VizRec: Recommending Personalized Visualizations

BELGIN MUTLU, Know Center GmbH
EDUARDO VEAS, National University of Cuyo
CHRISTOPH TRATTNER, Norwegian University of Science and Technology

Visualizations have a distinctive advantage when dealing with the information overload problem: Because they are grounded in basic visual cognition, many people understand them. However, creating proper visualizations requires specific expertise of the domain and underlying data. Our quest in this article is to study methods to suggest appropriate visualizations autonomously. To be appropriate, a visualization has to follow known guidelines to find and distinguish patterns visually and encode data therein. A visualization tells a story of the underlying data; yet, to be appropriate, it has to clearly represent those aspects of the data the viewer is interested in. Which aspects of a visualization are important to the viewer? Can we capture and use those aspects to recommend visualizations? This article investigates strategies to recommend visualizations considering different aspects of user preferences. A multi-dimensional scale is used to estimate aspects of quality for visualizations for collaborative filtering. Alternatively, tag vectors describing visualizations are used to recommend potentially interesting visualizations based on content. Finally, a hybrid approach combines information on what a visualization is about (tags) and how good it is (ratings). We present the design principles behind *VizRec*, our visual recommender. We describe its architecture, the data acquisition approach with a crowd sourced study, and the analysis of strategies for visualization recommendation.

CCS Concepts: • **Information systems → Content ranking**; **Personalization**; • **Human-centered computing → Collaborative filtering**; *Graphical user interfaces*; *Empirical studies in visualization*; *Visualization design and evaluation methods;*

Additional Key Words and Phrases: Personalized visualizations, visualization recommender, recommender systems, collaborative filtering, crowd-sourcing

ACM Reference Format:
Belgin Mutlu, Eduardo Veas, and Christoph Trattner. 2016. VizRec: Recommending personalized visualizations. ACM Trans. Interact. Intell. Syst. 6, 4, Article 31 (November 2016), 39 pages.
DOI: http://dx.doi.org/10.1145/2983923

31

1. INTRODUCTION

Despite technical advances in search engines and content services, information overload still remains a crucial problem of many application fields. Finding the right piece of information in huge information spaces is a tedious, time-consuming task.

The reviewing of this article was managed by special issue associate editors Nava Tintarev, John O'Donovan, and Alexander Felfernig.
This work is funded by the EC 7th Framework project EEXCESS (grant no. 600601) and partially by the European Horizon 2020 research project AFEL (grant no. 687916) and CONICET. The Know-Center GmbH is funded within the Austrian COMET Program managed by the Austrian Research Promotion Agency. The study was carried out during the tenure of an ERCIM "Alain Bensoussan" fellowship program by the third author.
Authors' addresses: B. Mutlu, Knowledge Visualization Group, Know Center GmbH; email: bmutlu@know-center.at; E. Veas, Faculty of Exact and Natural Sciences, National University of Cuyo; email: eveas@know-center.at; C. Trattner, Norwegian University of Science and Technology; email: ctrattner@know-center.at.

DOI: http://dx.doi.org/10.1145/2983923

Visualizations have shown to be an effective way to deal with the overload problem, with the opportunity to display and explore a huge set of data points simultaneously.

However, creating useful visual representations of data requires expert knowledge. It requires knowledge of the shape and structure of the data. It also requires expertise about visualization design to apply the right visual encodings. Analysts rarely have knowledge of both data and visualization principles and, most often, novices lack both. Still, popular visualization tools require manual specification of visual information, which involves the following: selecting variables of interest, selecting transformations, and designing visual encodings [Stolte and Hanrahan 2002; Wickham 2009]. All together, it is a tedious task that interrupts the exploration flow [Wongsuphasawat et al. 2015]. To date, only a few approaches have attempted to automatically generate visual representations starting just from the data [Mutlu et al. 2014; Nazemi et al. 2013], albeit with limited success. Despite their usefulness, these approaches are ineffective in terms of dealing with highly heterogeneous data. They also ignore the fact that the choice of visual representation involves as much user preferences and needs: Popular visualization tools require several human choices to tailor the end result to the user's preference. Beyond visualization, recommender systems address the personalization issue basing on knowledge about interests and previous choices of users. A number of questions arise to this respect. Which aspects of the visualization and underlying data are important for the user? How should these aspects be captured? Which strategies should be used to recommend visualizations based on them?

This work contributes fundamentals on the design of a visualization recommender (see Section 3), in particular, the design of recommendations based on perceptual guidelines to prune the number of combinations to a manageable size, the information design to represent user's needs and preferences, and the studies validating different recommendation strategies therewith. When generating visualizations, arbitrary selection of data fields and subsequent naïve choice of visual encoding inevitably lead to a combinatorial explosion. We introduce a novel approach—called *VizRec*—that relies on perceptual guidelines to reduce the number of recommendations. Considering just visual encoding rules reduces the combinatorial problem but still leads to a large set of possibilities, valid in terms of representing the data visually but without considering which type serves the user's needs best [Mutlu et al. 2014]. To promote just the relevant views, appropriate filtering and recommendation strategies are needed [Wongsuphasawat et al. 2015]. Our goal is to suggest those visualizations that a user would select as part of her analysis workflow. Therefore, we investigate which information lets us anticipate the choice of visualization for data analysis and how to represent such information and use it for recommendation. We investigate a collaborative filtering approach based on a multidimensional scale to gauge different quality aspects of the visualizations suggested. We analyze the effects these aspects have on the filtering of visualizations. Content-based filtering is analyzed as a means to suggest visualizations according to the information needs of the user by collecting vectors of tags describing what aspects of the data each visualization represents. A combination of both aspects, what a visualization is about (tags), and how good it is (ratings) is studied using a hybrid recommendation strategy. Furthermore, an extensive evaluation of visualization types in the context of three data repositories was conducted in Amazon Mechanical Turk (see Section 4). The evaluation serves multiple purposes as follows: to understand the variability in choice of preferred visualizations, to analyze assessments of quality of visualizations and their descriptions, and to study different recommendation methods.

It is our hope that these results provide motivation to contemplate these personalization aspects as part of the design of visualization recommendation.

2. BACKGROUND AND RELATED WORK

Recommending visualizations is a relatively new strand of research and only a small amount of effort has been made so far to tackle the challenge. However, a broad body of research formulates perceptual guidelines for visual communication that an expert uses to create a visualization. Section 2.1 summarizes the visual encoding principles behind our approach. Rule-based systems have been developed to generate visualizations, for example, based on correspondence of data attributes and visualization properties [Stolte and Hanrahan 2002; Cammarano et al. 2007; Mackinlay et al. 2007]. Although partially successful, these systems fall short of automating the whole process and leave decisions that require certain expertise to the user (e.g., mapping of variables to data). Yet, the greatest flaw of these methods is not accounting for user preferences or task.

Alternatively, methods aiming for adaptive visualization build a user context to configure the visual display [Nazemi et al. 2013; Ahn and Brusilovsky 2009]. These methods work bottom-up, analyzing user actions to determine her behavior, and thereby predict the desired configuration of the visual display. In contrast, established methods in recommender systems personalize suggestions relying on users to express preferences by rating items, while tagging drives content-based recommendations. Our approach extends recommender system methods to visualization, building user and content-based stages for a visualization recommender. A large number of preferences and annotations for our empirical methods were acquired with a crowdsourced study. Crowd sourcing studies offer the possibility to reach a high participation rate. They have been applied in a handful of occasions to study aspects of visualization. Section 2.5 highlights crowd-based works that illuminated our study design.

2.1. Visualization Compositing Guidelines

Visualization can be considered from information theory as visually coding and communicating information [Chen and Jänicke 2010]. Bertin's work on semiology offers a systematic study of visual representations [Bertin 1983]. It defines and characterizes visual variables that compose visualizations. Carpendale analyzes visual variables for computational information visualization [Carpendale 2003]. Building on semiology, Mackinlay developed a formal language to generate graphical presentations for relational information and defined *expressiveness* (whether a graphical language can express the desired information) and *effectiveness* (whether the graphical language exploits the capabilities of the output medium and the human visual system) [Mackinlay 1986]. Card and Mackinlay categorized data in terms of its attributes (e.g., nominal, ordinal, quantitative) and analyzed their mapping to visual variables in scientific visualizations [Stuart and Jock 1997]. Engelhardt systematically analyzed syntactic structure and information type in graphic representations [von Engelhardt 2002]. The structure and design of any graphical representation have a perceptual connotation with cognitive implications [Ware 2012]. These contributions build our understanding of the visual encoding principles that help us design visualizations [Munzner 2014]. Voigt et al. documented these principles in a descriptive ontology [Voigt et al. 2013a].

2.2. Rule-Based Approaches

The evolution in the formalization of visual encoding theory and principles not only improves our understanding of the process but also contributes to the formulation of generative methods for visualization. Following Mackinlay [1986], the initial referent for the automated generation of visualizations is Polaris, the backbone engine in the early version of *Tableau* [Stolte and Hanrahan 2002]. The system automatically suggests visualizations for tables in relational databases and coordinates the interaction between them. But the mapping of data onto visual properties of a visualization is not performed automatically; instead, it has to be formulated by the user. Conversely, Cammarano et al. describe a method for automatic mapping of data attributes to visual attributes based on schema matching [Cammarano et al. 2007]. Using this system, the user formulates a query and obtains a result set of visualizations including map, time, or scatterplot. Once selected a type of visualization, the system searches for the attributes in the data space that best fit the requirements of the chosen visualization.

