framework' section)

Van de Sompel was a central figure in the development of the OpenURL framework, and it is perhaps no surprise that its potential role in building a recommender system is already envisioned in this early work:

> An appealing side-effect of the OpenURL framework derives from the fact that by clicking OpenURLs in distributed information resources, users are not only requesting extended services. They are also enabling their service component to record the request, and as such to accumulate a log of the users' actions across resources. Since it is fair to assume that users will mainly click OpenURLs for *preferred* works, the resulting log is a collection of user preferences that can become the knowledge base upon which to build recommendation services. (Van de Sompel & Beit-Arie, 2001, 'The OpenURL framework' section)

Note that this quotation includes an assumption about users' preferences. In other words, the users' movement across the information environment is not understood as random. In chapter 5 I will show how this assumption is developed and built upon in later works by Bollen and Van de Sompel, so that it comes to be a statement about relatedness, and essential for how bX exploits the harvested usage data.

4.4.1 The OpenURL ContextObject

I previously mentioned that the harvested usage data is represented and exported as OpenURL ContextObjects. On an abstract level these are defined as "an information construct that binds a description of a primary Entity – the referenced resource – together with descriptions of Entities that indicate the Context" (NISO, 2010, p. 6). In other words it combines metadata about some entity with metadata about context. In the case of bX the 'primary Entity' is normally a journal article.

Table 3 shows the different parts of the ContextObject. A researcher is reading article X in an online academic database. From this article she clicks on Rieder (2017) in the reference list. A ContextObject capturing this usage event, might contain the following:

Entity	**Description**	**Example**
Referent	The referred resource, of which the ContextObject is created.	An article, in this instance Rieder (2017).
ReferringEntity	The resource referring the Referent.	The above article was in the reference list of the article X, which the researcher is reading online. Article X is the ReferringEntity.
Requester	The resource that request some service regarding the Referent.	The researcher. Represented for example by an IP address or a randomly generated anonymous session ID.
ServiceType	Type of service requested.	Full-text access to Rieder (2017).
Resolver	The target for the service request.	The link resolver of the researcher's institution.
Referrer	The resource that generates the ContextObject.	The academic database Web of Science.

Table 3: The OpenURL ContextObject, based on (NISO, 2010, pp. 11–12).

This data is typically encoded as XML following the predefined XML ContextObject schema. Thus all ContextObjects can be represented in a standardized way and generate standardized data, which more easily can be manipulated by services such as bX.

I will now try to bring the contents of the chapter so far more into life by presenting a common real-life scenario. It is worth to have in mind that the events depicted are the type of 'usage events' captured in the architecture developed by Bollen and Van de Sompel, namely OpenURL requests. And it is this data that bX ultimately generates its recommendations from. The scenario also functions as a repetition of the overall technical structure.

4.5 Example scenario

Imagine a researcher at an academic institution. Sitting by the computer in her office she has access to several different scholarly databases, an online library catalogue (in this case Oria, an instance of the Ex Libris discovery system Primo), and online search engines like Google Scholar.

The researcher's institution also subscribes to a link resolver (Alma link resolver), making it possible to retrace her 'travels' through the information environment, seeing how she gets from A to B, or from A to C, and harvest this information[12]. Multiply this with thousands of users at her own institution, and then with thousands of users at hundreds of other institutions, and we can see the foundation of the bX Article Recommender. To make context-sensitive services, and, by extension, have context-sensitive usage data, a link resolver is needed. The link resolver is the centre of the whole process, "indeed, a linking server logs OpenURL requests of all users of the community originating from many of the available distributed information sources" (Bollen & Van de Sompel, 2006, p. 299). To repeat the illustration from earlier, the researcher's simplified information environment can be presented as following (Figure 4):

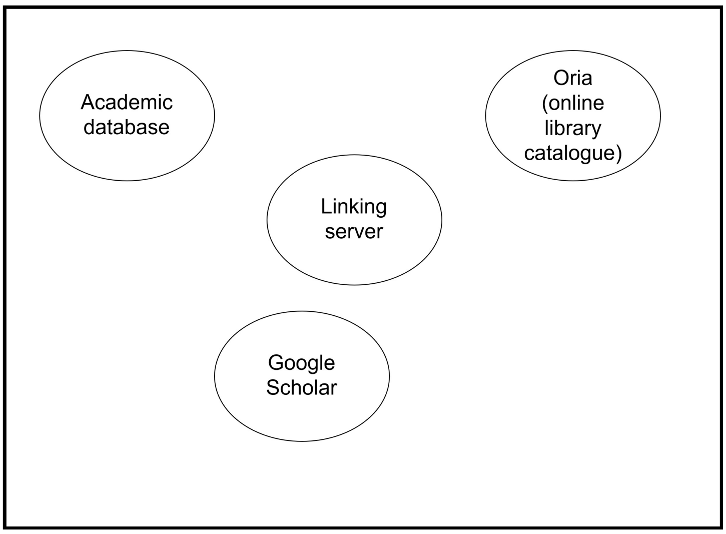
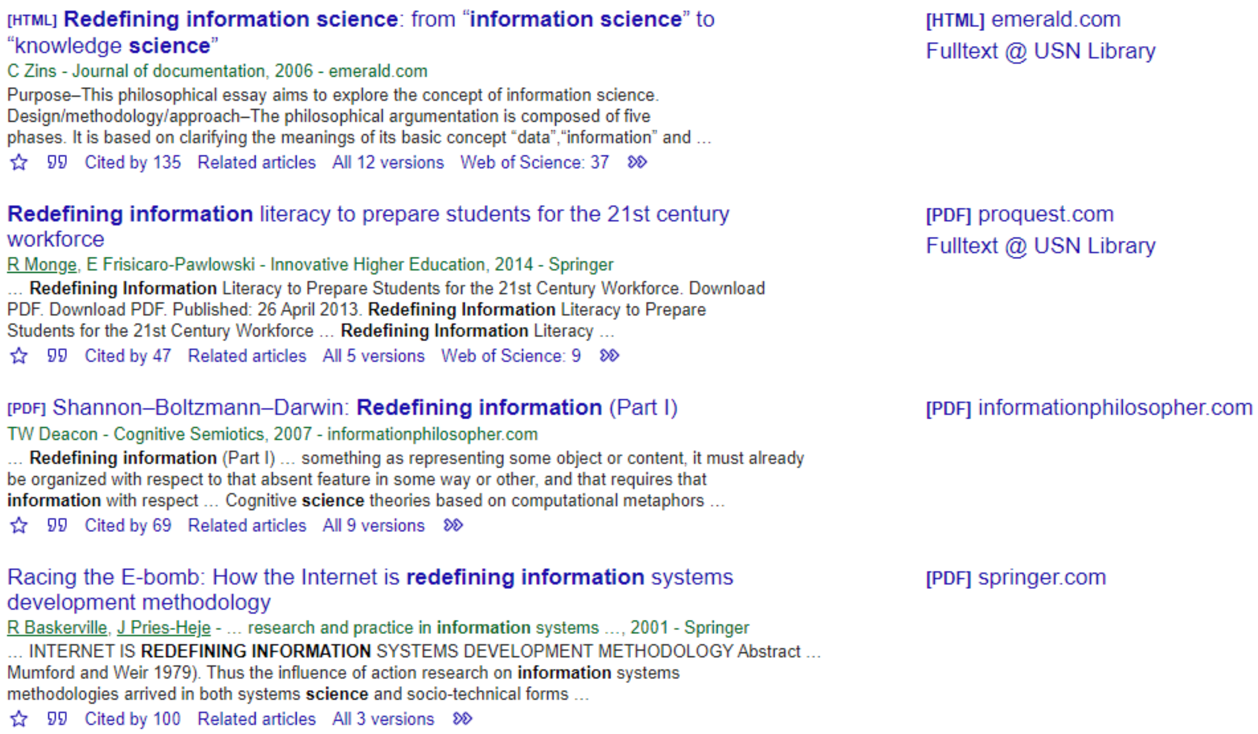


Figure 4: The information environment

---

12 It should be noted that this data is anonymized. The privacy aspect of the architecture is discussed in Bollen and Van de Sompel (2006) – but this is not something I concentrate on in my thesis. There exists much research on privacy and recommender systems, since recommendations often are connect to *personalisation* (see for example Friedman et al., 2015)

Say that the researcher is aware of an article whose title include the phrase “redefining information science”. She searches for it in Google Scholar, and the following results are presented to her (Figure 5):


[HTML] **Redefining information science**: from “**information science**” to “knowledge **science**”
C Zins - Journal of documentation, 2006 - emerald.com
Purpose–This philosophical essay aims to explore the concept of information science. Design/methodology/approach–The philosophical argumentation is composed of five phases. It is based on clarifying the meanings of its basic concept “data”,“information” and …
Cited by 135 Related articles All 12 versions Web of Science: 37
[HTML] emerald.com
Fulltext @ USN Library

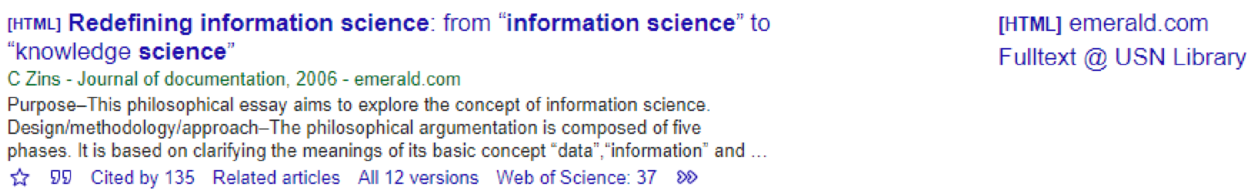
Redefining information literacy to prepare students for the 21st century workforce
R Monge, E Frisicaro-Pawlowski - Innovative Higher Education, 2014 - Springer
… **Redefining Information** Literacy to Prepare Students for the 21st Century Workforce. Download PDF. Download PDF. Published: 26 April 2013. **Redefining Information** Literacy to Prepare Students for the 21st Century Workforce … **Redefining Information** Literacy …
Cited by 47 Related articles All 5 versions Web of Science: 9
[PDF] proquest.com
Fulltext @ USN Library

[PDF] Shannon–Boltzmann–Darwin: **Redefining information** (Part I)
TW Deacon - Cognitive Semiotics, 2007 - informationphilosopher.com
… **Redefining information** (Part I) … something as representing some object or content, it must already be organized with respect to that absent feature in some way or other, and that requires that **information** with respect … Cognitive **science** theories based on computational metaphors …
Cited by 69 Related articles All 9 versions
[PDF] informationphilosopher.com

Racing the E-bomb: How the Internet is **redefining information** systems development methodology
R Baskerville, J Pries-Heje - … research and practice in **information** systems …, 2001 - Springer
… INTERNET IS **REDEFINING INFORMATION** SYSTEMS DEVELOPMENT METHODOLOGY Abstract … Mumford and Weir 1979). Thus the influence of action research on **information** systems methodologies arrived in both systems **science** and socio-technical forms …
Cited by 100 Related articles All 3 versions
[PDF] springer.com

Figure 5: Result list from Google Scholar (screenshot).

The first result is the article the researcher looked for:


[HTML] **Redefining information science**: from “**information science**” to “knowledge **science**”
C Zins - Journal of documentation, 2006 - emerald.com
Purpose–This philosophical essay aims to explore the concept of information science. Design/methodology/approach–The philosophical argumentation is composed of five phases. It is based on clarifying the meanings of its basic concept “data”,“information” and …
Cited by 135 Related articles All 12 versions Web of Science: 37
[HTML] emerald.com
Fulltext @ USN Library

To the right there are two hyperlinks: [HTML] emerald.com and Fulltext @ USN Library. These are context-sensitive URLs, and are provided because Google Scholar is a OpenURL compliant service. By identifying the IP address[13]Google Scholar ‘knows’ that the researcher belongs to an institution that has valid full-text access to this particular resource. This information – what online resources the researcher’s institution has access to – is exported between Alma (the library system) and Google Scholar at a weekly basis (Ex Libris, n.d.-h). This illustrates how link resolvers extend the

13 Or the user has manually stated that she belongs to the USN Library via the Google Scholar settings. This can be done without logging in and is useful when working from outside of campus.

content of the online library collection beyond the library catalogue, and into different parts of the information environment.

The researcher then proceeds to clicks on the “Fulltext @ USN Library” link. This takes her to the library catalogue of the USN Library. When she clicked the link in Google Scholar an OpenURL request was sent to the link resolver. The resolver then processes it, searches the collection for the corresponding resource, adds relevant services to the result page and displays it to the end-user. The services added to the result page includes links to full text and recommendations generated by bX (Ex Libris, n.d.-a). The process is illustrated by the figure below (Figure 6).

