Interactive Recommender Systems For A Professional Social Network

— Master Thesis —

Scientific Computing Group
Department of Informatics
Faculty of Mathematics, Informatics and Natural Sciences
University of Hamburg

Author:	Mirko Köster
E-Mail-address:	0mkoeste@informatik.uni-hamburg.de
Mat. No.:	5616739
Course of Studies:	MSc Informatics
Supervisor:	Dr. Julian Kunkel
1st assessor:	Prof. Dr. Thomas Ludwig
2nd assessor:	Dr. Julian Kunkel

Hamburg, June 9, 2017

Contents

Preface

This thesis was written in cooperation with *XING* – a professional social network that has around 13 Million users[1] (see Section 2.2.1 for details). This involved a couple of unique advantages. For example, it allowed us to test our research on a real-world problem scenario – job recommendations – which is different from the movie domain that is typically studied by the recommender systems research community (cf. Chapter 2). It also gave us the opportunity to test our algorithms on a large-scale, multifaceted dataset. Moreover, it allowed us to follow an interdisciplinary approach and cooperate with people from different disciplines ranging from Data Scientists and Front-end Developers to Product Managers and Interaction Designers.

[1] see `https://xing.com/` for details

1. Introduction

1.1. Motivation and Goals

Today, humans are often overloaded with information available on the Web and in specific Web portals. Amazon[1] is selling more than hundred million products, music streaming platforms like Spotify[2] have more than 30 Million songs in their inventory and social networks like Facebook[3] provide access to more than a Billion of profiles.

Recommender systems support users in overcoming the information overload in such environments. For example, while browsing a product on Amazon, recommendations help to explore related products (e.g. *Customers who bought this item also bought...*), Spotify recommends songs, albums and playlists that may fit the user's musical taste and Facebook recommends profiles of people that the user may know.

Identifying those items that meet the current demands of a user is often a non-trivial challenge. Recommending items that are perceived as irrelevant by the user may moreover lead to dissatisfaction and to a loss of trust in the service. Recent research has revealed that user satisfaction can be increased by empowering the user to influence the recommender system. For example, by putting the user more into control over a movie recommender system, Harper et.al. [HXK+15] show that user satisfaction increased and that people were more satisfied with the interactive recommender system compared to the regular recommender system.

The goal of this thesis is to research interactive recommender systems in the domain of job recommendations. In particular, we focus on job recommendations on XING, a business network with more than 10 Million users in Germany, and aim to answer the following research questions:

Design: What kind of interactive recommender strategies are appropriate for the job recommendation domain? (see Chapter 3)

Implementation: How can the existing job recommendation system be transformed into an interactive recommender system? (see Chapter 4)

[1] see `https://amazon.com` for details
[2] see `https://spotify.com` for details
[3] see `https://facebook.com` for details

Evaluation: What is the quantitative impact of the interactive recommendation strategies on the system and what is the impact on user satisfaction? (see Chapter 5)

1.2. Contributions & sub tasks

In order to answer the above research questions, we will study and characterize interactive elements of recommender systems available in other platforms on the Web. We develop a conceptual framework to categorize interactive features and use this framework to design an interactive recommender system for job recommendations on XING.

We then implement selected strategies and integrate them into XING's existing job recommendation system. In particular, we add back-end functionality that allows users to specify and select their preferences for jobs, e.g. emphasize and de-amphasize interest into selected jobroles, skills, disciplines, willingness to commute / move, etc. Moreover, we extend the different components of XING's job recommender system so that these components respect and incorporate the selected preferences of the user.

Furthermore, we contribute to the interaction design and visual design of the front-end that allows end-users to interact with the job recommender system. Given this design, we also implement a prototype application that we use for conducting a user study in order to measure the qualitative impact of our interactive recommender system strategies on user satisfaction. This user study complements a quantitative evaluation that we conduct to measure the possible impact of the interactive elements at large scale.

The concrete tasks and contributions of this thesis can be summarized as follows:

- We define a conceptual framework for assessing interactive elements of recommender systems and apply this framework to compare related work and guide our design.
- We design interactive recommender strategies for the job recommendation design and cooperate with front-end and interaction designers to define a suitable user interface for interactive jobs recommendations on XING.
- We extend the actual job recommender system so that it provides the required back-end functionality (a) to allow users to specify their preferences and (b) to enable the recommender system algorithms to adapt to these preferences.
- We build a lightweight front-end prototype to allow stakeholders to experience the interactive recommender system.

- Based on our prototype, we design a user study and conduct a survey in order to collect feedback about the users' satisfaction with the interactive recommender system.
- We perform a large-scale quantitative evaluation in which we compare the interactive recommender system with XING's current job recommendation system.
- Given the data collected during the user study, we extensively evaluate the qualitative impact of the interaction recommender strategies on user satisfaction and compare it to the non-interactive recommender system.

1.3. Structure

In Chapter 2, we first give a high-level introduction into recommender systems before we dive into specifics of job recommendations and job recommendations on XING. In Section 2.3, we discuss related work and related systems. We analyze the interactive recommender strategies of these systems and categorizes them with respect to our conceptual framework along dimensions such as *user control* and *exploration*. We conclude the chapter in Section 2.4 with a status quo and limitations of XING's current job recommender system.

In Chapter 3, we overview our design of the interactive recommender system strategies. We present several strategies for turning XING's job recommendation system into an interactive recommender system and discuss these strategies using again our conceptual framework for categorizing interactive recommender strategies. We then describe how our selected strategy can be integrated into XING's job recommender system both regarding the required algorithms as well as the required front-end components. We conclude with the design of our user study that will allow us to evaluate the success of our designed strategies.

The implementation of our interactive recommender strategies into XING's recommender system as well as the implementation of our prototype that is used for conducting a user study and survey is presented in Chapter 4.

Chapter 5 details the evaluation of our interactive job recommender system. In Section 5.1, we report about the results of a large-scale quantitative analysis. The observations and results of our user study are reported and discussed in Section 5.2.

We conclude with a summary of our findings and an outlook about possible future works in Chapter 6.

2. Background

In this chaper, we first give a high-level introduction into recommender systems before we dive into specifics of job recommendations and job recommendations on XING. In Section 2.3, we discuss related work and related systems. We analyze the interactive recommender strategies of these systems and categorizes them with respect to our conceptual framework along dimensions such as *user control* and *exploration*. We conclude the chapter in Section 2.4 with a status quo and limitations of XING's current job recommender system.

2.1. Recommender Systems

Recommender systems are designed to suggest items of a given domain (e.g. books, movies, jobs, etc.) to users. They typically aid users who are not searching for something specific to overcome the "information overload problem" [RRSK10]. That is, that usually too many items are available and users are most likely overwhelmed by that quantity. This makes it hard for them to find relevant items. Most recommender systems are *personalized* and allow a given user to explore / browse through the item space. Examples of *non-personalized* recommender systems are Top 10 lists in magazines, which present the most favorite items over all users/customers. In this work, we focus on personalized recommender systems.

Basic approaches of recommender systems are *Content-based Filtering*, *Collaborative Filtering*, and *Hybrid Approaches* which combine these approaches [RRSK10].

Content-based Filtering exploits the attributes of users and items to generate personalized recommendations. They highly depend on well maintained features or metadata. If applied too strictly, this approach tends to overspecialize, e.g. in the domain of books, a user who specified he likes some specific genre might only get other books from this genre as recommendations.

Collaborative Filtering (CF) analyzes the relationships between users and items. For example, users may rate items and the corresponding rating data is used to construct a user-item matrix that captures how a given user rated a certain item. The rating data is then used to identify *similar* users or items. The recommendations are generated for a given user by retrieving items the most similar users liked (user-based CF), or items that are similar to items this user liked (item-based CF) [ERK11].

2.2. Job Recommendations

This section provides an overview over XING and its job recomender system. We describe the high-level architecture as well as the underlying algorithms.

2.2.1. XING

XING is a professional social network founded in Hamburg, Germany, in 2003 under the name *OpenBC* (Open Business Club) and was renamed to XING AG in 2006. As of March 2017 XING has more than 12,7 million users. Most of these users (> 10 million) live in the german-speaking region (D-A-CH: Germany, Austria, Switzerland) [XIN17b][XIN17a]. Users may create personalized profiles (including details about their current jobs, their work experiences, their skills and many more), connect with other users and find new jobs, groups with shared interests or events they might be interested in. Companies may create company profiles to be represented within the social network of XING. Additionally they may create job postings (on behalf of their employer / company).

At the moment, around 52% of the tracked traffic on job postings coming from logged-in users is generated by the job recommender systems and 48% by the search service. These job recommendations are shown at different places within the XING platform. For example, 2 job recommendations are prominently shown on the startpage (cf. Figure 2.1) The same site offers a sidebar on the right with mixed (e.g. members, events and jobs) recommendations (cf. Figure 2.2). 6 job recommenations are presented within the jobs section[1] (cf. Figure 2.3) and the ProJobs section[2] (cf. Figure 2.4), which can be expanded to show up to 20 recommendations. Furthermore, job recommendations are available on mobile devices inside the respective XING apps, which are available[3] for iOS, Android and Windows Phone.

2.2.2. XING and the job recommendation problem

XING tries to show the users' top recommendations of job postings. That is given a user, find the top k job postings she might be interested in and recommend those to her. A user in this context is described by her profile and her behaviour. As part of their profile, users may enter their CV and their current position, which skills they have and what they are interested in. Users' clicks on job postings are being collected while a user is using the platform. Additionally users can bookmark job postings and even leave apply intents. They can delete single recommendations and even leave explicit feedback by rating them on a scale from 1 to 5. These actions

[1] see `https://www.xing.com/jobs` for details
[2] see `https://www.xing.com/projobs/dashboard` for details
[3] see `https://mobile.xing.com/` for details

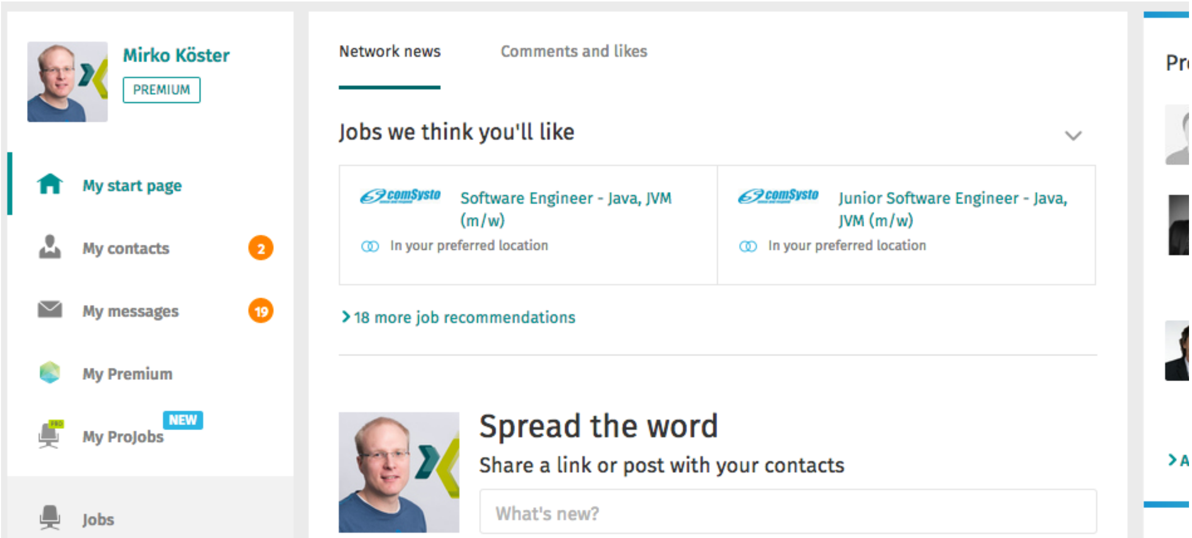


Figure 2.1.: Recommendations on the XING startpage

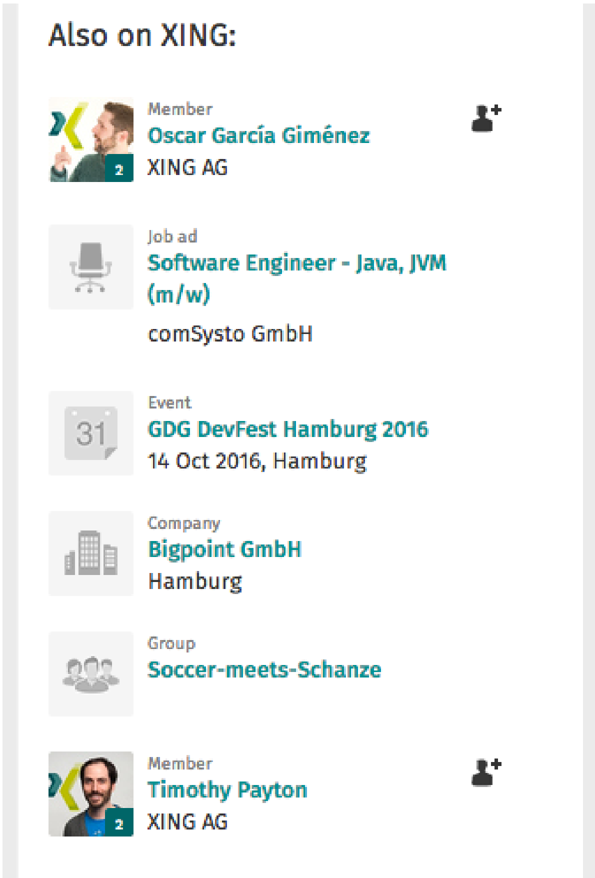


Figure 2.2.: Combined recommendations on the startpage (including one job recommendation)

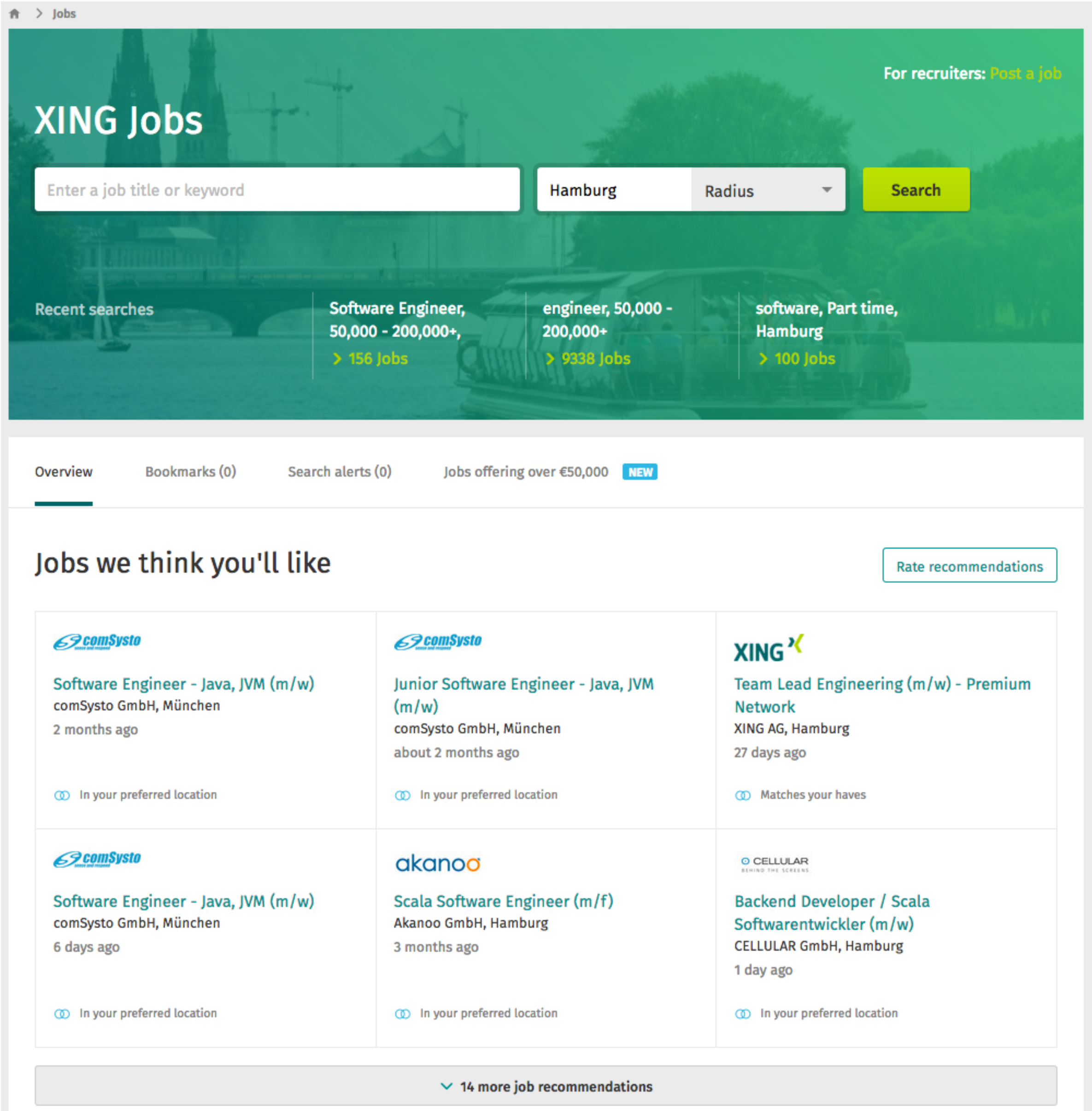


Figure 2.3.: Jobs landing page on XING

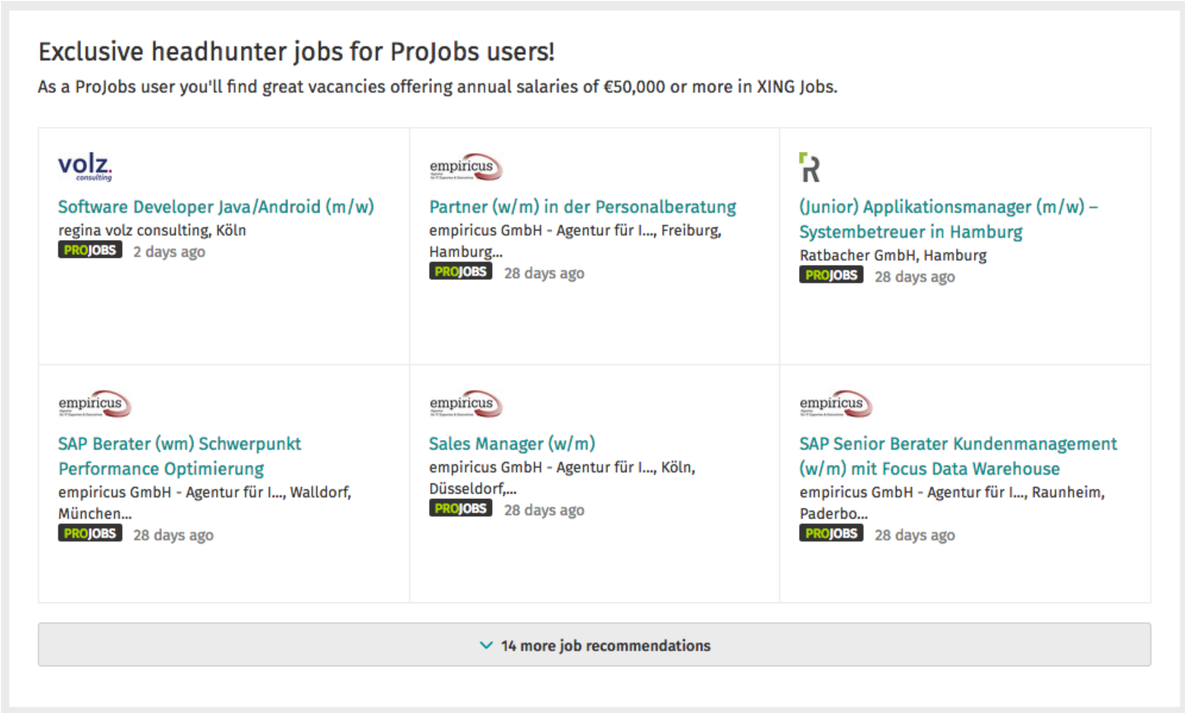


Figure 2.4.: ProJobs: exclusive job postings are being recommended

are described in more detail in Section 2.4.1 "Interactivity of XING's Recommender System".

2.2.3. Job Recommendation System

Architecture

To provide job recommendations to the users, XING runs several instances of the stateless job recommender service. By being stateless, subsequent requests from the same user do not have to be routed to the same instance every time, but can instead be served in a round-robin fashion. The service is written in Scala[4] using the Play Framework[5] and runs inside the Java Virtual Machine (JVM). Requests are handled in an asynchronous and non-blocking manner.

At the moment of this writing the job recommender service consists of 4 major backends, which are described in more detail in the next subsection. Upon reception of a user's request, the service calls those backends in parallel to get a list of recommendations from each of them. Each recommendation list is ranked and each item in those lists has a score assigned by the respective backend. The higher the score, the more likely is the user interested in that specific item (according to

[4]see `http://scala-lang.org/` for details
[5]see `https://playframework.com/` for details

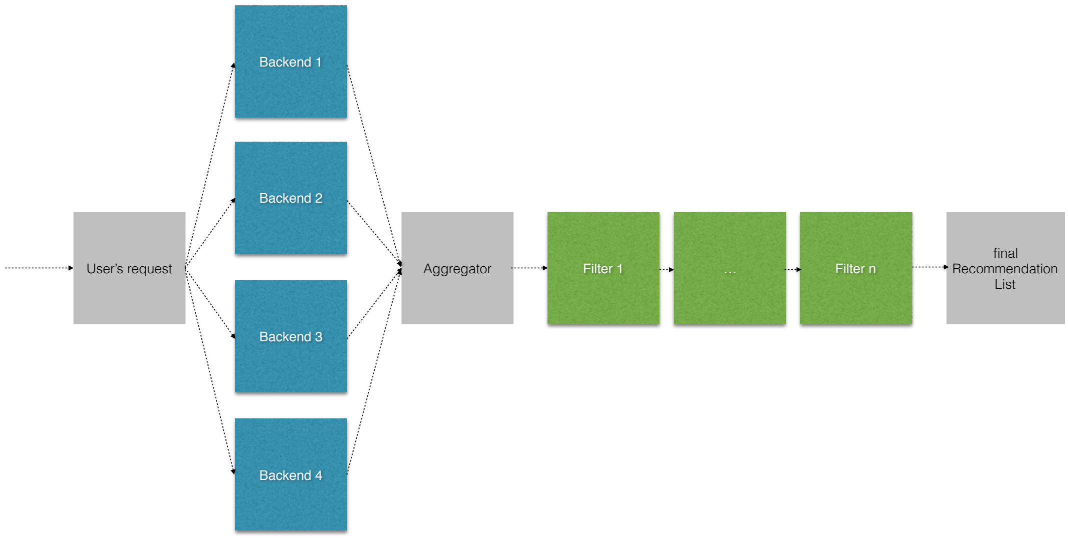


Figure 2.5.: High level architecture of the XING Job Recommender

the backend's strategy).

In the next step those recommendation lists are aggregated. Afterwards the aggregated list goes sequentially through several filters. A filter might remove items (e.g. deleted items) or change the score of items (e.g. to de-emphasize frequently shown items and to diversify). This might change the ordering of the list. Then the final list is returned to the user. Figure 2.5 shows the high level architecture of the XING Job Recommender.

2.2.4. Algorithms

Profile-based The profile-based sub-recommender uses profile information of a given user to create a search query, which is executed on elasticsearch on the job posting index. The query exploits the user's jobroles from her CV, her skills, location, industry, discipline, and salary information.

Interest Profile This backend generates an interest profile of a given user based on her click behavior on the platform. For instance common skills, jobroles and

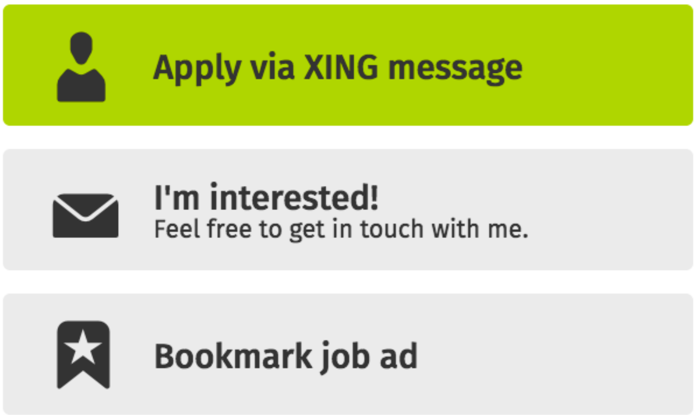


Figure 2.6.: Interactions on job postings which are considered positive

other information are extracted from job postings the user interacted with and are weighted by number of occurrences. This interest profile is then used to again create an elasticsearch query like the profile based sub-recommender.

More-Like-This component This sub-recommender is similar to the former in that it also uses the user's behavior on the platform to create an interest profile. But instead of taking all interactions into account, it uses only those which can be interpreted as strictly positive interactions, which can be seen in figure 2.6. These are *apply intention* (just clicking on 'Apply via XING message'), *apply* (by actually sending the application), *tell me more* (by clicking on 'I'm interested!', which sends an automated message to the job posting's contact person, asking for more information on the position), *bookmarking* a job posting (by clicking 'Bookmark job ad') and giving positive feedback to a job recommendation (see Section 2.4).

Pseudo Collaborative filtering Collaborative filtering (*CF*) approaches use the feedback (either explicit, e.g. in the form of ratings, or implicit, e.g. clicks) of the users and items. "The key idea is that the rating of a target user for a new item is likely to be similar to that of another user, if both users have rated other items in a similar way" [RRSK10].

User-based CF tries to find for a given user the n most similar users by comparing users based on their feedback, or, in other words, users who "have similar rating patterns". Each user is represented as a vector, where the length of these vectors is the number of items in the system. An entry in these vectors is the feedback on the corresponding item. Similarity between users could be defined by cosine similarity. These similar users' feedback is then used to predict a rating for a given item.

Item-based CF, on the other hand, looks at the n most similar items for a given item. Each item is represented as a vector, where the length of these vectors is the number of users in the system. An entry in these vectors is the feedback of the corresponding user. Again, similarity between items could be defined by cosine similarity. To predict the rating of a given user on that item, the ratings by the user on these most similar items are used.

In both cases, the vectors form the so called *user-item matrix*. This matrix is used to precompute recommendations for all users.

At XING, mostly implicit feedback in the form of clicks on job postings is available. Furthermore, the beforementioned matrix is too sparse for classical collaborative filtering. That is why users are clustered based on jobroles, skills and field of study. For each of these clusters the clicks are aggregated. The clusters are then treated as users in user-based collaborative filtering. At the time the recommendations for a given user are computed, the user's jobrole, skills and field of study are used for the lookup in the precomputed recommendations for these clusters.

A drawback of precomputing recommendations using collaborative filtering is that new items, i.e. job postings, cannot be recommended until the next iteration of the precomputation.

2.3. Interactive Recommender Systems

In this section we present work related to interactive recommender systems as well of current implementations from others (see Section 2.3.2 "State of the Art"). Afterwards, we discuss these different approaches.

2.3.1. Related Work

In the following, we present related work that is important to and served as inspiration for this thesis. Citations in this section are written in italic for better readability.

MovieLens

In September 2015 Ekstrand et.al. [EKHK15] published a paper about an interactive movie recommender system used at MovieLens[6], *a research site run by GroupLens Research at the University of Minnesota* [Mov17]. The paper is called "Letting Users Choose Recommender Algorithms: An Experimental Study". They prepared their recommender system to support and offer several different recommender algorithms. This new system allowed *users to choose the algorithm they want to provide their recommendations* and to *explore and/or switch among algorithms*. The research questions were (among others):

[6]see `https://movielens.org/` for details

- *Do users take advantage of a means to switch recommender algorithms?*
- *Is there a clear favorite* algorithm?
- How much do the recommender lists *from the algorithms differ* per user?
- Could *the user's choice of algorithm* be predicted or do the users need to be *in control in order to identify the algorithm with which they will be most satisfied?*

At the beginning of the experiment users were *randomly assigned to one of the algorithms as their initial condition.* They found that *a substantial portion of* their *user base (25%) used the recommender-switching feature* and that most of the users preferred *a matrix factorization algorithm, followed closely by item-item collaborative filtering.* 72.1% of these users *settled on a different algorithm than they had been assigned. 26.2% of users who switched recommenders only did so within their first hour of using the new system* and *44.1% of users only switched recommenders during their first session.* Furthermore they *did not find any effect of the user's initial algorithm on their final choice if the user tried different algorithms.*

These different recommender algorithms *did produce measurably different recommender lists for the users.* And they were *unable to predict the user's final choice of recommender.*

In September 2015 Harper et.al. [HXK+15] published a paper called "Putting Users in Control of Their Recommendations" about an interactive movie recommender system, which *puts some control in the hand of the users*, used at MovieLens, *a movie recommendation web site with several thousand active monthly users.* They hypothesize *that such a system could leave users feeling more satisfied with their recommendations and more in control of the process.* Their research questions were:

- *Do users like having control over their recommendations?*
- *Given control, how different are users' tuned recommendations from their original recommendations?*
- *Do users converge to a common "best tuning" setting?*

The recommender system offered users to tune one variable which they called *blending variable.* Each user was in one of two groups. The first group was offered to tune the variable *popularity* (show more/less popular movies) and the second group was offered to tune the variable *age* (show more/less recent movies) by changing the weight of those variables. Their *offline evaluation* gave them *some general understanding of the effects of blending on top-N recommendations,* e.g. *that even small weights on the blending variable will have a dramatic influence on the resulting recommendations* and that *adding a small amount of popularity dramatically increases the average rating, while adding age slightly lowers the average rating of items*.