Mackinlay et al. propose an influential, albeit conceptually different, approach in the ShowMe [Mackinlay et al. 2007] system. It integrates a set of user interface commands and functions aiming at automatically generating visualizations for *Tableau*.[1] ShowMe attempts to help the user by searching for graphical presentations that may address her task. Appropriate visualizations are selected based on the data properties, such as data type (text, date, time, numeric, Boolean), data role (measure or dimension), and data interpretation (discrete or continuous). We follow a similar approach and select visualizations based on visual encoding rules instrumented in a functional ontology with mapping algorithm to suggest visualizations for published data [Mutlu et al. 2014]. VizRec identifies all possible visualizations for the current data in advance and guarantees in this way the effective graphical presentation of the data based on their characteristics.

Generative approaches fulfill the requirements of *expressiveness*, expressing the information in a dataset visually. Section 3.2 illustrates how visual encoding rules narrow down the combinatorial of visual variables to a manageable number. *Effectiveness*, however, not only depends on the syntax and semantics of the graphical language, as Mackinlay puts it, but also on the capabilities of the perceiver [Mackinlay 1986]. In formative studies by Mutlu et al., users found that the initially suggested visualizations did not sufficiently emphasize the aspects they were interested in [Mutlu et al. 2014; Sabol et al. 2014]. What aspects of the data are important for the user at a given time? And which visualization preferably represents them?

ShowMe introduces a ranking of visualizations based on static ratings (scores) globally defined for every supported visualization type [Mackinlay et al. 2007]. Rather than using global ratings, our method allows us to personalize the resulting visualizations according to the interests of the individual user using a collaborative filtering (CF) based approach.

The closest approach to our suggestion is a system described by Voigt et al. [2013b], which uses a knowledge base of numerous ontologies to recommend visualizations. It is essentially a rule-based system that pre-selects visualizations based on the device, data properties, and task involved. Subsequently, the system ranks visualizations following the rules concerning visualization facts, domain assignments, and user context. One disadvantage of Voigt et al.'s approach is that both visualizations and data inputs have to be annotated semantically beforehand. Furthermore, the pre-selection and the ranking stages are rule based. More importantly, a large theoretical part of the work completely lacks empirical support. While user preferences, such as graphical representations and visualization literacy, are outlined as required in their approach, the actual collection and validation of user preferences are tasks for future work. In contrast, we present a complete approach using different recommendation strategies and supported by the collection of user preferences for personalization in a large study involving the general public, validated in an offline experiment and drawing conclusions based on the empirical evidence.

[1] http://www.tableausoftware.com/.

2.3. Behavioral Approaches

Nazemi et al.'s system suggests visualizations based on user preferences [Nazemi et al. 2013] incrementally gathered during interaction with the visualization system in the form of usage profiles for particular visualizations. Nazemi et al. follow a bottom-up approach, analyzing user interaction via visualization to describe user behavior. In contrast, we apply a top-down method to elicit user preferences by collecting ratings. These methods are complementary and can be deployed together. Similarly to us, Nazemi et al. utilize a personalized approach to suggest visualizations but only target the content from digital libraries (i.e., bibliographical notes and publications).

Ahn et al.'s work on adaptive visualization attempts to provide user-adapted visual representation of their search results [Ahn and Brusilovsky 2009]. The user context is a collection of user actions accumulated over time, such as the issued search queries, selected documents from the search results, and traversed links. The collection captures user interests beyond the query and in turn defines a user model, which is applied to visually highlight the relevance of a particular result set. In contrast, VizRec augments user queries with preferences in order to find the best representation of the information behind the queried content instead of only displaying relevant results as clusters.

Similarly to Ahn et al., building preferences to adapt visualizations to user interests has also been practiced in specific domains; for instance, Vartak et al. [2014] obtain preferences by capturing user interactions with the visualization system for digital libraries. Before recommending visualizations, the system runs analytics, looking for behavior patterns in the output of the user and then selects the visualizations that might be interesting or useful for the active user.

Despite these notable efforts, the problem of recommending visualizations is still insufficiently explored, and especially little research has been performed on generating and suggesting useful visualizations for heterogeneous multidimensional data.

Moreover, there seems to be a gap in the literature on doing this in a personalized manner, since previous work on recommender systems has shown that the one-size-fits-it-all principle typically does not hold. To contribute to this small body of research, we developed and evaluated VizRec, a novel visual recommender engine capable of recommending various types of visualizations in a personalized manner.

2.4. Generating Personalized Recommendations

One of the the most successful and prominent approaches to generate recommendations is CF [Schafer et al. 2007; Su and Khoshgoftaar 2009], which uses a collection of user preferences to generate recommendations. Basically, the preferences are collections of either explicit ratings on a 1–7 scale given by users to catalog items or implicit ratings, which are automatically inferred from a user's behavior. CF uses this repository of known preferences of a group of users to define predictions of unknown preferences for other users. Hence, the basic idea behind it is as follows: Users who had similar tastes or behaviors (e.g., reading, watching, buying, etc.) in the past will have similar tastes or behaviors in the future.

The CF algorithms represent the entire $m \times n$ user ratings as an matrix A. Each entry $a_{i,j}$ in the Matrix A represents ratings of the ith user on the jth item. To generate the top-n recommendations for the active user u, it is necessary to calculate the k most similar users or items (nearest neighbors) to user u. There are two different CF approaches to obtain the nearest neighbors, namely (1) memory-based (user-based) CF and (2) model-based (item-based) CF.

Given matrix A as input, the memory-based CF algorithms generate for the active user u prediction based on the ratings from similar user v, who rated the same items.

The prediction will be defined using the average ratings made by user u and user v and is a numerical value within the same scale like user's ratings, that is, from 1 to 7. Model-based CF algorithms pursue the same idea but use the similarity between items i and j rated by the active user u. Summarized, the prediction will be defined using (1) the average ratings made by user u and user v and using (2) the average ratings of similar items rated by the active user u. In both cases, the prediction is a numerical value and is within the same scale like user's ratings, that is, from 1 to 7.

A popular similarity measure in CF is the Pearson correlation, which measures the strength of the linear association between two variables and defines the direction (positive $+1$ or negative -1) of the association. To make a prediction, the memory-based algorithms add and subtract the neighbor's bias from the active user's average and use this as a prediction for the item i; in contrast, the model-based algorithms make a prediction by averaging the rating of similar items rated by the active user u. After the predictions are calculated, the items will be sorted in decreasing order based on their prediction value, put into a so-called top-n list, and recommended to the active user.

In contrast to the collaborative filtering–based recommender systems (CF-RS) that select items based on the similarity of user preferences, the content-based recommender systems (CB-RS) select items based on the correlation between the content of the items and the user's profile [Lops et al. 2011]. In CB-RS, the item content can be represented with a set of extracted terms or features. However, the personal comments and tags of a user can define her profile, since user-provided tags are assumed to describe user's tastes, needs, and interests. For VizRec, the items are chart descriptions in terms of visual encoding.

Usually, CB-RS uses a keyword-based Vector Space Model (VSM) together with basic TF-IDF weighting to determine the correlation between items and users. Transported in VSM, each item is represented as a vector of term weights, where each weight indicates the degree of association between the item and the term. Similarly to this, user profiles can be represented by profile vectors. Thus, using a cosine measure, the system can reveal the similarity between a profile and an item vector. Summarized, the generation of recommendations using content-based recommender systems is based on the matching of the attributes of an user profile (tags, comments, etc.) with the content properties (extracted terms, keywords, features, etc.) of an item.

However, the two presented recommender techniques, collaborative- and content-based filtering, have both advantages and shortcomings. The advantages of the CF include the content independency of the items being recommended, the low cost for knowledge acquisition and maintenance (no knowledge engineering is required), and ease of use. However, these recommendation techniques suffer from the so-called cold start problem [Schein et al. 2002]. The term *cold start* in the context of recommender systems generally characterizes the situation where a user has not yet provided her feedback to the system or when there is a new item transferred into the system, that is, no past information is available. Even though there is some feedback provided, the collaborative filtering mechanisms in particular sometimes fail to provide the results, since they become unable to find the corresponding user with similar tastes. This is often referred as s *data sparsity* problem [Good et al. 1999], implying that the collaborative filtering algorithm might be unable to form recommendations due to lack of information on user or item. In contrast, CB-RS do not require a direct user involvement in terms of, for example, providing ratings. Furthermore, these recommender techniques are capable of recommending items not yet rated by any user. However, the recommendations generated using content-based recommender systems can be too general, since the systems might capture only a certain aspect of the content. In this case the user might be recommended items similar to those she already rated or tagged without considering her interests changing over the time. One obvious solution for these problems is to combine different recommender systems to a hybrid recommender that uses the strength of all available recommender techniques. There are different methods for a hybrid design [Burke 2002; Jannach et al. 2010], including (i) weighted hybrids, (ii) switching hybrids, (iii) mixed hybrids, (iv) feature combination hybrids, (v) cascade hybrids, (vi) feature augmentation hybrids, and (vii) meta-level hybrids. Intuitively, for VizRec, a hybrid design combines data of what a visualization is about (tags) and how good it is at representing it (ratings) to propose between recommendations. We investigate the effects of three approaches, CF-RS, CB-RS, and Hybrid, in Section 4.