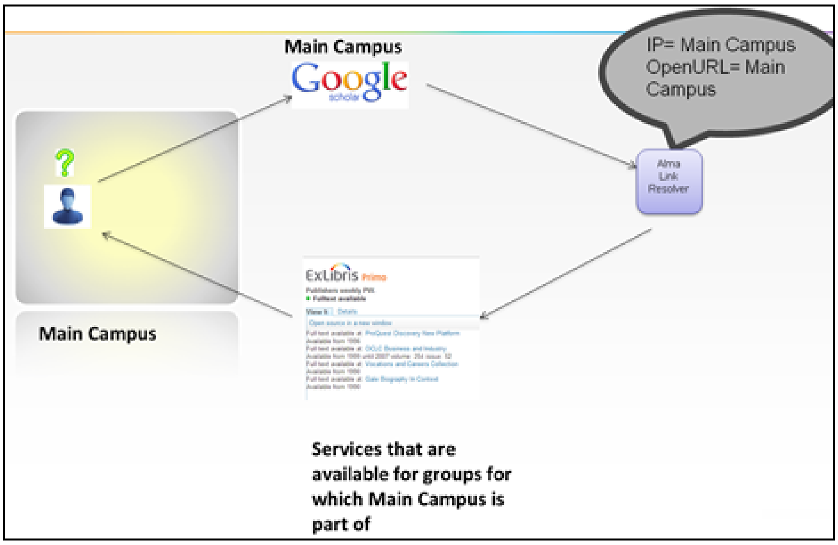


Figure 6: Google Scholar to library catalogue (taken from Ex Libris, n.d.-i).

In the described scenario two (OpenURL) links to the full text are added to the result page, because access is provided by two different databases (Figure 7). On the right hand side of the result page (omitted from the screenshot below) the recommendations generated by bX are displayed.

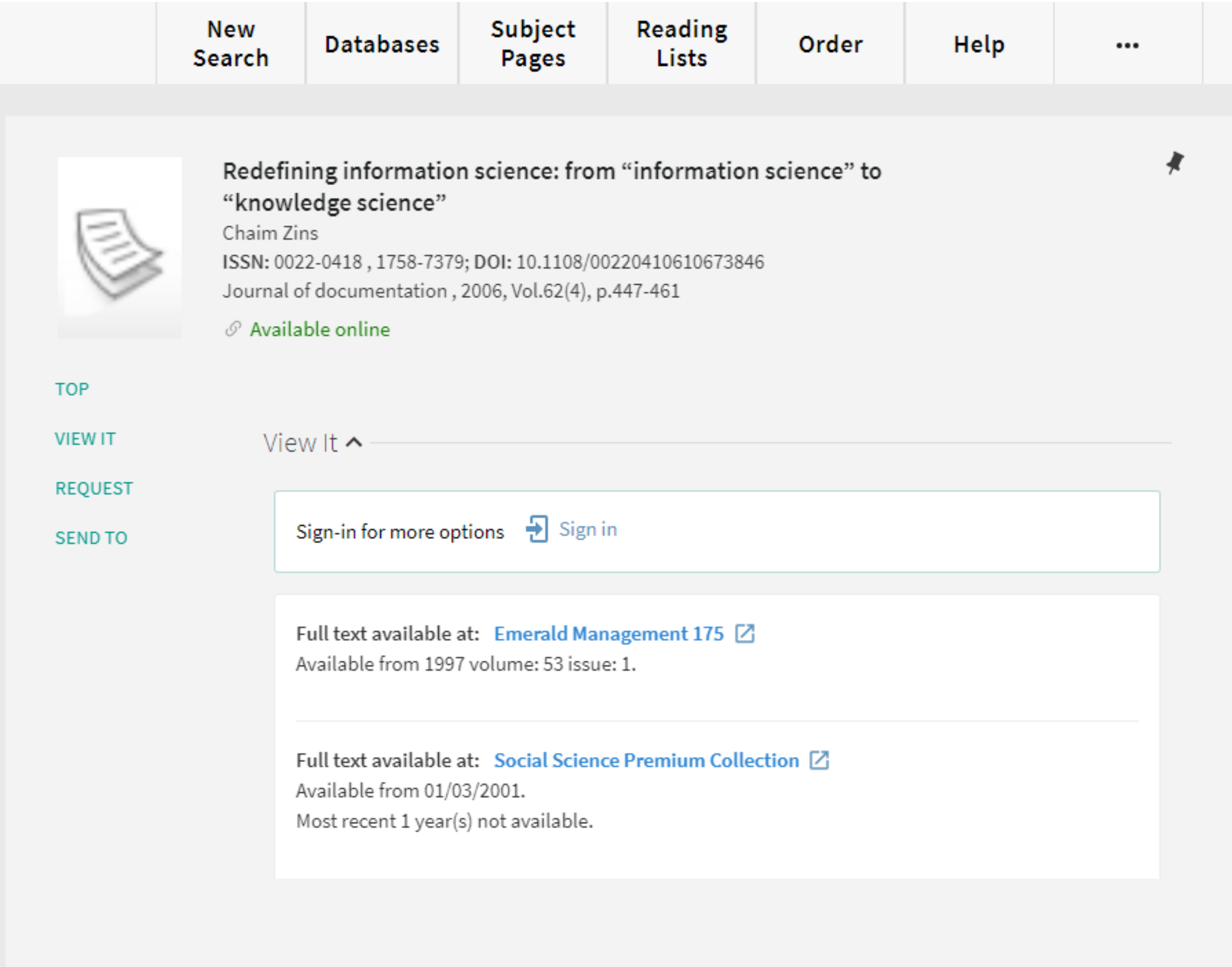


Figure 7: Extract of result page in Oria (screenshot).

If the researcher at this point looks at her web browser’s address bar, she will see an OpenURL. Although a little dense, some of the content is very much human-readable, as seen from the following excerpt. For extra ease of readability I have also decoded, split up and shortened the URL:

```
https://bibsys-almaprimo.hosted.Ex
Librisgroup.com/primo-explore/openurl?
sid=google
&auinit=C&aulast=Zins
&atitle=Redefining information science: from “information science”
to “knowledge science”
&id=doi:10.1108/00220410610673846
```

In a large degree this is metadata about the *Referent* (article title, author and DOI). But there is also a trace of where the request originated (sid=google), this is the *Referrer* – meaning that the request originated from Google Scholar. Seemingly this does not contain much information about the *who* and *when* of the request. But by clicking the link in Google Scholar, the researcher requested services from her institutions link resolver for the article “Redefining information science”. Links to the full text version of the article were inserted into the result page:

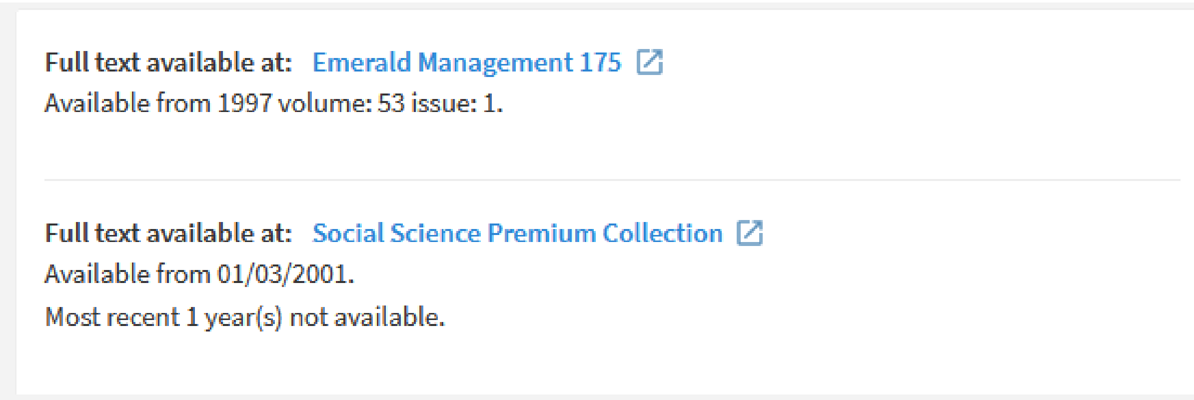


Figure 8: OpenURL links to full-text (screenshot).

These links can be more closely inspected via web browser tools, leading to the XML ContextObject behind them. The ContextObject contains much more metadata about the requester, f. ex. IP address and a timestamp for the usage event (year, date, hour, minutes), the services that can provide the user with full text access and so on. Two data points from the ContextObject are shown below as an illustration:

```
<key id="user_ip">user IP here</key>
<key id="rfr_id">info:sid/primo.Ex Librisgroup.com-
proquest_pasca</key>
```

The “rfr_id” stands for Referrer ID. This is the ID for the service that generated the ContextObject, in this case it is the library discovery system itself.

To conclude the scenario: If the researcher clicks on the ‘Emerald Management 175’ link, she is taken to the full text version in the Emerald Insight database. In this scenario, by analysing the link resolver logs, the movement from Google Scholar → Primo → Emerald Insight can be reconstructed. And this allows for recommender systems that recommend items that are often accessed one after another:

> Such a sequence of requests can be recorded by the linking server and hence exploited by click-stream based methods of log analysis to reveal temporal trends in user behaviour and recommending items which are often accessed in a particular sequence. Such temporal patterns would be very difficult, if not impossible, to reconstruct from the aggregation of the logs obtained from each of the individual information services in the user's environment. (Bollen & Van de Sompel, 2006, pp. 299–300)

The centrality of link resolvers cannot be understated. It is because of them that the researcher can be shown not only links to full-text, but also recommendations generated by bX.

The screenshot to the right is the list of recommendations presented to the researcher in the above scenario. Only the three first are shown here. These are recommendations created by bX; a full-scale version of the potential recommender system prototype that Bollen and Van de Sompel (2006) developed. The article titles are also themselves OpenURLs. If the researcher clicks on one of them, she is taken to a new result page in Oria, where links to the full-text are inserted and new recommendations are generated, if available. And this is logged by the link resolver, adding one more link to the existing chain.

Figure 9: List of bX recommendations (screenshot).

4.6 Summary

In this chapter I have approached research question 1: How does the bX Article Recommender capture and represent usage data? I have presented an infrastructure that can harvest huge amounts of *structured* usage data. And I have hopefully demonstrated how this architecture is reliant on the technologies of link resolvers, the OpenURL framework and the OAI-PMH. These are not technologies built specifically for bX, but I would argue that they make bX possible. Except for pinpointing that it is important to capture the context of the user, I have not really explored what happens at the data manipulation stage. This is the theme of the next chapter.

5 The exploitation of usage data

How does bX exploit harvested usage data to build its recommendations? Why did the researcher in the previous chapter see exactly those recommendations and not others? As Rieder (2017) points out: A working software system contains the result of many human decisions when it comes to how input data is to be tweaked, what parameters should be set, the outputs one wants produced and so on (p. 102). How can the bX Article Recommender be studied as an instance of *algorithmic literature,* a program written with a concrete purpose and in a concrete setting (Rieder, 2017, p. 101)? The guiding question for this chapter is research question number two: How does the bX Article Recommender exploit usage data to generate recommendations?

In the previous chapter I showed how usage data is harvested and represented with a purpose in mind. To be usable for bX the data must include information that makes it possible to recreate sessions by single users, and see what resources were requested and in what order. As such it is no clear cut line between the data harvesting and the data exploitation stage; *how* the data is harvested is affected by *how* the data is to be used. But it is perhaps possible to say that at the data manipulation stage bX' 'interested reading' of reality becomes clearer: "On this level of signification, data mining techniques attribute meaning to every variable in relation to a purpose" (Rieder, 2017, p. 111). Is it possible to gain more insight into what parameters/variables exists in bX, and what meanings are attributed to them?

To approach these questions I will again look at the architecture developed by Bollen and Van de Sompel, but also at earlier research by the two. I will also study the bX API documentation from Ex Libris, which describes some parameters that are at play in bX. Together these sources shed some light on how bX exploits its usage data and calculates its recommendations. Together with the last chapter on data harvesting, this chapter also forms a developmental argument of how bX has developed. Mason writes the following about this approach:

> You will construct a developmental argument if you want to explain how social phenomena, social relationships, social processes and so on have developed or come to be. Here, the logic of explanation is centred on the idea that a meaningful process of development, or a story, or a narrative, or an 'archaeology', can be invoked. (Mason, 2018, p. 221)

The previous chapter showed the technological dependencies of the data harvesting stage. This chapter will show that the ideas at the data exploitation stage also do not stem from a vacuum, i.e., *ex nihilo nihil fit*[14]. As such this chapter will also touch upon research question 3: What epistemological ideas and interpretations of the world operate in the bX Article Recommender?