Since the offline evaluation did not show *how users perceive these effects and what amount of popularity or age (if any) they choose to blend in*, they conducted an online study, which consisted of two parts. The users were first asked to *"tune" a list of recommendations* until they found their *favorite list of recommendations*. To *remove any cognitive bias* the UI did not show what exactly the user was tuning (more/less of age and popularity respectively) but a *more neutral "left" and "right"*. The system was set up such that *each step — a right or left click — changes four items in the user's top-24 list* which was *enough to feel the list changing, but not enough to be overwhelming*. And then they were asked to *complete two surveys [...] about their original list of recommendations, and another [...] about their adjusted list.* When tuning their recommendations users *used a median of 10 actions*. They had 381 participants and *85% chose a configuration one or more steps from the initial setting*. In the following survey users *strongly preferred their top-24 recommendation lists after using the experimental controls to adjust the popularity or age*. Additionally they *responded positively to a survey question asking if they would use a feature like this if it were a permanent part of the system*. Regarding the other two research questions, they found that the *median user in the pop condition changed out 12 (50%) of their original top-24 recommendations, while the median user in the age condition changed out 7 (29%)* and *there does not appear to be a "one size fits all" tuning value, which underscores the importance of giving the users control*.

2.3.2. State of the Art

In the following, we take a look at what others have done regarding interactive recomender systems. We do this by categorizing the different approaches.

Search-like

Interactive recommender systems in this category offer users to explore a given domain (e.g. job postings) along certain suggested topics. The procedure is similar to entering terms into a search engine, but goes more in the direction of query suggestions: these RS suggest topics the user might be interested in, so the user does not have to know beforehand exactly all the relevant terms for what he is looking for. One possible approach is to look at the cooccurrences of search terms and offer terms that are frequently used together (if a user enters *java*, the system might offer *software development* as another topic the user might be interested in).

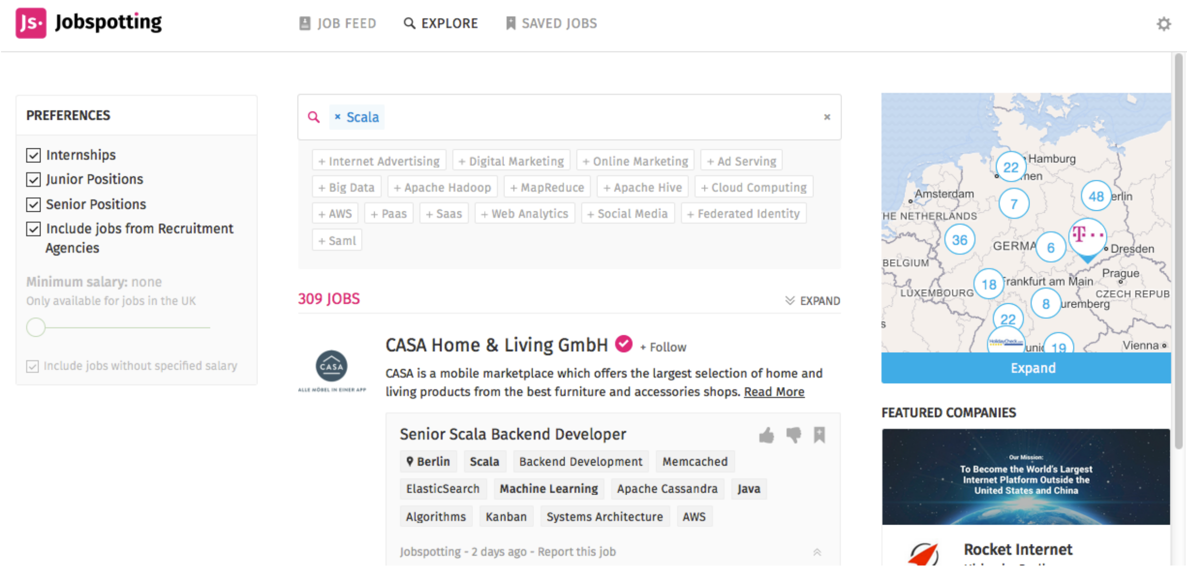


Figure 2.7.: Exploration at Jobspotting

Jobspotting Jobspotting[7] is a jobs portal that offers users to browse through their catalogue of job postings without creating actual profiles for the users. The interactive feature is called *browse* and offers a search bar. The user may enter skills, loactions or business fields. Once the user entered something, the system offers other terms the user might want to add to his search. Figure 2.7 shows an example browse session on Jobspotting. In the center of the screens the user may enter terms into the search bar. Below this search bar, the system offers other possibly relevant topics the user may add to her serach. The result list is displayed below the search bar and the suggested topics, and changes when the user alters the search. The individual items of the result list show some details about the respective job posting (like some description, location and skills). On the left, the user may choose the types of positions she is interested in (e.g. internships, junior/senior positions). And on the right a map is shown which displays the locations of the items from the result list.

Exploiting implicit feedback

Interactive recommender systems in this category analyze the users' behavior to offer recommendations. Different types of behavior and user interactions may be exploited such as clicks on items or even consumed items from a given domain (e.g. buying an item on a shopping site, watching a video on a movie site). The underlying recommender system analyzes the user behavior in order to understand whether a user liked an item or did not like an item. For example, the

[7] see `https://jobspotting.com/` for details

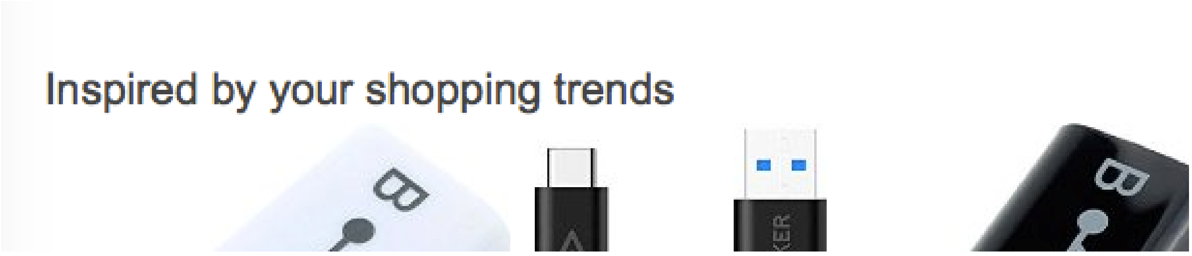


Figure 2.8.: Amazon - inspired by your shopping trends

impression of an item that was then also clicked by the user may be interpreted as positive feedback while an impression that did not receive a click may be interpreted as negative feedback. The recommender system can then, for example, recommend those items that are similar to the ones that the user “positively interacted with” and dissimilar to the ones that the user does not seem to like (item-to-item recommendations). Identifying similar / dissimilar items may be done by using metadata about the items the user interacted with (e.g. a user watched several action movies and the system might recommend to watch other movies of this category) or by using item-based collaborative filtering methods [SKKR01] that identify similar items based on the users that interacted with the items (e.g. two items may be considered as similar if many users interacted with both of these items).

Amazon Amazon[8] is the largest online shopping site. They offer a big variety of categories and a huge catalogue. Amazon uses recommendations in several places of the platform and in different situations. One variant which uses implicit feedback (recently ordered items) is called *Inspired by your shopping trends* and is shown in figure 2.8. This feature analyzes what the user recently bought and recommends other items based on these items.

Depending on the user’s context

Interactive recommender systems in this category react differently depending on the user’s context or last actions. A simple approach is to use Apriori or association rule mining [TSK05] as for example done by Bendakir and Aimeur [BA06] for course recommendations or even item-to-item collaborative filtering. These kind of recommendations even work quite well if the user is not logged in or doesn’t even have a user account, since only the current context is used.

Amazon Amazon uses these kind of recommendations in several variants in different places. Figure 2.9 shows Amazon’s analysis of the user’s current shopping

[8] see `https://www.amazon.com/` for details

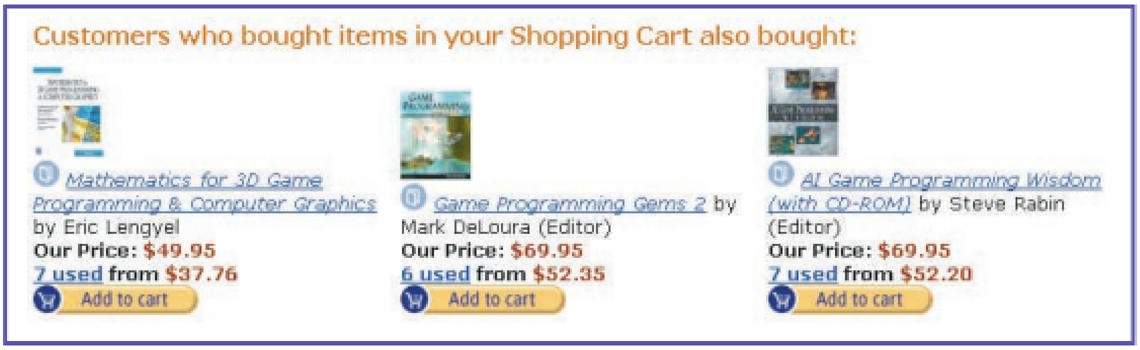


Figure 2. Amazon.com shopping cart recommendations. The recommendations are based on the items in the customer's cart: The Pragmatic Programmer *and* Physics for Game Developers.

Figure 2.9.: Amazon - Customers who bought items in your Shopping Cart also bought

cart and recommends items which where bought by other users who bought at least one item from the user's shopping cart (*Customers who bought items in your Shopping Cart also bought*).

Other recommendations are shown in the context of viewing a specific item. One feature (*Frequently Bought Together*) recommends to buy the currently viewed item together with items frequently bought with the said item (e.g. the user views a digital camera and amazon suggests to buy this camera and a SD-card for storage and a bag for protection/transportation because several other customers bought these three items together). Another feature (*Customers Who Bought This Item Also Bought*) is very similar. This recommender system also works in the context of viewing an item. But instead of recommending a whole group of items which are bought togehter, it shows several individual items which are frequently bought together with the first item but have themselves no relation (e.g. the user views a camera bag and amazon suggests several SD-cards since each of these have often been bought together with this bag). Figure 2.10 shows these examples.

And yet another variant analyzes the user's session and several items (*Recently Viewed Items*) and computes recommendations based on these. Figure 2.11 shows an example of a user who browsed through the category of digital cameras and their accessories and some recommendations based on these items (which are also from the same category).

Exploiting explicit feedback

Interactive recommender systems in this category take the users' feedback which is explicitly positive or negative and use this knowledge to generate recommendations.

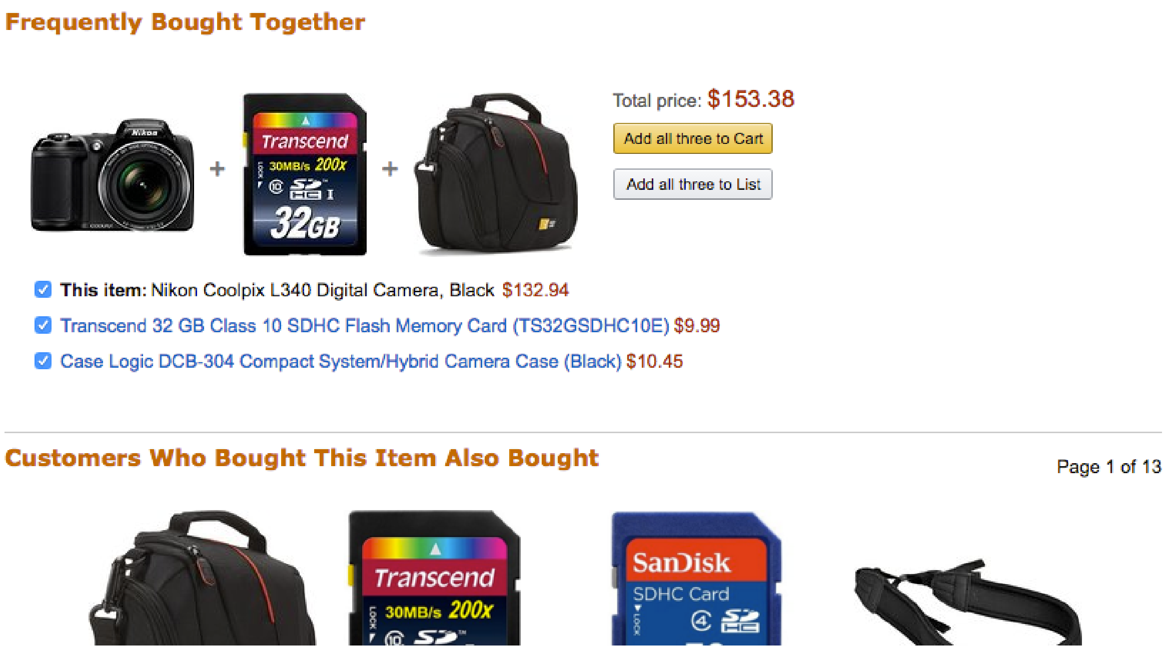


Figure 2.10.: Amazon - View Item

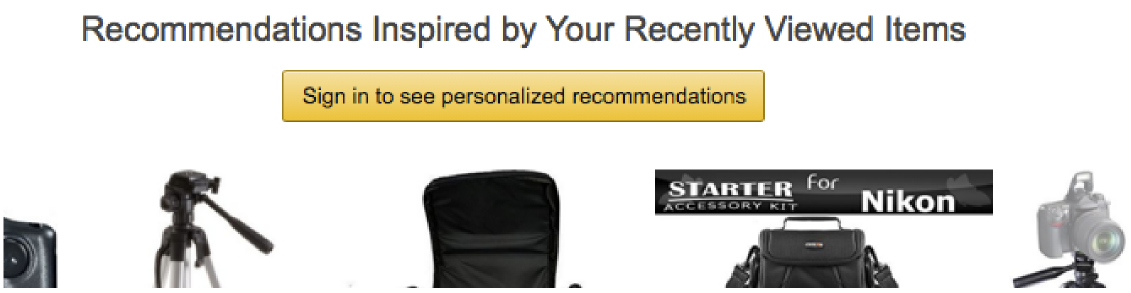


Figure 2.11.: Amazon - last viewed

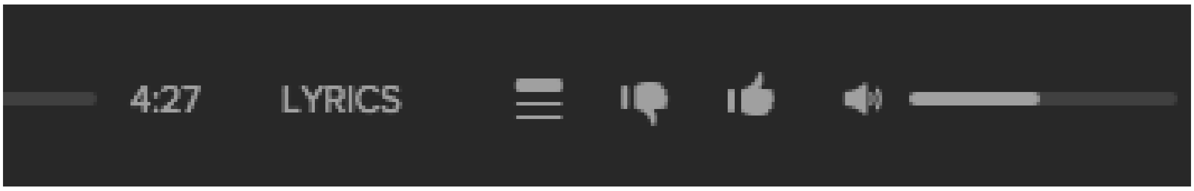


Figure 2.12.: Spotify - Radio

This is different from interactive recommender systems which use implicit feedback like clicks, since these can only assume that an interaction indicates positive feedback. It happens a lot that user's click on items for different reasons which indicate negative feedback. But a recommender system cannot easily distinguish a *positive* click from a *negative* click. An example of a *negative* click is a user who sees a bad recommendation and asks herself why this is recommended to her and clicks on this item simply out of curiosity.

Explicit positive or negative feedback often comes in the form of *more / less like this*, *+ / -* or *rating* (e.g. rating an item on a scale from 1 to 5 stars). This means that a user can tell the system for a specific item wheter she likes it or not. This feedback is then often used by Collaborative Filtering techniques (cf. Section 2.2.4). Another approach is to construct an interest profile from the feedback (i.e. extract the most common features of all items with positive feedback).

Spotify Spotify[9] is the largest music streaming service. Figure 2.12 shows their version of *more / less like this* (thumbs up / thumbs down). While the user listens to his personalized radio station, Spotify offers these buttons for the currently playing track. The service uses clicks on these to influence this radio station on the fly. Additionally clicking *thumbs down* skips immediatly to the next song.

Netflix Netflix[10] is the largest video / moview streaming service. Figure 2.13 shows their version of *more / less like this* (hot / cold). A movie or series is recommended to the user and the user may give explicit feedback by clicking the red right arrow (hot / positive) or the blue left arrow (cold / negative). This input is then used to generate new recommendations.

Tinder Tinder[11] is an online dating platform. Its most prominent feature is the interaction they offer to their users on mobile devices to tell the system whether they would like to meet some other user. The system shows photos of a given users. The user then decides by swiping the photo to the left or to the right.

[9] see `https://www.spotify.com/` for details
[10] see `https://www.netflix.com/` for details
[11] see `https://www.gotinder.com/` for details

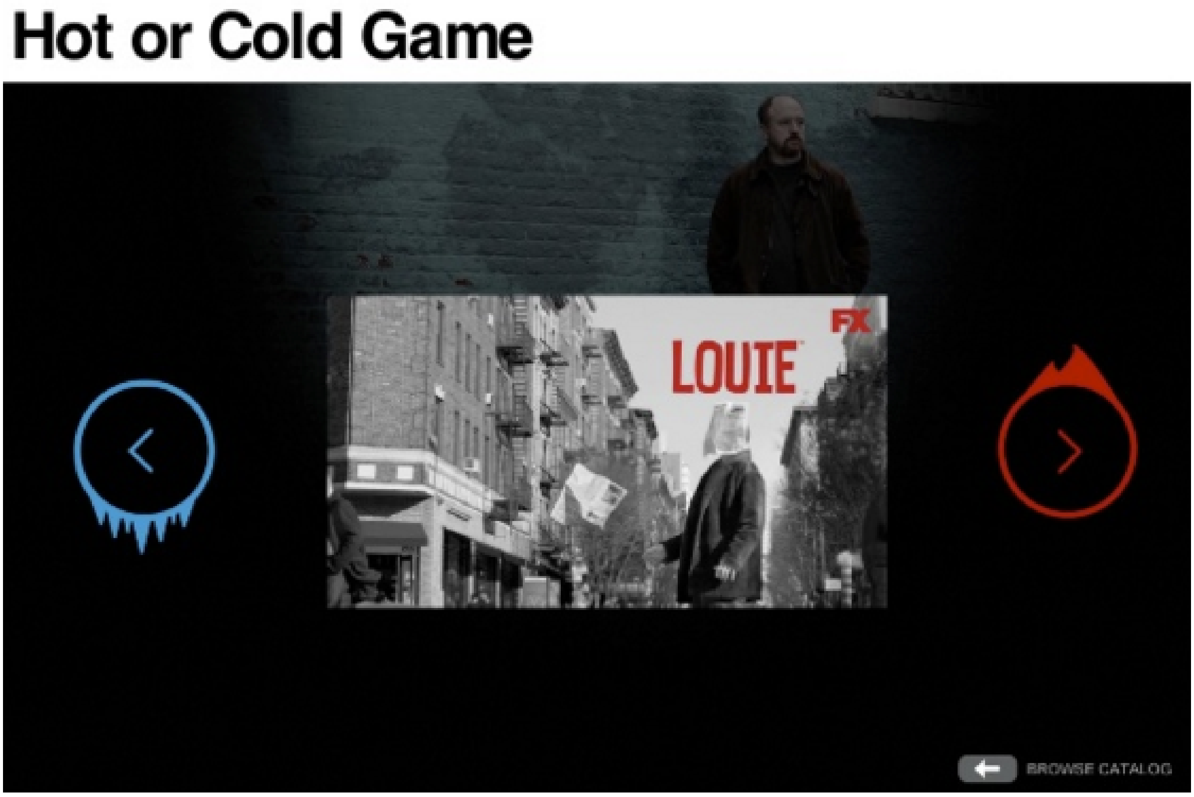


Figure 2.13.: Netflix - Hot or Cold

Approach	user control	effortless	impact	comprehensible	exploration
MovieLens_1 [EKHK15]	$+$	$+$	$+$	$--$	$\circ$
MovieLens_2 [HXK+15]	$+$	$+$	$+$	$-$	$\circ$
Search-like	$++$	$--$	$++$	$++$	$--$
Exploiting implicit feedback	$--$	$++$	$\circ$	$-$	$++$
Depending on the user's context	$+$	$++$	$+$	$\circ$	$++$
Exploiting explicit feedback	$\circ$	$-$	$+$	$\circ$	$\circ$

Table 2.1.: Comparison of approaches

2.3.3. Discussion

In the following, we compare and discuss the different approaches from Section 2.3.1 "Related Work" and Section 2.3.2 "State of the Art" (cf. Table 2.1). Explanations for the columns of Table 2.1:

user control	How much control does the strategy give the user over his recommendations?
effortless	Effort of the user to interact with the recommender system.
impact	How big is the impact of the user's actions?
comprehensible	How easy is it for a user to understand what implications his interactions have?
exploration	May the user just browse and explore the item collection? Or does she need a task (e.g. search for X)?

"MovieLens_1" [EKHK15] gives the users some control over their recommendations, with low effort and "measurably different recommender lists". But the system is still a black box and is still not comprehensible for the average user. The users' task is to choose one strategy for their recommendations. This task is considerably small, but still does not allow to just browse through the item space.

"MovieLens_2" [HXK+15] gives the users control over one variable and thus over their recommendations. But since the user is not told what exactly he is tweaking, the system still is not very comprehensible. The effort for the user is quite low and the impact of each change was designed to be "enough to feel the list changing, but not enough to be overwhelming". The users are not browsing the platform when tuning their recommendations, but the task of tuning one variable is manageable.

The users gain the most control over their recommendations with "Search-like", because they assemble their search query as they want or need it. In comparison, this requires the biggest effort by the users of these approaches. Usually users have a clear goal when using such systems and are looking for something specific. The users' actions have a possibly very high impact, but each step is still very comprehensible for the user.

On the other hand, "Exploiting implicit feedback" offers the least control, since the users might not even be aware of the effect of their behavior on recommendations while they are using the service. Users browsing the platform leave valuable implicit feedback with every action, but the impact of a single click is just moderate.

"Depending on the user's context" allows users to browse through the items with very little effort. Users have some control over the direction, and most recommended items are related to recent activities. But this is not always the case since recommendations might also depend on other users behavior.

"Exploiting explicit feedback" requires some effort by the users in the form of giving feedback on individual items. While the impact is usually quite high, the lists generated by techniques like Collaborative Filtering (cf. Section 2.2.4) also reflect other users' behavior and is not always transparent for the users.

2.4. Status Quo

This section describes what XING already has to offer regarding interactivity and job recommendations as well as the limitations of the current system.

2.4.1. Interactivity of XING's Recommender System

In the following, we present which means of implicit and explicit feedback are available, as well as their most important properties.

Implicit feedback

XING collects several types of implicit feedback on job recommendations (or job postings in general). Those are clicks, bookmarks, apply intents and deletes.

Users click on job postings for various reasons. They might be interested in the given position. But they might also click on job postings because they think this might be interesting for a friend or family member. But it might also happen that a user is just curious (e.g. about the job title of a job posting).

Users may bookmark job postings so they are able to find them again later. Bookmarking is considered positive implicit feedback. Apply intents on job postings are considered even stronger positive implicit feedback.

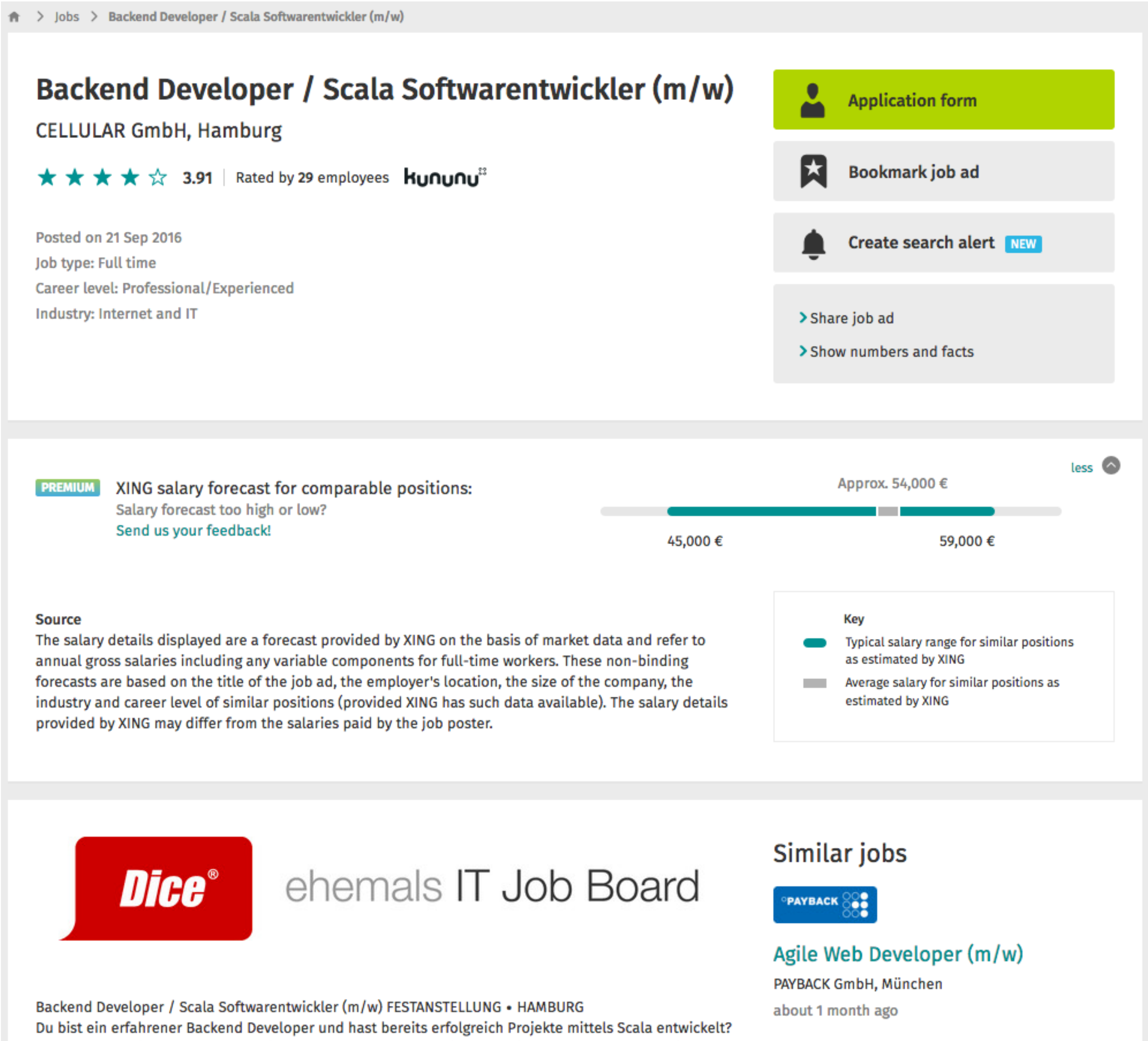


Figure 2.14.: An example of a job posting on XING

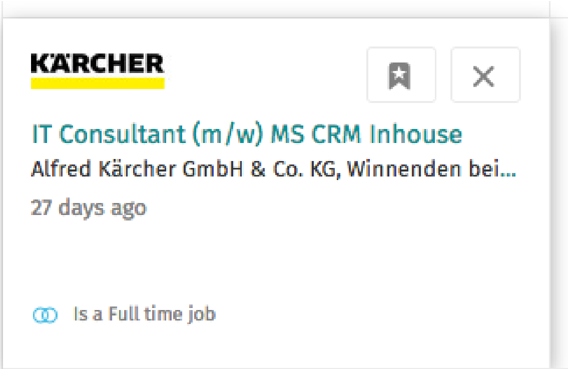


Figure 2.15.: A job recommendation with delete and bookmark buttons

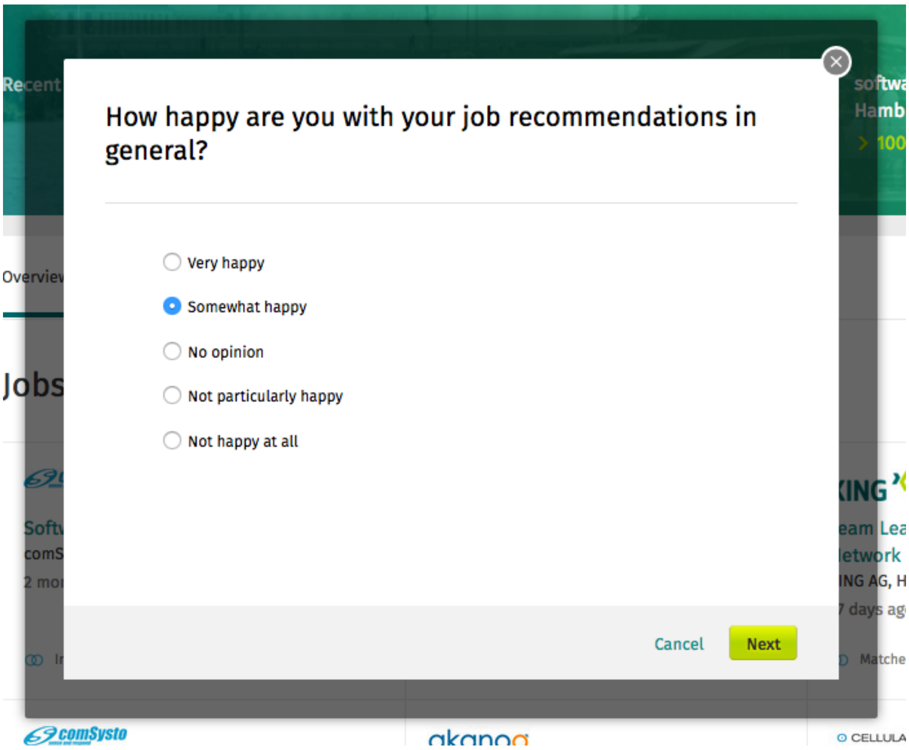
Deletes on the other hand are ambivalent. The reason behind this is the way the delete action is implemented in the UI. If a user clicks the delete button of a recommended job posting, it is replaced by a new recommendation. Most users use this feature to remove recommendations which in their opinion do not match their expectations. In this case those deletes should be considered negative implicit feedback. On the other hand some users seem to abuse the delete feature as a "show me more" feature as new recommendations appear after a delete. Those users might have actually disliked the deleted recommendation. But they might also have no opinion (neutral) or even have liked the item, but wanted to see new content (she might have seen the given recommendation already several times).

When XING implemented and tested (A/B test) a new feature called "LessLikeThis" based on the delete actions, this feature resulted in a significant worse click through rate (CTR) in the test group, thus indicating the ambivalent usage of the delete feature by users.