2.5. Crowd Sourcing Visualization Studies

Much of the data used in the studies in Section 4 were collected in a crowd-sourced visualization study. A concern with crowd-sourced studies is the lack of control over many experimental conditions, which may impact ecological validity. Nevertheless, perception studies in crowd-sourced platforms are viable, as evidenced by a growing number of successful studies in visualization and related fields [Kittur et al. 2008; Heer and Bostock 2010; Borkin et al. 2013; Lin et al. 2013]. Borkin et al. [2013] investigated memorability of visualizations. Considering visualizations much like a static picture, they performed a crowd-sourced study to determine which types of visualizations are better recalled. Investigating perceptual aspects of visualizations, Heer et al. replicated the influential experiments of Cleveland and McGill in the format of a crowd-sourced study [Heer and Bostock 2010]. Lin et al. performed a crowd-sourced experiment to determine semantically resonant colors, that is, colors that people associate with entities or effects, and derived guidelines for visualization [Lin et al. 2013]. Carefully designed tasks are mandatory to elicit valid data from crowd platforms. Kittur et al. discuss several design considerations for developing the tasks in crowd-sourced studies [Kittur et al. 2008]. One design recommendation is to have explicitly verifiable questions as part of a task. They found that asking tags for the content is useful because it requires users to process the content. Our intention was for participants to first analyze a visualization and then provide a rating for it. Hence, we used this guideline to setup a preparatory task where participants had to accurately study a visualization and prevent rash rating. Section 4 describes a crowd-sourced study designed to elicit user preferences related to automatically generated visualizations following aforementioned design recommendations.

3. THE VIZREC APPROACH

VizRec responds to a query with a list of personalized visualizations ordered in a top-n sorted manner. The query is a typical free-form text common in search engines. The response to the query is a dataset (containing relevant documents) compiled by a federated system from various associated sources, each with its proprietary data model. Before passing to VizRec, the data are structured after a common data model with a predefined schema. Within VizRec, two recommendation stages take place. First, a rule-based system applies visual encoding guidelines to generate a collection of visualizations appropriate for the data. Second, the collection is sorted and filtered according to user preferences using a recommendation strategy. The study in Section 4 investigates the strategies applied in the second recommendation stage.

Visual encoding guidelines are generic principles that establish relations between visual components of a visualization (e.g., x-axis of a bar chart) and elements of the data (e.g., whether a field is numeric, categorical, or a location, see Section 2.1). A preprocessing unit analyzes the data to structure them in terms of interesting data elements so visual encoding can take place. The three steps to generating personalized visualizations, summarized in Figure 1, are (1) preprocessing, (2) visual mapping, and (3) user preference filtering. This section further details and illustrates each unit with

Fig. 1. Schematic representation of the VizRec recommendation pipeline: The stages (a), (b), and (c) illustrate the preprocessing unit. Stage (d) illustrates the visual mapping process between the elements and the visual patterns, whereby the defined mapping combinations are shown in stage (e). For the personalized visualization recommendation VizRec uses the user preferences, user and item profiles, or a combined version of both (f). Finally, the recommendations will be presented to the user in a top-n manner (g). As shown, currently there are four types of visualizations integrated into the system (bar chart, timeline, line chart, geo chart).

a real example of generating visualizations for data obtained from MovieLens.[2] The example is an excerpt of the datasets used for the study in Section 4.

3.1. Preprocessing

The preprocessing unit models, extracts, and manages the input data. Furthermore, it addresses the task of prior organization of the visualizations into visual patterns that can be used to reify visualizations. The following describes how these stages go *from data to semantically enriched data* and *from visualization vocabulary to visual patterns* that can be used to actively derive appropriate visual encoding.

3.1.1. From Data to Semantically Enriched Data. Associated data sources, such as Mendeley, Europeana, ZBW (German National Library of Economics), ACM Digital Library, and so on, collect and index various kinds of information (books, journals, images, videos, etc.) in repositories structured according to a proprietary (often closed) data model. For instance, scientific digital libraries define the structure of literature archives in terms of some important metadata, such as title, abstract, author, keywords, following, for example, the Dublin Core metadata format.[3]

When it comes to working with the data from idiosyncratic data models in a holistic way, a unified data model offers the following benefits: (i) it simplifies the automated data processing, for example, in terms of extracting information, and (ii) enables interoperability with other applications, for example, the visualization tools. For VizRec, the input data are structured in a common data model following the specification in Orgel

[2] https://movielens.org/.
[3] http://www.dublincore.org/usage/documents/overview/.

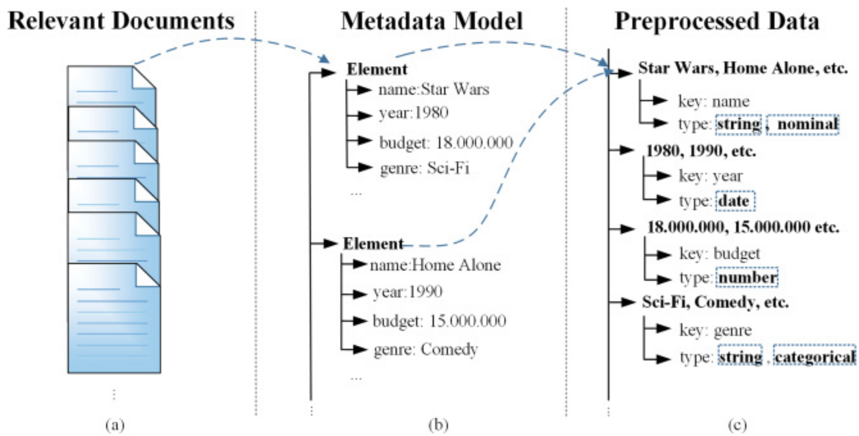


Fig. 2. Preprocessing: The input data (a) for VizRec are structured following a metadata model (b). The preprocessing unit is responsible, first, for the identification and extraction of the metadata elements and, second, for the data type analysis on the values of the metadata elements (c).

et al. [2015] (see Figure 2(b)). The unified model organizes metadata elements extracted from the original sources (such as title, content information, document type, and attributes). A mapping ontology defines the relation for each concrete metadata format and the unified data model used in VizRec.

The responses to the user's query are cached and separately translated into the unified model. The above-mentioned digital repositories have specific services to obtain their data with various interfaces for the access, such as JavaScript Object Notation (JSON), Resource Description Framework (RDF), or Extensible Markup Language (XML). In contrast, linked data have a graph structure connecting data that originate from different sources. Data obtained from *DBpedia* or MovieLens are translated locally in the common metadata model. Finally, a simple matching operator merges the metadata of each dataset together and presents them as a single dataset. As example, a user interested in the budget and income of movies at the end of last century obtains Listing 1 for the query *Top 10 successful movies filmed in 1960, 1970, 1980, and 1990*. The dataset merges *movie name*, *genre*, and *year* obtained from MovieLens with *budget* and *gross* information obtained from DBpedia. The above-mentioned metadata mapping methods are beyond the scope of VizRec and are only mentioned here for completeness. For further details, refer to Orgel et al. [2015].

Once the data are obtained, the preprocessing unit of VizRec carries out four important technical steps. First, metadata extraction, the metadata elements (i.e., movie name, genre, year, budget, and gross) are automatically identified and their values extracted following the metadata model (see Figure 2(b)). Second, data type categorization and extracted values are collected in series and a data analysis step categorizes them into standard data types, such as categorical, temporal, and numerical—represented by primitive data types string, date, and number, respectively (see Figure 2(c)). Third, semantic extraction, if required, using gazetteer lists specialized data types are derived, for example, spatial information like coordinates are obtained for metadata elements belonging to the term *country*. Fourth, enrichment, extracted elements enriched with categorized values are passed to the mapping algorithm to execute the mapping process (see Section 3.2).