5.1 Previous research by Bollen and Van de Sompel

To begin with I will look at research where Bollen and Van de Sompel are involved. Either works where both of them contributed, or works by Bollen/Van de Sompel together with others. Both of them have done quite a lot of work exploring potential uses of digital library usage data. Although it would have been possible to follow the historical, technological thread in these works even more in depth, I have mainly looked for things that can contribute to some informed guesses and descriptions of what is going on inside of bX today. But I want to be clear that this does not mean that the following will describe the internal algorithms of the bX Article Recommender. The relevance of this chapter is based on the assumption that the way bX works at least share the core ideas and technical details provided in Bollen and Van de Sompel (2006) and, by extension, in some of their previous work. In my eyes this assumption is strengthened by Ex Libris' own description of bX (as seen in the previous chapter), and by them stating that bX is based on research by Bollen and Van de Sompel.

Some of the works I will mention are directly quoted in Bollen and Van de Sompel (2006), while others are not. Still, I think it is fair to assume that ideas from their earlier works and projects have influenced the work from 2006, even though they are not cited. Also work B might be cited in Bollen and Van de Sompel (2006), and then work B might cite work C and so on. This is related to the idea of citation trails – a quite common method for discovering and identifying relevant literature.

In their prototype of a recommender system Bollen and Van de Sompel describe the necessity of creating a network of items relationships out of the usage data "in order to perform more sophisticated, network-based methods of *Referent* ranking and to create recommender services able to link one *Referent* to the other" (Bollen & Van de Sompel, 2006, p. 303). 'Referent' is here

14 Nothing comes of nothing, or, of nothing comes nothing

taken from the OpenURL framework, in the case of bX meaning a journal article. Here ‘network-based’ methods are introduced as a technique for exploiting the usage data and building a recommender service. They refer the reader to an earlier work for more details into the methodology.

This work is the article “Toward alternative metrics of journal impact: A comparison of download and citation data” (Bollen et al., 2005). This article presents a method for calculating journal impact based on usage data (in this case, article downloads). It also embodies a critique of the Institute for Scientific Information (ISI) Impact Factor as developed by Eugene Garfield (see for example Garfield, 2006 for the creator’s own origin story). Instead of creating a network of article relationships, as in bX, they create a network of journal relationships. They demonstrate how journal impact in a limited environment, such as a faculty, can vary strongly from the global ISI Impact Factor (Bollen et al., 2005).

The connection between this work and Bollen and Van de Sompel (2006) is further strengthened when the authors state that “the proposed methodology can be applied similarly to the construction of article networks” (Bollen et al., 2005, p. 1429). The methodology is based on the use of relational matrices – and the overarching field of network theory – in the calculation of ranking. But, before such calculations are applied, the harvested usage data goes through a filtering process. This is the first step in the data exploitation process. The guiding idea of this step is the concept of co-occurrence.

5.2 Co-occurrence

Bollen et al. (2005, pp. 1425–1426) introduce the concept of ’reader generated network’; a network of relationships between academic journals, created from download data. It is named so because it is the readers (users) that have generated the data of which the network is generated. This is opposed to an ‘author generated network’ which is based on citations: “More precisely, rather than determining how often authors have cited articles in journal B from articles published in journal A, we examine how often articles in journal B have been downloaded within the same session as articles in journal A” (Bollen et al., 2005, p. 1426). This is the central idea of co-occurrence, and that which ‘co-occure’ are article downloads. Another term for the concept is ‘co-retrieval’.

Their argument for the validity of the method is based on the ’retrieval coherence assumption’:

> The main principle underlying the generation of the RGN [reader generated network] is the Retrieval Coherence Assumption (RCA), namely the notion that when a DL [digital library] user downloads a set of documents he or she is often driven by a specific information need. From the RCA it follows that when we observe a reader sequentially downloading a set of articles, we can infer a certain probability that the downloaded articles, and thus the journals in which they appeared, are related. Their degree of relatedness can be determined on the basis of two factors. First, the closer the documents are located within a sequence of reader downloads, the more related they are expected to be (Jones, Cunningham, & McNab, 1998; Pirolli & Pitkow, 1999; Chi, Pirolli, & Pitkow, 2000). The RCA thus applies most reliably to the shortest retrieval sequences, i.e. pairs of documents downloaded one after the other. Second, the more frequently a particular pair of documents are downloaded by a group of readers, the greater the degree to which we can assume these documents to be related. (Bollen et al., 2005, pp. 1426–1427)

The authors say that for two downloads to be deemed as a co-occurrence, the time interval between the downloads must differ by less than one hour. If the downloads are requested by the same user within this time frame it is a co-occurrence, and assumed to be related per the retrieval coherence assumption (Bollen et al., 2005, pp. 1427–1428).

5.3 The relationship matrix

Bollen and co describe how sociology has long been interested in questions regarding the ranking of actors in a network according to their status. They contrast this approach, which is sensitive to more complex relationships, with the simpler ISI Impact Factor, which they label a ’frequentist approach’, in that it simply counts a number of occurrences (Bollen et al., 2005, p. 1422). The authors thus position their work and views within the realm of network theory.

In the methodology developed in Bollen et al. (2005) the actors are journals, and their status is based on their relationship to other journals. The filtering process outlined above, where usage logs are scanned, results in sets of co-retrieval events that ultimately are used to generate a relationship matrix. To further describe this process a short introduction to matrix terminology is necessary.

A matrix is a way to represent a network, organized into rows and columns. The dimension of a matrix is described by R×C, where R is rows and C is columns. Thus a 10×5 matrix has ten rows and five columns. A matrix is typically labelled by a bold, uppercase letter (matrix **A**). One particular point in the matrix, say row 3, column 2 is called a cell. A cell is referred to as $\mathbf{a}_{ij}$ where *i* is the row number and *j* is the column number. So the above mentioned cell would be described as $\mathbf{a}_{32}$ (Lizaardo & Jilbert, 2021, 4.1 From graph to matrix).

There are different types of matrices and the relationship matrix is one of them. The characteristic trait of a relationship matrix is that the value of a cell is determined by relationships. In a relationship matrix, the rows and the columns are the same cases, "so each cell, as the intersection of two cases (the row case and column case) gives us the value of the relationship between the cases" (Lizaardo & Jilbert, 2021, 4.2 Relationship matrices). Following is a 3×3 relationship matrix of journal articles (matrix **A**). The cell values are the strength of the relationship between two journals, based on download data, and created the same way as in the approach described by Bollen et al. (2005).

		1	**2**	**3**
Journal A	**1**	0	2	1
Journal B	**2**	0	0	4
Journal C	**3**	5	0	0

Table 4: Example of a relationship matrix (matrix A)

The column with Journal A, B, C is only included for readability. The journals are assigned an index number (Journal A = 1, and so on), which is used both in the rows and the columns. The first thing to notice is that the matrix is perfectly square; a relationship matrix will always have the same number of rows and columns. The above matrix can be described as a square matrix of order *n*, where *n* is the number of rows, in this case three.

The value of the cells in the diagonal ($\mathbf{a}_{11}$, $\mathbf{a}_{22}$, $\mathbf{a}_{33}$) are all zero. That is because the journals does not have a relationship with themselves. Furthermore, the value in $\mathbf{a}_{1,2}$ – the relationship between journal A and journal B – is different from the value in $\mathbf{a}_{21}$. One might think that these should be the same, but this is not the case. That is, the relationship between journal A and B is *not* the same as the relationship between journal B and A. The captured *network tie* is asymmetric (Lizaardo & Jilbert, 2021, 4.2 Relationship matrices). Matrix **A** shows that there were recorded instances where users downloaded an article from journal A and then downloaded an article from journal B ($\mathbf{a}_{12}$), but no recorded instances of users downloading articles from journal B and then journal A ($\mathbf{a}_{21}$).

This example network might also be described as a weighted network, with varying 'weight' between the edges. That is, different journal relationships have different weights. This introduces the notion of reciprocity. In the example above the relationship between Journal C and Journal A can be described as uneven. There is more reciprocity, or larger weight (5), in the dyad {Journal C, Journal A}, than in the dyad {Journal A, Journal C}, which only has a weight of 1 (Lizaardo & Jilbert, 2021, '3.4.1 Reciprocity').

5.3.1 Calculation of relationships

Returning to Bollen et al. (2005, p. 1428), they write that the set of co-retrieval events can be represented by the matrix R where every cell represents the strength of journal relationships between any given pair of journals. In other words, a relationship matrix as described above. The value of each cell is calculated by an algorithm. For every instance of a co-occurrence this algorithm readjusts that particular journal relationship weight, i.e. it increases the numerical value of the related cell in the matrix. The pseudo-algorithm for calculating relationship values is formalized as follows (p. 1428):

$\forall_{ij} a_{ij} = 0$

for (i = 1; i < n + 1; i++) {

$e_i = v_i, v_j, t(v_i, v_j)): r_{ij} += f(e_i)$

}

In natural language this states that every cell from the start is set to 0, i.e., there are no *a priori* recorded relationships between journals. It then loops over every recorded co-retrieval event in a set (download log) and adjust the weight of the recorded relationship between journal v_i and journal v_j accordingly. $e_i = v_i, v_j, t(v_i, v_j))$ is a co-retrieval event, where document v_i and v_j are downloaded, and $t(v_i,v_j)$ is the time in seconds between the downloads. r_{ij} is the corresponding cell in the relationship matrix. The function $f(e_i)$ determines the value to be added to the journal relationship weight (Bollen et al., 2005, p. 1428).

That means the function $f(e_i)$ plays an essential part, it decides the value that is to be added for each instance of a co-occurrence. The function will affect the relationship weight, and eventually, the output produced by the system. The authors themselves state that "the reinforcement function $f(e_i)$ can be varied according to the nature of the data set on which the algorithm is operating" (Bollen et al., 2005, p. 1428). In their proposed methodology they define the function $f(e_i) = 1$. In this case the value of a cell in the relationship matrix is the exact number of times the co-retrieval has happened (p. 1428).

It should be noted that as long as the function for adding weight merely equals a fixed number, that is, the value to be added is always the same, then it does not matter if this number is set to 1, 8 or any other. The dimensions between the values in the matrix remain the same. The value of the number to be added only matters in relation to a score threshold[15],where a recommendation needs to have a minimum score to be displayed. I will show that such a threshold exists in the bX Article Recommender.

Bollen et al. (2005) further state that a lone user only contributes a small amount to the calculated weight between journals. It is the aggregate effect of consistent download patterns (A and B often downloaded in the same session, one after the other) that establish significant journal relationships and strengthens the assumed relationships between journals (p. 1427). Furthermore the authors expected a large amount of zero-entries, that is journals with no recorded weight between them, and "indeed, only a small fraction of all possible, directed journal relationships can be meaningful and therefore matrix densities will be low" (p. 1429). Many co-retrievals happened

15 And perhaps for calculations purposes, where shorter numbers = less demanding calculations?

only once, the largest recorded relationship weight was 22 (p. 1430). The relationship matrix in the bX Article Recommender, which harvest usage data from institutions all over the world, will have item relationships with plenty more co-retrievals than 22. But it probably shares the feature of having a large amount of zero-entries and low matrix 'density'. The implications of these characteristics will be further discussed in chapter 6.

5.4 Other works

In 2003, Bollen co-authored a work that seeks to utilise usage data from a digital library to discover research trends (Bollen et al., 2003). It almost reads like an earlier version of the methodology for creating journal networks. It both introduces the retrieval coherence assumption and calculates weights between journal based on download sequences and time between downloads (Bollen et al., 2003, Introduction section). I think this strengthens the assumption that earlier works and ideas by the authors are part of the prototype recommender system (Bollen & Van de Sompel, 2006), and by extension, liken the solutions applied in the bX Article Recommender.

Interestingly this work hypothesises a more dynamic calculation for readjusting journal relationships weights. Instead of simply stating that $f(e_i) = 1$, the time between downloads can take the role as a variable: "The amount by which a journal relationship weight is increased can be varied according to the time passed between the two downloads or any other function" (Bollen et al., 2003, Networks of journal relationships section). Here a whole new possibility is introduced, namely that the time between downloads not merely functions as a threshold for what is and is not a co-occurrence, but also as something that affects the calculation of relationship weights. And, the remark 'any other function' illustrates that a developer indeed can add any parameter they want; for example that downloads on a Monday should add more weight than downloads on a Friday. Or, perhaps, that co-retrievals from certain journals or databases are worth more than others.