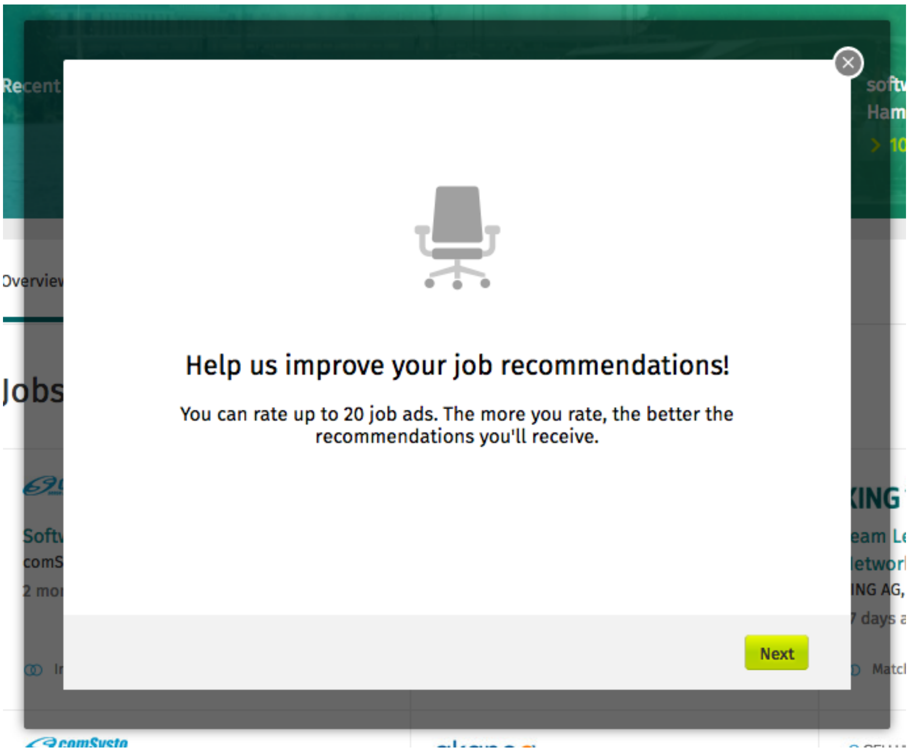
Explicit feedback

XING also collects explicit feedback on job recommendations in the form of the FeedbackApp. When a user chooses to participate, she first has to answer a general survey about her satisfaction with her job recommendations. Afterwards up to 20 of her recommendations are presented to her one after another. The user has to rate each recommendation on a scale from 1 to 5. When XING analyzed that feedback, they found that most users who used this feature were unhappy with their recommendations. And in contrast very few happy users gave feedback on their recommendations. We hypothesize that satisfied users see no benefit in using this feedback channel since they do not feel much need for improvement.

Furthermore a user may rate a given job posting after clicking on it. The feature is called *ITJFM* (*Is This Job For Me?*) internally and is presented to the users as "Are you a good match for this job?"; it is the next iteration of the Feedback

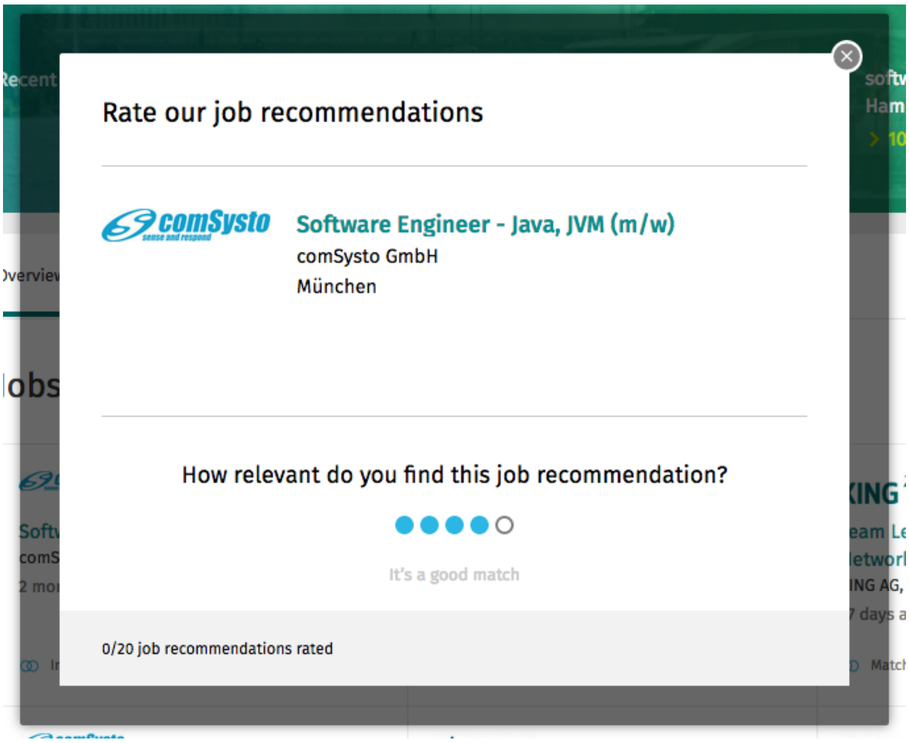


(a) FeedbackApp - survey

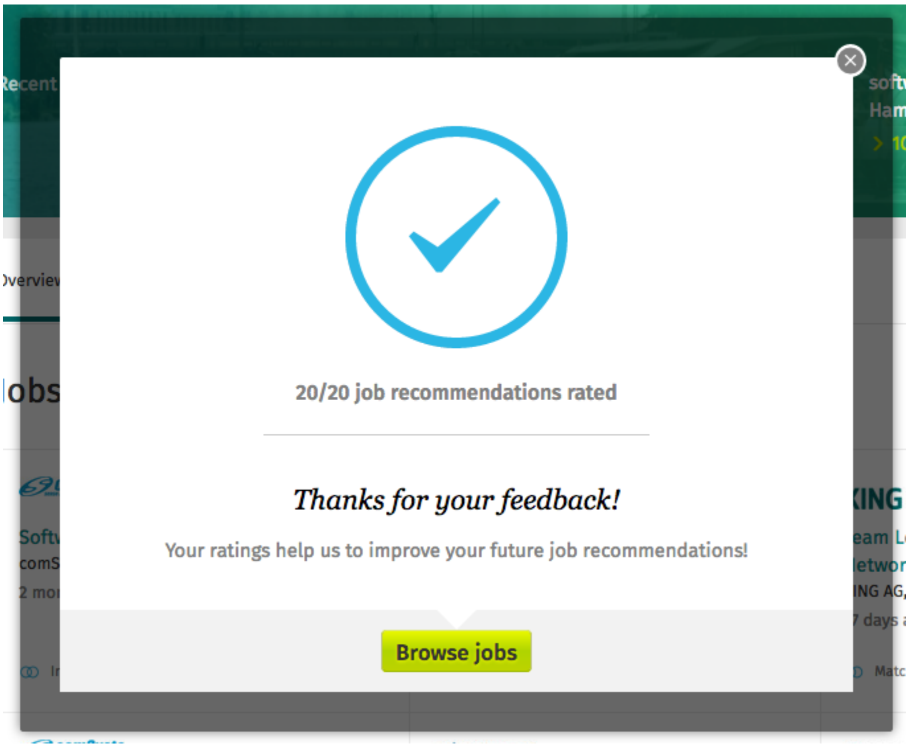


(b) FeedbackApp - introduction to ratings

Figure 2.16.: FeedbackApp (1)



(a) FeedbackApp - example rating



(b) FeedbackApp - rating session finished

Figure 2.17.: FeedbackApp (2)

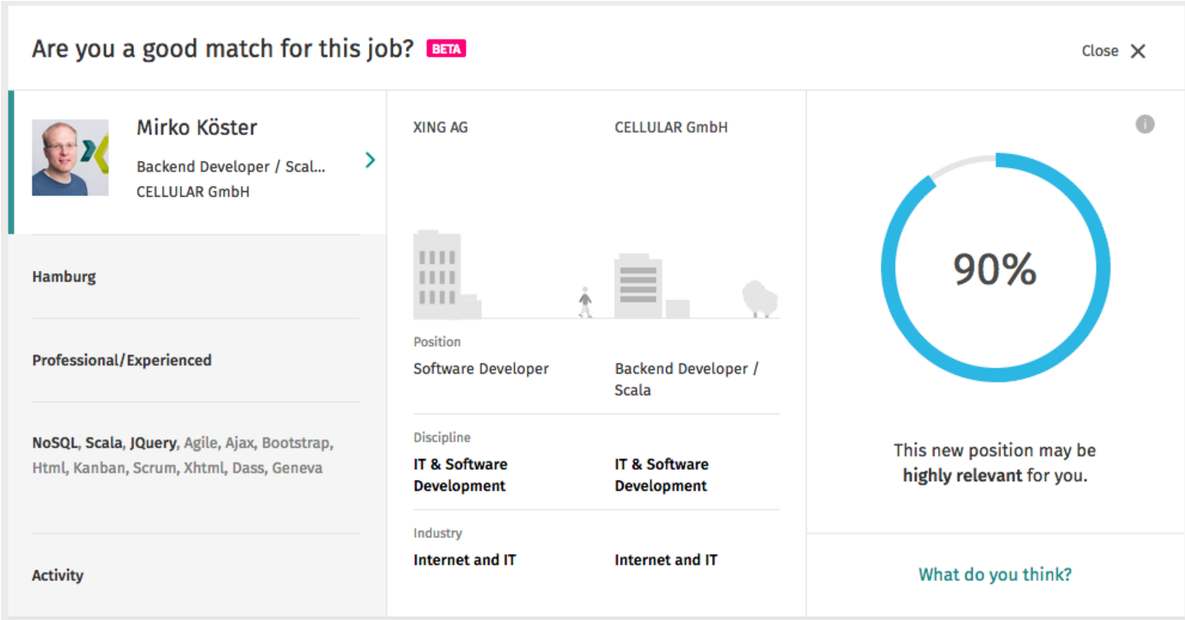


Figure 2.18.: Is This Job For Me? - Screen 1

app described earlier. ITJFM is embedded in the job postings' details page (see Figure 2.18 to Figure 2.22). The user sees what she has in common with the job posting (e.g. common skills, same career level, etc.) and a predicted rating based on those commonalities. The user may confirm or change the rating by clicking on it.

Statistics of the ratings indicate that users are more likely to use the Feedback-App if they are in general unhappy with their recommendations. The average rating and the distribution of ratings are significantly worse compared to ratings through ITJFM. We hypothesize that users want to interact with their recommendations and to tell XING what they think about them.

2.4.2. Limitations of the current system

In this section, we discuss limitations regarding *explicit control* and *enhancing user satisfaction*.

Explicit Control XING currently does not offer users means to explicitly control their recommendations. While there are means to influence job recommendations, those mechanisms do not put the user in direct control of her recommendations:

- Users can edit their XING profile but do not control how the job recommender is interpreting the changed profile. Moreover, the profile of a user rather describes her past and current experience (visible to the public) but does not

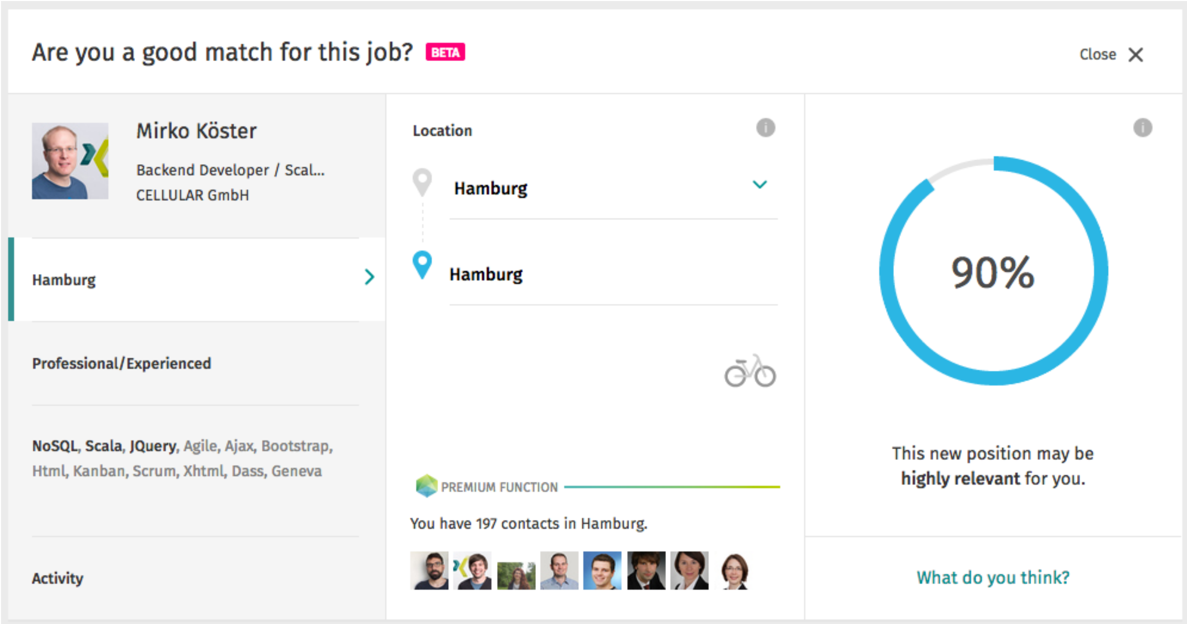


Figure 2.19.: Is This Job For Me? - Screen 2

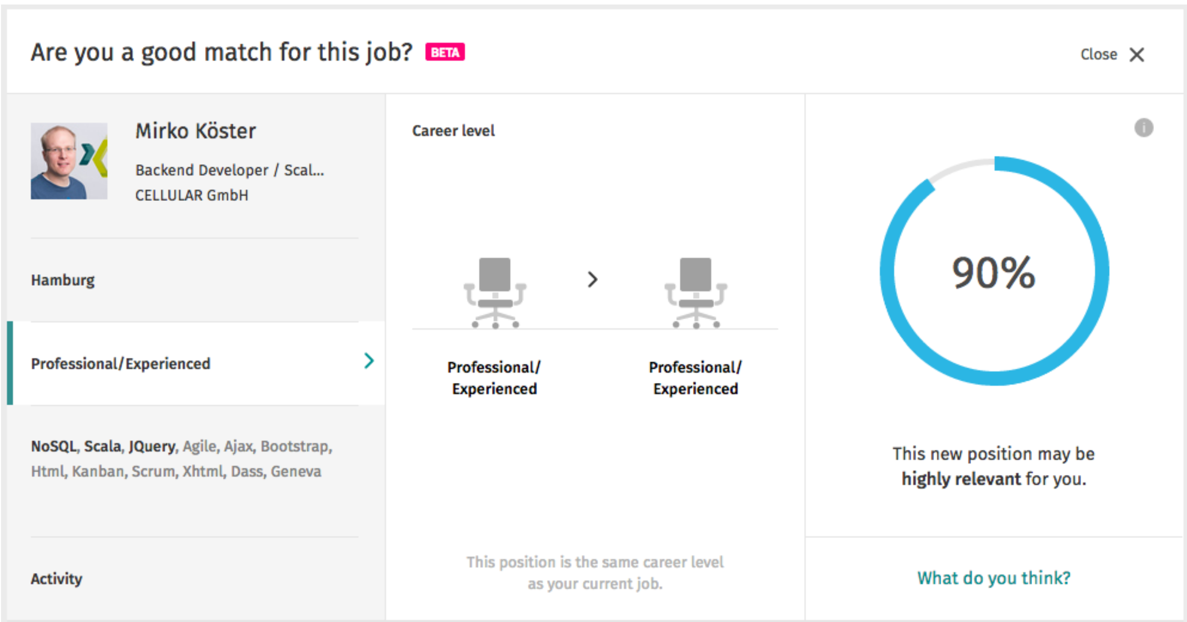


Figure 2.20.: Is This Job For Me? - Screen 3

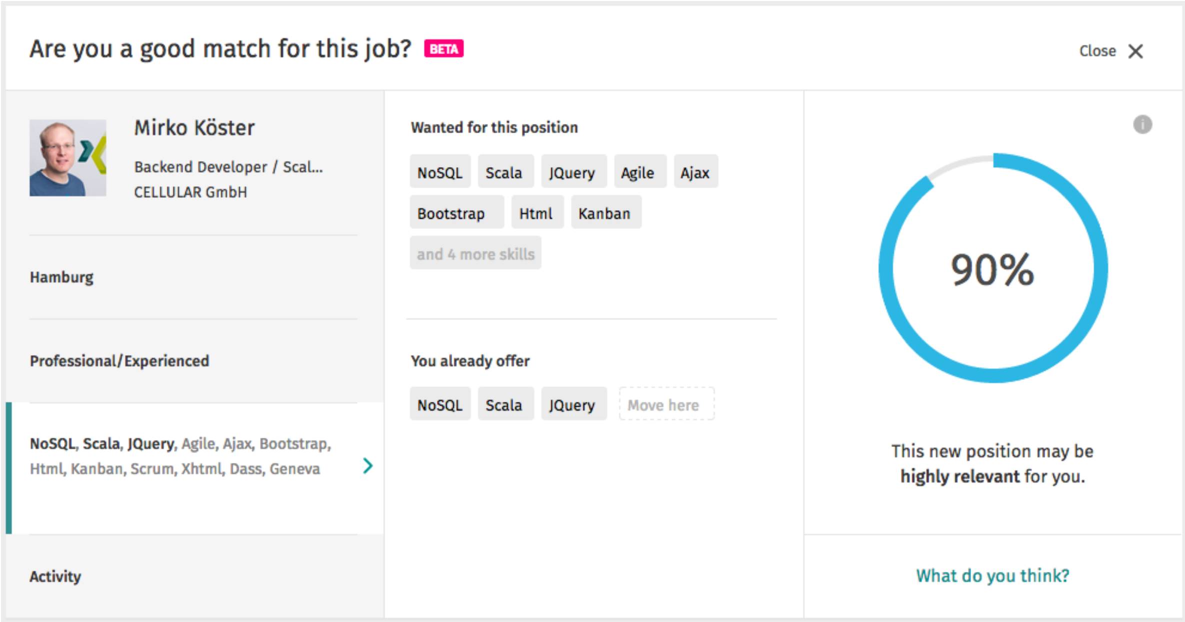


Figure 2.21.: Is This Job For Me? - Screen 4

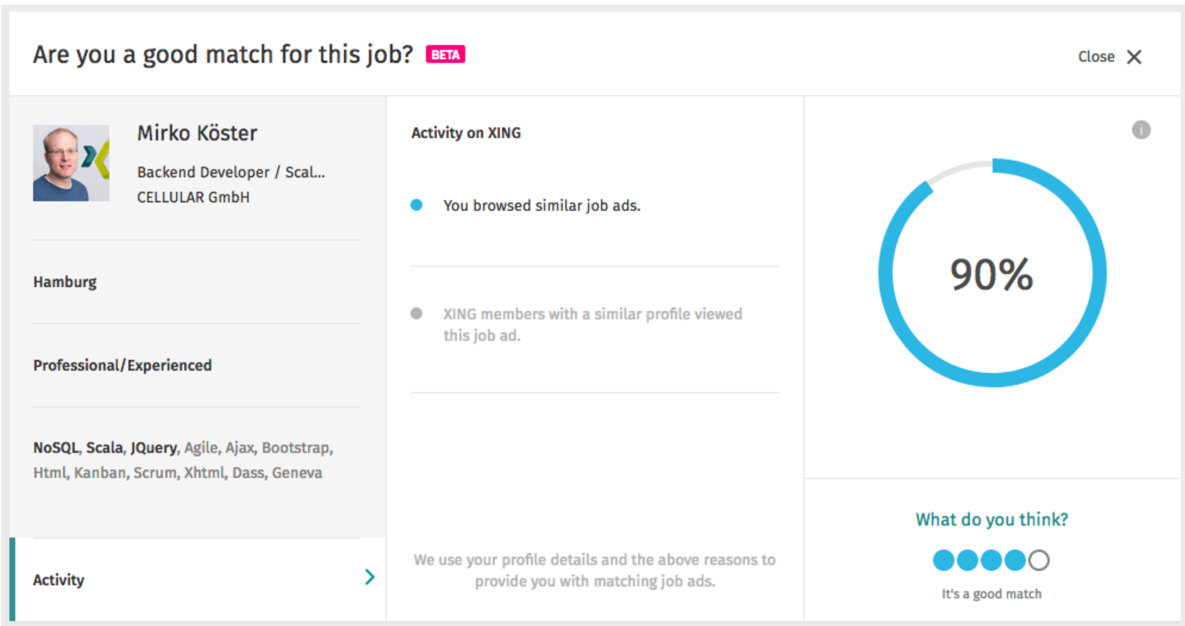


Figure 2.22.: Is This Job For Me? - Screen 5

provide means to describe wishes and preferences that users may have for their future career.

- People can delete or bookmark job recommendations. However, when those interactions are performed, the user does not get any feedback whether the corresponding interaction will influence the user's future recommendations or whether the recommender system at least understood the feedback and will learn from those interaction. As described above, *delete* interactions have, in fact, no impact on XING's recommender system as the so-called *less-like-this filtering* could not be implemented successfully.

Enhancing User Satisfaction XING allows its users already to give explicit feedback via their *Feedback App*. However, this Feedback App and the approach of asking users to give feedback and rate their recommendations in a survey-like manner has some drawbacks and shortcomings:

- it attracts a specific crowd of users, e.g. people who are dissatisfied with their recommendations are more likely to give feedback. This becomes obvious when comparing the average ratings that people provide in the Feedback App with those that people perform in *Is this Job For Me?* (ITJFM) for job recommendations. The average rating on job recommendations in ITJFM are more than 3 times higher than those provided in the Feedback App.

- the number of people who use the Feedback App is rather small. For example, around 4000 of the more than 2 Million unique monthly users typically give feedback on job recommendations. Therefore, most users are not motivated to give feedback.

- the Feedback App constitutes a rather artificial environment and does not allow for observing the natural behavior of the users. For example, the fraction of job recommendations that users skip and do not provide a rating is less than 1%. It thus seems that they feel forced or challenged to give feedback to all recommendations independently of the fact whether they have some opinion about the job recommendation or not.

- While the feedback that people provide via the Feedback App has a high impact on the future recommendations of the user, it does not convey the impression that the user feedback actually influences the user's recommendations. After the user has finished the feedback procedure, she will see a message entitled *"Thank you for your Feedback"* (cf. Figure 2.17b) and will then be redirected to an updated list of recommendations. Whether these updated recommendations relate to the feedback that the user just provided, is however not explained.

XING's Feedback App is thus geared towards collecting feedback about the current recommender system and is not geared towards enhancing user satisfaction. For example, the recommender system does not have an *interactive flavor* and does not give the user the impression that the system learns from user feedback and user interactions.

In this thesis, we aim to investigate solutions that tackle some of the aforementioned limitations. In the next chapter, we discuss strategies for giving users more explicit control over XING's job recommendation system.

3. Design

In this chapter, we provide an overview of our design of the interactive recommender system strategies. We present several strategies for turning XING's job recommendation system into an interactive recommender system and discuss these strategies using again our conceptual framework for categorizing interactive recommender strategies. We then describe how our selected strategy can be integrated into XING's job recommender system both regarding the required algorithms as well as the required front-end components. We conclude with the design of our user study that will allow us to evaluate the success of our designed strategies.

3.1. Possible Strategies

This section describes how the approaches of interactive recommender systems described in section 2.3.2 could be applied at XING for job recommendations.

3.1.1. Search-like

How could this strategy look like on XING?

Search-like A search-based strategy would allow the recommender system to create a dialogue with the user that aims at supporting the user to create search queries which in turn allow the users to explore and control the recommender system. As a starting point, some of the user's profile information could be displayed (e.g. the jobroles and cities from his last *x* CV entries, his skills, etc.). Then the user is allowed to choose any of these and even add his own jobroles, cities, skills, etc. by typing into a textfield with the aid of XING's auto completion service. Whenever the user is selecting a jobrole, city, skill or another filter element, a search query is updated and executed by the recommender engine. The results of the search query are presented to the user who then may further refine his search terms.

3.1.2. Exploiting implicit feedback

How could this strategy look like on XING?

Implicit feedback This is already done at XING. Implicit feedback (e.g. clicks) is used for precomputing the pseudo Collaborative Filtering as well as to generate the users' interest profile (cf. Section 2.2.4).

3.1.3. Depending on the user's context

How could this strategy look like on XING?

User's context The existing components (cf. Section 2.2.4) could be extended in a way that they focus on recent activities of the user (e.g. clicks and bookmarks) and extract information from these activities (e.g. the most common jobroles and skills from recently interacted job postings). This would be a minor modification of the existing system but would put emphasis on more recent activities, and thus give the user the feeling that his actions on the platform are being understood.

3.1.4. Exploiting explicit feedback

How could this strategy look like on XING?

Feedback One method could be to use the user's feedback on job recommendations (ratings). The positive ratings (e.g. 4 to 5 out of 5) are treated as input for the existing components as described in Section 2.2.4. The job postings which were rated negatively could be used in a *less like this* fashion by extracting common features (like jobroles and skills) and making sure these are not features the user has in her profile. This is important in cases where user receive job recommendations which are in general good fits, but the user disagrees in one feature. One example is a user who receives a lot of job recommendations with matching jobrole and skills but with job offers in cities she would not want to work at. If these recommendations were rated negatively, it is crucial to detect which features lead to the negative rating (in this example the cities). These features would afterwards be penalized by the job recommender.

This could also be exploited by gamification, e.g. by turning it into a *rating game*. The system would present a job posting from the user's recommendation list and ask the user to rate this item. This feedback is used immediatly by the job recommender system, which in turn presents the next unrated item. Additionally the system might ask the user if the interpretation of his ratings are correct, leading to even better feedback.

Strategies Another method could be to do something similar to what Ekstrand et.al. did in [EKHK15]. XING would predefine some strategies (e.g. a strategy could be a set of weights for the different job recommender components) and explain what the main differences are. The user would choose a strategy that best matches her needs. This strategy is then used by the job recommender system.

Backend weights Similarly the system might allow users to directly choose the weight of each component. The user might adjust the weights until he is satisfied with his job recommendations generated by the job recommender system using these weights.

Profile weights Furthermore the job recommender might allow users to define weights for their profiles' features. This might be done by presenting some sliders (ranging from -1 to 1, where -1 means *very unimportant*, 0 *neutral* and 1 *very important*) to the user. Thus the user may tell the system that the feature *jobrole* is very important and that *location* is very unimportant to her. This feedback would then be used by each of the job recommender system's components to generate better job recommendations for the user.

3.1.5. Job Recommendations Settings Page

This subsection describes the idea developed for this thesis.

settings This method will allow users to express their explicit feedback in the form of a settings page designed for job recommendations. The following categories of settings will be offered to the user:

- disciplines
- jobroles
- skills
- career level
- distance

The system will display some *topics* the user might be interested in (disciplines, jobroles and skills) and the user may emphasize (*more like this*) or de-emphasize (*less like this*) each item. The system will use this information to find more job postings including the emphasized topics, and use the de-emphasized topics to create a *soft* filter, which will not exclude job postings containing these topics but lowering its score. The career level setting will be offered in form of a range, so that users can express a minimum and maximum career level. The career level settings will be used as a *hard* filter, meaning a job posting not within this range will be removed. The distance setting lets the user choose a maximum distance from her current location. This setting will also be a *hard* filter which exludes job postings too far away from the user's location.

Approach	user control	effortless	impact	comprehensible	exploration	novelty at XING
Search-like	++	−−	++	+	−−	−
Implicit feedback	−−	++	+	−	++	−−
User’s context	−	++	○	○	+	○
Feedback	○	−	+	○	−	−
Strategies	−	○	+	−	○	+
Backend weights	−	−	+	−	○	+
Profile weights	○	○	○	+	−	+
settings	+	○	+	+	○	+

Table 3.1.: Comparison of possible strategies. cf. Table 2.1

Topics The list of topics will be generated by analyzing the user’s existing job recommendations and will not simple be taken from the user’s profile. Users might have been annoyed if they now had two places where they had to maintain that information. This would not have been a pleasent user experience. Instead we extract the topics (disciplines, jobroles and skills) from the user’s top k recommended job postings. The generated list is ranked by the number of occurrences and the item’s trustworthiness. This list of topics can be seen as a *summary* of the user’s job recommendations.

The advantage of this approach is that users can express directly what they like or dislike about their recommendations, thus boosting topics they like and even more importantly removing outliers from their list of job recommendations which are not interesting to the user.

3.1.6. Discussion

In this section, we compare and discuss these possible strategies similarly to Section 2.3.3. Compared to Table 2.1, Table 3.1 consists of one additional column: *novelty at XING*. This column shows whether a strategy is already available on the XING platform in some form or would be new.

The approach of "Search-like" would allow the most user control and would possibly result in the highest impact. The users need to know somehow what they are looking for, and they need put some effort into the system to get the results they expect. Since XING offers fulltext search with auto-completion for job postings (among others), this feature would not offer a completely new experience for users of the platform.

"Implicit feedback" is already used at XING (cf. Section 2.2.4).

Using "User's context" could be done while browsing the platform and is quite effortless. Since this would be a change to the existing system which uses less of the older behavioral data, it would not be a new feature and the impact would be negligible.

"Feedback" is partly used within the XING job recommender system in the form of the positive feedback (cf. Section 2.2.4). Implementing the "Less-like-this component" using the negative feedback would be new. The impact of these 2 should be quite high, but would require some work of the users, who would have to rate some items to improve their recommendations. This solution does not allow to browse the item catalogue. Additionally it might not be transparent to the users, which aspects of the rated job postings lead to the new list of recommendations.

The approaches "Strategies" and "Backend weights" are similar to the work of Ekstrand et.al. [EKHK15]. Although giving users some control, it might not be obvious to the users what happens behind the scenes once they chose a strategy or set the weights respectively. Finetuning the weights is even more work for the users than choosing one strategy. Allowing users to control their recommendations like that would be a new feature on XING.

"Profile weights" would afford some effort by the users and might lead to considerable impact on their lists of recommendations. Understanding the implications while using this new feature would be quite easy for users.

The approach of "settings" gives users direct control over their recommendations and would have a high impact, since users give explicit feedback on properties of their current list of recommendations. It requires some effort by the users to finetune their settings. The implications of these settings would be easy to understand for them. This approach would be something new on the XING platform.