3.1.2. From Visualization Vocabulary to Visual Patterns. Formally, a visualization can be broken down in a number of r visual components, each of which encodes a single

```
<?xml version="1.0" encoding="UTF−8" standalone="yes" ?>
<description> The top 10 successfully movies filmed at 1960, 1970, 1980 and 1990</description>
<results>
    <result>
        <facets>
            <provider>DBpedia</provider>
            <type>Linked Open Data</type>
            <moviename>Star Wars: Episode V</moviename>
            <genre>Sci-Fi</genre>
            <year>1980</year>
            <budget>18.000.000</budget>
            <gross>290.158.751</gross>
        </facets>
    </result>
    <result>
        <facets>
            <provider>DBpedia</provider>
            <type>Linked Open Data</type>
            <name>Home Alone</name>
            <genre>Comedy</genre>
            <year>1990</year>
            <budget>15.000.000</budget>
            <gross>285.761.243</gross>
        </facets>
    </result>
    ...
</results>
```


Listing 1: Exemplary input data for the *VizRec* obtained from MovieLens and enriched with data from DBpedia.

piece of information visually [Bertin 1983]. One can naïvely think that every visual component may encode any kind of data. Thus, the possible number of combinations for one visualization is the permutation relation [Gilson et al. 2008]:

$$C_r^n = \frac{n!}{(n-r)!}, \tag{1}$$

where n is the number of metadata elements in the dataset (i.e., number of fields). For example, a simple bar chart has three visual components: x, y, and *color*. The example dataset in Listing 1 has five metadata elements ($n = 5$), so the total number of combinations for the bar chart ($r = 3$) is

$$\frac{n!}{(n-r)!} = \frac{5!}{(5-3)!} = 60. \tag{2}$$

So the number of options a user would have to consider is rather high even for a simple chart, without considering alternative visualizations. The fact is that many of these combinations are perceptually incorrect, since visual components are often suited to represent only certain metadata given by the perceptual properties of the component and the characteristics of the metadata [Bertin 1983]. To prevent this, VizRec uses visual patterns to explicitly define which metadata element is related to which visual component of a visualization type [Rahm and Bernstein 2001].

In VizRec visualizations are described in a Visual Analytics vocabulary[4] (VA vocabulary) representing visualizations in a common persistence model that can be reused by various technologies. The VA vocabulary is an explicit conceptualization that describes the semantics of visualizations in pragmatic, simple facts that will aid the sensible mapping from data. It consists of two parts: (1) the model of an abstract visualization specifying structural components that any concrete visualization may

[4]http://code.know-center.tugraz.at/static/ontology/visual-analytics.owl.

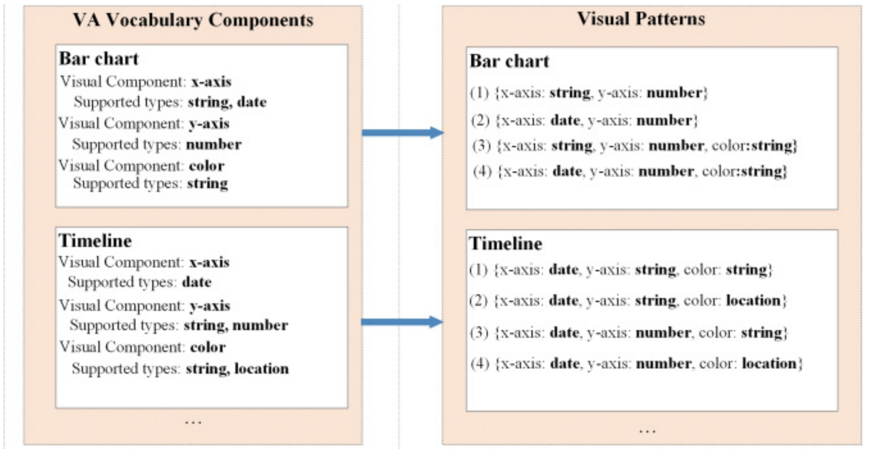


Fig. 3. Visual patterns for the bar chart and timeline defined in the description vocabulary.

have the following: (a) *name*, (b) *visual components*, (c) *description*; and (2) the model of a concrete visualization refining the abstract visualization model by reification of the visual components. Visual components are characterized by (i) *data type* (set of primitive data types that a visual component supports), (ii) *occurrence* (cardinality, that is, how many instances are allowed for the visual component), (3) *persistence* (whether a visual component is mandatory for the concrete visualization). Currently, the VA Vocabulary is used to describe four types of visualizations—bar chart, timeline, line chart, and geo chart. Additional visualizations can be integrated in a straightforward way by just following the specification of the vocabulary as shown in Listing 2.

Listing 2 describes the bar chart, with its three visual components, x, y and *color*, each of which has its own unique properties. Concretely, (i) each visual component supports a specific data type, (ii) the visual component x and y are mandatory, *color* is optional, and (iii) each component has to be instantiated only once to be able to produce valid combinations (in further text: mapping combinations or just mappings).

Visual patterns result from the fact that, depending on the properties of a visualization, a visual component can support different data types in different combinations. Having described visualizations in terms of visual components and supported data types, visual patterns can be derived, each describing one possible configuration of a visualization [Mutlu et al. 2014] (see Figure 3). In order words, the patterns specify the types of data that are required for each visualization to be instantiated. For instance, following bar chart description in Listing 2, two possible patterns for the bar chart are (1) $\{x - axis : string,\ y - axis : number\}$ and (2) $\{x - axis : date,\ y - axis : number\}$, representing the fact that the $x - axis$ can accept both type of data but not at the same time.

In addition, the presence of an optional visual component results in two additional patterns, concretely (3) $\{x - axis : string,\ y - axis : number,\ color : string\}$ and (4) $\{x - axis : date,\ y - axis : number,\ color : string\}$. These patterns will be instantiated only if there is a value that exists for *color*, otherwise the system would select patterns (1) and (2).

Using visual patterns, the system is able to generate all mapping combinations that are plausible for the data and perceptually correct regarding visual encoding guidelines. In the following, we further detail how VizRec operates to instantiate the appropriate visual patterns—the visual mapping process.

```
<rdf:RDF xmlns:rdfs="http://www.w3.org/2000/01/rdf−schema#" xmlns:vo="http://eexcess.eu/visualisation−ontology#
      " xmlns:va="http://code−research.eu/ontology/visual−analytics#" xmlns:rdf="http://www.w3.org/1999/02/22−rdf
      −syntax−ns#">
  <va:BarChart rdf:about="http://eexcess.eu/visualisation−ontologyBarchart">
    <rdfs:label>barchart</rdfs:label>
    <vo:hasVisualComponents>
      <vo:Axis rdf:about="http://eexcess.eu/visualisation−ontologyBarChartXAxis">
        <rdfs:label>x-axis</rdfs:label>
        <vo:supportedDataType rdf:resource="http://eexcess.eu/visualisation−ontology#string" />
        <vo:supportedDataType rdf:resource="http://eexcess.eu/visualisation−ontology#date" />
        <va:hasPersistence rdf:resource="http://code−research.eu/ontology/visual−analytics#Mandatory" />
        <va:hasOccurrence rdf:resource="http://code−research.eu/ontology/visual−analytics#One" />
      </vo:Axis>
    </vo:hasVisualComponents>
    <vo:hasVisualComponents>
      <vo:Axis rdf:about="http://eexcess.eu/visualisation−ontologyBarChartYAxis">
        <rdfs:label>y-axis</rdfs:label>
        <vo:supportedDataType rdf:resource="http://eexcess.eu/visualisation−ontology#number" />
        <va:hasPersistence rdf:resource="http://code−research.eu/ontology/visual−analytics#Mandatory" />
        <va:hasOccurrence rdf:resource="http://code−research.eu/ontology/visual−analytics#One" />
      </vo:Axis>
    </vo:hasVisualComponents>
    <vo:hasVisualComponents>
      <vo:Axis rdf:about="http://eexcess.eu/visualisation−ontologyBarChartColor">
        <rdfs:label>color</rdfs:label>
        <vo:supportedDataType rdf:resource="http://eexcess.eu/visualisation−ontology#string" />
        <va:hasPersistence rdf:resource="http://code−research.eu/ontology/visual−analytics#Optional" />
        <va:hasOccurrence rdf:resource="http://code−research.eu/ontology/visual−analytics#One" />
      </vo:Axis>
    </vo:hasVisualComponents>
    <va:hasDescription>
      <rdfs:label>Bar Chart is a diagram that presents the numerical values of variables by the length of
          bars.</rdfs:label>
    </va:hasDescription>
  </va:BarChart>
</rdf:RDF>
```

Listing 2: Description of the Bar chart using the VA Vocabulary.

3.2. Visual Mapping

The visual mapping process can be considered as a schema matching problem [Rahm and Bernstein 2001]. The basic idea behind schema matching is to figure out a semantic relevance between two objects in schemas under consideration. The result is a mapping comprising a set of elements, each of which indicates that certain elements of schema S1 are related to certain elements of schema S2. In our case, the schemas we deal with are, on the one hand, the metadata model, which describes the semantics of the input data, and, on the other hand, the VA Vocabulary, which describes the semantics of the visualizations. Hence, the schema matching in our context produces mappings, each of which describes the correspondence between a metadata element and a visual component of a visualization to define a possible configuration. In the following, we describe this process more in detail.

The relation from elements of the input data to components of a visualization is valid only if we can establish syntactic correspondences between the metadata and the visualizations. One possibility to identify this is to verify the data type compatibility. Data type compatibility in our context means having exactly the same data types, conforming to the *XSD data type definition.*[5] The preprocessing unit provides patterns for visualizations and a common model for the input data both including the data types of their elements. From the specifications of the visual patterns, the mapping operator compares the data types of the visual components and metadata with each other and builds a list of plausible mappings (see Figures 4(c) and (d)).

[5] http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.