The idea of building networks based on digital library usage logs was also fronted by Bollen in 2002. In this work the network is generated at document level, and is meant to assist with digital library management. Again the argument is that such a network gives a better picture of the research activities of the local user community, than traditional usage statistics like frequently downloaded articles etc. (Bollen & Luce, 2002). It also explicitly envisions a recommender system that uses a network to generate recommendations: "Second, given a previously generated document link structure, a class of novel recommendation systems can be implemented that does not require text-query matching for retrieval, but operates on network structure to generate document recommendations" (Bollen & Luce, 2002, Conclusion section). The authors also thank Van de Sompel for his contribution to the ideas and principles presented in the article, and that "this is particularly the case for his input concerning the generation of document relationships from user retrieval sequences recorded in DL server logs" (Bollen & Luce, 2002, Acknowledgments section).

More examples could be given, but I think the discussion in the chapter so far is sufficient enough to establish the fact that both Bollen and Van de Sompel shares a common reservoir of ideas and interests, and that they have operated within the same research areas and communities prior to their co-authored work in 2006 (Bollen & Van de Sompel, 2006). I will now summarize the findings of this chapter so far and what they, by extension, can say about the bX Article Recommender.

5.4.1 Summary

The following are the main findings from the mentioned research. As stated several times, I hold them to be a valid description of how the bX Article Recommender works at a general level.

- There exists a time parameter in the algorithm. There is a time threshold during which two articles must be downloaded to count as a co-retrieval and thus be seen as an expression of relatedness. In bX it is not article downloads per se that are analysed, but OpenURL requests (see page 30).
- A relationship matrix is created on basis of analysis of usage data logs. This allows for the appliance of graph theory and network calculations.
- There exists a reinforcement function in the algorithm that readjusts the value of a relationship between two articles. In the method for journal impact calculations (Bollen et al., 2005) this only a number and set to 1. It is possible that the bX algorithm uses a more complex and dynamic function. For example that the time between downloads affects the added weight, a possibility mentioned by Bollen et al. (2003).

- It is said that it is the repeated action by a community of users that is important. The lone user does not contribute much to establish relationships between articles; the single co-occurrence event does not add much weight to a relationship by itself.
- Both Bollen and Van de Sompel have done several works on building services based on usage data from online libraries. In these works the possibility of a recommender system for journal articles, at a global scale, has been envisioned.

What is new in the architecture (Bollen & Van de Sompel, 2006) compared to these earlier works, is the global scale and ease of interoperability that the OpenURL technology provides. Indeed the technology of link resolvers and OpenURL merely makes possible what Bollen and Van de Sompel 'prepared' in earlier works.

I will now shift the focus to freely available documentation from Ex Libris relating to bX. In some ways these data sources are closer to the real bX Article Recommender than the research reviewed above. To begin with I will look at the bX API documentation.

5.5 bX API documentation

An API (application programming interface) is a set of rules and features inside a software program that allows other software to interact with it. That is to say that a program can interact directly with another program, sidestepping a human user interface (*API - MDN Web Docs Glossary*, 2021). The bX API allows developers to build services on top of the bX data. This could for example be a small script that inserts a list of recommendations into a web site. In this thesis I have primarily concentrated on the bX Article Recommender as integrated into the discovery system Primo. These are the recommendations that users such as my example researcher (chapter 4.5) will see. Most end-users will not have access to the bX API. And even though I have not searched for examples of services built on top of the bX API, it is my impression that many libraries are content with having the bX Article Recommender as part of Primo. That is, as an integrated part of the library catalogue. A natural possibility in further research on bX would be to seek out institutions that have worked with the API and get their impressions and thoughts.

This does not mean it is of no interest to study the bX API-documentation. In fact, the API documentation is one of the few places where some parameters that bX uses are explicitly stated and described. Ex Libris offers four different APIs for bX: bX Hot Articles, Docid Service, Lookup service and OpenURL service. The Hot Articles API is used to return a list of the most used articles for a given subject field during each month, while the other three are related to recommendations pertaining to a particular item (Ex Libris, n.d.-b).

The first thing to note is that calls to the bX Recommender API are only available for subscribing customers. If the call does not contain a valid encrypted institution ID, no results are returned. An API-call can simply be initiated in the address bar of a web browser, or built into software. The documentation defines the syntax of an API-call (see for example Ex Libris, n.d.-c). An example call might look like the following (this contains one OpenURL-parameter):

```
http://recommender.service.Ex
Librisgroup.com/service/recommender/openurl?res_dat=token
%3Dxyz&rft_id=info:doi/10.1045/march2001-vandesompel
```

This examples contains the base URL for a link resolver, and a dummy for the institutional code (token%3Dxyz). It also contains a DOI for an article. The result of this API call would be a list of recommendations pertaining to this article.

5.5.1 Matching and caching

The introductory remarks to the bX APIs says that when someone initiates an API-call containing OpenURL parameters, then bX applies different matching algorithms to find out if the resource exists in the system. It is crucial that the OpenURL clarify the genre of the resource (book or article). If information about genre is lacking the process of generating recommendations is not started at all (Ex Libris, n.d.-g). If the OpenURL is not properly constructed, then it cannot be properly resolved by bX.

If the OpenURL request *is* properly constructed, bX tries to generate recommendations. In this step bX is depending on a cache for its recommendations (Ex Libris, n.d.-g). This is done to improve the speed and performance of the system. In this context 'cache' means that recommendations are already stored 'together' with the articles they apply to[16]. That also means that the recommendations are not necessarily generated on the fly. When the researcher in the scenario in chapter 4.5 looked up the article "Redefining information science", the recommendations to insert into the result page had been calculated *a priori.*

bX thus contains pre-calculated recommendations for almost every resource: "The results are intended to be reused and are considered valid for a system-defined number of days. After this period of time, they are considered expired" (Ex Libris, n.d.-g, Processing section). Recommendations are only calculated in real time if no recommendations pertaining to the requested article are found in the cache.

The above quotation speaks of a 'system-defined number of days'. This 'time' parameter of how often recommendations are to be re-calculated will affect the generated output. It can be imagined that if this number is set too high, then bX will not capture the here-and-now. On the other hand, if it is set too low, it might be too prone for changes. This parameter alone will affect which recommendations are displayed to end-users. I might be mistaken in the following, but lacking more information, I think it might be a valid concern: In this defined period of time recommendations pertaining to a request are essentially 'locked'. The same recommendations will be shown, and the longer the time period, then more effect this will have. Say that it is set to 14 days. In this time period the relationship strength between article A and its pertaining recommendations will strengthen themselves, because users will only see and be able to chose from the same, limited set of recommendations. If the recommendations are to change in the next re-calculation, link resolvers must have logged co-retrieval events between article A and other resources not found in the list of recommendations. These co-retrievals must then have happened in other parts of the information environment. But it is also possible that bX for example readjusts relationship weights negatively when recommendations are offered to a user but *not* clicked on. As said this reflection might be mistaken, but if so, it can at least help illustrate that more information and transparency could help with avoiding such misconceptions. I have not been able to find at

16 The non-technical definition and use of the word 'cache' ironically describes some of the experience of algorithms as black boxes: 'a hiding place especially for concealing and preserving provisions or implements', or as a verb: 'to place or store (something) in a hidden or secure place for safety or concealment' (https://www.merriam-webster.com/dictionary/cache).

what interval bX re-calculate its cache. Independent of this parameter, I anyway think that bX in some way embodies a "rich getting richer" effect, which I will return to in chapter 6.

5.5.2 Ranking

The documentation states that "the bX Recommender provides a ranked list of related resources for an item (predominantly articles). The relationship is created by the co-usage of resources. The rank within one relationship is calculated according to a co-usage network algorithm" (Ex Libris, n.d.-b). This again confirms the use of network based calculations, and the fact that lists are internally ranked.

The existence of the score parameter can be seen in the example output of an API-call, here in an XML-format:

```
<bX:score xmlns:bX="http://bX.ExLibrisgroup.com/docs/schema/bX">
96</bX:score>
```

This score likely stems from the relationship network described earlier in this chapter. However, it cannot be known if this number corresponds to the cell value in the relationship matrix, or if it goes through some modifying process before it is assigned as a 'score'. It can, for example, be imagined that relationships weights are grouped, so that cells in the relationship matrix with a value between 50-60 gets a bX score of 96 and so on. Anyway, the score is a result of the internal calculations of bX – and can only be observed via its output, and only via API access. Ranking scores are not provided for recommendations displayed in Primo. The documentation further defines some other parameters that are of interest, which I will now look at.

5.5.3 API-parameters

There are three defined input parameters for the bX Recommender APIs that are of special interest: source, maxRecords and threshold. The values of these can be set by the user initiating an API-call, and are shown in the table on the following page (Table 5).

Parameter	Description	Value	Default
source	Source of the co-retrieval values used to generate recommendations. Valid values are global or local, where local indicates that co-retrieval values are only from the user's institution.	string	Global
maxRecords	The maximum number of recommendations to return	number	15
threshold	The minimum score a recommendation must have in order to be included in the results. Scores may range from 0 to 100.	number	50

Table 5: Excerpt of bx-API input parameters (modified from Ex Libris, n.d.-c).

The source parameter shows that an institution can adjust the API-call so that it only retrieve lists of recommendations based on local usage data. This might imply that it exists two different relationship matrices, one global and one local, for every subscribing institution. Or that bX can manipulate its global matrix to provide recommendations based on local usage when needed. It is not clear how this works in relation to the bX Article Recommender in Primo – does it show recommendations based on global or local usage? In the bX registration guide it is said that an institution can chose to either just subscribe and get access to global recommendations, or they can subscribe *and* provide local usage data. The latter is said to create a richer recommendation database and make local usage count. If an institution chose to only subscribe to recommendations from bX, they are not able to retrieve recommendations based on local usage data (Ex Libris, n.d.-f). It is not clear if this refers to API usage alone. If not, then it might be possible for libraries to adjust bX so that recommendations shown in Primo are based on local usage data only. In a configuration guide for SFX – a link resolver offered by Ex Libris – there seems to be a possibility for subscribers to set a minimum rank score for a recommendations and to set the recommendation source. There also is a choice if bX only is to show recommendations that are available in full-text (Ex Libris, 2013, pp. 31–33). It remains unclear if this opportunity exists for the members of the BIBSYS-consortium.

In relation to this I searched for an article (doi: 10.1108/00220410610673846) directly from OsloMet's instance of Oria, and then via Google Scholar. I was presented with different recommendations in the two cases (Figure 10). The list of recommendations to the left is from the search within Oria, the right is from being redirected from Google Scholar to OsloMet's Oria:

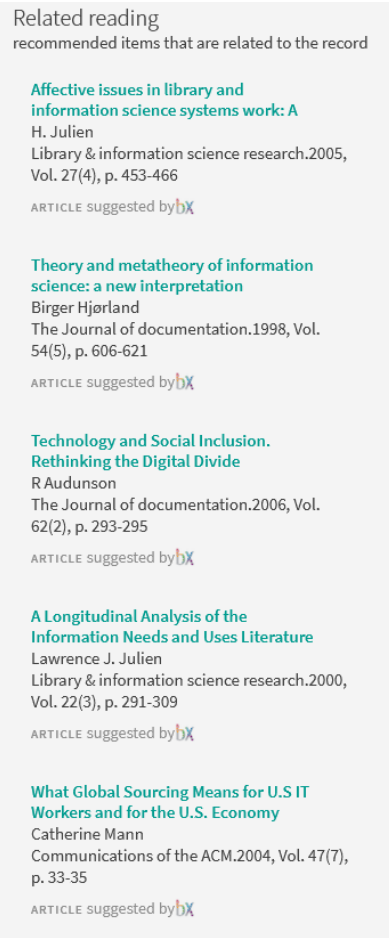


Figure 10: Comparison of lists of bX recommendations, November 2021 (screenshot).

I do not know if the leftmost list is purely based on local usage data (stemming from OsloMet, or all the libraries in the BIBSYS-consortium) or if there is some mix of global and local usage data, where the local is prioritized in some way. In the same way I do not know for certain if the list to the right (Google Scholar) is based purely on global usage data. It is also possible that the difference in the two lists comes from some error while resolving the OpenURL. If nothing else, this can yet again illustrate that lack of clear information can lead to confusion or unfounded speculation.

The next parameter in table 5 (page 55), maxRecords, shows that each resource potentially can generate many recommendations (a maximum number is not stated). The threshold-parameter also limits what and how many recommendations are shown, in that it defines a minimum score. It also shows that scores are calculated on a scale of 0-100. Even with this knowledge many questions can be asked that are not answered by the documentation. What is the difference between a score of 50 and 60? What is the consequence of changing the threshold a little bit? How does such changes differ for popular articles that have many recorded co-occurrences, and those with only a few recorded instances? Does the bX Article Recommender itself use a default threshold of 50?