3.2. User Interface

We want to give the user explicit control over job recommendations. This is why we came up with the idea of a preferences page for job recommendations. Figure 3.1 shows a draft (in German) of the settings page, which incorporates the ideas developed in Section 3.1.5; this draft was created in cooperation with a UX designer at XING. The first setting in this draft is *career level*, which can be controlled with using a slider with two handles to define the range. Then *max. distance* can be set to one of several predefined values. The next section would be the size of the company in terms of number of employees. In the scope of this thesis this feature was dropped, but might still be a good idea to pursue. Finally, the list of suggested topics is presented and the user is asked to mark them as important.

The right sidebar shows a preview of the recommendations generated using these settings. This preview could be refreshed on every change, thus offering real interactivity. While the setting themselves could be offered on the small screens of mobile devices, this preview would most likely be omitted on mobile, since the available space is a limiting factor.

3.3. Design of the evaluation

In the following we investigate the effect of user adjusted recommendations. Therefore, we discuss the methodology and design a study comparing the adjusted recommendations to random recommendations and the current system used at XING. We measure the effect of the user adjusted recommendations using an offline evaluation and a questionnaire. In the offline evaluation, we measure the potiential impact of our proposed method. In other words, how much do recommendation lists change when using the adjusted recommendations. Futhermore, we designed a questionnaire to show that adjusted recommendations potentially lead to higher user satisfaction.

3.3.1. Methodology

In order to evaluate the proposed method, the preferred approach would be to make the settings publically available and perform a split test [KHS07] to measure the effect. Meaning that all XING users are divided equally into a *control group*, which would get the same job recommendations as before, and a group for which the settings take effect, so the results of the two groups could be directly compared. This test could answer the question, *whether offering settings to the users lead to more clicks on job recommendations for those users who adjusted their settings.* This procedure leads to a possibly big sample size, which is advantageous when performing significance tests. In addition, if carefully split, the two groups would have the same properties as the whole population of XING users. This would allow to generalize the conclusions.

Unfortunately, this approach is not feasable, since the prototype developed for this thesis is not available outside of XING's network and developing a frontend, that is ready to be integrated into the XING platform, is out of the scope of this thesis. Taking that into consideration, the user study has to be performed at XING itself.

Offering a prototype frontend of the settings to XING employees and performing a split test on them would be possible. But as internal studies had shown, XING's employees are not representative for its userbase, e.g. the distribution of jobroles is different and also the click behavior differs, since they interact with the platform as part of their job. This prevents using clicks on job recommendations to measure the effect of settings as proposed above. Additionally generalizing the results to all XING users gets difficult.

Another approach is to design an interactive questionnaire and ask XING's employees to participate. This questionnaire would ask for explicit ratings after applying the settings. This avoids the problem with using clicks for evaluation. The problem with generalization stays the same, though. Since this approach is feasable and avoids some of the problems the other two proposed evaluation methods have, we choose to design and implement it. The design is described in section 3.3.3 and the implementation details are described in the next chapter.

But first, the design of an offline evaluation is depicted in the next section. It aims at evaluating the possible impact of the proposed method.

3.3.2. Offline evaluation

We will perform a quantitative evaluation that we conduct to measure the possible impact of the interactive elements at large scale. Therefore, the output of the current job recommender system will be compared to the output of the interactive recomender system, which has to be called with random (but controlled) settings. To measure the impact of settings, the intersection of the two generated lists will be used as our metric.

3.3.3. User study

The user study is done in the form of an interactive questionnaire. The process of the questionnaire is as follows:

1. We identified a participant (by his XING account)
2. We let the user rate a list of personalized topics such as disciplines / jobroles / skills.
3. In the next step the participant adjusts his settings in order to generate personalized recommendations. First, the participant picks the career level range of interest. Furthermore, the participant specifies location based preference. Finally, the participant selects topics of interest and topics he dislikes.
4. We present the participant three lists of up to 20 recommendations. The first is the original list of job recommendations generated by the current system. The second list of recommendations is generated using the participant's preferences. And finally a list of random recommendations which serves as a *control group* mechanism.

5. The participant rates each of these lists, which are displayed in random order.

6. This step is repeated with the addition of rating the individual job postings of the 3 lists.
   - This is introduced because people found it rather difficult to rate those lists with up to 20 items without some cognitive aid.

7. Finally, the participant may leave additional (optional) feedback as free text.

Possible values while rating the list of topics and the three lists of job recommendations are *very low*, *low*, *medium*, *high* or *very high*. The scale of ratings is thus *ordinal*. Using the questionnaire, we investigate whether the tuned job recommendations perform better than the original and random recommendations.

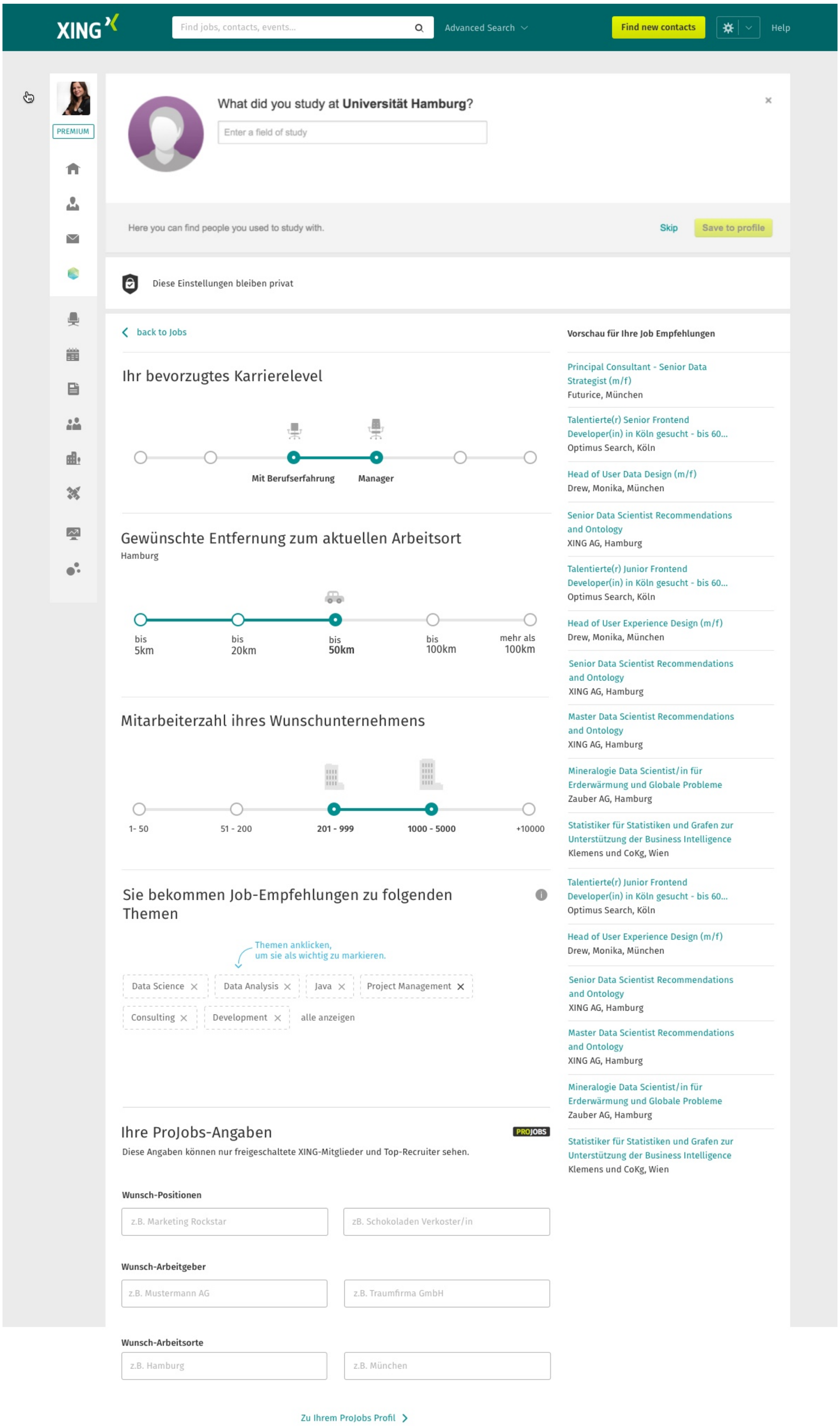


Figure 3.1.: Draft of Job Recommendations Setting page

4. Implementation

This chapter describes how the concept introduced in section 3.1.5 "Job Recommendations Settings Page" was realized and how the user study is working.

4.1. Topics of interest

As described in section 3.1.5 in the last chapter, a list of *topic*s the user might be interested in should be generated and displayed. This is done by summarizing the user's list of job recommendations and extract the most common topics. A topic in this context is one of the following:

- disciplines
- jobroles
- skills

To generate and suggest topics to the user, the regular job recommender system is called to get a list of job recommendations. Afterwards the most common disciplines, jobroles and skills are extracted and ranked by the number of occurrences. As a byproduct of this thesis, a *topic recommender* REST service was created, which creates this list of topics and returns them with some additional information as a json response. Each recommended topic consists of the following information:

item_id a unique id identifying the topic

score a score between 0 and 1 indicating the relevance of this item to the user

position position within the list of recommended topics (0 based)

reasons an array of reasons (= the job recommender system's backends) which contributed to this topic

labels a map of languages to a representing label of the given topic

trustworthiness a score that reflects the probability of the topic to appear in a negatively rated job recommendation

```json
{
  "total": 255,
  "collection": [
    {
      "item_id": 4607683,
      "score": 1,
      "position": 0,
      "reason": [
        "whi",
        "interactions",
        "collaborative",
        "mlt"
      ],
      "labels": {
        "en": "IT & Software Development",
        "de": "IT und Softwareentwicklung"
      },
      "trustworthiness": 0.578237
    },
    {
      "item_id": 24432,
      "score": 1,
      "position": 1,
      "reason": [
        "whi",
        "interactions",
        "collaborative",
        "mlt"
      ],
      "labels": {
        "en": "Software Development",
        "de": "Softwareentwicklung"
      },
      "trustworthiness": 0.502728
    },
    ...
  ]
}
```

4.2. Changes to the existing job recommender system

The existing job recommender system was extended to accept further parameters representing the user's preferences. The following additional parameters are supported now:

disciplines a comma separated list of dicipline IDs, a colon as a separator and a setting[1]

jobroles a comma separated list of jobrole IDs, a colon as a separator and its setting[1]

skills a comma separated list of skill IDs, a colon as a separator and its setting[1]

min_career_level the minimum career level a recommended job posting needs to have[2]

max_career_level the maximum career level a recommended job posting needs to have[2]

max_distance the maximum distance from the user to each recommended job posting's location[3]

These parameters correspond with the possible settings described in section 3.1.5. An example for the skills parameter might be "123:1,42:-1", which means that the skill with the ID 123 was emphasized and the skill with the ID 42 was de-emphasized by the user.

After parsing these additional parameters, the adjusted job recommender system uses these settings in each of its backends and even in some filters (e.g. *max_-distance* is used in the distance filter to remove job postings which are in locations too far away from the user). The existing job recommender system was not able to use *negative* aspects, so this feature was added as part of this work.

4.2.1. Preparatory work

As described in section 2.2.4 "Algorithms", three of the four existing backends build an elasticsearch query which is executed on an index especially maintained for the job recommender system. During the analysis of these query generators it becaume obvious that the generated queries not only do not support negative weights to de-emphasize, but are somewhat inefficient as they create some part

[1]positive or negative, indicated by 1 or -1
[2]a value between 1 and 6
[3]in *km*

of the query that is supposed to act as a filter (at least one jobrole or skill has to match), but is not an actual elasticsearch filter. This means that unneccessary work is done by the elasticsearch cluster since scores are calculated for these filters which are never used. Additionally these three backends built slightly differently structured queries.

As a preparation for this work these query generators where unified and support for negative weights was introduced.

4.2.2. elasticsearch queries: usage of the user's settings

The user's *distance* and *career level* settings are used to build elasticsearch filters, which act as *hard filters*, i.e. any job posting not matching these criteria won't be included in elasticsearch's result list. The minimum and maximum career level settings are used to generate a range filter. The distance setting is used to generate a *geo location query*. The user's current location (or inferred location, depending on the backend) in the form of *geo coordinates* is used as the filters centre and the radius is taken from the setting. This way, only job postings with locations within that area are found and returned.

The user's *topic* settings are used in two places. The topics (jobroles and skills), which are emphasized (positive), are used in the filter part of the generated query, so that at least on of the user's origninal or emphasized jobroles or skills have to match. All emphasized topics are then used in the query part of the elasticsearch query. The de-empahsized topics (negative) are weighted negatively, so job postings containing these topics will get penalized by elasticsearch.

The *pseudo collaborative filtering* backend cannnot make use of the settings for career level, distance and negatively rated topics directly. It may use the positively rated topics to lookup precomputed suitable job recommendations.

Career level and distance settings are again used in the end by the filters.

4.3. Implementation of the evaluation framework

In the following, we present the questionnaire we designed as well as the data it collects.

4.3.1. User interface for the evaluation

The questionnaire guides the participants through the following steps:

Step 0 The goals of the questionnaire are explained and the participant has to enter his XING user ID (see Figure 4.1 "Questionnaire: Welcome page").

Step 1 In this step the personalized list of topics (up to 25) is generated for the participant, who is asked to rate this list considering *how useful, relevant or important these topics are for him or his profession* (see Figure 4.2 "Questionnaire: List of topics"). Valid ratings are *very low*, *low*, *medium*, *high* or *very high*.

Step 2 Afterwards, the participant has to adjust his job recommendation preferences (see Figure 4.3 "Questionnaire: User's settings"). He has to specify his range of career levels. Career level is ordinal data and allows the following values: *student or intern*, *entry level*, *professional experienced*, *manager*, *executive*, and *senior executive*. Furthermore, we ask the participant to specify how far he is willing to commute or move. Possible settings are: *5 km*, *20 km*, *50 km*, *100 km*, *200 km*, or *no limit*. And finally he selects the topics of interest, topics he dislikes and topics he has no opinion of. This is done by assigning one of the following values to each of the personalized topics: *positive*, *negative*, or *neutral*. Then he might proceed to the next step

Step 3 Three lists of recommendations are generated. First, the *original* list using the existing XING job recommender system. Second, a random list of job recommendations, generated by taking the original top 60 job recommendations of the participant, shuffling them and using the top 20 of these as the *random* list of job recommendations. And third, the *tuned* list of job recommendations generated by using the settings from the step before. For each job posting in these lists, we present the *title*, the *city*, the *career level* (e.g. "3 - professional experienced"), and the *discipline* (e.g. "Engineering & Technical") (see Figure 4.4 "Questionnaire: Rate lists of recommendations"). The *company* is omitted on purpose to remove any possible bias. The participan is asked to decide which of the 3 recommendation lists he likes most and to rate each of the lists (using the same five values as in step 1, ranging from *very low* to *very high*). While rating, he is asked to consider how *relevant* the job postings in each list are for him, how well they meet his preferences and how well the lists are ordered according to relevance.

Step 4 Then the same three lists are presented. Additionally, we ask the participant to rate each individual job posting (*thumbs up* or *thumbs down*) (see Figure 4.5 "Questionnaire: Rate lists of recommendations as well as individual job postings"). Ratings for job postings present in more than one lists are applied to all of these once the user clicks on the corresponding button. Clicking on *thumbs up* marks the job posting green, while clicking on *thumbs down* marks it red. Below each list a statistic of how many items he rated positively or negatively in the corresponding list is shown. This serves as visual aid for the participant to decide again how to rate the three lists. It is possible to rate them differently than in step 3.

Step 5 The last step thanks the participant for his feedback (see Figure 4.6 "Questionnaire: Thanks for your participation").

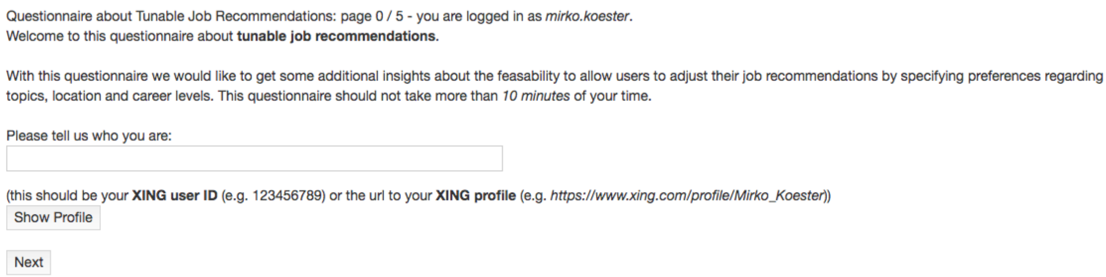


Figure 4.1.: Questionnaire: Welcome page

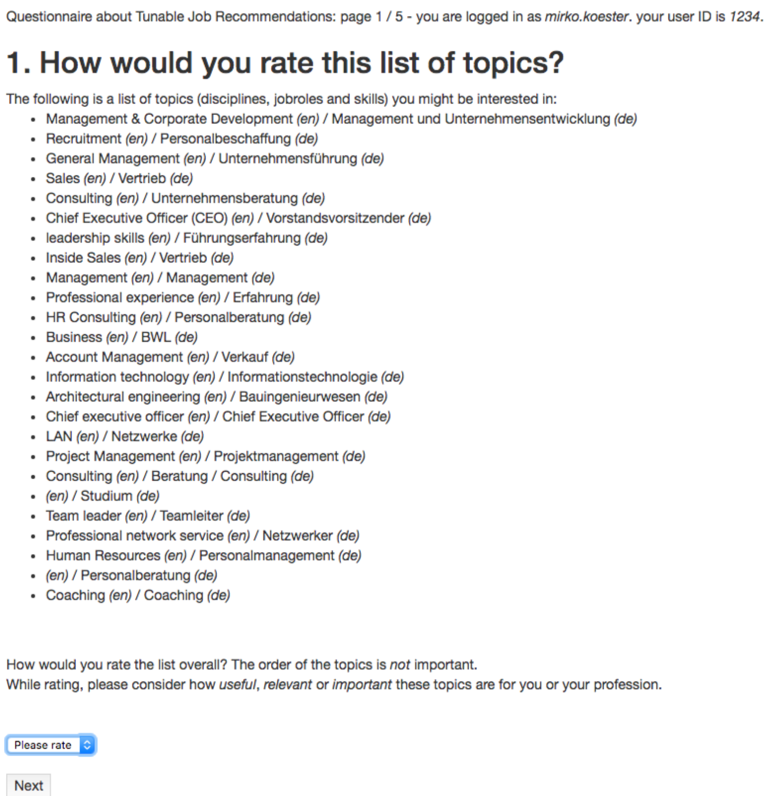
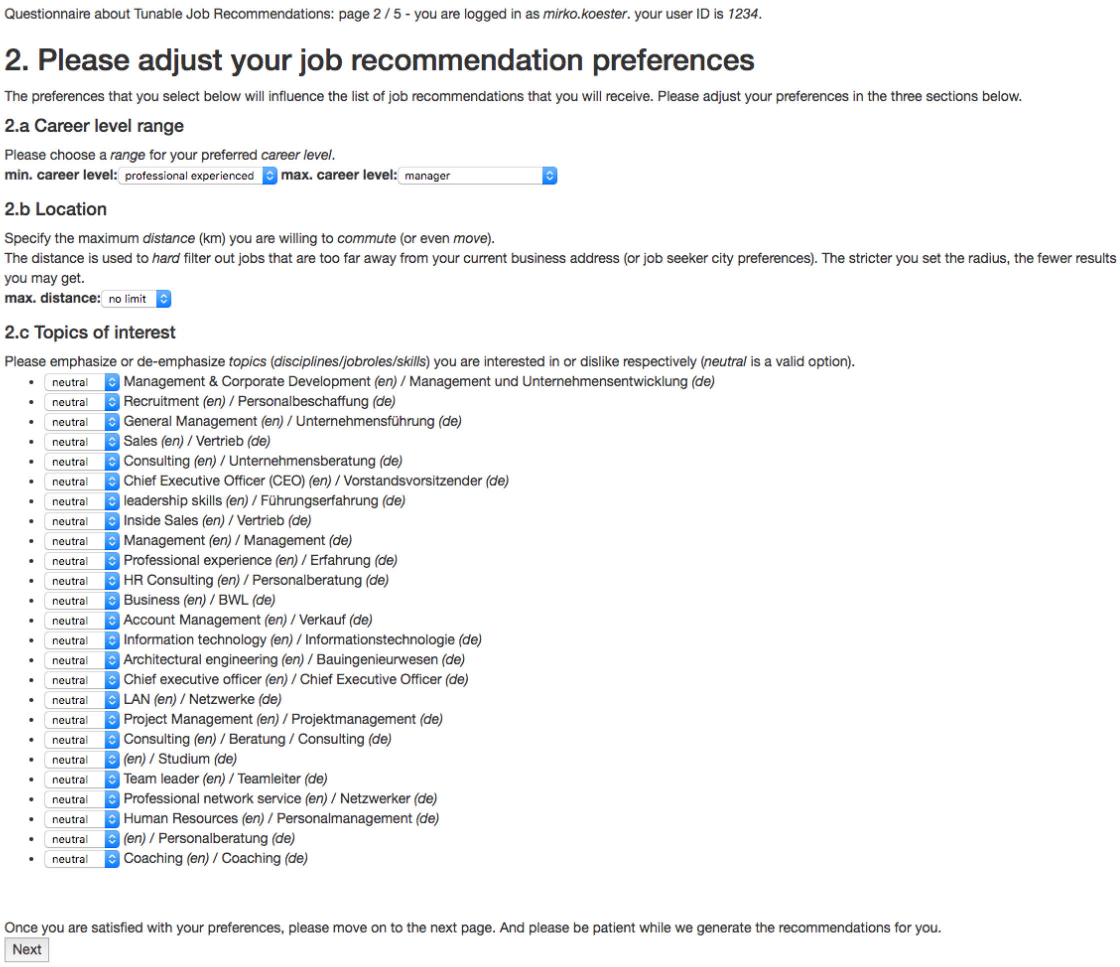


Figure 4.2.: Questionnaire: List of topics

4.3.2. Data collected by the questionnaire

During the participation, the following data is stored:

- XING user ID


Questionnaire about Tunable Job Recommendations: page 2 / 5 - you are logged in as *mirko.koester*. your user ID is *1234*.

2. Please adjust your job recommendation preferences

The preferences that you select below will influence the list of job recommendations that you will receive. Please adjust your preferences in the three sections below.

2.a Career level range

Please choose a *range* for your preferred *career level*.
min. career level: professional experienced **max. career level:** manager

2.b Location

Specify the maximum *distance* (km) you are willing to *commute* (or even *move*).
The distance is used to *hard* filter out jobs that are too far away from your current business address (or job seeker city preferences). The stricter you set the radius, the fewer results you may get.
max. distance: no limit

2.c Topics of interest

Please emphasize or de-emphasize *topics* (*disciplines/jobroles/skills*) you are interested in or dislike respectively (*neutral* is a valid option).

- neutral Management & Corporate Development *(en)* / Management und Unternehmensentwicklung *(de)*
- neutral Recruitment *(en)* / Personalbeschaffung *(de)*
- neutral General Management *(en)* / Unternehmensführung *(de)*
- neutral Sales *(en)* / Vertrieb *(de)*
- neutral Consulting *(en)* / Unternehmensberatung *(de)*
- neutral Chief Executive Officer (CEO) *(en)* / Vorstandsvorsitzender *(de)*
- neutral leadership skills *(en)* / Führungserfahrung *(de)*
- neutral Inside Sales *(en)* / Vertrieb *(de)*
- neutral Management *(en)* / Management *(de)*
- neutral Professional experience *(en)* / Erfahrung *(de)*
- neutral HR Consulting *(en)* / Personalberatung *(de)*
- neutral Business *(en)* / BWL *(de)*
- neutral Account Management *(en)* / Verkauf *(de)*
- neutral Information technology *(en)* / Informationstechnologie *(de)*
- neutral Architectural engineering *(en)* / Bauingenieurwesen *(de)*
- neutral Chief executive officer *(en)* / Chief Executive Officer *(de)*
- neutral LAN *(en)* / Netzwerke *(de)*
- neutral Project Management *(en)* / Projektmanagement *(de)*
- neutral Consulting *(en)* / Beratung / Consulting *(de)*
- neutral *(en)* / Studium *(de)*
- neutral Team leader *(en)* / Teamleiter *(de)*
- neutral Professional network service *(en)* / Netzwerker *(de)*
- neutral Human Resources *(en)* / Personalmanagement *(de)*
- neutral *(en)* / Personalberatung *(de)*
- neutral Coaching *(en)* / Coaching *(de)*

Once you are satisfied with your preferences, please move on to the next page. And please be patient while we generate the recommendations for you.
Next

Figure 4.3.: Questionnaire: User's settings

- topics
  - id and type (discipline, jobrole or skill) and position of each topic
  - rating for the list of topics
- settings
  - min. and max. career level
  - max. distance the participant is willing to commute or move
  - setting (positive, neutral or negative) for each topic
- job recommendations
  - 3 lists of recommendations (original, tunded and random)
    - id and position of each job posting
  - inital ratings of each lists of recommendations

- rating (positive or negative) of each job posting
- adjusted ratings of each lists of recommendations

- comment (optional)

4.3.3. Statistical hypothesis testing

We use the responses from the participants of the study to evaluate the three conditions. Therefore, we use the ratings to construct a list of ranked conditions for each participant. We evaluate if user adjusted recommendations rank significantly higher than the other conditions. We compare the average rank in each condition to find the best scoring condition and use the *Wilcoxon signed-rank test* [Wil45] to show significance. We decided to use a non-parametric test since it does not assume any specific distribution over the ranks. Furthermore, the test is designed specifically to show statistical significance for paired ranking data. Bortz and Schuster [BS10] make the same argument for using the Wilcoxon signed-rank test. The statistical test will be done with $\alpha = 0.05$. The *null-hypothesis* H_0 and the *alternative hypothesis* H_1 will be defined in 5.2.1 "User experiment".

Wilcoxon signed-rank test To perform this test, one needs to rank the data from the experiment.

Given n data points with paired, ordinal conditions x_i and y_i, $1 <= i <= n$. First, the differences D_i of these conditions are computed, as well as the *signs* S_i of these differences. Then, the data is ordered by the absolute differences $|D_i|$ and the *rank* R_i is assigned. In case several data points have a same values for $|D_i|$, the *mean* value of their ranks is assigned to all of these as the new rank.

Afterwards, two values W_+ and W_- are computed. W_+ is the *sum* of all ranks where $S_i = 1$, and W_- is the *sum* of all ranks where $S_i = -1$. Data points with $D_i = 0$ are eqally assigned to these two groups, i.e. one half gets assigned $S_i = 1$ and the other half $S_i = -1$.

$$W_+ = \sum_{i=1}^{n} I(S_i = 1) \cdot R_i \tag{4.1}$$

$$W_- = \sum_{i=1}^{n} I(S_i = -1) \cdot R_i \tag{4.2}$$

I is the *indicator function*[4].

Finally, W is computed as the *minimum* of W_+ and W_-. W is then used to determine whether to reject the null hypothesis H_0 or not. To reject H_0 and accept

[4]see `https://en.wikipedia.org/wiki/Indicator_function` for details

the alternative hypothesis H_1, W has to be less than a critical value. These critical values are precomputed for small n.

$$W = min(W_+, W_-) \tag{4.3}$$

α, p-value, β and power

When executing statistical hypothesis testing, two general types of errors could occur [BS10].

Type I errors, or *false positive*, happen when H_0 is true but is rejected. The tolerated probability of Type I errors is denoted as α. The actual probability of falsely rejecting H_0 is the *p-value* generated by the statistical test. H_0 is rejected and H_1 is accepted if $p <= \alpha$.

Type II errors, or *false negative*, happen when H_0 is false but is not rejected. The probability of Type II errors is denoted as β. The complementary probability $1 - \beta$ is called *power*.

Execution

The test will be done using available functions in R[5] (wilcox.test[6]) or GNU Octave[7] (wilcoxon_test[8]).

4.3.4. Considerations

Furthermore, the statistical test has to be able to deal with relatively small sample sizes. Wilcoxon provided a lookup table for *p-values* of 0.055 or less starting with just 7 samples. A small sample size is sufficient to reject H_0 if there is statistically significant evedience to do so.

Increasing the sample size lowers the probability of *Type II errors* (or increases the *power* of the test).