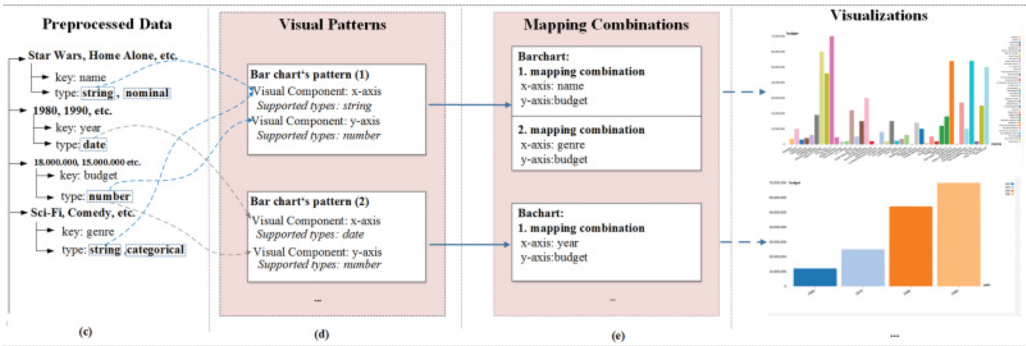


Fig. 4. Visual mapping process.

Beyond the data type compatibility, a valid mapping needs to account for structural compatibility, since visualizations have either fixed or varying number of visual components. To generate a visualization, the mapping operator has to instantiate every mandatory visual component while the pattern(s) including optional components can be ignored if there is no corresponding data element. Formally, each pattern i defines for each visual component j which r_j metadata element should be selected from n_j metadata elements:

Note that n_j is a subset of n that complies with data type compatibility for the j visual component. To obtain the total number of combinations M_i, generated for a particular pattern i, we multiply every suitable $\binom{n_j}{r_j}$ notation of a pattern:

$$M_i = \prod C_{n_j}^{r_j}, \tag{3}$$

$$\frac{n_j!}{r_j!(n_j - r_j)!} = \binom{n_j}{r_j} = C_{n_j}^{r_j}. \tag{4}$$

Thus, the final number of combinations M for a visualization is nothing else but the sum of every M_i:

$$M = \sum \{M_i\}. \tag{5}$$

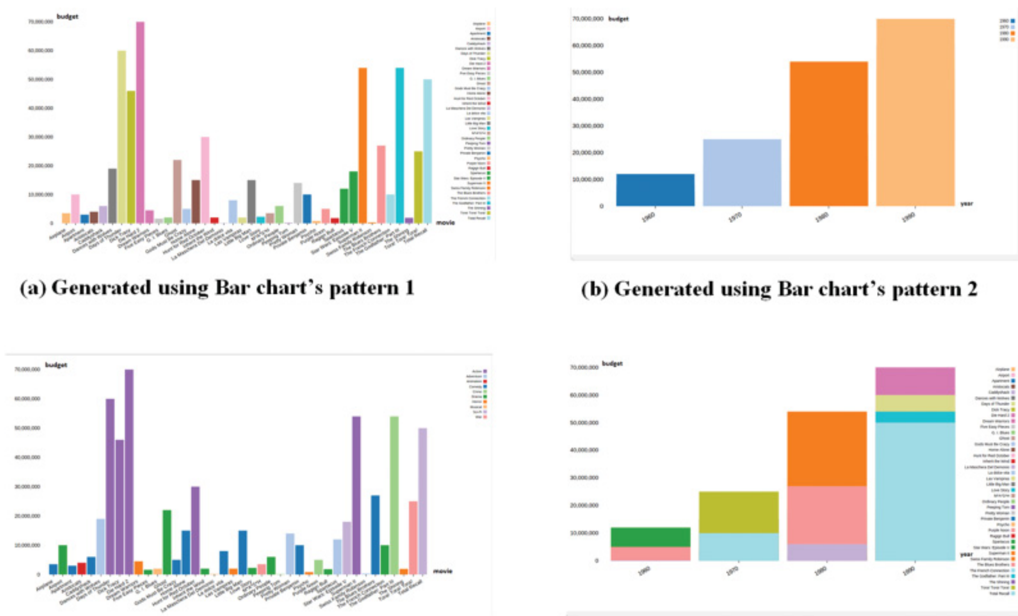
Continuing the example about successful movies, VizRec considers the following facts: (i) the underlying dataset contains two *string* values (*movie name*, *genre*), one *date* (*creation year*), and two *numbers* (*budget and gross*); and (ii) only the patterns, which accept categorical/nominal (*string*), temporal (*date*), and numerical values (*number*) are appropriate. Thus, geographical visualizations will not be further considered by the system.

According to the pattern description from Listing 2, the bar chart complies with facts (i) and (ii). Using visual patterns, the system selects bar chart pattern (1) counting exactly one element with data type *string* and one with data type *number*, producing:

$$M_1 = C_2^1 \times C_2^1 = \binom{2}{1} \times \binom{2}{1} = 4 \tag{6}$$

mapping combinations, containing, for example, $\{x - axis : movie\ name,\ y - axis : budget\}$ (see Figure 5(a)). For pattern (2) the system selects one metadata element with the data type *date* and one with datatype *number*, obtaining

$$M_2 = C_1^1 \times C_2^1 = \binom{1}{1} \times \binom{2}{1} = 2 \tag{7}$$



(a) Generated using Bar chart's pattern 1

(b) Generated using Bar chart's pattern 2

(c) Generated using Bar chart's pattern 3

(d) Generated using Bar chart's pattern 4

Fig. 5. Some of the bar chart combinations generated for the dataset *movies* using the bar chart patterns (1), (2), (3), and (4). The patterns are listed in Table I.

mapping combinations containing, for example, $\{x-axis : creation\ year, y-axis : budget\}$ (see Figure 5(b)). For pattern (3) VizRec selects one element with data type *string*, one with data type *number* and another one with data type *string*, so

$$M_3 = C_2^1 \times C_2^1 \times C_2^1 = \binom{2}{1} \times \binom{2}{1} \times \binom{2}{1} = 8 \tag{8}$$

mapping combinations are determined containing for example, $\{x-axis : movie\ name, y-axis : budget, color : genre\}$ (see Figure 5(c)). Note that these mappings contain redundant mapping, that is, an element can be selected for more than one visual component. For instance, for this pattern the element *movie name* can be mapped once on the $x-axis$ and once on the *color* (see Table I Pat.num (3)) since both components support the datatype *string*. Redundant mapping with optional visual channels reinforces aspects of the data—in this case, the entity movie name.

Applying the same approach for pattern (4) results in 4 possible mapping combinations containing, for example, $\{x-axis : creation\ year, y-axis : budget, color : movie\ name\}$ (see Figure 5(d)). The total number of perceptually valid combinations is 18 (see Equation (5) and Table I). Comparing with the naïve result (60) the number of visualizations to consider is reduced considerably. Yet, the example concentrated only on a single chart (bar charts), each type of chart adds another number of visualizations that may be useful for the user. Furthermore, some users may be more inclined to use one type of chart than other to spot what they are looking for in the data. In the following section, we consider recommendation strategies to filter results according to user preferences.

Having obtained all valid mapping combinations (see Figure 1(e)), the mapping operator maps the values of the selected metadata elements to the corresponding visual components of a visualization and presents them to the user as a set of appropriate

Table I. Mapping Combinations Defined for the Exemplary Dataset *Movies* Using Bar Chart's Visual Patterns (1)–(4). The Visualizations Shown in the Last Columns Are Generated for the First Mapping Combination of Each Pattern to Give an Example for Instantiated Mapping Combinations

Pat. num.	Visual Patterns	Mappings	Vis.
1	*{x-axis:string, y-axis:number}*	*{x-axis:movie name, y-axis:budget}* *{x-axis:movie name, y-axis:gross}* *{x-axis:genre, y-axis:gross}* *{x-axis:genre, y-axis:budget}*	
2	*{x-axis:date, y-axis:number}*	*{x-axis:creation year, y-axis:budget}* *{x-axis:creation year, y-axis:gross}*	
3	*{x-axis:string, y-axis:number, color:string}*	*{x-axis:movie name, y-axis:budget, color:genre}* *{x-axis:movie name, y-axis:gross, color:genre}* *{x-axis:movie name, y-axis:budget, color:movie name}* *{x-axis:movie name, y-axis:gross, color:movie name}* *{x-axis:genre, y-axis:budget, color:movie name}* *{x-axis:genre, y-axis:gross, color:movie name}* *{x-axis:genre, y-axis:budget, color:genre}* *{x-axis:genre, y-axis:gross, color:genre}*	
4	*{x-axis:date, y-axis:number, color:string}*	*{x-axis:creation year, y-axis:budget, color:movie name}* *{x-axis:creation year, y-axis:gross, color:movie name}* *{x-axis:creation year, y-axis:budget, color:genre}* *{x-axis:creation year, y-axis:gross, color:genre}*	

visualizations. The various mapping combinations present different analysis scenarios and thus can cater to wider range of user needs and interests.

The pseudo-code 1 in Listing 1 summarizes the essential steps performed by the mapping algorithm. Initially, for a given dataset (cf. Figure 1), relevant visual patterns are identified from the existing visualization collection. Based on those patterns, the schema matching part of the algorithm identifies the concrete configurations for visualizations that are compatible with the data provided in the data types and the structure. Candidates complying with these rules are valid mapping combinations that are in further steps of the VizRec pipeline used for the detailed, personalized filtering.