To summarise, the API documentation has helped with identifying some parameters in bX that affect the output:

- bX keeps pre-calculated recommendations for almost every resource in a cache. These are valid for a system-defined number of days. It is only if no recommendations are found in the cache, that recommendations are calculated in real time.
- There is a score parameter of a number between 0-100. That is, given article A as input, what is the score of the other articles in relation to article A?
- There is a minimum threshold score for a document to be included in the recommendations.

The above is true for the *API,* and the API is not necessarily the same as the product of bX Article Recommender. Based on the publicly available documentation for configuring bX (Ex Libris, n.d.-d, n.d.-f), I cannot find that the subscribing libraries have the possibility to look at or adjust any of the parameters that affect the outcome of bX for end-users – other than (perhaps) deciding if the recommendations should be based on local or global usage data.

5.5.4 Summary

In this chapter I have gained some insight into how bX utilises the harvested usage data, and also of some of the parameters existing inside it. Still, it leaves the impression that merely knowing that certain parameters exists is of limited value. This is particular true if their value is related to other parameters. If it is uncertain what parameters *mean* in relation to each other, and what their values expresses in the context of bX, then how much does one really know? In the next chapter I try to interpret the epistemological views contained within the material I have looked at so far.

6 The epistemology of bX

This chapter concerns research question number three:

> What epistemological ideas and interpretations of the world operate in the bX Article Recommender?

With 'epistemological ideas and interpretations of the world' I mean to ask what knowledge claims are made and on what basis, in the material I have studied so far. This chapter and this research question is about the dynamic movement of software capturing the world, and the world seeping into software (Rieder, 2012, 1. Introduction section). Or, how epistemological statements about the world implicitly exists in software.

I am studying the process outlined in the previous two chapters, from harvesting usage data to generating recommendations, in the light of research question 3. I will make 'stops' at different places in the process where, in my view, some epistemological statements can (implicitly) be identified. These are related to the concept of co-occurrence, the time threshold, the concept of networks, and to ranking and parameters. I have chosen examples were I think it is clear such statements are made. In these examples the role of human decision making in creation of software and algorithms threads forth, and the link between human decisions and algorithmic output can be established. My first stop is the idea of co-occurrence. In my eyes this is a fundamental epistemological idea that the bX architecture builds upon.

6.1 The idea of co-occurrence

The previous chapter introduced the concept of retrieval coherence assumption (RCA): "When the same user retrieves two documents during the same session, this serves as an indication that both documents may be related to the same information need" (Bollen et al., 2003, Introduction section). Here a user's actions are analysed from the overarching concept of a 'session'; a limited interval of time. This limitation enables the methodology to inject user actions within a session with meaning. The RCA implicitly supposes that the limited time interval makes a user's action less random; they are more likely to stem from the same motivation ('information need'). And if the same patterns can be analysed with different users at different times, it strengthens the impression:

> Their [pair of articles] degree of relatedness can be determined on the basis of two factors. First, the closer the documents are located within a sequence of reader downloads, the more related they are expected to be (Jones, Cunningham, & McNab, 1998; Pirolli & Pitkow, 1999; Chi, Pirolli, & Pitkow, 2000). The RCA thus applies most reliably to the shortest retrieval sequences, i.e. pairs of documents downloaded one after the other. Second, the more frequently a particular pair of documents are downloaded by a group of readers, the greater the degree to which we can assume these documents to be related. (Bollen et al., 2005, p. 1427)

In fact the RCA can be said to take on the role of an axiom, understood as “a proposition laid down as one from which we may begin; an assertion that is taken as fundamental” (Blackburn, 2008, p. 32). The RCA prepares the mechanical reasoning, the reduction of a phenomena to bits and bytes, which Rieder (2017, p. 103) labels as a ‘datafied reality’. In this case it is used to break down the complex concept of ‘relatedness’ and make it a subject for computation. It could be interpreted as an epistemological statement that says:

> **Epistemological statement 1 (ES1)**: User actions that happens within a certain time frame are driven by the same underlying information need. Thus when a user requests different journal articles these articles are related because they confirm to the same information need. By studying the sequential order of user’s requests something can be known about the relationship between two journal articles. If a particular sequence of requests are observed over time this further strengthens the claim that the requested articles are related[17].

Note that nowhere in this statement, or in the bX architecture as far I know, is metadata, contents or semantics relating to the articles considered. But in my eyes EP1 is also a statement about semantics, about meaning. It does not matter that the recommendations are based on analysis of

17 I here adopt Bollen and co’s simple use of ’information need’. They avoid complications by not defining nor discussing the concept any further. Of course, this is a complex concept with a long history in LIS.

usage data alone, the assumed relationship is at some level, indirectly, based on the contents of the journal articles, on semantics. It is the users that are the final judges of the validity of the recommendations. The first step in their judgement of potential usefulness is largely based on metadata about author, title, publication, page numbers, year, as can be seen from the recommendations at page 56 (Figure 10). Except for the statement that these are 'recommended items related to the record', this is the only information a user has to go on. Will a user click on a recommendation only because it is there, if the sum of this metadata does not seem relevant? Would the judgement change if the user had knowledge about how bX generate recommendations? These would be interesting questions to pose in a user oriented study.

If a user follows through on a recommendation and find it not to be semantically coherent with the work it pertains to, he will probably be a little frustrated. In my view bX's goal of providing recommendations of related/relevant works, implies that the content of the recommended works should be semantically/topically related. And that ES1 is a way to mechanize and establish (semantic) relatedness between two journal articles, even though contents or metadata is not considered at all. But the mechanisation of ES1, the way that bX works, does not differ between clicks on recommendations. To the system a click is a click, no matter the underlying motivation. The fact that when a user first has clicked on a recommendation, then the assumed relatedness has already been established and further strengthened, is a potential weakness of the system. The user has vouched for the relatedness, even though he ended up finding the recommended work to be of no use.

Another potential weakness that is latent in ES1 is the following: It is possible that someone publish an article (article B) and want to 'attach' it to another popular work (article A). Then this someone could probably, via different methods, work to affirm and strengthen the relationship weight between Article A and Article B. In the end the researcher's article might show up among the recommendations pertaining to article A, and potentially bring more readers to his own work. Though bX probably has some mechanisms in place to monitor for suspect behaviour.

In light of the above discussion it is worth to have in mind that ES1 and its assumptions is *not*, and does not try to be, a philosophical investigation of the concept of ’relatedness’. Rather, it is a product of the highly pragmatic world of software making. This observations of the pragmatic side of software making is laid out by Rieder in his analysis of the Bayes’ classifier (Rieder, 2017, p. 110). In the case of bX, the main concern is to create a functioning recommender system that produces good results. But even though Bollen and Van de Sompel’s goal is not a philosophical one, that does not mean statements such as the RCA cannot be read, interpreted and scrutinized from a philosophical/epistemological viewpoint. It is merely an indicator that one should be wary to see them (primarily) intended as such. After all, reduction of meaning is something unavoidable when writing software (see the theory chapter). But, assumptions such as the RCA still has consequences, and these can and should be objects of critique.

So, even though bX is built on the assumptions in ES1, it has to make some ‘epistemological’ concessions when faced with practicalities of software making. Say that a user, within a session, generates the sequence of Article A → Article B → Article C by downloading them in this order. According to the ES1, article A and C should also be related, and the relationship weight for Article A-C readjusted. But, in bX only sequential pairs seems to be considered (Bollen et al., 2005, p. 1428; Bollen & Van de Sompel, 2006, p. 303). Why? I cannot say for certain, but if multinomial download sequences were to be considered, it probably would require heavier calculations and lead to a more resource demanding system. The limitation to only analyse pair-wise retrievals can thus be a quite pragmatic simplification, even though one can argue it is inconsequential in light of ES1.

Another good example of choices one has to make when creating software – choices with consequences for output – is the time threshold; the parameter that decides what constitutes a co-occurrence. What should this be set to? From the point of view of system development this is also a practical question. It relates to the bX Article Recommender as a piece of ‘algorithmic literature’, as a system designed with a concrete purpose and in a concrete context, balancing between technological demands and making the ‘best possible’ recommendations.

I do not know what the time threshold is set to in bX, but in the methodology developed by Bollen et al. (2005) it is set to one hour. Between themselves the authors must have discussed this number, though no such reflections have made their way into the article. This threshold value seems like a little thing, and after all it has to be set to something. But here the concept of 'mechanical reasoning' is in play. Here human actors must set the value of a parameter, a value which will have large consequences for the output generated by the system. If the time threshold was set to 90 minutes more instances of co-occurrence would probably be captured. If lowered to 30 less would be captured, but would it indicate a stronger relatedness? These are questions the authors must have asked themselves. In the case of the bX Article Recommender it is unfortunate that this parameters is not public. The existence of it is known, but the visibility of it is limited by not knowing its value. It lessens the understanding of bX, and the possibility for subscribers and potential subscribers to reflect upon it.

I want to dwell a little longer on the 'retrieval coherence assumption'/ES1 and 'information need'; that when a user downloads a set of documents he is driven by an information need. And, by extension, that the documents are related to the same information need. Furthermore, if a pair of documents are downloaded by different users, the stronger the indication that the documents are related. Still, say that twenty researchers all have downloaded article A and then article B. It is hard to imagine that one could formulate a description of an information need they all would agree to. If so the relationship between article A and article B captured in the relationship matrix does not reflect a certain information need. Behind articles with a strong relationship weight it is probably safer to say they represent the same, or closely related, research area, or one particular common research topic within a field. The single user has been driven by an information need, but bX recommendations cannot be traced back to any particular information need. This might seem like hair-splitting, but I think it shows that the inclusion of 'information need' in the retrieval coherence assumption is redundant. It is the repeated co-occurrence across users that is the indicator and argument for relatedness. In the mechanization of the RCA/ES1, the single user and his information need disappears. bX only counts, it does not interpret information needs. In a work on analysis of usage logs to help with digital library management, Bollen and his co-author find that the lone user is too random. It is precisely the culmination of usage data across time and different users that makes it a valid tool:

> User preferences and satisfaction tend to be highly transient and specific. User search focus can shift from one scientific domain to another between, or even within, retrieval sessions. Analysis of user preferences and user satisfaction therefore needs to focus on more stable characteristics of a given user community such as the community's perspective on general document impact and the relationships between documents in a collection. (Bollen & Luce, 2002, DL evaluation techniques section)

However, this might seem like a contradiction of the retrieval coherence assumption, where the same session is said to pertain to the same information need. Indeed Bollen and Luce formulates a variation of the assumption that does not refer to the concept of information need:

> Our central assumption is that when an individual user is searching for documents in a DL [digital library], this search focuses on a given subject matter. Therefore, when a user retrieves two documents within a short period of time, it adds support to the claim that some level of similarity exists between these documents. When many users repeatedly retrieve the same pair document within a short period of time, this is an even stronger indication that the two documents are related or similar. In other words, the frequency by which two documents are retrieved in temporal proximity over a population of DL users corresponds to the strength of the relationship between these documents. (Bollen & Luce, 2002, 'Hebb's law' section)

This in my eyes is a better definition of the retrieval coherence assumption, one that does not refer to information need but relates a search session to a "given subject matter". Even though the actions of a single user can be unfocused within the same session, the assumption is that mere size, that is, the number of occurrences, will correct this behaviour. In this there is something of the notion of the wisdom of the crowd, where a crowd of people are closer to the 'truth' than a single person. This reflection on the retrieval coherence assumption emphasise that bX is not about the single user event, it is about a community of users.

In other words, co-occurrence are not interesting on the level of the individual users; a user is merely a small actor in a larger network. That means that a co-retrieval event can be plotted into a matrix, and on this matrix of document relationships, “document and journal impact measures can be derived from the graph-theoretical properties of these networks and applied to any documents for which retrieval requests have been registered” (Bollen & Luce, 2002, ‘A user-centered approach’ section). The assumptions contained in epistemological statement 1 can be said to prepare the philosophical ground, so that bX can apply networked based calculations as an algorithmic technique. This is the theme of the next section.

6.2 Networks

In his study of *PageRank*, Rieder writes that “the choice of this specific case is further motivated by the immense influence the network model has gained in many scientific disciplines, which can be explained, at least partially, by the considerable means for calculation it provides” (Rieder, 2012, 1. Introduction section). Modelling something to a network is thus very useful for applying mechanical reasoning. Rieder shows how inherent ideas in PageRank developed from citation analysis and classical bibliometrics, but then, with inspiration from sociology and graph theory, expanded into the social: “*PageRank* is the mechanism by which the Web is no longer treated exclusively as a document repository, but additionally as a *social system*” (Rieder, 2012, ‘3.1 Flatlands’ section).