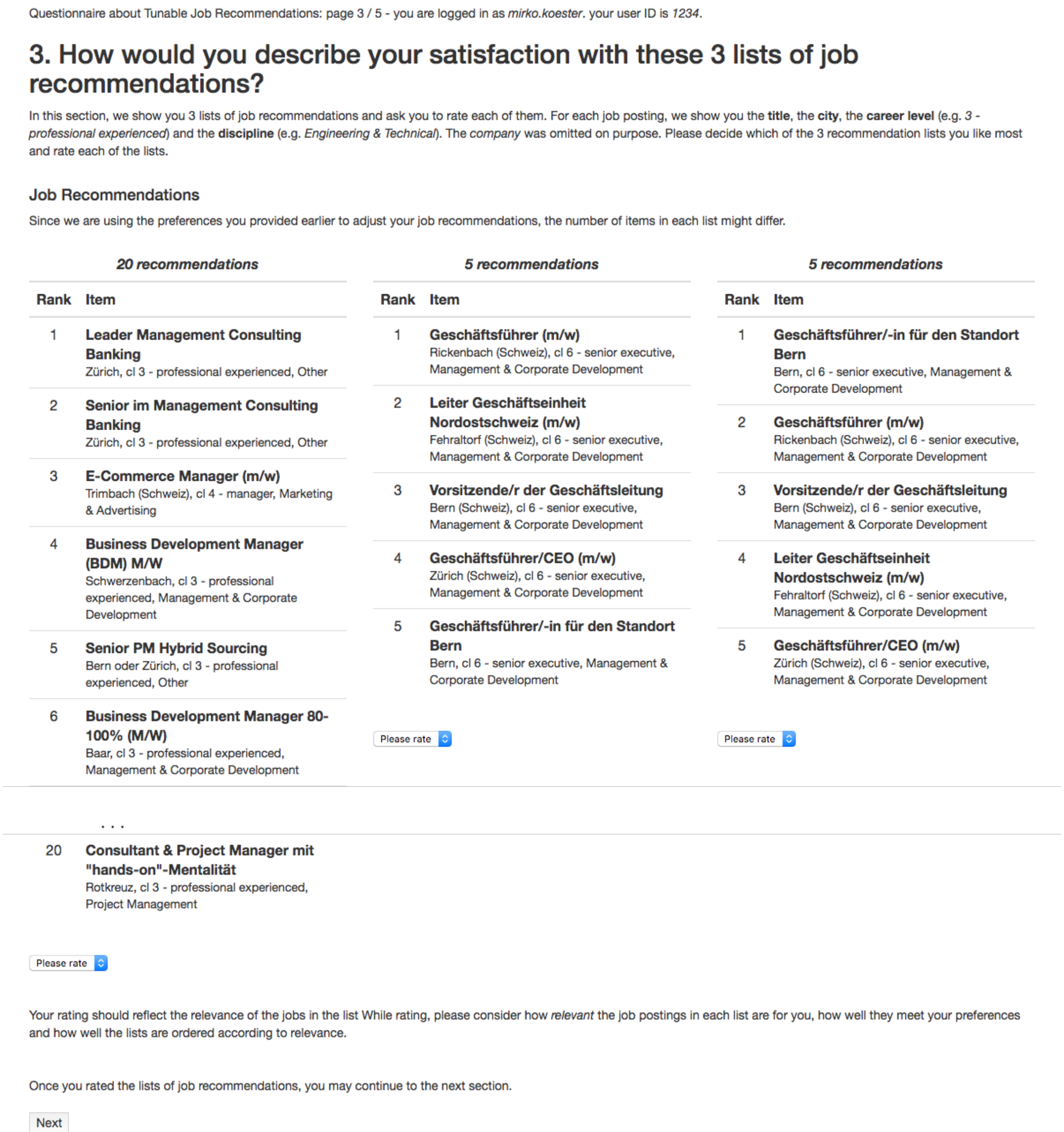
The number of *ties* (this is, the number of participants who rated both lists equally) may also be a factor. While the test itself can handle these ties, having many of them might already indicate that there is no significant difference in the performance of both methods.

Since the prototype of the job recommender is only available inside of XING's internal network and is not available to the public, the user study will be conducted asking XING's employees to participate. This most likely leads to a *bias*, since

[5] see `https://www.r-project.org/` for details
[6] see `https://stat.ethz.ch/R-manual/R-devel/library/stats/html/wilcox.test.html` for details
[7] see `https://www.gnu.org/software/octave/` for details
[8] see `https://www.gnu.org/software/octave/doc/interpreter/Tests.html` for details

XING’s employees are not representative for its userbase, as already mentioned in section 3.3.1 "Methodology". This has to be taken into account when drawing conclusions from the user study’s data as it might not be generalizable to the whole platform.

As noted in section 4.3.1 "User interface for the evaluation", both steps 3 and 4 omit the company of the job postings. While browsing the XING platform and looking at job recommendations, the company (and its logo) is prominently shown (see Figure 2.1). But as the user has no way of influencing the company, we decided not to show it in the questionnaire. This is done to avoid some bias, because the participant’s rating should be based on factors he could influence. Displaying the company might lead to a different rating in case the participant likes or dislikes a given company.



Questionnaire about Tunable Job Recommendations: page 3 / 5 - you are logged in as *mirko.koester*. your user ID is *1234*.

3. How would you describe your satisfaction with these 3 lists of job recommendations?

In this section, we show you 3 lists of job recommendations and ask you to rate each of them. For each job posting, we show you the **title**, the **city**, the **career level** (e.g. *3 - professional experienced*) and the **discipline** (e.g. *Engineering & Technical*). The *company* was omitted on purpose. Please decide which of the 3 recommendation lists you like most and rate each of the lists.

Job Recommendations

Since we are using the preferences you provided earlier to adjust your job recommendations, the number of items in each list might differ.

20 recommendations

Rank	Item
1	**Leader Management Consulting Banking** Zürich, cl 3 - professional experienced, Other
2	**Senior im Management Consulting Banking** Zürich, cl 3 - professional experienced, Other
3	**E-Commerce Manager (m/w)** Trimbach (Schweiz), cl 4 - manager, Marketing & Advertising
4	**Business Development Manager (BDM) M/W** Schwerzenbach, cl 3 - professional experienced, Management & Corporate Development
5	**Senior PM Hybrid Sourcing** Bern oder Zürich, cl 3 - professional experienced, Other
6	**Business Development Manager 80-100% (M/W)** Baar, cl 3 - professional experienced, Management & Corporate Development
	. . .
20	**Consultant & Project Manager mit "hands-on"-Mentalität** Rotkreuz, cl 3 - professional experienced, Project Management

Please rate

5 recommendations

Rank	Item
1	**Geschäftsführer (m/w)** Rickenbach (Schweiz), cl 6 - senior executive, Management & Corporate Development
2	**Leiter Geschäftseinheit Nordostschweiz (m/w)** Fehraltorf (Schweiz), cl 6 - senior executive, Management & Corporate Development
3	**Vorsitzende/r der Geschäftsleitung** Bern (Schweiz), cl 6 - senior executive, Management & Corporate Development
4	**Geschäftsführer/CEO (m/w)** Zürich (Schweiz), cl 6 - senior executive, Management & Corporate Development
5	**Geschäftsführer/-in für den Standort Bern** Bern, cl 6 - senior executive, Management & Corporate Development

Please rate

5 recommendations

Rank	Item
1	**Geschäftsführer/-in für den Standort Bern** Bern, cl 6 - senior executive, Management & Corporate Development
2	**Geschäftsführer (m/w)** Rickenbach (Schweiz), cl 6 - senior executive, Management & Corporate Development
3	**Vorsitzende/r der Geschäftsleitung** Bern (Schweiz), cl 6 - senior executive, Management & Corporate Development
4	**Leiter Geschäftseinheit Nordostschweiz (m/w)** Fehraltorf (Schweiz), cl 6 - senior executive, Management & Corporate Development
5	**Geschäftsführer/CEO (m/w)** Zürich (Schweiz), cl 6 - senior executive, Management & Corporate Development

Please rate

Your rating should reflect the relevance of the jobs in the list While rating, please consider how *relevant* the job postings in each list are for you, how well they meet your preferences and how well the lists are ordered according to relevance.

Once you rated the lists of job recommendations, you may continue to the next section.

Next

Figure 4.4.: Questionnaire: Rate lists of recommendations

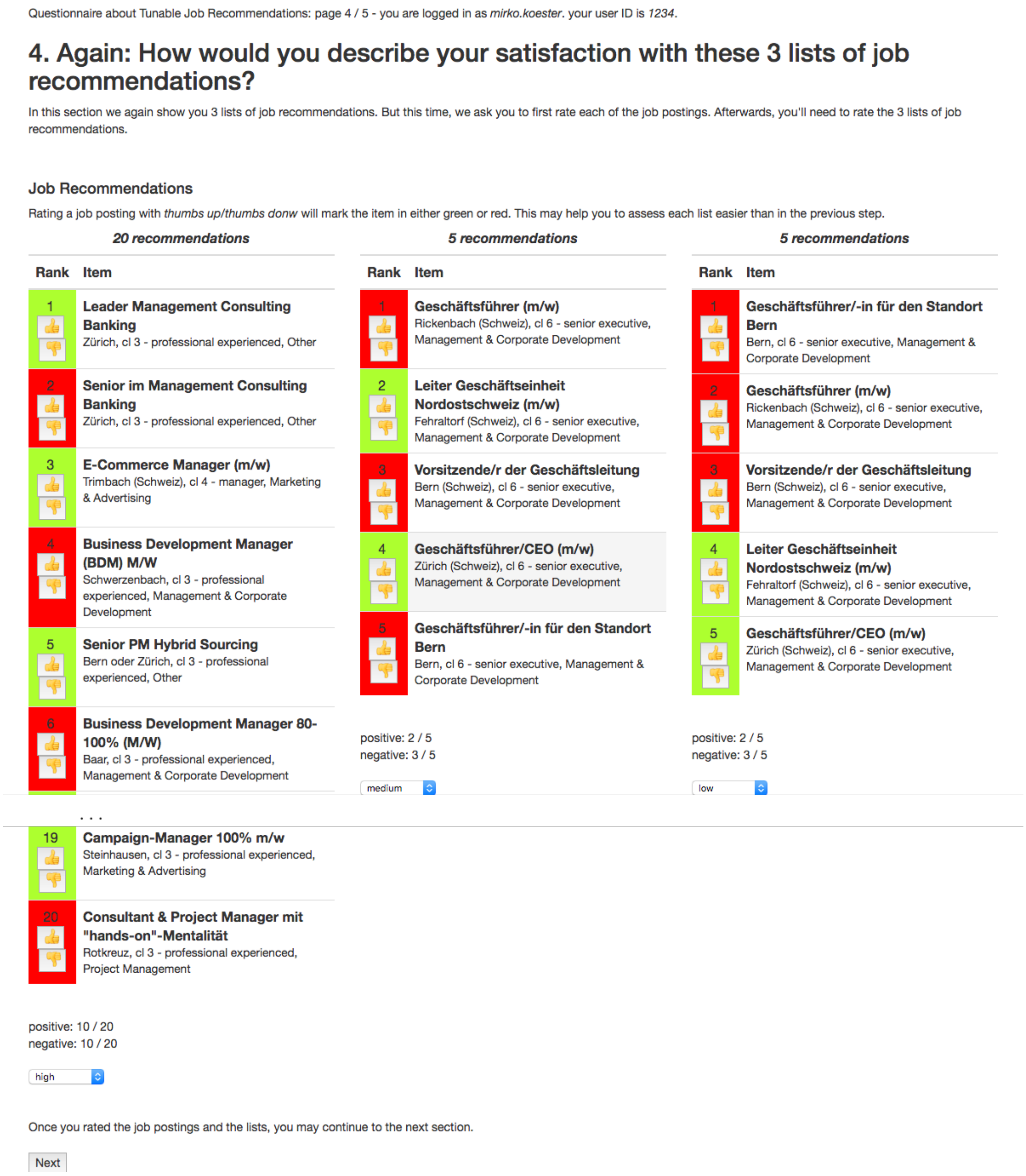


Figure 4.5.: Questionnaire: Rate lists of recommendations as well as individual job postings

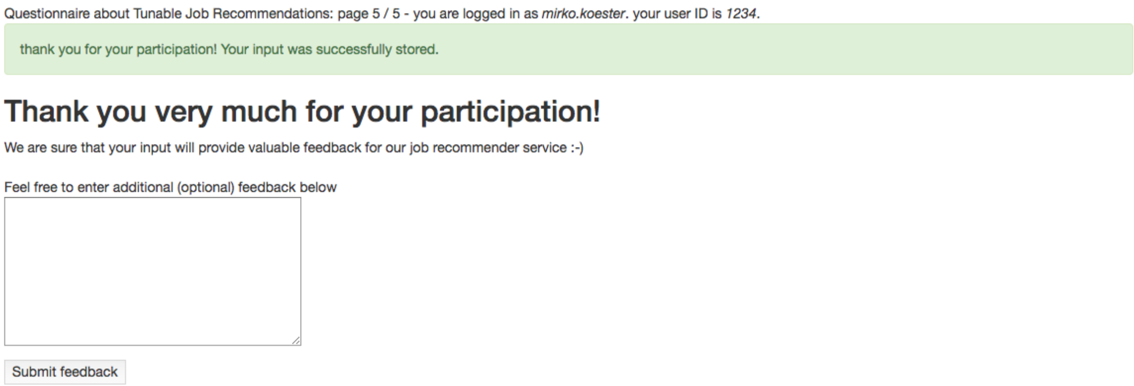


Figure 4.6.: Questionnaire: Thanks for your participation

5. Evaluation

This chapter details the evaluation of our interactive job recommender system. In Section 5.1, we report about the results of a large-scale quantitative analysis. The observations and results of our user study are reported and discussed in Section 5.2.

5.1. Quantitative analysis

We performed an offline evaluation to determine that we use adequate weigths for the *positive* and *negative* topics, as well as to measure the possible impact of the settings as described in Section 3.1.5 "Job Recommendations Settings Page".

Tables 5.1 to 5.4 show the results of this offline evaluation of the impact of reco settings. The evaluation was run four times with 2000 XING users. For each user, the jobs recommender was called twice. Once in the default way and once with customized settings. Each evaluation was run with different values and methods to generate the settings for the users. Afterwards the two result lists were compared for each user and the mean of the intersection of the top k recommendations (values for k were 4, 10, 20 and 50) over all users was computed.

Evaluation 1 used the top $\lceil 0.15 \cdot \#topics \rceil$ topics (see Section 3.1.5) as positive settings and (if we could suggest at least 3 toppics) the bottom $\lceil 0.15 \cdot \#topics \rceil$ topics as negative settings. Evaluation 2 used $\lceil 0.33 \cdot \#topics \rceil$ and $\lceil 0.20 \cdot \#topics \rceil$ respectively. Evaluation 3 used the same percentages, but chose the topics randomly. Evaluation 4 additionally used the origins (i.e. which backends as described in Section 2.2.4 contributed to each topic) of the topics to calculate the impact of each sub-recommender.

Tables 5.1 to 5.4 represent evaluations 1 to 4. The 4 evaluations show similar behavior. The values of the mean intersection (%) for the top $k = 4$ are between 39.8 and 42.4, meaning that on average and over all 4 offline evaluations more than half of the top 4 recommendtaions are replaced when settings are used. Those values are between 52.1 and 54.6 for the top $k = 10$, between 65.1 and 66.6 for the top $k = 20$ and between 69.0 and 71.2 for the top $k = 50$ (lower values mean more new items in the top k when settings are used).

These evaluations say nothing about the impact on the quality of job recommendations when using settings, as the settings where generated arbitrarily. But they show that using settings have an impact on recommendations, so using set-

Top K	Intersection (%) mean
4	42.4
10	54.6
20	66.6
50	71.2

Table 5.1.: Offline evaluation of settings impact (1)

Top K	Intersection (%) mean
4	39.8
10	52.1
20	65.1
50	69.0

Table 5.2.: Offline evaluation of settings impact (2)

Top K	Intersection (%) mean
4	40.7
10	53.6
20	65.1
50	69.3

Table 5.3.: Offline evaluation of settings impact (3)

Top K	Intersection (%) mean
4	42.4
10	54.5
20	65.5
50	69.6

Table 5.4.: Offline evaluation of settings impact (4)

tings would actually make a difference for XING's users. XING would be able to offer the personalized settings (i.e. topics) to every user who already receives job recommendations.

5.2. Qualitative analysis

5.2.1. User experiment

The questionnaire designed in Section 3.3.3 "User study" and implemented as described in Section 4.3 "Implementation of the evaluation framework" allows for analyzing the settings of the participants as well as the settings' impact on job recommendations, including

- distribution and average of ratings for the topics list
- do participants who rate their original list of recommendations *negatively* also rate their list of topics *negatively*?
- How many users where able to select k topics *positively*?
- impact of 'max. distance in km' on number of recommendations
- distribution of ratings for the 3 lists of recommendations
  - initial ratings
  - ratings after rating the individual job postings

- do the ratings of the 3 lists of recommendations correlate with the ratings of the individual job postings?
  - looking at the *top K* (with $K \in \{2, 6, 20\}$[1]) recommendations, is the ration of *positively* and *negatively* rated job postings in each list reflected in the rating of each list?

We had 58 participants. Three of them did not finish the questionnaire – they stopped after they rated the three lists for the first time (after step 3). This means, that we have data about the topics (list of topics as well as their settings) and how they rated the three lists of recommendations initially from 58 participants. 55 participants finished the whole questionnaire. The data can be found in Appendix A. And 10 people left a comment regarding the questionnaire. The comments can be found in Appendix B.

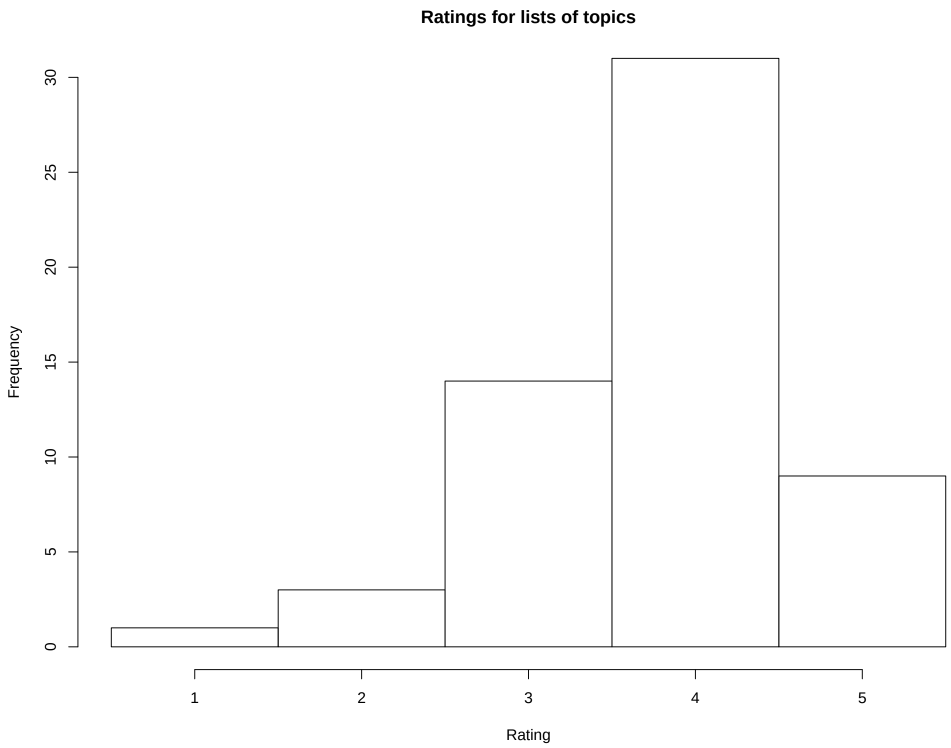
In the following, we present our analysis of that data. Graphs showing ratings (*very low* to *very high*) use the number 1 to 5 to represent these ratings.

Topics

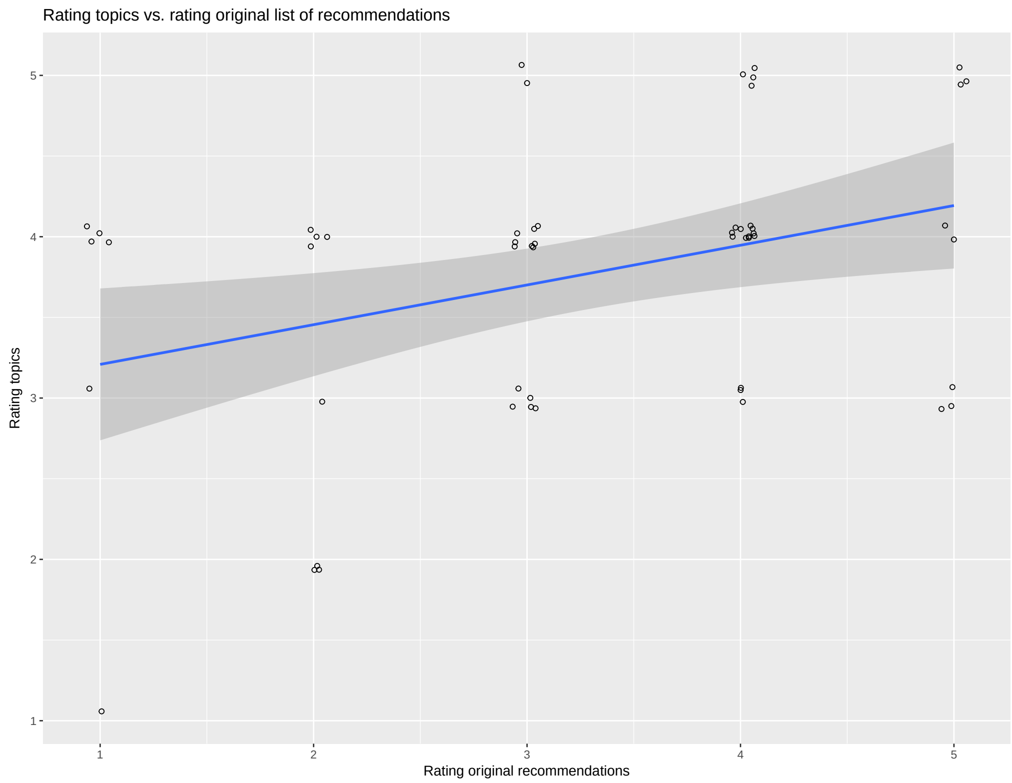
Looking at the ratings on the lists of topics by the initial 58 participants (cf. Figure 5.1a), they rather liked their suggested topics with a *median* of *high*. 9 rated the list as *very high*, 31 as *high*. Just one participant did not like the list at all, and three rated it as *low*. The remaining 14 rated their list as *medium*.

Figure 5.1b shows the correlation between how participants rated their list of original recommendations and how they rated their list of topics. The two are weakly positively correlated (≈ 0.35). Since the list of topics is generated from the list of original recommendations, we expected a higher correlation. But it seems that quite some users still like their topics, even if they do not like their list of recommendations. We conclude, that our approach of generating the topics works rather well and represents the user.

[1] These are the most common numbers of job recommendations shown on the XING platform: 2 recommendations are shown on the startpage, 6 and 20 recommendations are presented in the jobs section.



(a) Ratings - list of topics



(b) Correlation between original recommendations and topics

Figure 5.1.: Topics

Ratings

In this section, we analyse how participants rated their lists of recommendations.

Table 5.5.: Initial ratings

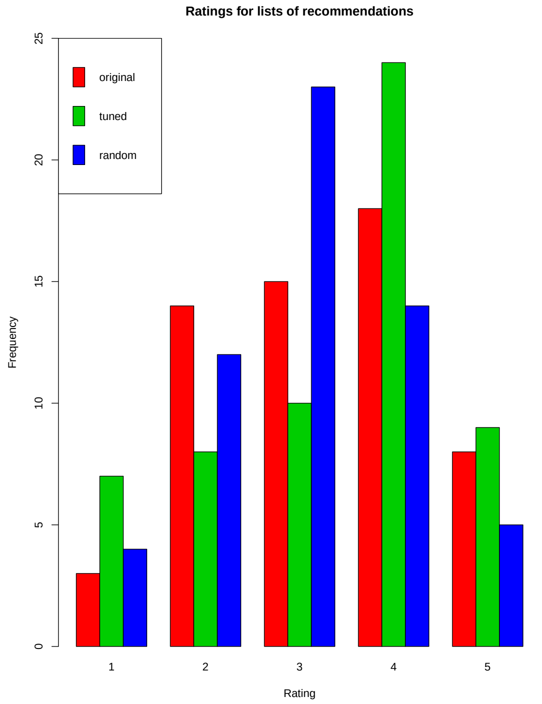
	very low	low	medium	high	very high
original	3	14	15	18	8
tuned	7	8	10	24	9
random	4	12	23	14	5

Table 5.6.: Ratings in step 4

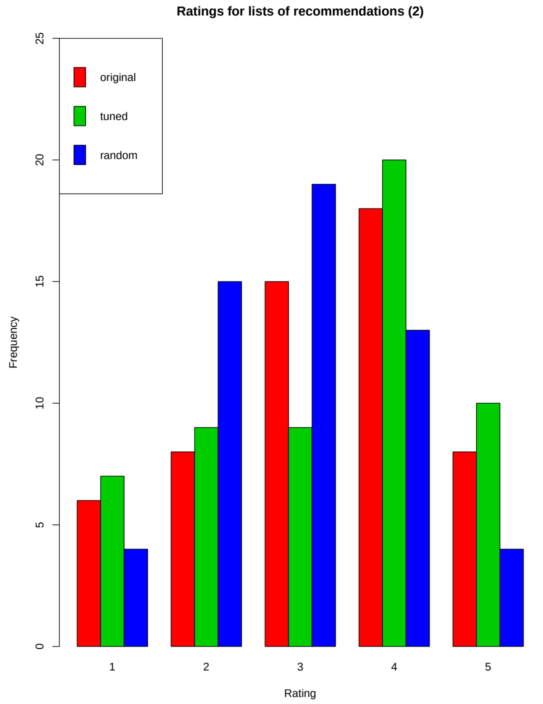
	very low	low	medium	high	very high
original	6	8	15	18	8
tuned	7	9	9	20	10
random	4	15	19	13	4

The participants were asked to rate their lists of recommendations twice. Once in step 3 and once in step 4, after they rated the individual job postings. Table 5.5 and Figure 5.2a show the initial ratings. The distribution of ratings on random is almost bell-shaped, while the other two distributions are skewed towards higher ratings. Table 5.6 and Figure 5.2b show the ratings from step 4. The distributions look similar. As Figure 5.4a shows, most participants rated the corresponding lists the same in both steps (33/55 or 60% for original, 36/55 or $\approx 65\%$ for tuned and 28/55 or $\approx 51\%$ for random). For the list of original recommendations, 12 people chose to increase their rating by 1, 2 decreased their rating by 2 and 8 decreased it by 1. For the list of tuned recommendations, 7 and 3 people chose to increase their rating by 1 and 2 respectively, 2 decreased their rating by 2 and 7 decreased it by 1. For the list of tuned recommendations, 18 people chose to increase their rating by 1, 1 decreased his rating by 2 and 8 decreased it by 1. Since the participants changed their initial ratings between $\approx 35\%$ and $\approx 49\%$ of the times, we conclude that rating a list (or even three of them) is not an easy task for humans and offering them some visual aid in the form of rating and highlighting individual job postings was beneficial.

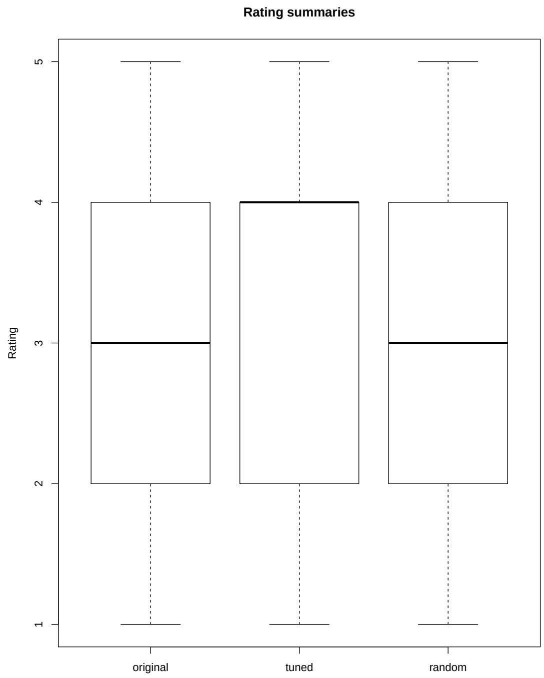
From this point on we only use the ratings from step 4.