3.3. User Preference Filtering

Visual patterns together with rule-based mapping algorithms generate all mapping combinations that are plausible for the data. However, not all of them represent what the user needs or prefers. For example, the bar chart in Table II with the item-id 541 shows the yearly distribution of each movie's budget (generated for bar chart pattern (2)) without displaying to which movie the budget belongs to. Thus better mechanisms for selecting the visualization are required. The first approach we investigate is CF [Schafer et al. 2007], which relies on explicit feedback provided by the user in form of ratings. Ratings alone do not tell much about the content of the data that a visualization represents. To take this aspect into account, we investigate content-based filtering [Lops et al. 2011]. Content-based filtering requires metadata information, for example, in the form of keywords, comments, or tags provided by the user. VizRec uses tags, as they have been shown to be useful in many recommender or information retrieval scenarios [Larrain et al. 2015]. Finally, VizRec includes a hybrid recommendation approach that combines information on what a visualization is about (tags) and how good it is (ratings). This section further provides technical details of the recommendation strategies included into VizRec.

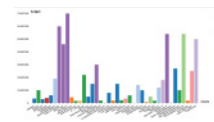
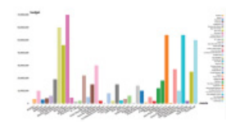
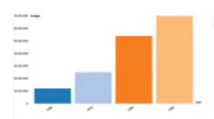
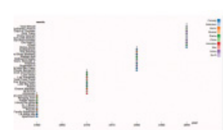
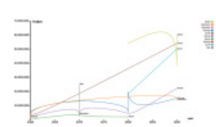
3.3.1. Collaborative Filtering. To filter the mapping combinations M based on the user preferences (see Figure 1(f)), we employ a simple user-based CF approach utilizing ratings [Su and Khoshgoftaar 2009]. The basic idea behind CF is to find a user with

ALGORITHM 1: Simplified Algorithm for Determining Appropriate Mapping Combinations

```
Data: set(data_element) // retrieved and preprocessed content from data sources
Result: set(mapping_combination)
// result set
set(mapping_combination) ← empty set;

// map containing visualizations and their visual patterns
map(visualization, set(visual_pattern)) ← empty map;

// first step: collect all available visual patterns
set(visualization) ← get all visualizations from repository;
while set(visualization) not empty do
    visualization ← take current visualization from the set;
    set(visual_components) ← get visual components from visualization;

    // The generation is based on component attributes: occurrence, persistence
    set(visual_pattern) ← generate patterns out of set(visual_components);

    // Store generated patterns
    map(visualization, set(visual_pattern)) ← append pair (visualization,set(visual_pattern));

// second step: identify mappings based on visual patterns
while map(visualization, set(visual_pattern)) not empty do
    visualization ← take current vis. pair (visualization, set(visual_pattern)) from set;
    while set(visual_pattern) not empty do
        visual_pattern ← take current pattern from set;
        while set(data_element) not empty do
            // The structure is evaluated based on a number of visual components
            // within a pattern
            if structural match between (visual_pattern) and (data_elements) then
                // Datatype match is performed between visual components
                // and individual elements of the current data
                if datatype match between (visual_pattern) and (data_elements) then
                    // Elements of the current data are mapped (linked)
                    // to the corresponding visual components of the current pattern
                    mapping_combination ← map elements to visual components;
                    set(mapping_combination) ← append mapping_combination;
                else
                    continue;
            else
                continue;
```

similar preferences to the active user, who has rated the item x that the active user has not seen yet. Hence, the average ratings of the similar users are applied to predict if the active user will prefer the item x. In a nutshell, the algorithm needs to identify users similar to the active user, k-nearest neighbors, respectively, who share active user's tastes. To calculate the k-nearest neighbors, we construct a $m \times n$ matrix A where each entry $a_{i,j}$ represents the rating of the ith user on the jth item (mapping combination). Each rating is a numerical scale, for example, from 1 to 7. Having constructed the matrix A we employ the Pearson correlation coefficient to calculate the similarity

Table II. Input Data for the Calculation of the K-Nearest Neighbors and Generating Predictions for the Active User, Including Item-id, User-id, and Ratings

Item-id	**User-id**	**Rating**	**Item**
254	1 6 10	4.0 4.5 5.5	
960	1 6 10	6.5 5.5 4.0	
541	1 6 10	2.5 3.0 3.5	
721	1 6 10	1.0 2.5 2.0	
360	6 10	5.5 6.5	
...	...	...	...

between the active user u and the user v using Equation (9):

$$sim(u,v) = \frac{\Sigma_{i \in I}(r_{u,i} - \bar{r_u})(r_{v,i} - \bar{r_v})}{\sqrt{\Sigma_{i \in I}(r_{u,i} - \bar{r_u})^2}\sqrt{\Sigma_{i \in I}(r_{v,i} - \bar{r_v})^2}}, \tag{9}$$

where I is the set of items rated by users u and v and $\bar{r_u}$ is the average rating of the active user u. Once the k-nearest neighbors are detected, VizRec combines the preferences of the neighbors to generate the predictions or the top-n recommendations for the active user, the set R, respectively, following Equation (10):

$$pred_{cf}(u,i) = \bar{r_u} + \frac{\Sigma_{v \in N} sim(u,v)(r_{v,i} - \bar{r_v})}{\Sigma_{v \in N} sim(u,v)}, \tag{10}$$

where $\bar{r_u}$ and $\bar{r_v}$ are the average ratings of the user u and v. To do so, VizRec selects from the set of M only those mappings that the active user might prefer and presents them to her as recommendations (see Figure 1(g)). The list of recommendations R for the active user is nothing else but a subset of M.

The cooperation of the mapping algorithm with CF offers two important benefits: first, the definition of perceptually valid visualizations for the active user's dataset and, second, the recommendation of only the valid visualizations that the active user might prefer.

To clarify this, we consider our example about the top-ranked movies again. The visual mapping unit defines for this dataset a list with 18 possible bar chart configurations. However, the system contains also alternative visualizations, such as timeline and line chart, which are also appropriate for the current dataset.[6] When including their mapping combinations, 6 for timeline and 4 for line chart, respectively, the total sum of available visualizations would be 28. Since the preferences of the active user u are known by the system, it can reduce the list on those to which the user's preferences matches the best. To do so, the system, first, performs a k-nearest-neighbor search by employing the Pearson correlation measure to detect those users who are the most similar to the active user u. Table II shows an excerpt of our example rating database containing ratings for items (visualizations) that has been seen by the active and/or other users whereby the active user is assigned the user-id 1. Second, using Equation (9) for the Pearson correlation and the Table II with item-ids and user-ids and their ratings, the system reveals following similarity values for the active user u:

$$sim(1,6) = \frac{(4.0-3.5)(4.5-4.2)+\cdots+(1.0-3.5)(2.5-4.2)}{\sqrt{(4.0-3.5)^2+\cdots+(1.0-3.5)^2}\sqrt{(4.5-4.2)^2+\cdots+(2.5-4.2)^2}} = 0.9806,$$

$$sim(1,10) = \frac{(4.0-3.5)(5.5.0-4.3)+\cdots+(1.0-3.5)(2.0-4.3)}{\sqrt{(4.0-3.5)^2+\cdots+(1.0-3.5)^2}\sqrt{(5.5-4.3)^2+\cdots+(2.0-4.3)^2}} = 0.6154.$$

Having detected the similarity values for *user*6 and *user*10 the system tries to predict whether the active user u might prefer the visualization with the id 360 she has not seen before:

$$pred_{cf}(1,360) = 3.5 + \frac{0.9806(5.5-4.2)+0.6154(6.5-4.3)}{0.9806+0.6154} = 5.1465.$$

Considering the prediction value for item 360, we can assume that this item might be one of the 10 items being recommended to the active user. To finally define the list of top-10 recommendations, we apply this approach to every user and item the active user has not seen before. Using the similarity values, we rank each item i of the k most similar users to the active user and present her only the visualizations with the highest ranking.

3.3.2. Content-Based Filtering. CF-RS needs the user's interests beforehand, which should be in common to at least a few other users. When CF-RS cannot find similar users, that is, the case when the user or the item is new to the system, a CB-RS is a suitable alternative. The simplified workflow of the VizRec CB-RS is illustrated in Figure 1(f). In a nutshell, the VizRec CB-RS generates recommendations by analyzing the relevant content and, concretely, the information we know about the active user and the information we extracted from the items. Following the basic principles of CB-RS, the recommendations are produced based on the content similarity, in our case between the interests of the active user, that is, her profile, and the content of the candidate items (visualizations).