Bollen et al. make a similar observation when they write that “social network and citation analysis have, in the past decade, successfully converged on WWW search engines” (Bollen et al., 2005, p. 1424). In *PageRank* websites are not only ranked based on match between a query and content, but by the place of the website within a larger, social network. The authors further write that “our efforts to devise an alternative set of journal impact metrics are an attempt to bring the benefits of this approach to the domain of journal impact ranking” (Bollen et al., 2005, p. 1424). Here Bollen and co ‘calls’ on the use of network calculation as an algorithmic technique, as a tool. As shown in the previous section, this is possible because the data is harvested on a technical and epistemological fundament that has prepared the use of network theory. By anchoring their work in network theory, they at the same time implicitly adopts the inherent epistemological view and interpretations of the world that network theory manifests. And, as Rieder pointed out, it makes calculations on a big scale possible.

The concept of networks is a huge part of the bX Article Recommender. Indeed, the social feature is something Ex Libris themselves emphasise in their description of bX. And in the same way that web search is no longer just a question of a match between a query and a textual representation of a document, bX is described by Ex Libris to go 'beyond' the original query: "While the initial article serves as an entry point, the recommended material can provide new inspiration for learning and broaden the scope of research, going beyond the initial search query" (Ex Libris, n.d.-e).

Network theory allows for the exploration of the wider characteristics of a network, where "network architecture is a property not of parts but of the whole" (Buchanan, 2002, p. 185). Bollen and Van de Sompel ground the argument for the validity of their method on network theory. One of the research questions asked in Bollen et al. (2005, p. 1426) is as follows:

> *Question 1*. Can valid networks of journal relationships be derived from reader article download patterns registered in a DL's server logs?

To provide support for its validity, they use matrix calculations to describe features of the generated network. Among other to compare its characteristics to a randomly generated network (Bollen et al., 2005, p. 1426). The idea here is that if the network characteristics resembles other known networks of similar type, then their methodology is probably valid, i.e., their journal relationship network represents the reality in a good way.

Among other they describe the generated relationship matrix as sparse and as having a low matrix density. Of the 3 577 772 possible journal relationships in their data, only 0.176 % of them had a value larger than 0, "indeed, only a small fraction of all possible, directed journal relationships can be meaningful and therefore matrix densities will be low" (Bollen et al., 2005, p. 1429). That a matrix is sparse means that most of the cell values are zero, "by contrast, if the number of nonzero elements in a matrix is relatively large, then it is commonly referred as a dense matrix. The fraction of zero elements (non-zero elements) in a matrix is called the sparsity (density)" (Yan et al., 2017,

p. 1881). Furthermore for every cell with a value larger than 0, Bollen et al. (2005) found the mean link weight to be 1.195, i.e., the mean relationship weight in their matrix was 1.195. The minimum and maximum values (over 0) was 1 and 22 (p. 1429).

All these different numbers are taken as descriptions of the generated network's 'small-world' features. No formal proof can be established for the validity of the reader generated network, but small-world features is seen as a criteria that might support it (Bollen et al., 2005, p. 1430). They report that co-retrieval frequencies was skewed. The largest weight-value was 22, while 5250 had a weight of 1, that is, 5250 co-retrievals happened only once. Accordingly they find the network to have a scale-free topology. They also report other mathematical findings as evidence that their network is a small-world (p. 1431). What, then, is a small-world network with a scale-free topology, and why does it help with justifying their approach? Buchanan writes the following of small-world networks:

> What distinguishes a small-world network is not only that it has a low number of degrees of separation but also that it remains highly clustered. We might say that the fabric of the network is densely weaved, so that any element remains comfortably and tightly enmeshed within a local web of connections. Consequently, the network overall can be viewed as a collection of clusters, within which the elements are intimately linked, as in a group of friends. A few "weak" links between clusters serve to keep the whole world small. (Buchanan, 2002, p. 199)

In terms of bX the cluster feature of small-world networks is of most interest. Though the network as whole might be 'connected' through the notion of weak links, it does not really affect the output produced by bX. I would argue that recommendations from bX are centred around clusters and strong relationships, and that the so called 'long tail' does not matter much.

Scale-free networks also has the feature that there is no expected number of links between members of the network:

> This power-law distribution is special in that there is no "typical" number of links. In other words, the network has no inherent bias to produce elements with an expected number of

> links; rather this number varies widely over a huge range. That is to say, there is no inherent "scale" for the number of links, and the network is scale-free. (Buchanan, 2002, p. 215)

This description aligns with the relationship network created by bX, in which none of the possible article relationships have an inherent value. The network could thus be described as scale-free. But I do not think that the feature of being scale-free means that bX of today inherent no bias. When bX today recalculate its cache of recommendations, lots of relationship values are already established. It is probably more likely that articles in a 'cluster' enhances their score, then some article in the long tail. This is why I think bX embodies a "rich getting richer" effect. Although Vellino (2015) argued that bX is biased toward newer material, this is not to say that recently published articles quickly will show up. My interpretation of bX's nature is that it will take time before an article generates recommendations itself, or that it shows up *among* recommendations.

As previously mentioned, Bollen et al. (2005) see the small-world features as a mean for validation because their network then assembles other, similar networks such as citation graphs or WWW hyperlinks (p. 1437). Thus established social network metrics can be used on the usage data. This seems like using a method to validate using a method. But the point is that established matrix calculations are used on the matrix data, and the results shows similarities to known networks and patterns observed elsewhere. Networks exhibiting small-world features can be found many place, also in nature (see Buchanan, 2002 for many interesting examples).

The use of networks thus seems to draw on an epistemology of the social. An implicit epistemological statement can be formulated:

> **Epistemological statement 2 (EP2):** A relationship matrix is a way to represent real-world relations. Matrix calculations can be used to characterize specific relationships, or features of the network as a whole. In the case of bX, these calculations are knowledge statements about relationships between journal articles, and of the network as a whole.

The use of a relationship network allows bX to utilise a wide range of established techniques to interpret and (possibly) manipulate the network data further. Its network features also have a practical side from a computational viewpoint. To identify the network as sparse, for example, allows the application of certain computational algorithms that are not well suited used on other types of networks (see for example Yan et al., 2017). Though details of the bX algorithm at this level are non-existent in the material I have studied.

Since the data is structured as a relationship matrix, bX could implement more features into the bX Article Recommender that rely on matrix calculations. For example allowing users to explore network features such as centrality and betweenness, or offer end-users alternative entries for creating recommendations or exploring the network behind them. Since such features are not part of bX today, it could be said that it does not fully utilise the calculative power of the network when it comes to building services for end-users.

It must also be stated that the use of a relationship matrix is not dependent on data being generated on the basis of co-occurrence events. Other methods of data generation could have been used instead. The number of overlapping subject headings in the metadata, to give an example. As long as a methodology can express a relationship between two journal articles as a number, it can be used to create a relationship matrix. Although obvious, it means that the knowledge that bX uses a relationship matrix alone does not say much. The cell values in the matrix must be epistemological interpreted to say anything about their meaning, and thus it is necessary to know on what basis they are generated and what they express. This shows the importance of studying not only how data is represented, but also how they are generated.

6.3 Ranking

The next stop is the concept of ranking. I assume that bX recommendations are ranked by score. The API sets the minimum score threshold to 50 by default, but it may range from 0-100 (Ex Libris, n.d.-c). Since it is not clear if this threshold is used in the bX Article Recommender in Primo, it remains uncertain if an article with only a few, low-scoring relationships, would generate any recommendations.

The maximum number of recommendations shown in Primo seems to be 10. Therefore it exists a cut of point between recommendation number 10 and 11. Say that they have a score of 86 and 79 respectively. What is the difference? The latter is invisible to the user, but might it not be just as relevant? Can such small differences in the score be said to be meaningful? I think this maximum number of recommendations is yet another result of practical choices. It is (probably) not that the creators of bX understands the maximum possible number of related works to be 10, it is just that the number has to be set somewhere[18]. But the choice do matter.

If there is a defined score threshold, and if the algorithm only uses recorded co-retrievals to calculate a score, it seems to be a danger that several meaningful relationships will be hidden for a long time. It will then be tougher for journal article B, that is semantically close to article A, to show up in the recommendations pertaining to article A, if the weight is low. What is needed is that the co-occurrence article A→ article B[19] happens somewhere else in the information environment, and enough times so that it eventually is recommended by bX. It will not get any help from people re-enforcing the relationship by clicking on it in a list of recommendations, because it will not be there in the first place.

One possible way to sidestep this could be that users had the possibility, if interested, to access a full list of every recorded relation for an article to look through on his own. This could be a simple 'Show all'-link in the list of recommendations, or perhaps to be exported as a spreadsheet. In some way this is what the API does, but far from everyone has access to that. A choice has been made to limit the list of recommendations, and it is of course unavoidable that such lists will miss relevant articles.

6.4 Parameters

Rieder (2012) describes how parameters sink into a system and then become invisible. He writes of the Google search engine that "neither the ranking principles, nor their parameters are amendable to user intervention" (Rieder, 2012, 3. Two moments of commitment section). In the same way the current implementation of bX allows no *direct* user feedback. When bX first was released, it

18 And maybe it is connected to the knowledge that search engine users seldom look at more than the top results.
19 That is, article A is the *ReferringEntity* and article B the *Referent*. See Table 3, page 34, on the ContextObject.

included a way for the users to give feedback to a recommendation with thumbs up and down buttons, as seen in the following screenshot (Figure 11).

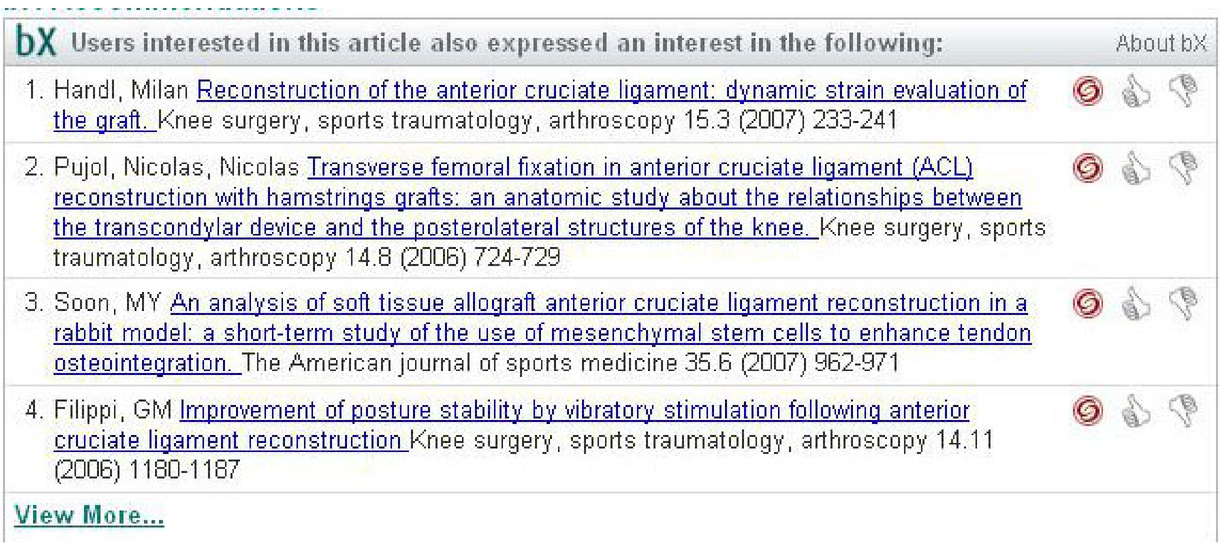


Figure 11: Screenshot of older version of bX. Taken from (Uhmeres, 2010).

Why is there no such function in bX of today? Did no one use them, did their presence imply that the algorithm was not good enough and that user correction was needed? Is their removal a sign that Ex Libris are more confident in the results bX produces? Or, does it imply that users give an implicit relevance judgment when they chose *not* to click on a recommendation? And, if so, that the calculated weight is readjusted negatively? There might also be some mechanisms in bX that distinguish if the user clicked on a recommendation and just viewed the metadata, or if he also ended up requesting full-text access, and that the weight to be added is readjusted accordingly. And if there is no way for bX to adjust a relationship weight negatively, then how easily can users really affect the recommendations?

The above are just speculations, and when it comes to the *utilisation* and *values* of parameters in the bX Article Recommender this is where it seems to end up. Jöran Beel, in his doctoral thesis on research paper recommender systems, writes of bX:

> Sadly, there is only little detail on the exact algorithms, which apparently are patent-pending. In the paper from which bX originated, only superficial information can be found,

> but it seems that bX is applying a simple count of co-occurrences to provide recommendations. (Beel, 2015, pp. 238–239)

The paper he refers to is the prototype by Bollen and Van de Sompel (2006). When it comes to details of the actual, working algorithm, the findings of this thesis seems to line up well with Beel's observation. But, it is my hope that this chapter has demonstrated that it still is possible to say something about epistemological commitments contained in bX – and that this endeavour has showed how the output generated by bX, like output by all recommender systems, reflects lots of human decisions. As I wrote in the theory chapter: Before a recommender system can do its thing, lots of interpretive, epistemological questions have already been asked and answered, whether consciously or not. I hope this chapter has helped illustrate how that is. Even though some of the parameters of bX are known, their ability to provide a peek into the inner workings of bX, must be said to be limited. They are perhaps not quite sunk into the system or invisible, but are lingering on the surface, teasing with their presence, while concealing their inner nature.