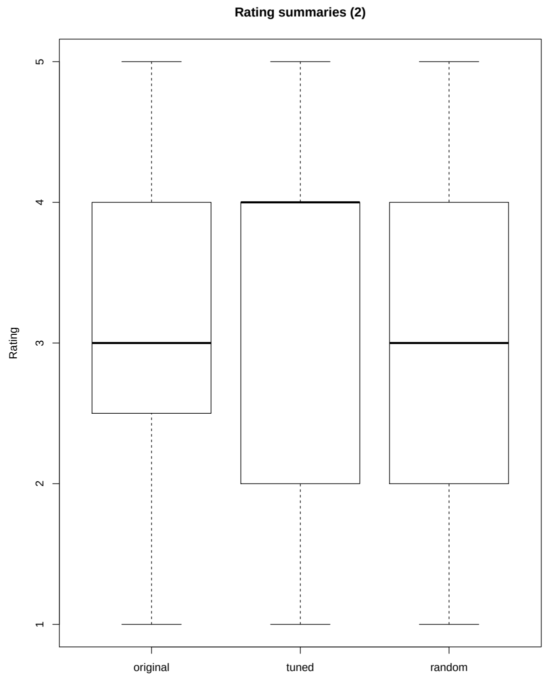
(a) Ratings



(b) Ratings (2)

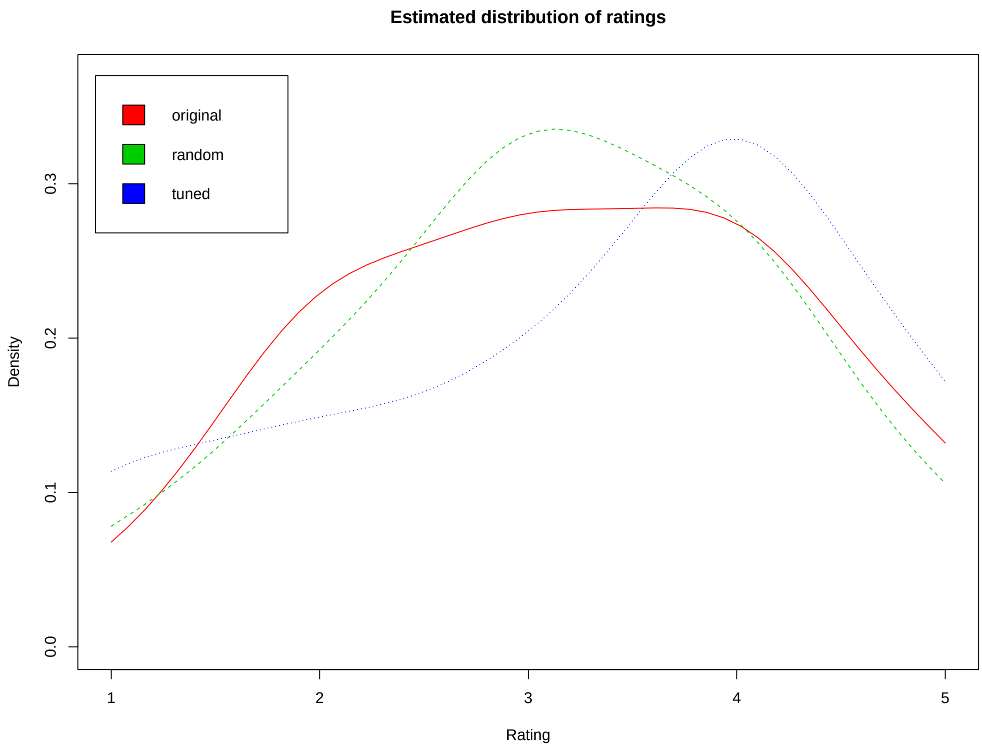


(c) Ratings - summaries

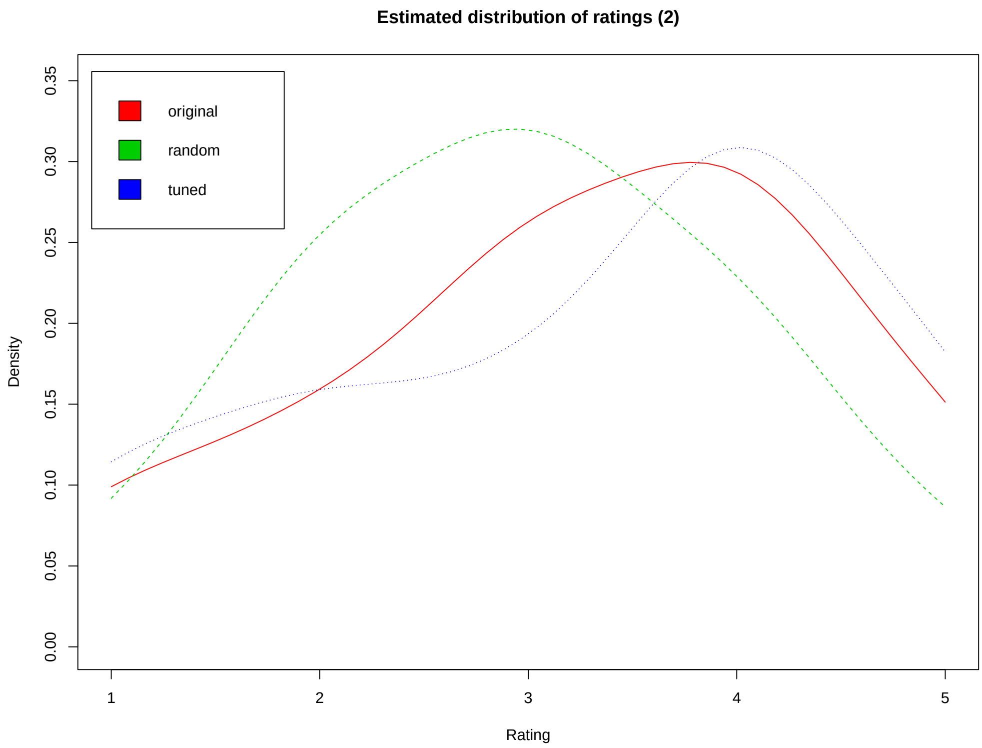


(d) Ratings - summaries (2)

Figure 5.2.: Ratings for lists of recommendations

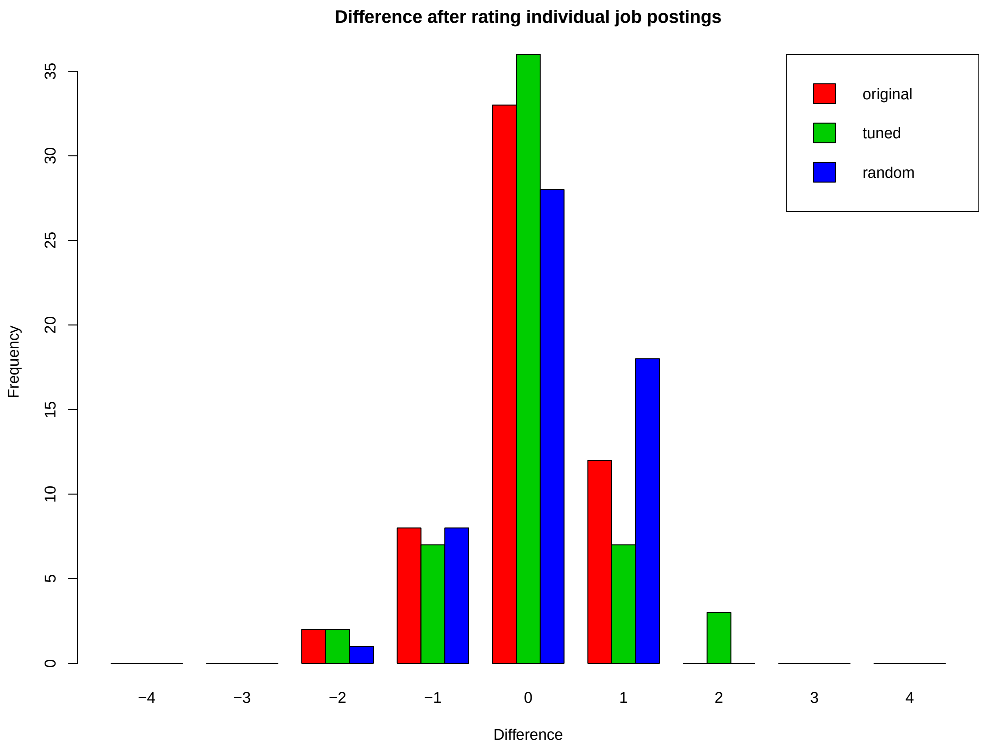


(a) Ratings (cf. Figure 5.2a)

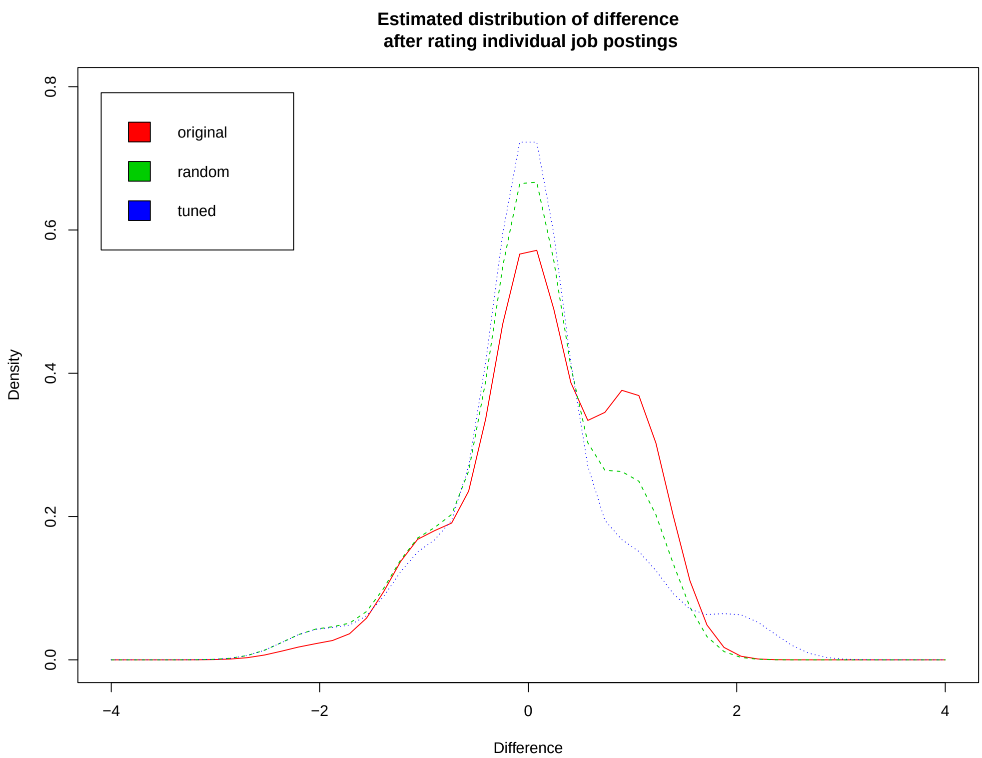


(b) Ratings (2) (cf. Figure 5.2b)

Figure 5.3.: Estimated distribution of ratings



(a)



(b)

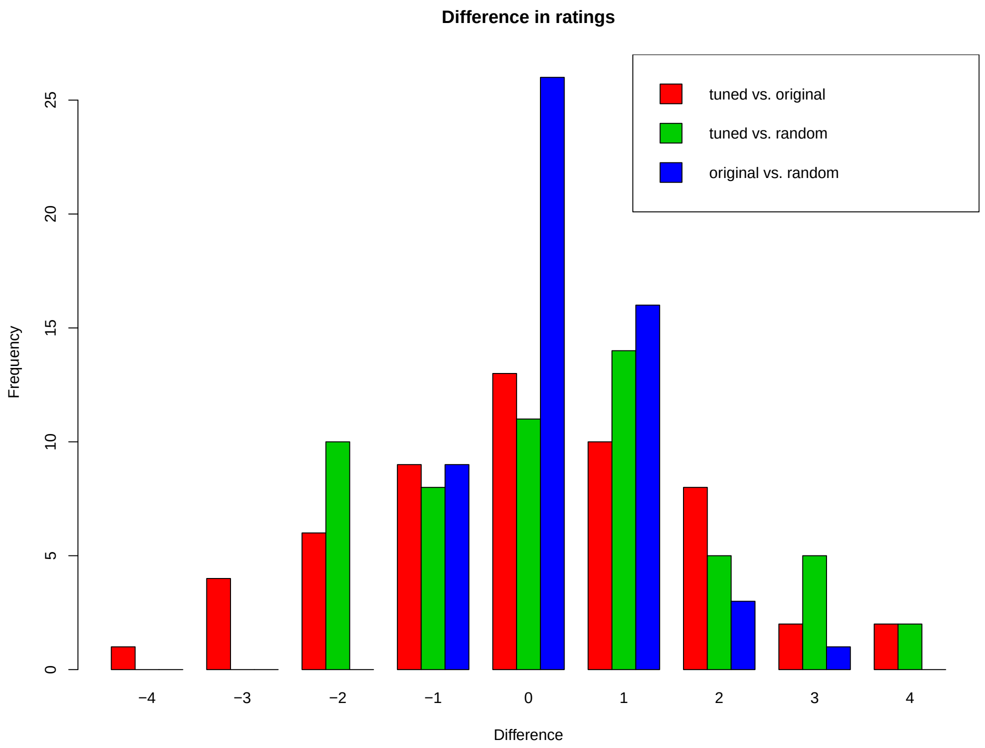
Figure 5.4.: Differences after rating individual items

Next, we are going to look at how the 3 lists of recommendations performed in comparison. The test for statistical significance will be done in Section 5.2.1 "Statistical hypothesis testing".

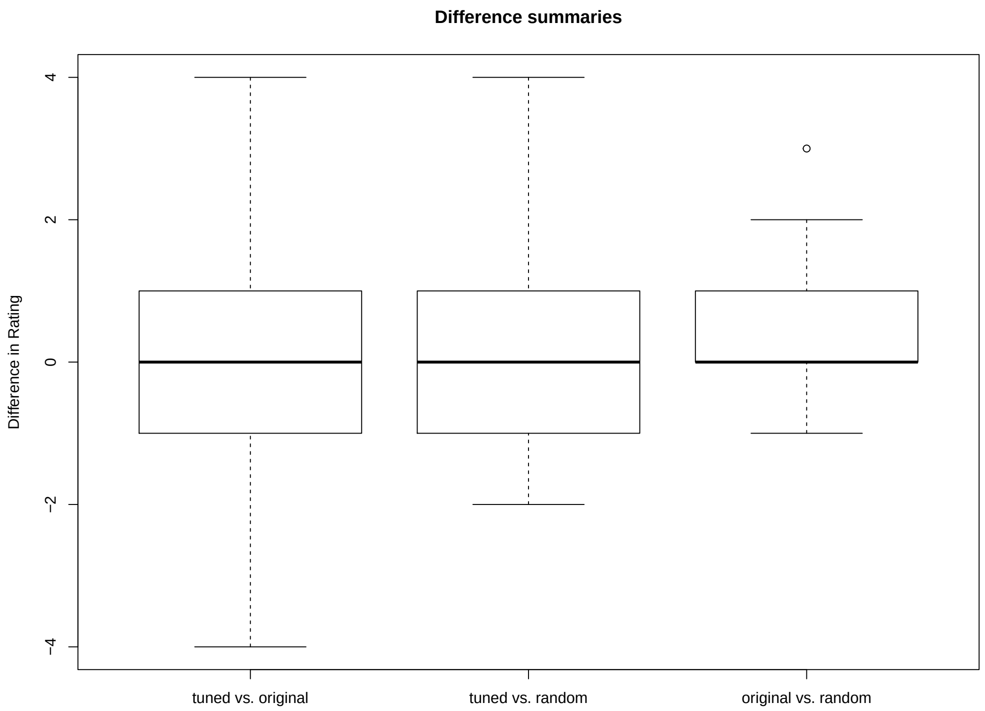
Table 5.7.: Performance comparison of the 3 lists of recommendations

	-4	-3	-2	-1	0	1	2	3	4
tuned vs. original	1	4	6	9	13	10	8	2	2
tuned vs. random	0	0	10	8	11	14	5	5	2
original vs. random	0	0	0	9	26	16	3	1	0

Table 5.7 and Figure 5.5a show these comparisons. First, we look at the *random list*, which served as a control group. *Original. vs. random* is skewed to the right, which means that the *original list* performed slightly better than the *random list*. Only 9/55 people liked the random list better, while 20 people preferred the original list. *Tuned. vs. random* is slightly skewed to the right, which means that the *tuned list* performed a little better than the *random list*. 18/55 people liked the random list better, while 26 people preferred the tuned list.



(a) Rating Differences



(b) Rating Differences - summaries

Figure 5.5.: Differences in ratings

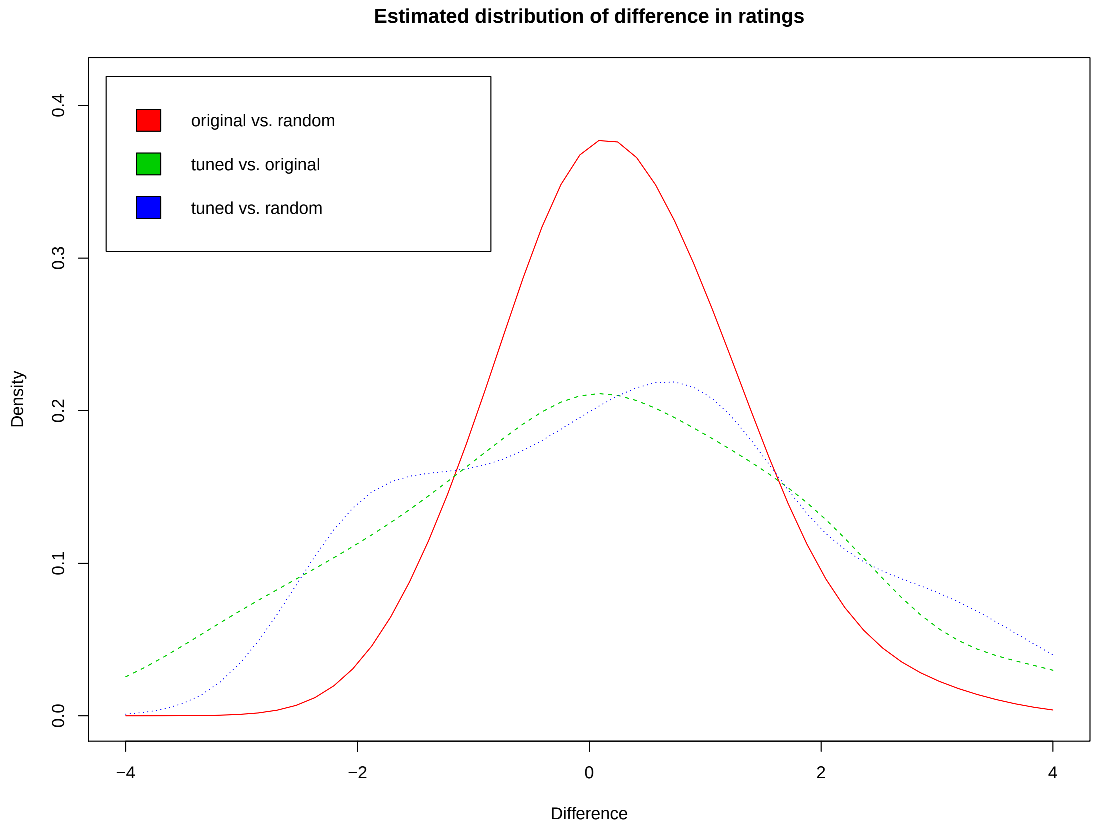


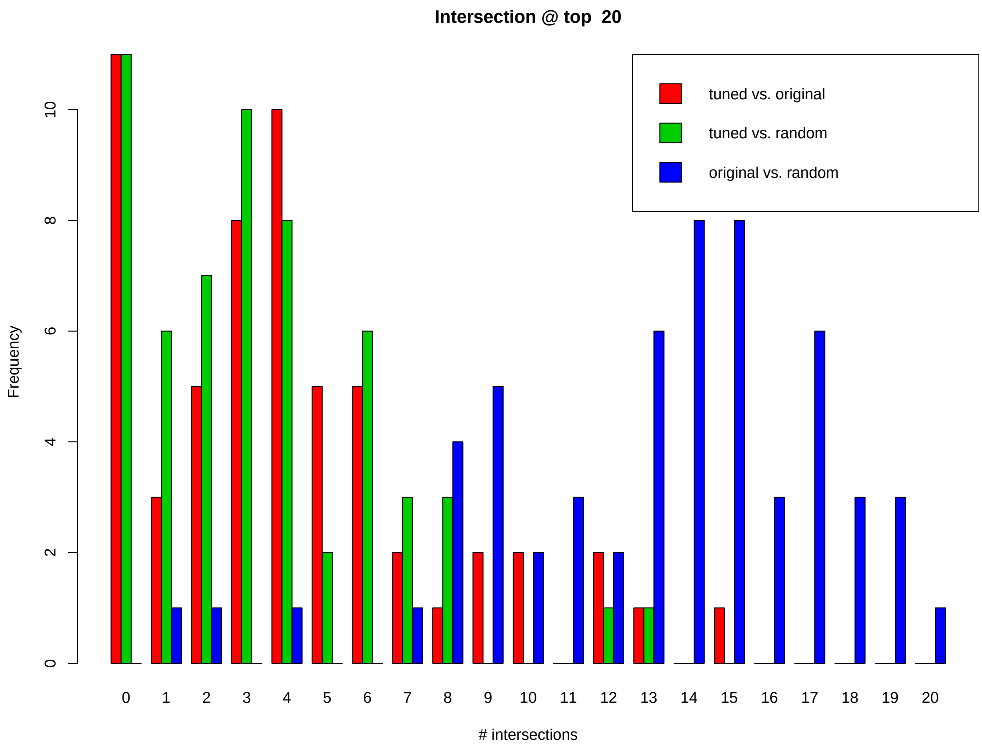
Figure 5.6.: Estimated distribution of differences (cf. Figure 5.5a)

Items

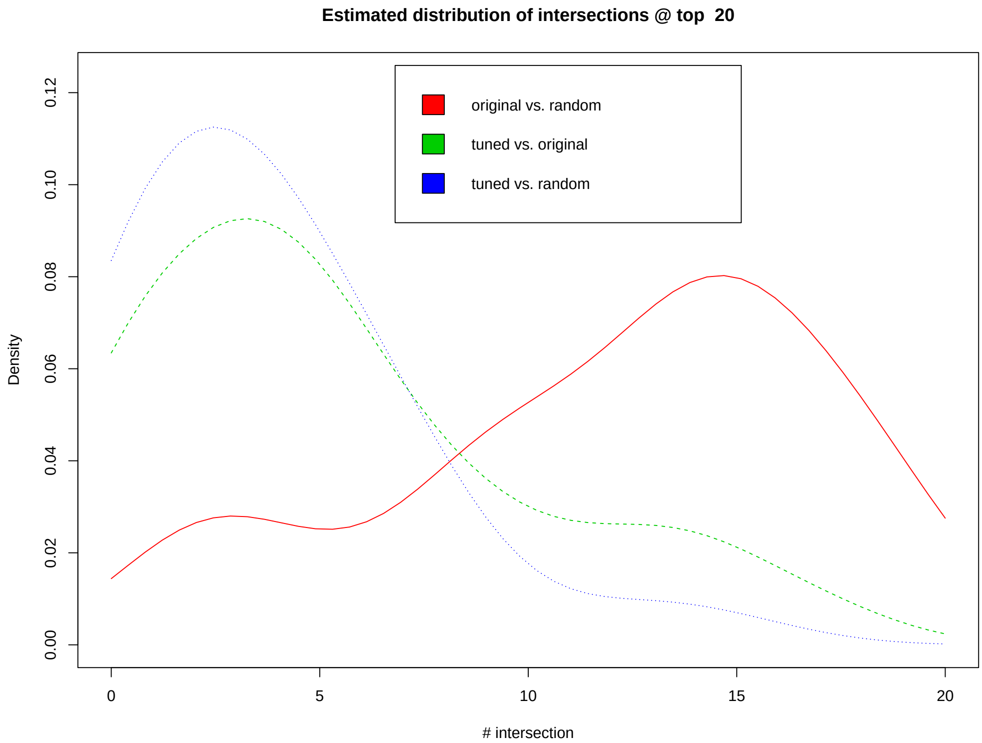
Next, we are looking at how much the users' lists of recomendations changed between the 3 methods. We describe the intersection in the top 20 as well as in the top 6.

Figure 5.7 shows that for the top 20 and for *original vs. random* the overlap is rather big, as the majority of users have an intersection of 10 to 20 items. Comparing the *tuned* method to the other two, one can see that the overlap is much smaller. Most users have an intersection of 0 to 10 items. This confirms our conclusion from the offline evaluation. The settings have a big impact of the recommendations.

Doing the same for the top 6, one can see in Figure 5.8 that the effect of *random* is stronger, meaning the intersection is smaller on average. Since *tuned* is clearly skewed to the left, i.e. there is little overlap in the lists of recommendations, the impact of settings on the head of the is also strong.



(a) Intersection



(b) Estimated distribution

Figure 5.7.: Intersection of lists of recommendations @ top 20

Now we look at how the ratio of positive rated items is distributed in the top 20 and top 6. First, for the top 20 (see Figure 5.9), we can see that all three distributions are somewhat bell-shaped, indicating that the three methods on average produce balanced lists of recommendations.

This is not the case for the top 6, as one can see in Figure 5.10. *Random* is still nearly bell shaped, but shifted to the left, which means the top 6 produced by this method is rather worse than with the other two methods, which are both skewed to the right. This is a good sign, since intuitively we expect any recommender system to put the best recommendations to the top of the list.

Finally, we analyse the correlation between the ratio of positive items and the ratings of the three lists of recommendations. Exemplary, we look at the *orignal* list of recommendations and the top 20 and top 6, since all other combinations showed very similar results. Figure 5.11 visualizes this strong correlation. This is expected, as we asked the participants to consider their ratings on individual postings when rating the whole list in step 4 of the questionnaire.

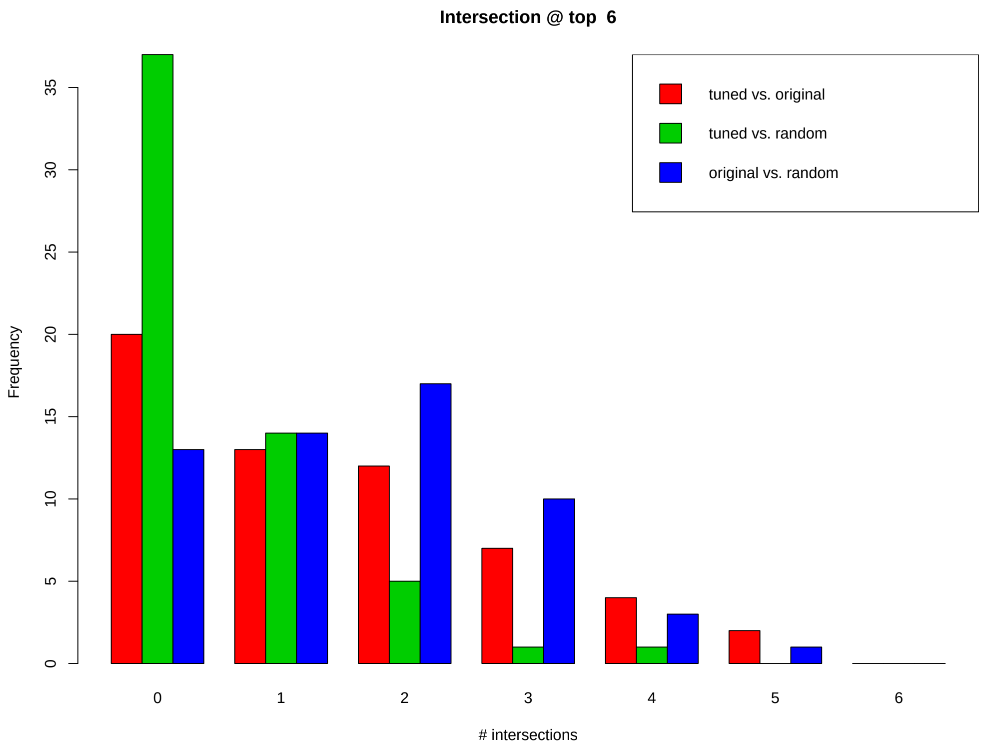
Settings / Topics

In this section, we investigate how users selected their settings. Figure 5.12a and Figure 5.12b tell us, that every participant selected at least 3 topics. The median participant selected 15 topics, while 50% of them selected between 11 and 18 topics. This tells us, that the way we generate these topics works well.

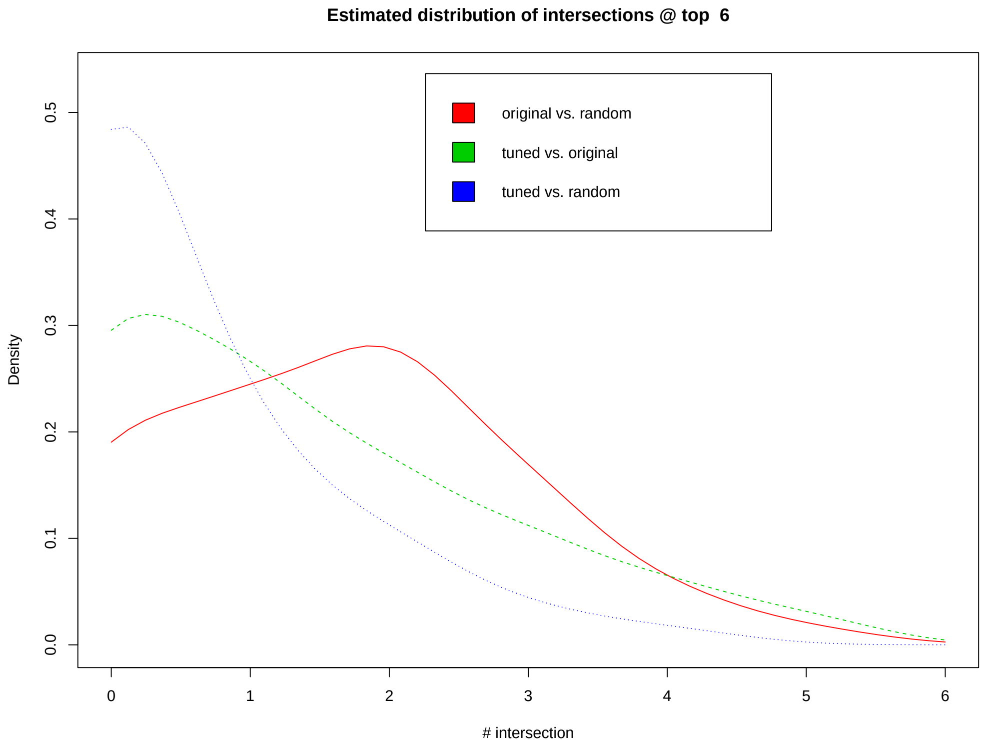
When split up by type (discipline, jobrole and skill) as depicted in Figure 5.12c and Figure 5.12d, we can see that most users selected exactly one discipline as well as one or two jobroles. This makes sense, since people are usually just interested in very few disciplines and jobroles. For skills, it is different, as people usually need a lot of skills in their professional life. The median participant selected 11 skills, while 50% of them selected between 8 and 14 skills.

When split up further by positive and negative selected settings (cf. Figure 5.13), we see that more than 50% of the participants marked all their selected disciplines and jobroles as positive. Furthermore, more than 75% marked at least one discipline and jobrole as positive. This is somewhat expected, as these topics are extracted from the users's job recommendations; and we showed earlier that most participants are rather happy with their recommendations, which only makes sense when the jobroles and disciplines match most of the time.

Skills on the other hand are marked more diverse. The average participant marked $\approx 52\%$ of the selected skills as positive, while 50% of them marked between $\approx 34\%$ and $\approx 71\%$ as positive.