User and Item Profiles: Each visualization generated in VizRec is described with a mapping from metadata to visual components. The metadata elements provide

[6]Geo charts were not generated, since the dataset does not contain spatial information.

basic information about the content of the visualization they describe. Thus, we build the item profiles based on the current set of mapping combinations (visualizations) the active user observes from the Visual Mapping stage. For instance, when the user's dataset is about the *movies* and the mapping combination $\{x - axis : movie\ name,\ y - axis : budget,\ color : genre\}$ (cf. Figure 5(c)) is one of the candidates, VizRec uses the related metadata elements, *movie name*, *budget*, and *genre*, to profile this particular mapping combination. Yet a user annotates items with tags that describe the content of the items and thus serve also as appropriate inputs for their profiles [Bogers and Van den Bosch 2009; Lin et al. 2015]. To take this into account, VizRec extends the item profiles with tags supplied by users in the past. To relate the item profiles with the content in the repository, we build an item profile by aggregating the tags of all users per item (visualization). The benefits of the tag aggregation among all users is (i) to obtain more valuable information about individual items than focusing just on information from a single source and (ii) to consider the preferences of the community and not just those of a single user, which in turn increases the likelihood that we select items that are relevant for the active user (cf., Bogers and Van den Bosch [2009]). In summary, we build a profile of an item (visualization) based on (i) the metadata they contain, including (if available) (ii) the tags user supplied to this item in the past. Similarly to the item profiles, the user profiles are built on the user-applied tags. Usually, a user's tags reflect her interest and needs. Thus, we profile each user with her tags, which are assumed to describe her interest in a topic or/and an item.

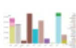
One important concern regarding tags for the user and item profiles is a normalization process that is executed before storing tag information in repository. This process involves (i) removing commoner morphological and inflectional endings from English words (e.g., movies $\longrightarrow$ movie, comedies&comedy $\longrightarrow$ comedi) using the Porter stemmer algorithm [Karaa and Gribâa 2013]; (ii) removing stop words (standard tokenizer) and punctuations (keyword tokenizer); and, finally, (iii) the lowercase filtering. This step helps to avoid that the words represented in various language forms are interpreted differently [Lops et al. 2011].

Similarity Estimation and Item Ranking: To determine the correlation between visualizations and users, we transform the content of the user profiles and item profiles into the Vector Space Model (VSM) with the Term Frequency-Inverse Document Frequency (TF-IDF) weighting schema. As mentioned in Section 2.4, VSM is a common technique to vectorize the content and in this way to enable their analysis, such as classification and clustering, for example. In our case, VSM consists of user profile (tags) and item profile (mapping combinations), both represented in the form of vectors. Concretely, using this scheme, each mapping combination (e.g., *movie name*, *budget*, *genre*) is defined as an n-dimensional vector, where each dimension corresponds to a term or, more precisely, to the TF-IDF weight of that particular term. To clarify this, let $M = \{m_1,\ m_2,\ m_3, \ldots, m_N\}$ be a set of mapping combinations and $T = \{t_1,\ t_2,\ t_3, \ldots, t_n\}$ a set of terms in M. Each mapping combination m_i is represented as a vector in a n-dimensional vector space, that is, $m_i = w_{1,i}, w_{2,i}, w_{3,i}, \ldots, w_{n,i}$, where $w_{k,i}$ denotes the weight for the term t_k in a mapping combination m_i, that is,

$$w_{k,i} = tf - idf_{t_k,m_i} \;=\; tf_{t_k,m_i} \times idf_t \;=\; tf_{t_k,m_i} \times \left[log_e \left(\frac{N}{df_t + 1} \right) + 1 \right], \tag{11}$$

where the former factor of the product is an occurrence frequency of the term t_k within a mapping combination m_i, and the latter indicates the distribution of the term among the both profiles (i.e., so particular and commonly occurring terms can be discriminated from each other). We apply the same weighting scheme to define the user profile. Having

Table III. An Excerpt of the *Movies* Dataset with Generated User and Item Profiles

User profile		
ID	**Terms (frequency)**	**TF-IDF Vector (weight)**
25	movie(5), revenue(2), genre(12), gross(8), collections(2), films(2), most(1), successful(1), film(2), top(1), grosser(1), movies(5), years(1), earnings(1), year(2), money(1), decades(2), profit(1), box(1), office(1), profits(1), decade(1), genres(2)	movi:(4.130), revenu:(3.974), genr:(5.791), gross:(5.420), collect:(4.548), film:(3.498), most:(3.216), success:(4.602), top:(3.061), grosser:(4.314), year:(2.670), earn:(2.928), monei:(3.503), decad:(4.298), profit:(4.232), box:(2.810), offic:(2.810)
Item profile		
ID	**Terms (frequency)**	**TF-IDF Vector (weight)**
..	...	...
 *38*	genre(1), movie(1), gross(1), ... , barchart(1)	genr:(0.749), movi:(0.749), gross:(2.174), ... , barchart:(1.510)
..	...	...
 *46*	genre(1), year(1), gross(1), ... , barchart(1)	genr:(0.749), year:(1.541), gross:(2.174), ... , barchart:(1.510)
 *47*	genre(1), year(1), budget(1), ... , barchart(1)	genr:(0.749), year:(1.541), budget:(2.174), ... , barchart:(1.510)
..	...	...

defined the profiles, it is now possible to estimate their similarity. To do so, we use the weighting information in vectors and apply the *cosine similarity* measure [Lops et al. 2011], defined as follows:

$$sim(m_i, m_j) = \frac{\sum_k \mathbf{w}_{k,i}\mathbf{w}_{k,j}}{\sqrt{\sum_k (\mathbf{w}_{k,i})^2}\sqrt{\sum_k (\mathbf{w}_{k,j})^2}}, \tag{12}$$

where m_j denotes the tag collection of the current user. The result of this measure is a cosine value of the angle between two vectors, in our case between the mapping combination and the tag collection. The retrieved values are then used as scores to rank the relevant visualizations.

Returning to the example of top-ranked movies filmed in a certain period of time, but now including the shooting location (country) and the population of each country in the results, the mapping algorithm would produce a total of 55 visualizations for four types of visualizations for the active user—including geo chart. Yet, to ascertain which of these visualizations the current user would prefer the most, we define the user's profile by aggregating her tags. An excerpt of the user profile is shown in the top part of Table III. Subsequently, the item profiles are defined by extracting relevant terms (metadata elements) from the individual visualizations. The summarized terms represent here the actual content in each particular visualization, for instance, *genre*, *movie*, and *gross* dimensions of the dataset are displayed in the bar chart (the second row). Note that this profile information can be augmented by additional tags if available. An excerpt of the item profile generated from metadata elements of each visualization is shown in the bottom part of Table III.

The TF-IDF vectors are shown next to the terms in Table III. For example, the tag *genre* in the first mapping combination from the table has a TF-IDF weight of *0.749*. The term occurs only once in this mapping, that is, $tf_{genre,mapping38} = 1$, and in 25 other mappings of 55 overall mappings, that is, $df_{genre} = 25$ and $N = 55$, respectively, so

Table IV. Recommendations for the Weighted Hybrid Recommender

Item	rec_1 score	rec_1 rank	rec_2 score	rec_2 rank	$hybrid$ score	$hybrid$ rank
Item1	0.3	2	0.7	1	0.5	1
Item2	0.5	1	0.3	3	0.4	2
Item3	0.2	3	0.5	2	0.35	3
Item4	0		0.1	4	0.05	

$tf-idf_{genre,mapping38} = 1 \times (log_e(\frac{55}{25+1}) + 1) = 0.749$. Note that here stemmed words are used instead of original ones (i.e., movie $\longrightarrow$ movi).

In the final step, scores are assigned to individual mapping combinations by comparing vectors of these mappings with vectors from the user profile using Equation (12). Concretely in this example, the best matching could be found between mappings 25 (user profile) and 46 (item profile). In a nutshell, their cosine value would be estimated as follows:

$$sim\left(\begin{bmatrix}0.749\\1.541\\\cdots\\1.510\end{bmatrix}, \begin{bmatrix}4.130\\3.974\\\cdots\\2.810\end{bmatrix}\right) = \frac{0.749 \cdot 4.130 + 1.541 \cdot 3.974 + \cdots + 1.510 \cdot 2.810}{\sqrt{0.749^2 + \cdots + 1.510^2} \cdot \sqrt{4.130^2 + \cdots + 2.810^2}} = 0.398.$$

Hence, the mapping combination 46 with the tags *genre*, *year*, *gross*, and *barchart* might be in this case one of the top-n preferred visualizations for the active user. Finally, having obtained the similarity values for each mapping combination we define the top-n visual recommendations, following the equation:

$$pred_{cb}(m_i, m_j) = \Sigma_{m_i, m_j \epsilon M} sim(m_i, m_j). \tag{13}$$

3.3.3. Hybrid Filtering. The two pieces of information used for recommendation separately describe what a visualization is about (CB-RS) and how good it is (CF-RS). A combination of these pieces of information in a single recommendation strategy would arguably supply more meaningful recommendations in varying situations (e.g., when the user or item is new, when the user's interest changes). In general, there exist different methods for a hybrid design [Burke 2002; Jannach et al. 2010] (see Section 2).