In the next chapter I will try to utilize some of my findings by looking at the concept of 'serendipity' in relation to bX. This is to show how my thesis potentially can inform further studies of different aspects of bX.

7 Example – bX and serendipity

'Enrich your discovery experience' – this is the slogan for the bX Article Recommender. In this chapter I will look at the way Ex Libris describes bX, especially the claim to serendipitous discovery. On the product page, the first, and most prominent, passage of text is the following:

> bX captures anonymous usage information from millions of scholars around the world, then leverages this data to enrich and expand the user discovery experience with relevant recommendations for articles and ebooks. Starting from an article of interest, bX provides users with other relevant articles for the same topic. While the initial article serves as an entry point, the recommended material can provide new inspiration for learning and broaden the scope of research, going beyond the initial search query. Depending on their nature, recommendations can help narrow or widen learning topics, provide new keywords to describe the topic, and allow the user to find items by chance through serendipitous discovery. (Ex Libris, n.d.-e)

The recommendations are said to provide inspiration for learning, present a broader research scope and expand upon the initial textual query. They widen and narrow 'learning topics', provide the searcher with ideas for new keywords – but only manually, the user needs to look at the recommended article and its keywords himself – and, lastly, the recommendations can allow for serendipitous discovery. In the following I will look closer at the claim that bX creates possibilities for serendipitous discoveries.

First of all, it must be stated bX is *not* a tool for thorough literature searches, nor is it presented as such. It is fronted more as an add-on to an existing library discovery environment. It wants to be of value, but it is not a substitute for anything else. The goal is not recall, but rather some type of precision, to use the classic information retrieval terms. And it is perhaps more of use in early phases of a research process.

In a blog post about bX and serendipity these features of bX are emphasised. Written by an unknown Ex Libris employee, bX is said to help with finding articles with other keywords, from other authors etc., bX is even seen as an instance of knowledge sharing, "that adds not only a mechanism to share knowledge, but exposes connections that I would not have found myself so easily – really lucky findings" (Ex Libris, 2012). This tie between serendipity and luck is also seen in the following dictionary definition: "Luck that takes the form of finding valuable or pleasant things that are not looked for; the faculty or phenomenon of finding valuable or agreeable things not sought for"[20]. A document by Ex Libris on core principles for designing library discovery services states: "Serendipitous discovery means expanding the result set from obvious information to related yet important information that the user may not know of. … Serendipitous discovery partners with exploration" (Botzer, 2015, p. 7). Here the use of recommendations are seen as facilitating a form of exploration, and as I have touched upon earlier, usage generated recommendations are seen to reflect the wisdom of the crowd: "In this way, a user benefits from the selections made by other users and the wisdom of the crowd" (Botzer, 2015, p. 7). In another blog post bX is viewed as a tool for less experienced users:

> Because this is not simply a metadata match, these recommendations can be especially helpful for patrons who may not know all the right keywords to try when searching for a concept, something librarians are great at helping with that can be lost in self-guided search. (Ransom, 2016)

This feature is also emphasised in a report on a user study conducted by Ex Libris. It describes how bX recommendations can take the user on a trail of discovery. It is said to be time saving, and to enrich the user's understanding of his research topic (Stohn, 2015, p. 10). In this case the mentioned user is a student. Generally, bX as tool for learning seems first and foremost to be tied to the student user group.

The Ex Libris document on design (Botzer, 2015) reads that serendipitous discovery is to expand upon an 'obvious' initial result to include related, important information that might be unfamiliar to the user. If one hold this to be true, can bX recommendations then by nature be seen as facilitating serendipitous discovery? One could argue against such a position, and state that serendipity is not so easily instrumentalised. An article on serendipity in the context of information science concludes with the following:

20 Definition from https://www.merriam-webster.com/dictionary/serendipity

> As argued in the paper, we cannot 'engineer' nor 'design' serendipity per se. We cannot design environments that always lead to serendipity – as serendipity is a highly subjective and situational phenomenon. Serendipity for one person is thus not necessarily serendipity for another person, and serendipity at one point in time is not necessarily serendipity at another time for the same person. However, even if we cannot 'design serendipity', we can design for serendipity. That is, we can design affordances for serendipity – seen from the designers' point of view. From the users' point of view, serendipity must always be encountered in unplanned ways in order to be serendipitous. (Björneborn, 2017, p. 1070)

Such a stance would say that bX recommendations cannot *always*, if at all, be serendipitous. It is not up to the system to decide or the developers to decide that. It can only happen in the meeting between the user and the recommendations. I agree with this view that serendipity is a highly subjective experience. As a consequence it is also a hard concept to subject to mechanical reasoning; it is hard to break it down into bites and bytes, although different methods that tries to design for serendipity do exist. Kotkov et al. (2016) is an interesting work that surveys different methods in which a recommender system can design affordances for serendipity. To summarise so far: I don't see the bX Article Recommender as providing possibilities for serendipitous discovery in any exceptional way, or at an exceptional rate. Indeed, it is hard to see that the bX Article Recommender does anything particular to enhance the possibility for serendipitous discoveries.

Because, if bX' method of generating a list of recommendations are solely based on what I have outlined in this thesis, it is hardly designed *for* serendipity. Rather, its small-world network features will create hubs that will grow stronger over time; the rich getting richer. As I have discussed several times: This design makes it hard for new recommendations to 'break through' and show up in a list of well established recommendations. But without access to more documentation I do not *know* this. I do not know if bX has implemented some way of enhancing the possibility for serendipitous discovery. It might be that recommendations displayed by bX are not always those with the highest relationship weight in relation to the source article. You could also argue that, if one takes serendipity to be a highly subjective notion, then it cannot really be denied that bX allows the users to find items by chance, through serendipitous discovery. But if so, does not every recommender system inherent this possibility? Does not any information retrieval system inherent this possibility? If one agrees to this, then Ex Libris' claim that bX allows user to find items via serendipitous discovery is not very substantial if this is not reflected in bX' design.

Vellino (2015), in his empirical study comparing recommendations by bX with generations from a recommender system based on citations, studied the semantic distance between seed articles and the recommended articles. This semantic distance was meant as a proxy for a user study focusing on serendipity, where recommendations from 'semantically' similar journals was taken as an implication of less serendipitous recommendations (Vellino, 2015, p. 600). He also makes the argument that bX does not differ between users levels of expertise when generating the usage data, and that it does not necessarily reflect smaller academic fields in a good way. I think the outline of networks in this thesis strengthens such critiques:

> For instance, while "bX" can take advantage of a voluminous amount of globally distributed usage data, this data may not reflect, even in the aggregate, the interests of specialists in any given field. Usage data from OpenURL logs is indiscriminate between expert researchers and undergraduate university students. In addition, a dependence on usage information makes such a recommender unable to address the recommendation needs of users interested in the end of long tail of sparsely researched areas. One consequence of this is that the publication dates of "bX" recommendations should typically be skewed towards the present. (Vellino, 2015, p. 602)

bX created recommendations for 30 % of the seed articles (9,453 seeds). For 12 % of the seeds both systems created recommendations, but none of the recommendations were the same. Both systems recommended articles semantically similar to the seed article. Vellino (2015) found that bX had an inherent bias for recommending more recent articles, where the average publication data was -0.6 years from the seed article, that the recommended articles are written *after* the seed article was published (p. 605-606). Whether bX today makes any adjustments for this is an open question. An experiment similar to Vellino's would have to be run.

To conclude: When reading Ex Libris' presentation of bX, the reader cannot know whether bX is designed *for* serendipitous discovery; i.e. if the concept is tried operationalised in the algorithm and system itself. As long as this is unknown, then the claim of providing chances for serendipitous discovery is shared with every other system of information retrieval and discovery.

The above discussion of serendipity is an example of one way this thesis can function as groundwork to further studies of bX (and, to some extent, other recommender systems). In the next chapter I will outline other such possibilities for further research.

8 Further research

When it comes to further research on bX, a natural direction would be to do more empirical oriented work, such as Vellino (2015). This thesis can inform such a work, and help in deciding *what* to look for, and in interpreting the results. The bX API could be used to generate recommendations, and allow for a more systematic observations of them. The results could be screened for possible biases, patterns etc. For example, what is the effect of the network's small-world features? If you track the recommendations for a given article, will you see the same recommendations show up among those? Is there a limited number of articles that will repeat and reinforce themselves? Such efforts might be seen as try to 'reverse-engineer' bX– which developers of recommender systems typically do not permit. Such concerns would have to be taken into consideration if one wants to use the API.

Another interesting possibility would be to look further into the user-interface that surrounds bX, and see if it influence user behaviour. In the material I have examined there is an absence of discussion of how human computer-interaction might influence the harvested usage data (although Greenberg & Bar-Ilan (2017) touches upon it). For example, what role does the user interface play in determining what requests are clicked on? Is the first recommendation in the list clicked on more often than the last because it is more easily accessible? Even though, if asked in hindsight, the user might find the seventh article in the list to be more relevant? What would happen if you moved the list of recommendations around in the interface? Or if users could access a list that gave more detail and context to the recommendations? Such research would lean more towards users studies.

Of course, bX can also be studied in the light of many theoretical concepts. Examples might be relevance, trust and power. This could be a study of bX, or research paper recommender systems in general. For example, what is the role of trust in library and information science, and in the adaption of recommender systems in particular? (Marsh & Dibben, 2003; Wang & Benbasat, 2005). How does trust in research paper recommender systems compare to studies on information behaviour and trust among students and researchers? Or, if we agree with Saracevic's notion that "underlying all information systems is some interpretation of the notion of relevance" (Saracevic, 1975, p. 321), then one could try to write forth bX' notion of relevance – which would be tightly knit to the retrieval coherence assumption and what it says about *relatedness.* Such a work would have to analyse the connection between these two concepts.

It is also possible to do a more critical, involved and activistic study of bX. There is already a field of a more critical, practice-oriented, research on libraries and library systems (see for example Barron & Preater, 2017; Nicholson & Seale, 2017; Reidsma, 2019). This body of research includes studies of other systems by Ex Libris. I have in my thesis raised some questions and concerns which can be elaborated further, and linked to such more critically involved studies.

A possible objection to such studies could be that bX after all is a pretty harmless tool, that yield some helpful results from time to time. If it actually helps users, and is not of particular annoyance to the rest, does it really matter if insight into its internal algorithms are sparse? Well, first of all it is a commercial product. When first launched, the price was set to 3000 USD per year for single sites, and for a consortia with 31-50 sites it was 1950 USD per site per year (Ex Libris, 2010). Subscribers should be interested in if it works properly. More importantly, I think, such studies raise some more fundamental questions of the lack of control for libraries over their tools. Should for example libraries use and pay for a system where few details are known about how the system works? By using bX it becomes part of the library services and, by extension, the library vouch for its quality. How does this align with the typical mantra of academic libraries providing access to 'quality assured material'? Should libraries accept low levels of transparency from its vendors? Is the fact that inner details of algorithms often are seen as trade secrets something one just have to accept? These are interesting questions which can be asked by research of this nature, and which this thesis might help inform.

Of course, it would also be natural to try to contact Ex Libris directly and seek to clarify some of the questions that have arisen in this thesis. That might also help correct eventual misconceptions and misunderstandings from my side. Still, I think this thesis has been a necessary groundwork to know what questions to ask and how to interpret the answers.

9 Conclusion

In this thesis I have at several points conceded that I cannot know for certain how the bX Article Recommender works, and that the finer details of its algorithms remain unknown. Even so, I think I have been able to present a general, technical structure of bX, that is not without merit. Technologies developed for other means – link resolvers, OpenURL, OAI-PMH – make bX possible. At some level I have been able to form arguments of how bX harvest, represents and exploits usage data. I have also shown that how this data is *used* contains epistemological statements and interpretations of the world from which they stem. Parameters have been identified, and I have demonstrated how their values might potentially affect the output. At the same time I have questioned the value of this knowledge. Knowing that certain parameters exist is not the same as knowing how they are used and operationalised in the working algorithm. As long as the parameters are unavailable to end-users, they remain hidden in the algorithms. That means that there is no way for the lone user to adjust the parameters according to his own needs or understanding; it also makes it harder for users to criticize their use or seek to improve it.