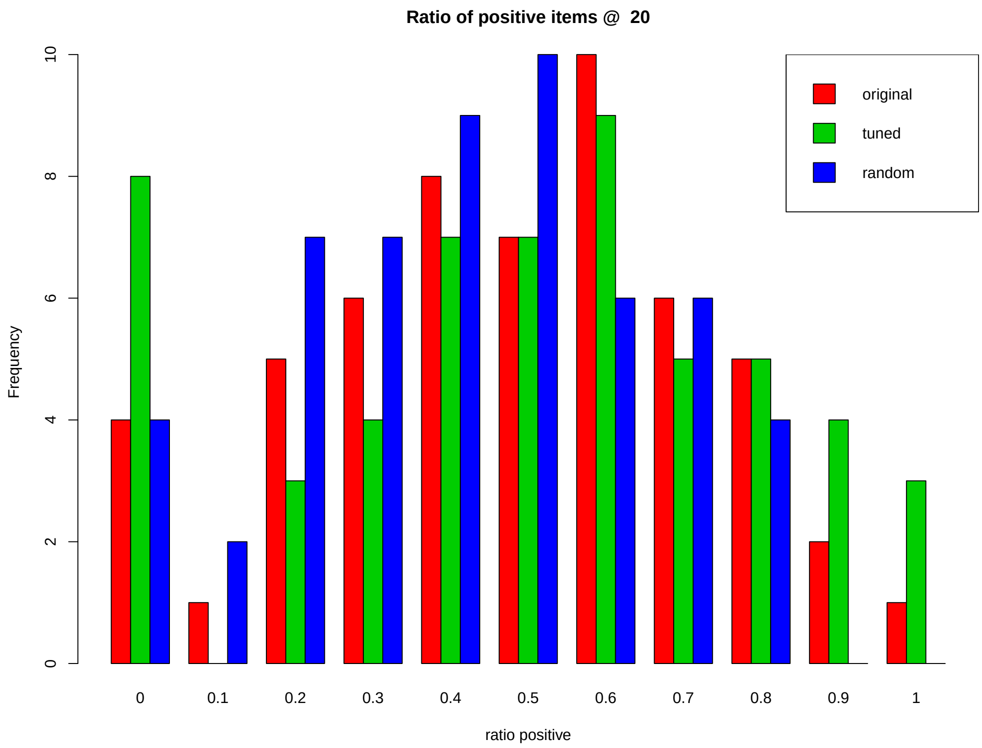


(a) Intersection

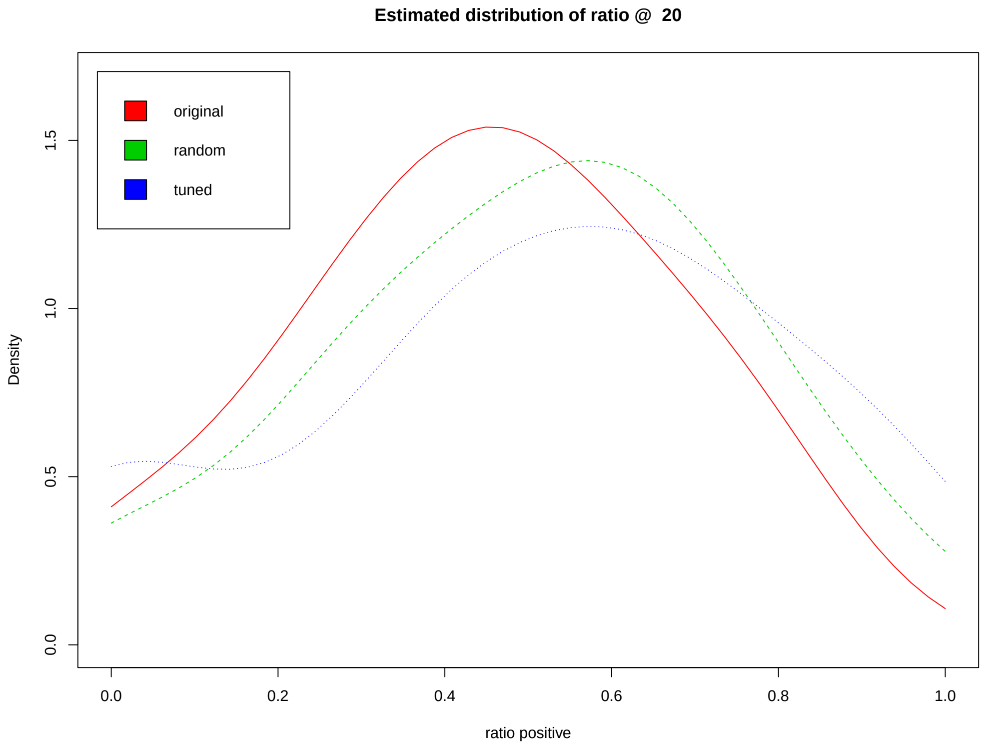


(b) Estimated distribution

Figure 5.8.: Intersection of lists of recommendations @ top 6

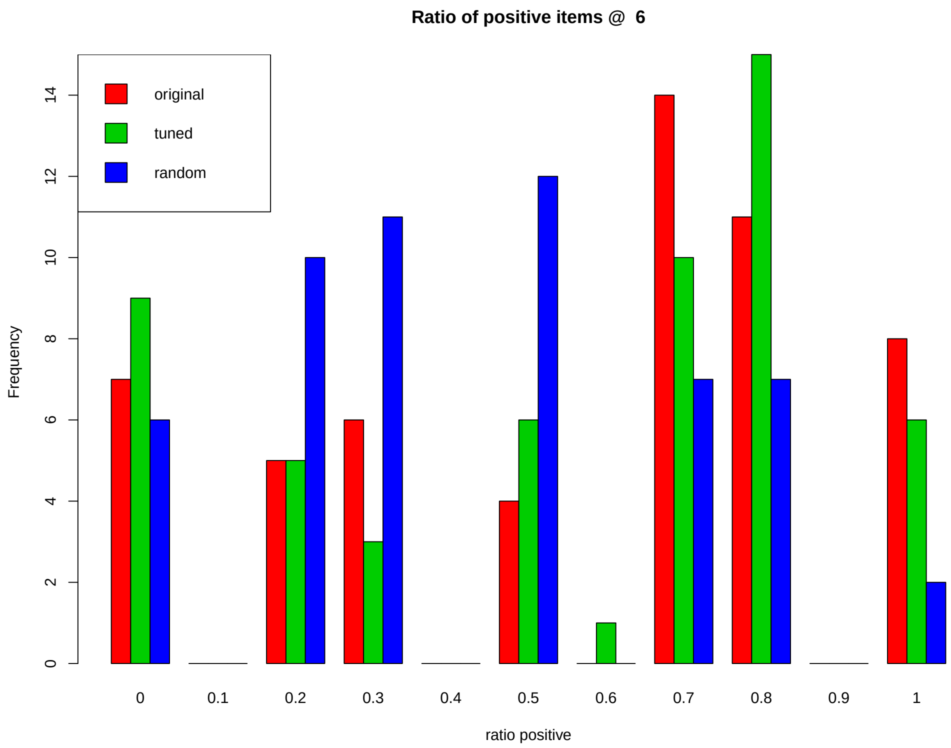


(a) Ratio of positive items

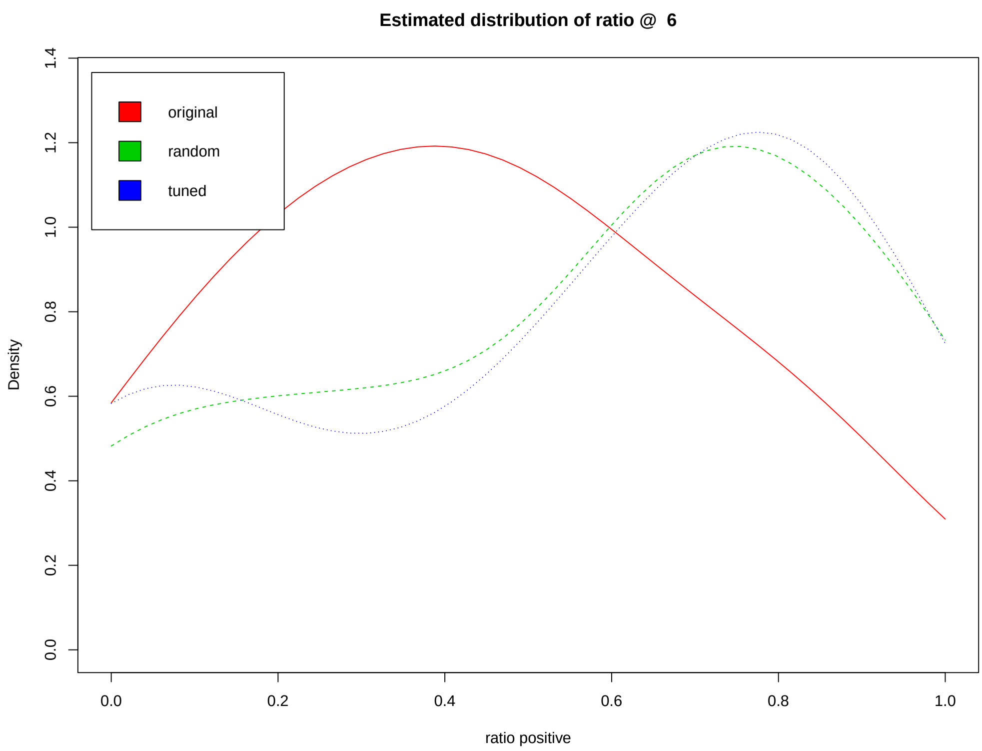


(b) Ratio of positive items

Figure 5.9.: Ratio of positive items @ top 20

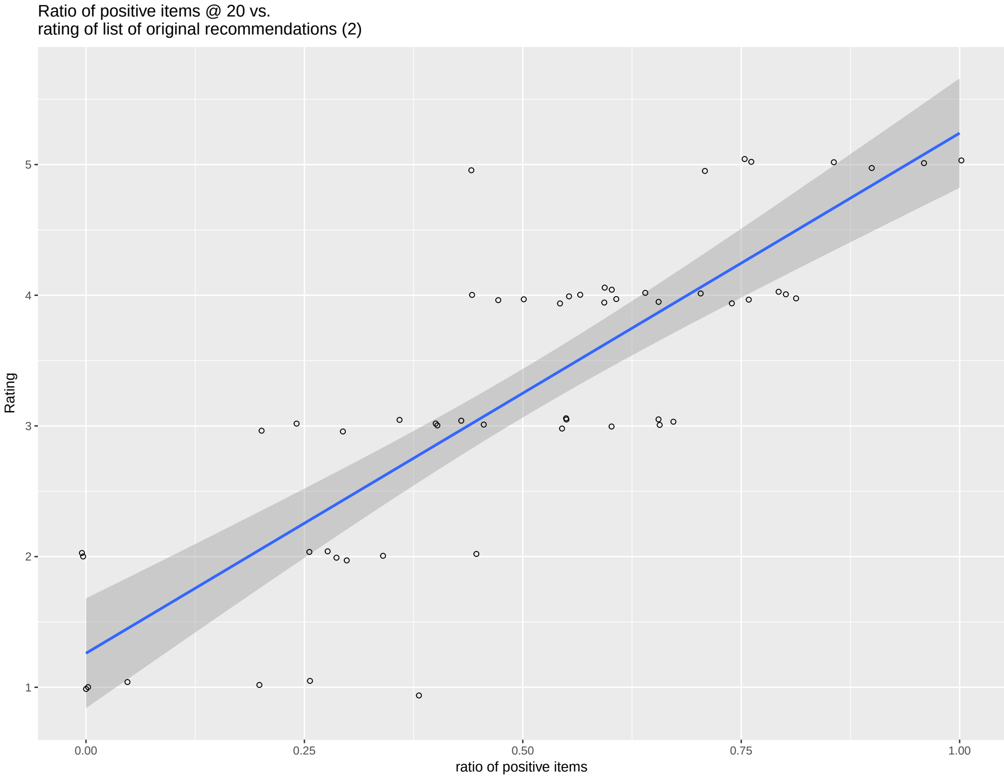


(a) Ratio of positive items @ top 6

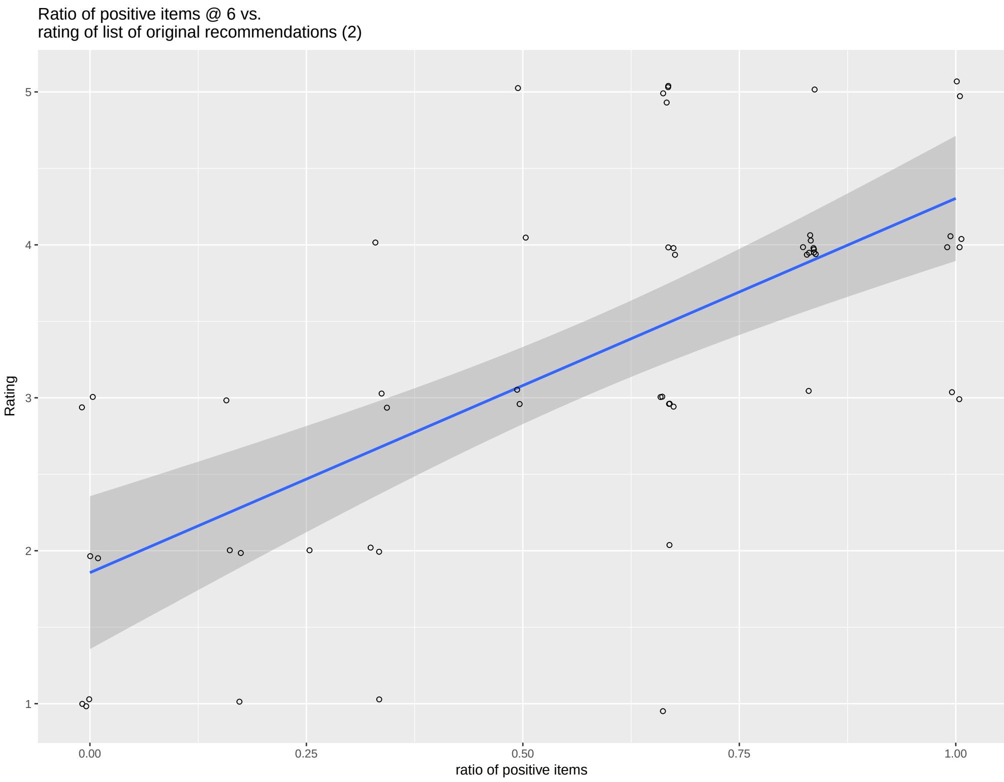


(b) Ratio of positive items

Figure 5.10.: Ratio of positive items @ top 6

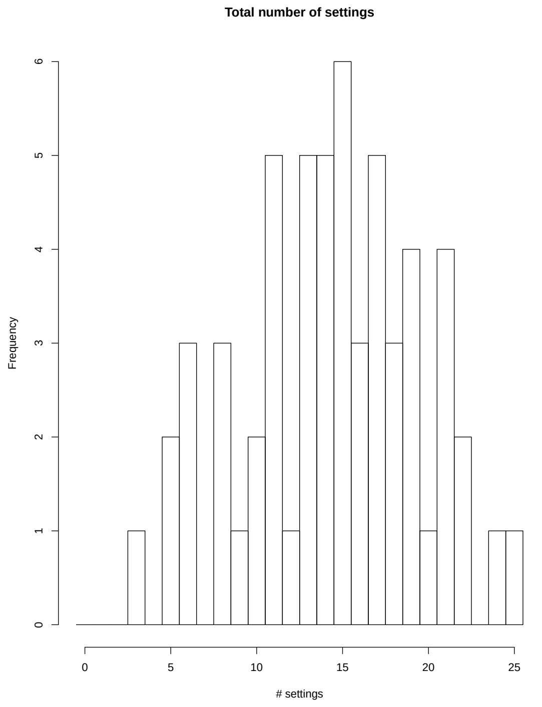


(a) Correlation @ top 20

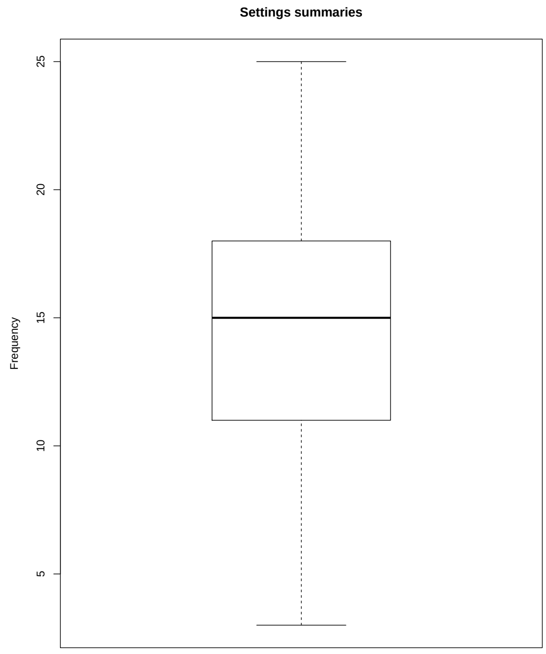


(b) Correlation @ top 6

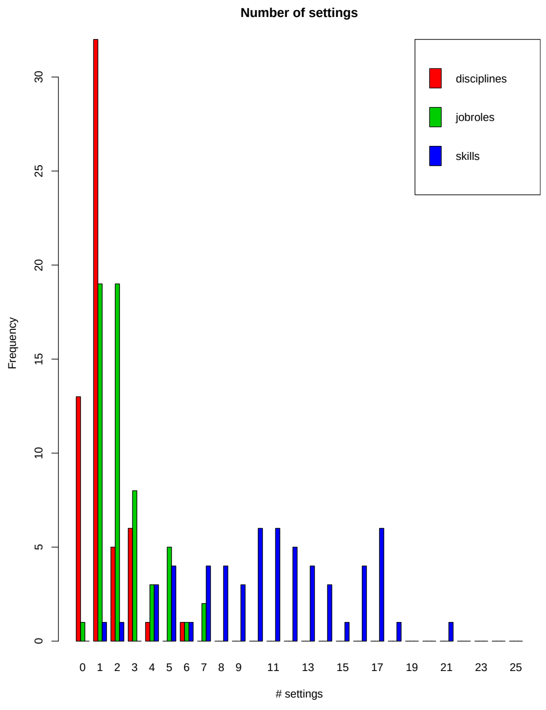
Figure 5.11.: Correlation between rating of list of recommendations & ratings of individual job postings



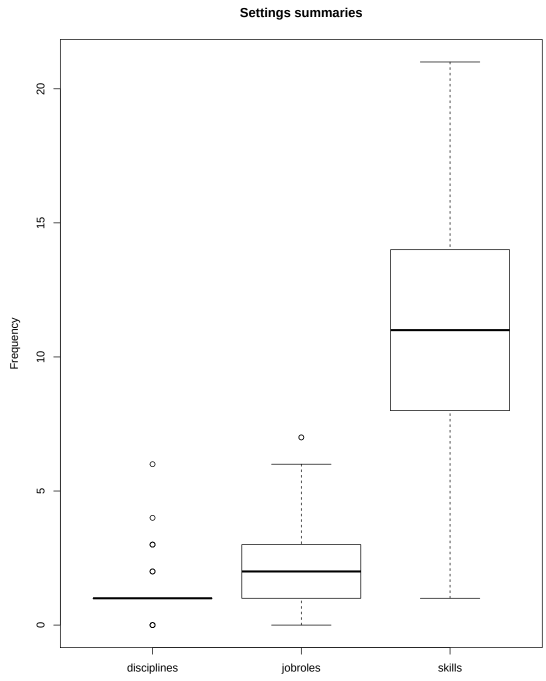
(a) Number of settings per user



(b) Number of settings per user

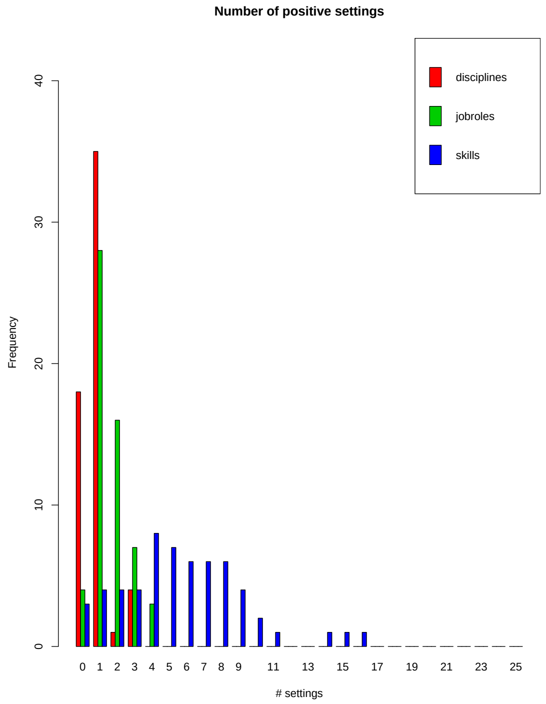


(c) Number of settings per user (per type)

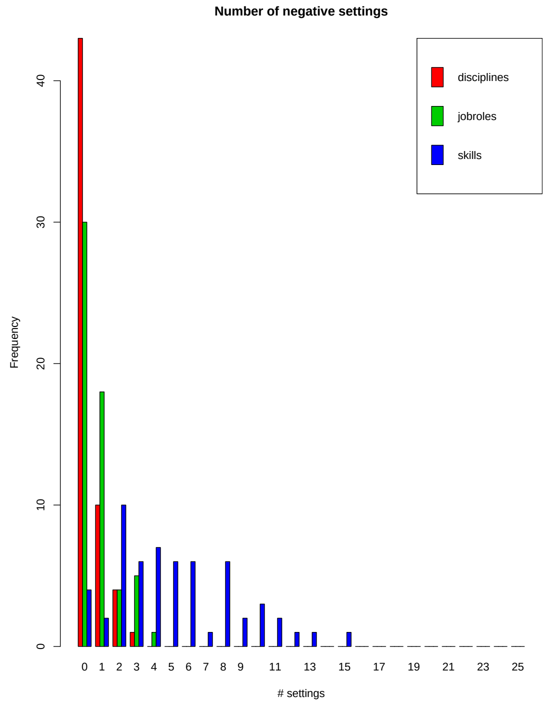


(d) Number of settings per user (per type)

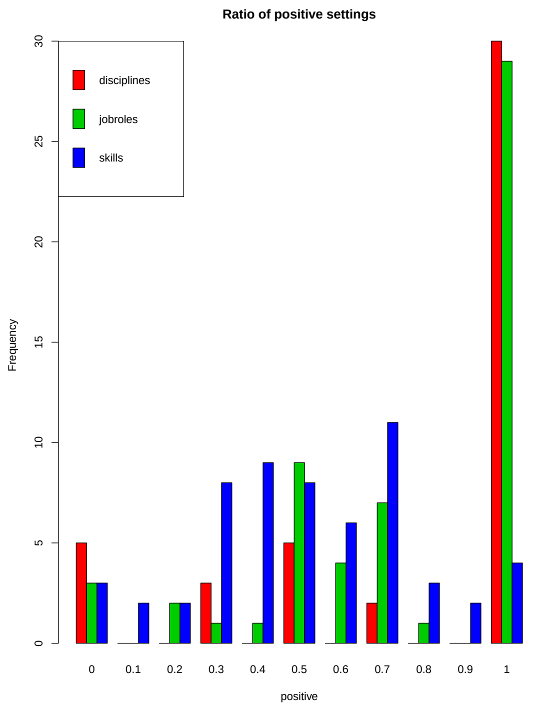
Figure 5.12.: Settings / Topics



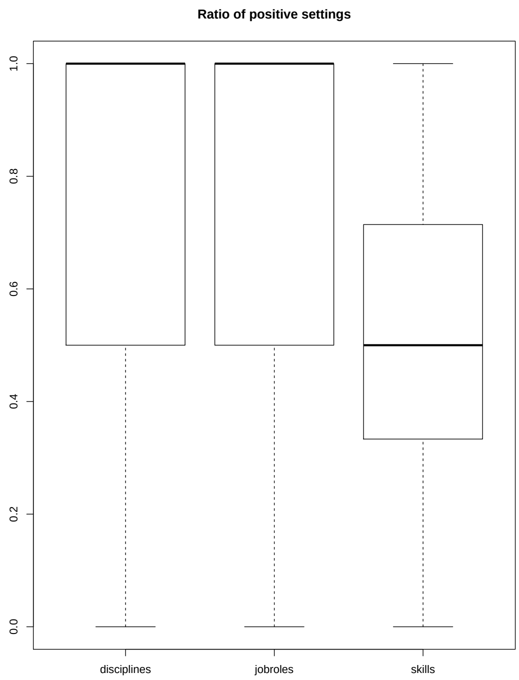
(a) Number of positive settings per user (per type)



(b) Number of negative settings per user (per type)



(c) Ratio of positive settings per user (per type)



(d) Ratio of positive settings per user (per type)

Figure 5.13.: positive/negative Settings / Topics

Statistical hypothesis testing

We are going to perform the statistical hypothesis tests using the *Wilcoxon signed-rank test* as described in Section 4.3.3. We are comparing all of the conditions pairwise to each other. These are the pairs as well as the hypothesis we are going to use:

- *original* vs. *tuned*
  1. H_0: The *original* method performs statistically significantly better than or equal to the proposed (*tuned*) method.
  2. H_1: The proposed method performs better than the *original* method.
- *random* vs. *tuned*
  1. H_0: The *random* method performs statistically significantly better than or equal to the proposed (*tuned*) method.
  2. H_1: The proposed method performs better than the *random* method.
- *random* vs. *original*
  1. H_0: The *random* method performs statistically significantly better than or equal to the *original* method.
  2. H_1: The *original* method performs better than the *random* method.

In the following, we perform one of them (*original* vs. *tuned*) manually. For the others, we will present the results generated by R using *wilcoxsign_test* from the *coin* library[2].

We have 55 data points in our data. First, for every pair of ratings, the *differences* are computed. Afterwards, for every *difference* we compute its *absolute value* as well as its *sign*. The data is then sorted by *absolute difference*. This data can be found in Table 5.8 and Table 5.9. The data presented in those tables is already sorted as described here.

The next step is to assign the *rank* to each data point. Since several data points share the same *absolute difference* (0 to 4), we assign the mean rank to each of the groups. Then we use the *sign* of each data point to determine to which case it belongs (W_+ or W_-). Since we have 13 data points with an *absolute difference* of 0, we split those data points into two groups of 7 and 6 and assign each group to one of these cases.

Then we compute W_+ and W_- by summing up all *adjusted ranks* for each of the cases. We get $W_+ = 795$ and $W_- = 745$. This leads to $W = min(W_+, W_-) = 745$.

The critical value for $\alpha = 0.05$ is ≈ 570. Since $745 \not< 570$, H_0 cannot be rejected.

[2] see `https://cran.r-project.org/web/packages/coin/coin.pdf` for details

This means, that there is not enough evidence that the proposed method works better than the existing method.

Performing the three tests using R, we get *p-values* of 0.4069 (*original* vs. *tuned*), 0.1064 (*random* vs. *tuned*) and 0.01545 (*random* vs. *original*). These values confirm what we just computed manually. We also have to accept that the random method works better than or equal to the proposed method. But we reject H_0 for the last case and find that the original method performs better than the random method.

Table 5.8.: Wilcoxon signed-rank test (1/2) sorted by *absolute difference*

user	rating original 2	rating tuned 2	difference	abs. difference	sign	rank	rank adjusted	contribution to W_+	contribution to W_-
10	4	4	0	0	0	1	7	0	7
17	4	4	0	0	0	2	7	0	7
20	4	4	0	0	0	3	7	0	7
22	4	4	0	0	0	4	7	0	7
31	4	4	0	0	0	5	7	0	7
33	4	4	0	0	0	6	7	0	7
35	4	4	0	0	0	7	7	0	7
36	5	5	0	0	0	8	7	7	0
38	4	4	0	0	0	9	7	7	0
43	2	2	0	0	0	10	7	7	0
45	4	4	0	0	0	11	7	7	0
52	4	4	0	0	0	12	7	7	0
53	5	5	0	0	0	13	7	7	0
1	2	1	-1	1	-1	14	23	0	23
2	3	2	-1	1	-1	15	23	0	23
4	2	3	1	1	1	16	23	23	0
7	5	4	-1	1	-1	17	23	0	23
8	4	3	-1	1	-1	18	23	0	23
11	3	2	-1	1	-1	19	23	0	23
12	2	3	1	1	1	20	23	23	0
15	5	4	-1	1	-1	21	23	0	23
18	3	4	1	1	1	22	23	23	0
21	3	4	1	1	1	23	23	23	0
23	3	4	1	1	1	24	23	23	0
24	4	5	1	1	1	25	23	23	0
25	3	2	-1	1	-1	26	23	0	23
27	3	4	1	1	1	27	23	23	0
28	3	2	-1	1	-1	28	23	0	23

Table 5.9.: Wilcoxon signed-rank test (2/2)
sorted by *absolute difference*

user	rating original 2	rating tuned 2	difference	abs. difference	sign	rank	rank adjusted	contribution to W_+	contribution to W_-
30	4	5	1	1	1	29	23	23	0
34	4	3	-1	1	-1	30	23	0	23
48	3	4	1	1	1	31	23	23	0
50	2	3	1	1	1	32	23	23	0
3	3	5	2	2	1	33	39.5	39.5	0
5	3	5	2	2	1	34	39.5	39.5	0
6	4	2	-2	2	-1	35	39.5	0	39.5
9	2	4	2	2	1	36	39.5	39.5	0
13	4	2	-2	2	-1	37	39.5	0	39.5
14	1	3	2	2	1	38	39.5	39.5	0
16	3	1	-2	2	-1	39	39.5	0	39.5
26	1	3	2	2	1	40	39.5	39.5	0
40	1	3	2	2	1	41	39.5	39.5	0
42	3	1	-2	2	-1	42	39.5	0	39.5
46	3	1	-2	2	-1	43	39.5	0	39.5
47	3	5	2	2	1	44	39.5	39.5	0
54	2	4	2	2	1	45	39.5	39.5	0
55	5	3	-2	2	-1	46	39.5	0	39.5
19	4	1	-3	3	-1	47	49.5	0	49.5
29	1	4	3	3	1	48	49.5	49.5	0
37	2	5	3	3	1	49	49.5	49.5	0
39	5	2	-3	3	-1	50	49.5	0	49.5
44	4	1	-3	3	-1	51	49.5	0	49.5
51	5	2	-3	3	-1	52	49.5	0	49.5
32	5	1	-4	4	-1	53	54	0	54
41	1	5	4	4	1	54	54	54	0
49	1	5	4	4	1	55	54	54	0

Making use of the ratings on individual items Since we got a negative result from the hypothesis tests, we want to analyse and exploit the participants' ratings on the individual job postings. We are looking at the *mean reciprocal rank* (mrr) of the first negatively rated job posting from each of the lists. The lower this value for

Table 5.10.: Using the participants' ratings on individual items (MRR)

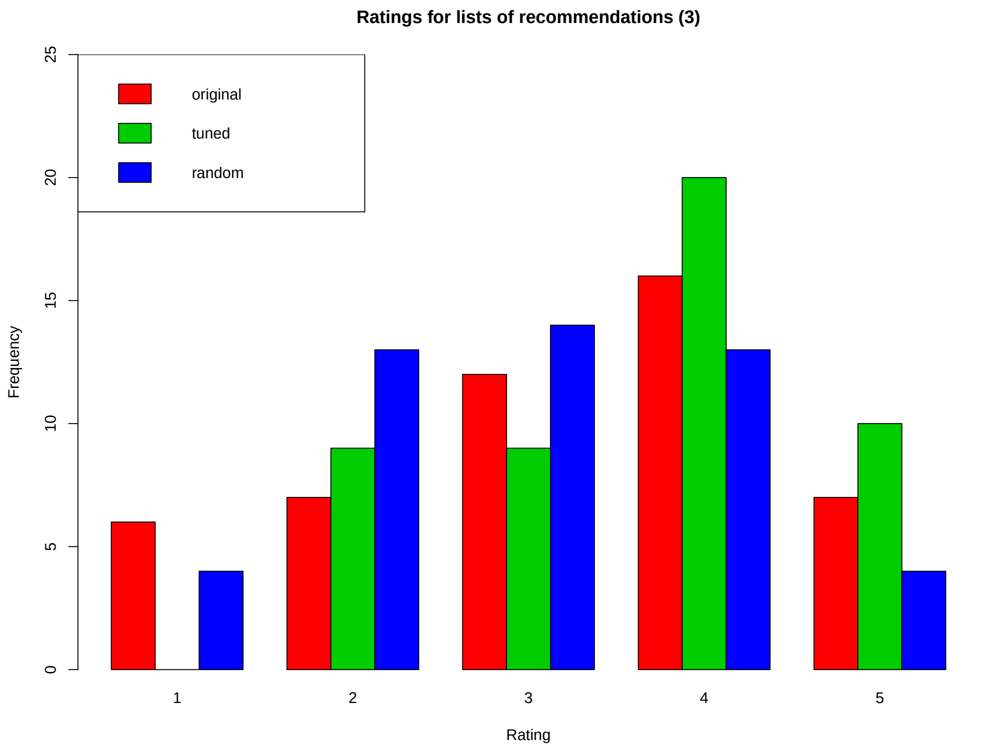
	mean reciprocal rank	mean rank	ratio negative in top 2	ratio negative in top 6	ratio negative in top 20
original	0.5268405	3.541667	0.375	0.4097222	0.428125
tuned	0.5025794	4.520833	0.3333333	0.3472222	0.328125
random	0.6899802	2.145833	0.4791667	0.5381944	0.4739583

a given list of recommendations, the better the method to generate this list, since on average more good items are at the head of the lists. Additionally we compute how many negative items are on average in the top 2, top 6 and top 20. The results are based on data from 48 participants and can be found in Table 5.10. Since we had 7 participants with no elements in the *tuned* list of recommendation, we removed these to be able to fairly compare only non-empty lists of recommendations. As you can see, *tuned* has the lowest mrr-value. Furthermore, it also has the best values for the *ratio of negative items* in the top 2, top 6 and top 20.