Even though one in the end is bound to stand outside the 'black box', the thesis has hopefully demonstrated that it can still be meaningful to study it. Rieder's theoretical framework alerted me to focus on the output. Even with a limited material, this helped me identify places where the human impact springs forth – places where decisions have to be made, balancing between pragmatic concerns and the 'best' possible solutions, all with consequences for what recommendations are generated or not. As such there is no objective algorithm or recommender system. The thesis set out to study bX from both a technical and epistemological perspective. I have been able to do that, and to show that such an approach can be useful in studying algorithms and recommender systems.

I want to end with a short remark on 'simple' systems. When systems such as bX or search engines are characterized as 'simple', it is mainly the user interface that can be understood as such. Behind the simple list of recommendations or search box are *complex* systems. I think this thesis shows how the simplicity in some degree comes from the fact that meaningful choices are removed from the end-users. Complex, epistemological related, decisions of what parameters to use and what they should be set to are made by the developers themselves. The results are ‘simple’, easy to use and impressively fast systems. But, what is the cost?

Literature

API. (2021). MDN Web Docs. https://developer.mozilla.org/en-US/docs/Glossary/API

Aravind, S. R. (2018). *A task-based scientific paper recommender system for literature review and manuscript preparation* [Nanyang Technological University]. https://doi.org/10.32657/10220/46243

Barner, K., & Tal, S. (2012). How do users search and discover? *Library Philosophy and Practice*. https://digitalcommons.unl.edu/libphilprac/754

Barron, S., & Preater, A. (2017). Critical system librarianship. In K. P. Nicholson & M. Seale (Eds.), *The politics of theory and the practice of critical librarianship* (pp. 87–110). Library Juice Press.

Beel, J. (2015). *Towards effective research-paper recommender systems and user modeling based on mind maps* [Otto-von-Guericke-Universität Magdeburg]. http://dx.doi.org/10.25673/4222

Beel, J., Gipp, B., Langer, S., & Breitinger, C. (2016). Research-paper recommender systems: A literature survey. *International Journal on Digital Libraries*, *17*(4), 305–338. https://doi.org/10.1007/s00799-015-0156-0

Beer, D. (2017). The social power of algorithms. *Information, Communication & Society*, *20*(1), 1–13. https://doi.org/10.1080/1369118X.2016.1216147

Beyene, W. M. (2012). *Personalization and usage data in academic libraries: An exploratory study* [Master thesis, Oslo and Akershus University College of Applied Sciences]. https://hdl.handle.net/10642/1265

Björneborn, L. (2017). Three key affordances for serendipity: Toward a framework connecting environmental and personal factors in serendipitous encounters. *Journal of Documentation*, *73*(5), 1053–1081. https://doi.org/10.1108/JD-07-2016-0097

Blackburn, S. (2008). *The Oxford dictionary of philosophy* (2nd ed., rev). Oxford University Press.

Bollen, J., & Luce, R. (2002). Evaluation of digital library impact and user communities by analysis of usage patterns. *D-Lib Magazine*, *8*(6). https://doi.org/10.1045/june2002-bollen

Bollen, J., Luce, R., Vemulapalli, S. S., & Xu, W. (2003). Usage analysis for the identification of research trends in digital libraries. *D-Lib Magazine*, *9*(5). https://doi.org/10.1045/may2003-bollen

Bollen, J., & Van de Sompel, H. (2006). An architecture for the aggregation and analysis of scholarly usage data. *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, 298–307. https://doi.org/10.1145/1141753.1141821

Bollen, J., Van de Sompel, H., Smith, J. A., & Luce, R. (2005). Toward alternative metrics of journal impact: A comparison of download and citation data. *Information Processing & Management*, *41*(6), 1419–1440. https://doi.org/10.1016/j.ipm.2005.03.024

Botzer, M. (2015). *Delivering the experience that users expect: Core principles for designing library discovery services*. Ex Libris. https://knowledge.exlibrisgroup.com/Primo/Product_Materials/Solution_Overviews_and_White_Papers/White_Papers_and_Studies

Buchanan, M. (2002). *Nexus: Small worlds and the groundbreaking science of networks* (1st ed). W.W. Norton.

Chen, X., Wu, C., & Gao, Y. (2012). An interaction model for literature recommendation based on cognitive principle. *2012 Eighth International Conference on Semantics, Knowledge and Grids*, 157–164. https://doi.org/10.1109/SKG.2012.19

Ex Libris. (n.d.-a). *Alma resolver augmentation*. Ex Libris Knowledge Center. Retrieved 15 June 2021, from https://knowledge.exlibrisgroup.com/Alma/Product_Documentation/010Alma_Online_Help_(English)/090Integrations_with_External_Systems/030Resource_Management/210Alma_Resolver_Augmentation

Ex Libris. (n.d.-b). b*X APIs*. Ex Libris Developer Network. Retrieved 14 May 2021, from https://developers.exlibrisgroup.com/bx/apis/

Ex Libris. (n.d.-c). b*X OpenURL service*. Ex Libris Developer Network. Retrieved 28 June 2021, from https://developers.exlibrisgroup.com/bx/apis/bx_recommender/bx_openurl_service/

Ex Libris. (n.d.-d). b*X Recommendations*. Ex Libris Knowledge Center. Retrieved 25 February 2021, from https://knowledge.exlibrisgroup.com/Primo/Product_Documentation/Primo/Interoperability_Guide/080Ex_Libris_Services/010bX_Recommendations

Ex Libris. (n.d.-e). b*X Recommender*. Ex Libris. Retrieved 14 August 2021, from https://exlibrisgroup.com/products/bx-recommender/

Ex Libris. (n.d.-f). b*X Registration Guide*. Ex Libris Knowledge Center. Retrieved 25 February 2021, from https://knowledge.exlibrisgroup.com/bX/Product_Documentation/bX_Registration_Guide

Ex Libris. (n.d.-g). *Introduction to bX Interoperability*. Ex Libris Developer Network. Retrieved 28 June 2021, from https://developers.exlibrisgroup.com/bx/apis/bx_recommender/introduction_to_bx_interoperability/

Ex Libris. (n.d.-h). *Publishing electronic holdings to Google Scholar*. Ex Libris Knowledge Center. Retrieved 3 May 2021, from https://knowledge.exlibrisgroup.com/Alma/Product_Documentation/010Alma_Online_Help_(English)/090Integrations_with_External_Systems/030Resource_Management/150Publishing_Electronic_Holdings_to_Google_Scholar

Ex Libris. (2010). b*X Overview*. http://registration.service.exlibrisgroup.com/customer/overview.do?product=bx

Ex Libris. (2012, July 5). *Serendipity, discovery and bX*. https://exlibrisgroup.com/blog/serendipity-discovery-and-bx/

Ex Libris. (2013). b*X-SFX configuration guide*. https://knowledge.exlibrisgroup.com/Primo/Product_Documentation/Primo/Interoperability_Guide/080Ex_Libris_Services/010bX_Recommendations

Friedman, A., Knijnenburg, B. P., Vanhecke, K., Martens, L., & Berkovsky, S. (2015). Privacy aspects of recommender systems. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender systems handbook* (2nd ed., pp. 649–688). Springer. https://doi.org/10.1007/978-1-4899-7637-6

Garfield, E. (2006). The history and meaning of the journal impact factor. *JAMA*, *295*(1), 90. https://doi.org/10.1001/jama.295.1.90

Greenberg, R., & Bar-Ilan, J. (2017). Library metrics – studying academic users' information retrieval behavior: A case study of an Israeli university library. *Journal of Librarianship and Information Science*, *49*(4), 454–467. https://doi.org/10.1177/0961000616640031

Haider, J., & Sundin, O. (2019). *Invisible search and online search engines: The ubiquity of search in everyday life*. Routledge.

Kotkov, D., Wang, S., & Veijalainen, J. (2016). A survey of serendipity in recommender systems. *Knowledge-Based Systems*, *111*, 180–192. https://doi.org/10.1016/j.knosys.2016.08.014

Lizaardo, O., & Jilbert, I. (2021). *Social networks: An introduction*. http://olizardo.bol.ucla.edu/classes/soc-111/textbook/_book/

Marsh, S., & Dibben, M. R. (2003). The role of trust in information science and technology. *Annual Review of Information Science and Technology*, *37*(1), 465–498. https://doi.org/10.1002/aris.1440370111

Mason, J. (2002). *Qualitative Researching* (Second edition). Sage.

Mason, J. (2018). *Qualitative researching* (3rd edition). SAGE Publications.

Munson, D. M. (2006). Link resolvers: An overview for reference librarians. *Internet Reference Services Quarterly*, *11*(1), 17–28. https://doi.org/10.1300/J136v11n01_02

Nicholson, K. P., & Seale, M. (Eds.). (2017). *The politics of theory and the practice of critical librarianship*. Library Juice Press.

NISO. (2010). *The OpenURL framework for context-sensitive services*. National Information Standards Organization.

Noble, S. U. (2018). *Algorithms of oppression: How search engines reinforce racism*. New York University Press.

Ponsford, B. C., Stephens, J., & Sewell, R. R. (2011). Improving OpenURL menus: User testing of revisions to SFX® menus. *Serials Review*, *37*(3), 162–170. https://doi.org/10.1016/j.serrev.2011.03.016

Ransom, J. (2016, September 22). Library exploration through serendipitous discovery. *Ex Libris*. https://exlibrisgroup.com/blog/library-exploration-through-serendipitous-discovery/

Reidsma, M. (2019). *Masked by trust*. Litwin Books.

Ricci, F., Rokach, L., & Shapira, B. (Eds.). (2015a). *Recommender systems handbook* (2nd ed.). Springer. https://doi.org/10.1007/978-1-4899-7637-6

Ricci, F., Rokach, L., & Shapira, B. (2015b). Recommender systems: Introduction and challenges. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender systems handbook* (2nd ed., pp. 1–34). Springer. https://doi.org/10.1007/978-1-4899-7637-6_1

Rieder, B. (2012). What is in PageRank? A historical and conceptual investigation of a recursive status index. *Computational Culture*, *2*. http://computationalculture.net/what_is_in_pagerank/

Rieder, B. (2017). Scrutinizing an algorithmic technique: The Bayes classifier as interested reading of reality. *Information, Communication & Society*, *20*(1), 100–117. https://doi.org/10.1080/1369118X.2016.1181195

Rieder, B. (2020). *Engines of order: A mechanology of algorithmic techniques*. Amsterdam University Press. https://doi.org/10.2307/j.ctv12sdvf1

Saracevic, T. (1975). Relevance: A review of and a framework for thinking on the notion in information science. *Journal of the American Society for Information Science*, *26*, 321–343.

Stohn, C. (2015). *How do users search and discover?* Ex Libris. https://knowledge.exlibrisgroup.com/Primo/Product_Materials/Solution_Overviews_and_White_Papers/White_Papers_and_Studies

Van de Sompel, H., & Beit-Arie, O. (2001). Open linking in the scholarly information environment using the OpenURL framework. *D-Lib Magazine*, *7*(3). https://doi.org/10.1045/march2001-vandesompel

Van de Sompel, H., & Hochstenbach, P. (1999a). Reference linking in a hybrid library environment: Part 1: frameworks for linking. *D-Lib Magazine*, *5*(4). https://doi.org/10.1045/april99-van_de_sompel-pt1

Van de Sompel, H., & Hochstenbach, P. (1999b). Reference linking in a hybrid library environment: Part 2: SFX, a generic linking solution. *D-Lib Magazine*, *5*(4). https://doi.org/10.1045/october99-van_de_sompel

Van de Sompel, H., & Hochstenbach, P. (1999c). Reference linking in a hybrid library environment: Part 3: generalizing the sfx solution in the 'SFX@Ghent & SFX@Lanl' experiment. *D-Lib Magazine*, *5*(10). https://doi.org/10.1045/october99-van_de_sompel

Van de Sompel, H., Young, J. A., & Hickey, T. B. (2003). Using the OAI-PMH ... Differently. *D-Lib Magazine*, *9*(7/8). https://doi.org/10.1045/july2003-young

Vellino, A. (2015). Recommending research articles using citation data. *Library Hi Tech*, *33*(4), 597–609. https://doi.org/10.1108/LHT-06-2015-0063

Wang, W., & Benbasat, I. (2005). Trust in and adoption of online recommendation agents. *Journal of the Association of Information Systems*, *6*(3), 72–101. https://doi.org/10.17705/1jais.00065

Yan, D., Wu, T., Liu, Y., & Gao, Y. (2017). An efficient sparse-dense matrix multiplication on a multicore system. *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, 1880–1883. https://doi.org/10.1109/ICCT.2017.8359956

List of figures and tables

Unless otherwise noted the figure or table is created by the author.

Figures

Tables