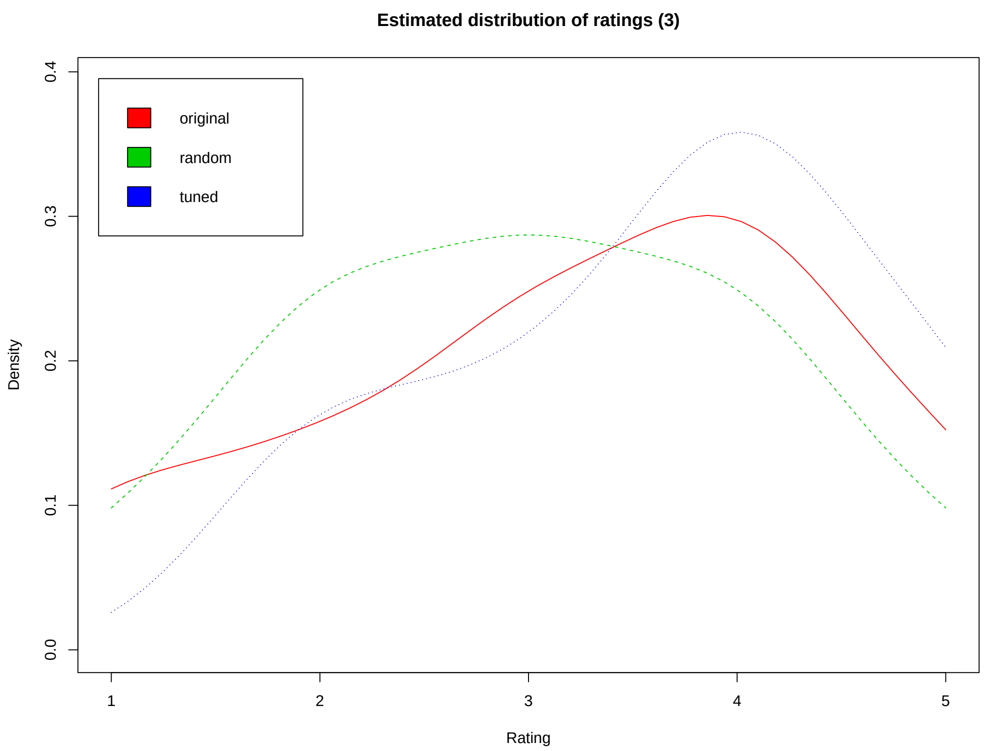
Since this is a promising result, we analysed what the users with empty lists of tuned recommendations have in common. Those 7 participants set a small value for *max distance* (between 5 and 50 km) while adjusting their settings and are XING employees working in Spain. Since XING is focused on the german-speaking parts of Europe, it has not a lot of job postings to offer in other regions. For these 7 participants it meant that all recommendations where removed because XING had no recommendable job posting in close proximity to those people during the time of the user study. All 7 participants rated the list of tuned recommendations as *very low*.

With this finding, it makes sense to repeat some of the analysis we did before in this chapter on the data set with these 7 participants removed. Figure 5.14 shows the distribution of ratings for the 3 lists of recommendations using the reduced data set. Compared to Figure 5.2b and Figure 5.3b, it seems encouraging to also repeat the hypothesis tests.

Performing the three tests on the reduced data set using R, we get *p-values* of 0.04517 (*original* vs. *tuned*), 0.005332 (*random* vs. *tuned*) and 0.06792 (*random* vs. *original*). This time, we reject H_0 for both tests involving *tuned* recommendations, meaning the proposed method performs better than the original and the random method, when we exclude people with too strict distance settings.



(a) Ratings



(b) Estimated density

Figure 5.14.: Ratings of users who got at least one tuned recommendation (48 participants)

6. Summary, Conclusions and Future Work

In this thesis, we research interactive recommender systems and present a method to offer interactive recommendations in the form of *recommender settings*. Specifically, this is done in the domain of *job recommendations* at XING, a professional social network. These settings allow users to tune some aspects of the job recommender system, i.e. their preferred career level, whether they are willing to commute or even move to a new location, and which *topics* (skills, jobroles and disciplines) they like or dislike. These topics are explicitly not taken from the users' profiles, as profiles on XING rather reflect the CV of the user, i.e. things that the user did in the past but not what the user aims to work on in the future. Instead, we generate the topics from the job recommendations we already offer, which are influenced by the users' profiles, their behavior on the platform as well as from their previously specified recommender settings. These topics can thus be seen as a *summary* of the users' job recommendations. By tweaking the recommendation settings, the actual job recommendations immediately change which in turn has an influence on the selectable topics thus allowing the user to interactively refine the recommendation settings and explore the item space.

We implemented our recommender settings approach in the back-end of the actual job recommendation service, thus turning XING's job recommender into an interactive recommender service. Moreover, we implemented a prototype application that allows users to experience the interactive job recommendations. Given both the adjusted job recommender service and our prototype, we conducted both a large-scale *quantitative evaluation* as well as a *user study* in which we collected qualitative feedback and analyzed the impact on user satisfaction. The conclusions from these evaluations are summarized in the subsequent section.

6.1. Conclusions

The quantitative evaluation on the impact of the recommender settings revealed that even a moderate number of settings lead to significantly different lists of recommendations. For example, by expressing positive and negative preferences into a small set of topics, we see that the average intersection between the preference-adjusted recommendations and the regular recommendation list is below 60% for the top 10 recommendations. The quantitative analysis thus indicates that our approach of recommender settings thus allows the user to actively influence the job recommendations.

In order to understand the impact on user satisfaction, we also conducted a user study using our prototype application of the adjusted job recommender system. We designed an interactive questionnaire and asked XING employees to participate. The findings of this user study and questionnaire, which was finished by 55 participants, can be summarized as follows:

- Participants rather liked their suggested topics, i.e. $\approx 69\%$ rated them *high* or *very high*, $\approx 24\%$ as *medium* and just $\approx 7\%$ as *low* or *very low*.
- Every user selected at least 3 topics he liked or disliked. The median participant selected 15 topics, while 50% of the users selected between 11 and 18 topics.
- Overall, the proposed method leads to a slightly higher user satisfaction. The difference in the overall user satisfaction is however not significant[1].
- Regarding the relavance of the items in the recommendation lists, we also see that the proposed method achieved better results than the existing system: non-relevant items had a lower chance of appearing at the top of the recommendation list (with a *p-value* of 0.04517).

Regarding the research questions raised in the introduction of this thesis, we thus conclude the following:

Design: We identified so-called *recommender settings* as an appropriate strategy for making recommender systems interactive in the job recommendation domain.

Implementation: We successfully implemented this strategy into XING's recommender service and developed a prototype application that allowed us to test our method with users.

Evaluation: Our quantitative evaluation shows that recommender settings have an influence on the recommendation lists and lead to different types of recommendations compared to the non-interactive system. More importantly, the results from the user study show that the interactive recommender leads to higher user satisfaction and to a significant enhancement regarding the ability of ranking relevant items at the top of the recommendation list.

[1] The test for significance was done via hypothesis testing using the *Wilcoxon signed-rank test*.

6.2. Future Work

Given the positive results of our analysis, we are currently—in collaboration with front-end developers—implementing our proposed method of the recommender settings as a new feature on the XING platform. We plan to introduce it to the XING users with an A/B test so that we are able to evaluate the impact of the recommender settings, i.e. check whether the click-through rate for users who adjusted their settings is significantly higher compared to the other group of users who do not benefit from the recommender settings.

The impact of negative or positive topics has to be reviewed. For this study, these values were defined during the offline evaluation. They were chosen so that their impact is high enough to change the lists of recommendations noticeable. These weights could either be learned once enough feedback data about tuned recommendations is collected. Alternatively, different values could be defined, which then have to be validated, e.g. again with A/B tests.

The impact of the strict distance filter has to be analyzed as well once the interactive recommender system is deployed to a larger user base. Alternative approaches are non-strict filtering or simply not offering this setting for certain types of users for whom the chances are low that relevant jobs are available close to their current location.

Another open question is how users will interact with the recommender settings and the interactive recommender system over time. For example, how frequently will users revisit the recommendation settings? And will people who made use of the recommender settings also continue to be more satisfied with their job recommendations in the long run?

Appendices

A. Data user study

Table A.1.: Data from user experiment (1/6)

user	rating topics	rating original 2	rating tuned 2	rating random 2	min. career level	max. career level	max. distance	# disciplines	# jobroles	# skills	# disciplines positive	# disciplines negative	# jobroles positive	# jobroles negative	# skills positive	# skills negative
1	2	2	1	2	5	6	5	6	2	17	3	3	1	1	7	10
2	3	3	2	4	4	6	50	1	1	17	1	0	0	1	8	9
3	5	3	5	2	3	4	50	1	1	4	1	0	1	0	3	1
4	2	2	3	2	4	6	50	2	1	16	1	1	1	0	6	10
5	5	3	5	3	3	4	20	1	4	12	1	0	3	1	6	6
6	4	4	2	4	3	4	50	3	7	8	3	0	4	3	5	3
7	5	5	4	3	3	4	20000	1	1	4	1	0	1	0	2	2
8	4	4	3	3	3	4	20	1	3	13	1	0	3	0	9	4
9	4	2	4	2	3	4	100	0	1	10	0	0	1	0	7	3
10	5	4	4	5	3	4	20	0	1	7	0	0	0	1	5	2
11	4	3	2	3	4	5	5	3	2	11	2	1	1	1	8	3
12	3	2	3	2	4	5	200	1	2	5	1	0	2	0	0	5
13	3	4	2	3	3	4	5	2	6	9	1	1	3	3	8	1
14	1	1	3	1	3	3	5	0	2	13	0	0	0	2	5	8
15	5	5	4	4	3	4	20000	1	2	21	1	0	2	0	16	5
16	3	3	1	2	3	5	20	1	3	16	1	0	2	1	5	11
17	5	4	4	3	3	4	50	1	1	16	1	0	1	0	14	2
18	4	3	4	4	3	4	20	1	3	7	1	0	2	1	4	3
19	4	4	1	3	3	4	5	1	3	17	1	0	2	1	9	8
20	4	4	4	4	3	4	20000	1	7	11	1	0	4	3	8	3
21	4	3	4	3	3	4	5	0	2	14	0	0	2	0	10	4
22	4	4	4	5	3	4	200	1	1	14	1	0	1	0	9	5
23	4	3	4	3	4	5	20000	1	3	13	0	1	2	1	8	5
24	4	4	5	2	3	4	50	1	1	12	1	0	1	0	4	8
25	4	3	2	2	3	4	50	2	2	11	0	2	1	1	4	7
26	4	1	3	2	3	4	50	4	3	8	3	1	3	0	6	2
27	3	3	4	3	4	5	20000	3	2	10	1	2	1	1	1	9
28	4	3	2	4	4	6	5	1	5	12	1	0	3	2	6	6
29	4	1	4	1	3	3	20	1	2	12	1	0	1	1	4	8
30	4	4	5	4	3	4	20000	0	1	13	0	0	1	0	7	6

Table A.2.: Data from user experiment (2/6)

user	rating topics	rating original 2	rating tuned 2	rating random 2	min. career level	max. career level	max. distance	# disciplines	# jobroles	# skills	# disciplines positive	# disciplines negative	# jobroles positive	# jobroles negative	# skills positive	# skills negative
31	3	4	4	4	4	5	50	1	2	10	1	0	1	1	4	6
32	3	5	1	3	3	4	20	1	1	17	1	0	1	0	5	12
33	3	4	4	4	3	3	20000	0	5	1	0	0	4	1	1	0
34	5	4	3	3	4	6	20000	1	1	10	1	0	1	0	8	2
35	4	4	4	3	2	3	50	1	2	7	1	0	2	0	7	0
36	3	5	5	5	3	4	20000	1	2	8	1	0	2	0	4	4
37	4	2	5	3	4	6	20000	0	5	5	0	0	2	3	0	5
38	4	4	4	4	3	4	20	3	4	15	3	0	3	1	15	0
39	5	5	2	4	3	4	20000	2	2	17	1	1	1	1	6	11
40	3	1	3	2	3	4	20	1	4	16	0	1	1	3	1	15
41	4	1	5	1	2	3	50	1	3	5	0	1	1	2	3	2
42	3	3	1	3	3	3	20	0	0	5	0	0	0	0	1	4
43	2	2	2	2	1	6	50	3	1	18	1	2	1	0	5	13
44	5	4	1	3	3	4	50	0	1	2	0	0	1	0	0	2
45	4	4	4	5	1	3	200	0	1	4	0	0	1	0	2	2
46	4	3	1	3	3	3	20	1	1	9	1	0	1	0	4	5
47	4	3	5	2	1	2	50	0	2	11	0	0	2	0	11	0
48	3	3	4	3	3	4	20000	2	5	10	1	1	1	4	6	4
49	4	1	5	1	3	3	20	1	2	11	1	0	1	1	3	8
50	4	2	3	2	3	4	20	0	1	14	0	0	1	0	10	4
51	4	5	2	4	3	4	5	1	2	11	1	0	2	0	3	8
52	4	4	4	3	2	3	200	1	2	10	1	0	2	0	4	6
53	4	5	5	2	3	4	20000	0	2	12	0	0	2	0	9	3
54	4	2	4	2	1	2	20	1	5	7	1	0	3	2	5	2
55	3	5	3	4	3	4	20	1	2	8	0	1	2	0	2	6

Table A.3.: Data from user experiment (3/6)

user	rating original	rating tuned	rating random	ratings item random
1	2	1	2	-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
2	4	4	5	11-1-111111-1-1-111-1-11-111
3	3	5	2	-1-1-1111-1-1-1-111-1111-1111
4	3	4	3	1-11-1-1-1-1
5	3	5	3	1-1-11-1-11-11-1-1-1-1-1-1-1-1-1-1
6	5	3	4	11111-11111-1
7	5	5	4	111-1-1-111-1-11111-1111-11
8	5	3	4	1-11-11-111-111-11-1111-1-1-1
9	2	4	2	-11-1-1-1-111-1-11-1-11-111-111
10	3	5	4	111111-11-1-1111-111-1111
11	4	4	4	-11111111-1-1-11-11-11-1-1-1-1
12	2	3	3	-11-1-1-11-1-1-1-1-1-11-1-1-1-1-1-11
13	5	3	4	-1111-11-11-11111-11-111-11
14	1	3	1	-11-1-11-11-1-1-1-111-1-1-1-11-11
15	5	4	4	-11-1-1-1-11-11-1-1-11-1-1-1-1111
16	4	1	2	-1-1-11-11-1-1-1-1-1-1-1-1-1-11-1-11
17	4	4	4	1-11-1-1-1-11-1-1-1-11-1-11-1-1-1-1
18	2	2	3	11-1-11-1-111-1-11111111-11
19	4	1	3	1-11111-1-1-1-1-111-1-1-11-11-1
20	4	4	5	-11-11-1-1111111111111-11
21	2	4	2	-1-11-11-1-1-1-1-111-11-11-11-1-1
22	4	3	4	11-11111111-11111-11111
23	3	4	4	-1-11-1-1-1-1111-1111-1-1111-1
24	4	4	3	-1-11-1-1-1-1-1111-11-1-1-111-11
25	3	2	2	1-1-11-111-1-11-11-1-1-1-1-11-1-1
26	2	4	3	-1-11-1-1-1-1-1-1-11-11-11-1-1-11-1
27	3	4	3	-11111111-1111-1-1-1
28	3	2	4	111-1-1-1-11-11-11-1-1-1
29	1	4	1	-1
30	3	4	5	11-1-1-11-11-111-1-11-1-11111

Table A.4.: Data from user experiment (4/6)

user	rating original	rating tuned	rating random	ratings item random
31	3	4	3	1-1-1-11-111-1-1-1-111-1111-1-1
32	5	1	2	-1-111
33	4	4	4	1-1-1-1-111111-1-1-11-1
34	4	3	3	-11-111-11-1-1111-1-1-111-1-11
35	4	4	3	-1-1-1-1-11-1-11111-1-1111-11-1
36	4	4	4	111111-1111-111-1111-111
37	2	5	3	-11-111-1-1-11-1-1-1-1-1-11-1-1
38	2	2	3	11-1111-1-11-11-11111111
39	5	2	2	11-11-1111-11111
40	2	3	3	-1-1-1-1-11-1-1-1-1-1-1-1-1-1-1-1
41	2	4	1	-1-11-1-1-11111-1-1-1-11-11-1
42	3	2	3	11-11-1-1-1-1-1-1-1-1-11111-1-1-1
43	2	2	2	-11-1-1
44	4	1	3	-111-1111-11111-1-1-1-1
45	3	4	5	11111-11-1-111-11111111-1
46	3	1	3	-1-1-1-1-1-1-11-1-1-1-1-11-1-1-1-1-11
47	4	5	3	-11-1-11-1-11111-111-11-1-1-11
48	3	3	3	111-1-111-11-1-1-1-1-1
49	2	5	2	-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
50	2	3	2	-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
51	4	2	5	1-1-11111-11-11-1111-11111
52	4	4	3	-1-1-11-1-11-1-1-11-111-111-11-1
53	5	4	3	-1-111-11-1111-1111-11-1-11-1
54	2	4	2	1-1-1-1-1-1-1-1-1-11-1-111-1-1-1-11
55	3	5	4	111-1-11-11-1-1-11-11-111111

Table A.5.: Data from user experiment (5/6)

user	ratings item original	ratings item tuned
1	-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1	
2	1-1-111-11111-1-111-1-1-1-111	-1-1-1-1-11-1-1-1-1111-111-111
3	-111-111-11-111-1-111111-11	1111-1-111111-1-111-1-11-11
4	1-1-1-11-1-1	1-1-111
5	11-1-1111-11-1-1-1-1-11-1-1-1-1-1	11111111111111-1-1-111-1
6	1-1111111-111	-1-111-1111-1-1-1-1-1
7	11111111111111111-11	1111-111111111-1-111111
8	11-111111-111-1-1-1111-1-1-1	1-1-11-11-1-11-11-1111-111-1-1
9	-1-11111-11111-1-1-1-1-11-1-1-1	11-1111111-11111111111
10	1111-111111-1-11-1-1-1111-1	111-11-1111111-11-11-1-1-11
11	11111-11-1-1-11-1111-1-1-11-1	-11-1-1-11-1-1-1
12	-1-1-11-1-11-1-1-111111-1-1-1-1-1	-1-1-11-1-111111-111-1-1-1-1-1-1
13	1111-1-1111111-11-1-1-1-111	11-11-11-1-1-1-1-1-1-1
14	-1-111-1-1-1-1-11-1-1-1-11-1-1-1-1-1	111-11111-1-1-1-11-11-1-1-11-1
15	1111-1-11-1-11-1-1-11-11-1-1-11	1-111111-1-1111-111-11-11-1
16	111111-1-1-1-1-1-1-1-1-1-1-1-1-1-1	
17	1-1111-11-1-1-111-11-11-1-1-11	11111-11-1-1-11-1-11-11-1-11-1
18	1-11-111-11-1-1111111-11-11	1111-11-111111-11-1-1-1-11-1
19	11111-1111111-11-1-11-1-11	
20	11111-111-1111-11-111111	111111111111-1111111
21	-11-1-11-11-1-1-1-1-11-1-1-1-1-1-1-1	111-1-111-1-111-11-1-1-1-1-1-1-1
22	11-1111-111-1111111-1111	-11111-11111-1-1-1-111-1-1-11
23	-1-1-1-1-1-1-1-1-111-1-11111111	-11-11-1-11111111111-111
24	11111-111-11-111111-1-111	11111-1-1-1-111111-111111
25	1111111-1-1-1-1-11-111-1-1-11	1111-1-1111-1-1-1-1-1-1-1-1-1-1-1
26	1-1-1-1-1-1-11-1-1-111-11-1-1-1-1-1	-111-1111-1-1-1-1111-1-1-1-1-11
27	1-1-1-11111-1-111111	11111-1
28	-11-11-1-1111-1-1-11-1-1	1-1-1-1-1-1-1111
29	-1	1-1111111-1111-1
30	1-11-1-1-11-1-111111111-11-1	-1-11-1111111111-1-1-11111

Table A.6.: Data from user experiment (6/6)

user	ratings item original	ratings item tuned
31	11-11-1-11-1-1-1111-111-111-1	-1111-1-11-1-11-1-1-111-1-1-11
32	11	
33	1111111-1-1-1-1-1-1-1-1	11111-1-1-1-1-1-11
34	111111-11-11-1-11-11-1-111-1	-1-111-11-1-1-11-1-11-1-1-11-1-11
35	111111-1111-1-1-11-1-1-1-11-1	11
36	1111-111111111-1111111	111-11111111111-111111
37	-1-1-1-11-1-1-1-1-1-1-111-111-1	11-111-111-11
38	1111111111-1-1-1111-1-1	-111111111-11-111-1-1-1-111
39	111-11-1-1111111	1-1-11-1111111-11-1-111
40	-1-1-1-1-1-1-1-1-11-1-1-1-1-1-1-1-1	-1-11-1-111-1-1-1-1
41	-11111-1-1-1-111-1-11-1-1-1-1	11-111111
42	11-1-111-111-11-1-1-1-1-1-11-1-1	
43	-1-11-1	-1-1-1-11-1-1-1-1-11-1-11-1-1
44	1111-1-1111-11-11-1-1-1	
45	1111-11-1-11-1111-1-1-1-1111	1111-111111-111-1-1-11-1-1-1
46	-1-1-1-1-1-11-1111-1-1-1-1-1-11-1-1	
47	1-111-1111-111111-1-1-1-1-11	1111
48	-1-1-1-11-1-1-1111-111	-1-1-1-11-1-1-111111-1111
49	-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1	11
50	-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1	-1-1-1-1-1-1-111-1-11-1-1
51	111-1-11-11-11111111-1111	-1-1
52	111-111-11-111-1-1-11-1-1-1-1-1	111-11-1111-111-11111-1-11
53	1111-1-1111111111-11111	111111-1111-111111111-1
54	-1-111-1-1-11-1-11-1-1-1-1-11-11-1	111-111-11111-1111-1-1-1-1-1
55	11-1-11-1-11-111-111111111	1111-1-1-111-111-11-1-11-1-1-1

B. Comments user study

- Hello, I felt that the presented recommendations
  - did not fit the right career level (almost all were too junior for what I thought I had specified)
  - did partly not reflect my location preference

  [...]
- Impressed by the quality of the recos. However, as I am not flexible for location, giving me jobs outside of HH after I defined this feels unmentorlike (I realize that this might be part of your research).
- It is kind of hard to select the settings: I have only one chance to decide about the topics I like or do not like. In case I would see the impact of my preferences then I could in fact change things until I end up with a recommendation list that really fits.
- It seems that there is no bigger eye on the interests I have on my profile. As well as it is not taken into account what my current as well as my former jobs where. I'm working with Ruby, fine, so I can tell that I can work with it. But this does not mean I'm a Ruby Developer. Future Me was a very good approach to know what people want and where they want to go. Unfortunately there is nothing of this in the job recos.
- I still find very difficult to compare whole lists of recommendations, especially with 20 items in each of them.
- too many jobs in other cities (although I set a radius); too many jobs in fields that I sorted out before
- To be able to "create", change and rate recos this simple would be great for users
- Many of the job recommendations exceeded distance limit by hundreds of Kilometers. Most of the recos fit my profile but I had to rate them as "negative" because of the distance problem.

- Some jobs really lacked a description - I didn't really know how I feel about "Software Engineer" position - it's very generic. Also there's not so much jobs in Barcelona and since I selected 5km range I had only two very generic job descriptions for the last list.

- I feel the job recommender system 1&3 were very good. It had the precision and diversity(location, role) of the results from my chosen skill. The system 2 gave some good results along but some recommendation were very far off from my requirements.

List of Figures

List of Tables

Bibliography

[BA06] Bendakir, Narimel ; Aimeur, Esma: Using Association Rules for Course Recommendation. In: *In Proceedings of the AAAI Workshop on Educational Data Mining* (2006), July 16–17, S. 31–40

[BS10] Bortz, Jürgen ; Schuster, Christof: *Statistik für Human- und Sozialwissenschaftler*. 7. Aufl. Berlin, Heidelberg : Springer, 2010. – ISBN 978–3–642–12769–4

[EKHK15] Ekstrand, Michael D. ; Kluver, Daniel ; Harper, F. M. ; Konstan, Joseph A.: Letting Users Choose Recommender Algorithms: An Experimental Study. In: *Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2015 (RecSys ’15). – ISBN 978–1–4503–3692–5, 11–18

[ERK11] Ekstrand, Michael D. ; Riedl, John T. ; Konstan, Joseph A.: Collaborative Filtering Recommender Systems. In: *Found. Trends Hum.-Comput. Interact.* 4 (2011), Februar, Nr. 2, 81–173. http://dx.doi.org/10.1561/1100000009. – DOI 10.1561/1100000009. – ISSN 1551–3955

[HXK+15] Harper, F. M. ; Xu, Funing ; Kaur, Harmanpreet ; Condiff, Kyle ; Chang, Shuo ; Terveen, Loren: Putting Users in Control of Their Recommendations. In: *Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2015 (RecSys ’15). – ISBN 978–1–4503–3692–5, 3–10

[KHS07] Kohavi, Ron ; Henne, Randal M. ; Sommerfield, Dan: Practical Guide to Controlled Experiments on the Web: Listen to Your Customers Not to the Hippo. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA : ACM, 2007 (KDD ’07). – ISBN 978–1–59593–609–7, 959–967

[Mov17] MovieLens: *About MovieLens*. https://movielens.org/info/about. Version: 2017

[RRSK10] Ricci, Francesco ; Rokach, Lior ; Shapira, Bracha ; Kantor, Paul B.: *Recommender Systems Handbook*. 1st. New York, NY, USA : Springer-Verlag New York, Inc., 2010. – ISBN 0387858199, 9780387858197

[SKKR01] Sarwar, Badrul ; Karypis, George ; Konstan, Joseph ; Riedl, John: Item-based Collaborative Filtering Recommendation Algorithms. In: *Proceedings of the 10th International Conference on World Wide Web*. New York, NY, USA : ACM, 2001 (WWW '01). – ISBN 1–58113–348–0, 285–295

[TSK05] Tan, Pang-Ning ; Steinbach, Michael ; Kumar, Vipin: *Introduction to Data Mining, (First Edition)*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 2005. – ISBN 0321321367

[Wil45] Wilcoxon, Frank: Individual Comparisons by Ranking Methods. In: *Biometrics Bulletin* 1 (1945), Dezember, Nr. 6, 80–83. http://dx.doi.org/10.2307/3001968. – DOI 10.2307/3001968. – ISSN 00994987

[XIN17a] XING AG: *Meldung - XING Corporate*. https://corporate.xing.com/de/newsroom/pressemitteilungen/meldung/xing-steigert-e-recruiting-erloese-im-ersten-quartal-2017-um-40-proze
Version: 2017

[XIN17b] XING AG: *XING Daten und Fakten*. https://recruiting.xing.com/en/daten-und-fakten/. Version: 2017

Eidesstattliche Erklärung

Ich versichere, dass ich die Arbeit selbstständig angefertigt und keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle Stellen, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, habe ich in jedem einzelnen Fall unter genauer Angabe der Quelle deutlich als Entlehnung kenntlich gemacht. Dies gilt auch für alle Informationen, die dem Internet oder anderer elektronischer Datensammlungen entnommen wurden. Ich erkläre ferner, dass die von mir angefertigte Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung im Rahmen meines Studiums war. Mir ist bewusst, dass die nachgewiesene Unterlassung der Herkunftsangabe oder die Nutzung als parallele Prüfungsleistung als Täuschungsversuch bzw. als Plagiat gewertet und mit Maßnahmen bis hin zur Zwangsexmatrikulation geahndet wird. Die von mir eingereichte schriftliche Fassung entspricht jener auf dem elektronischen Speichermedium.

Ich bin mit der Einstellung der Arbeit in den Bestand der Bibliothek des Fachbereichs Informatik einverstanden.

Hamburg, den 9.6.2017 ________________________

Mirko Köster