For the current investigation, we have chosen a weighted hybridization design as a first approach in VizRec to utilize the strength of both collaborative filtering and content-based recommender techniques in the straightforward way. Concretely, a weighted hybrid recommender defines the score of a recommended item from the results of all integrated recommender techniques by computing a weighted sum of their scores. When linearly combining the scores, the collaborative and content-based recommender obtain equal weights. Thus, we use the uniform weighting scheme with $w_1 = w_2 = 0.5$ for our hybrid recommender and define a new ranking for the recommended items by combining their (normalized) scores from collaborative and content-based recommender following the Equation (14):

$$pred_{hyb}(u, i) = \Sigma_{j=1}^{n} w_j rec_j(u, i). \tag{14}$$

To clarify this process, we consider the Table IV containing scores and rankings for five exemplary items. According to this table, rec_1 (CF-RS) produced for $Item1$ rank 2 and rec_2 (CB-RS) the rank 1 considering the scores 0.3 and 0.7. When linearly combining those scores following the Equation (14), we obtain for $Item1$

$$pred_{hyb}(u, Item1) = 0.5 \times 0.3 + 0.5 \times 0.7 = 0.5 \tag{15}$$

as the final score. Having computed the hybridized scores for the remaining items, $Item1$ will be finally ranked highest following $Item2$ and $Item3$.

4. EVALUATION

In this section, we investigate the performance of different recommendation strategies. To do so, we design a study on a crowd-sourced platform to elicit preferences and tags for a fix number of visualizations associated with three different datasets. This section describes in detail the data sources, the method, and metrics used and the studies of recommendation strategies.

4.1. Datasets and Mappings

The study used the following three open-source datasets:

MovieLens[7] *dataset* (Movies): This dataset comprises information about the top-ranked movies for the years 1960, 1970, 1980, and 1990. It has 40 entries, which are selected from items of the respective dataset and are characterized by the elements (movie) name, genre, budget, gross, creation year, shooting location (country), and population of the country. Based on this, the mapping unit produced four types of visualizations using the method described in Section 3.2 with the following mapping frequencies: 32 bar charts, 9 line charts, 13 timelines, and 1 geo-chart. Hence, a total of 55 mapping combinations were generated.

EU Open Linked Data Portal[8] *dataset* (Eu): The *Eu* dataset collects the percentage of the population looking for educational information online in the years 2009–2011 for 28 EU countries. It has 91 entries characterized by elements (country) name, year, language, population, constitutional form, and value (in percentages) of the population looking for educational information. The mapping unit suggested 30 possible mapping combinations, concretely 15 bar charts, 6 line charts, 8 timelines, and 1 geo chart.

Book-Crossing dataset[9] (*Books*): This dataset contained 41 randomly chosen books published between 1960 and 2003 and characterized by the elements name, country, publisher, and year. The mapping unit suggested three visualization types: bar chart with two combinations, geo chart with one combination, and timeline with three combinations, a total of seven mapping combinations.

4.2. Procedure

Our experimental approach was to gather user preferences for visualizations obtained from the rule-based system and to test different recommendation approaches to suggest visualizations. A crowd-sourced study was designed to obtain personalized scores for each visualization suggested by the visual recommender. Before giving a score, a participant had to perform some cognitively demanding task with the visualization (i.e., a minimal analysis). Based on the experiments conducted by Kittur et al. [2008], this preparatory task should bring participants to accurately study the chart and prevent a random or rash rating. We designed the task as follows: (1) a participant was given a one-line description of a dataset originating the visualization; (2) looking at the visualization, she had to write tags (at most five) and (3) rate the visualization. Figure 6 shows an example of a Human Intelligent Task (HIT). The score system used a multidimensional scale adapted from a list of usability factors presented in Seffah et al. [2006] and Zheng et al. [2007]. Providing a multidimensional rating scale should assist a user in considering various aspects of a visualization and thus to specify the subjective ratings for the considered visualization. The rating scale contained the following factors: (1) cluttered, (2) organized, (3) confusing, (4) easy to understand, (5) boring,

[7]https://movielens.org/.
[8]https://open-data.europa.eu/en/linked-data.
[9]http://www2.informatik.uni-freiburg.de/∼cziegler/BX/.

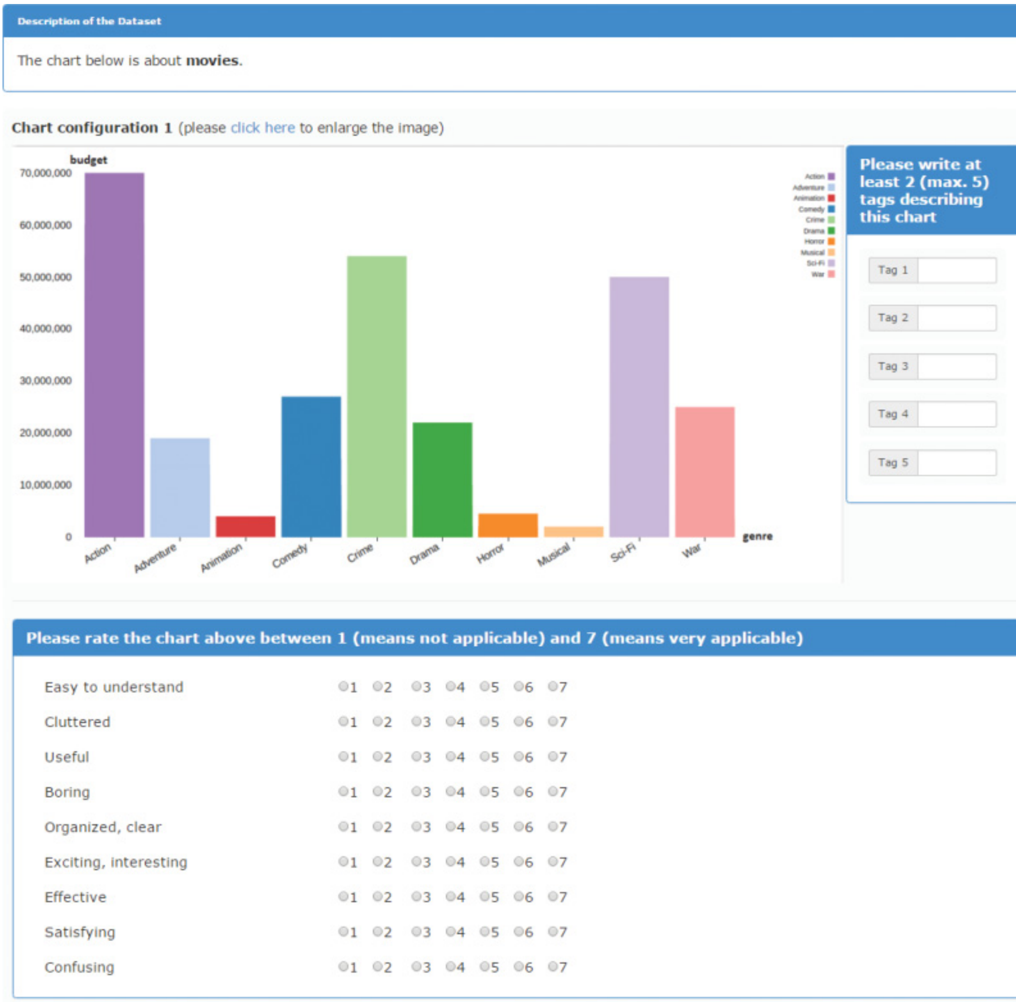


Fig. 6. Crowd-sourced experiment task (HIT). Participants were motivated to carefully observe the visualization with the study task, in terms of writing tags for this visualization. Thereafter, they had to rate it in a multidimensional scale.

(6) exciting, (7) useful, (8) effective, and (9) satisfying. Note that dimensions 1–6 are duplicated with opposing sentiment (e.g., cluttered vs. organized). Opposing dimensions were used to ensure meaningful ratings for scales with complex meaning. Dimensions were rated on a 7-point Likert scale (1 = not applicable–7 = very applicable).

Since the visualization scores were intended for the offline experiment, each participant had to rate more than one visualization. We experimented with varying sizes of HITs, collecting 10 and 5 tasks. After a pilot study, these turned out to take overly long (around 15 minutes), we settled for collecting three (3) charts per HIT. Suggested combinations were distributed in 32 HITs, each of which contained three randomly chosen mapping combinations. Pilot studies also helped to streamline dataset descriptions, task descriptions, and instructions across the study. After accepting a HIT, the participant (worker or turker) received a tour to complete a task, which showed a visualization and corresponding tags and ratings in the exact same format as the subsequent study. When ready, the worker started the first task in the HIT by pressing a button. Workers were allowed to write *not applicable* or NA for tags but were alerted if they failed to write any tags. The rating dimensions were not assigned a score until the worker did it. Workers could only proceed if they had rated all dimensions. A HIT with three visualizations/ combinations was compensated with $1.00. A worker rated a minimum of three visualizations, but to ensure a more realistic training set for the CF-RS, workers were allowed to perform more than one HIT. Only expert workers who consistently achieved a high degree of accuracy by completing HITs were allowed to take part in the study.

4.3. Evaluation Protocol

A set of studies was carried out to analyze the variability in preference scores. To compute the overall score for a visualization for each worker, the scores in opposing dimensions (clutter, confusing, boring) were inverted and then all dimensions were averaged together according to the following equation: $SC = (\sum_{i=1}^{k} \rho_k D_k)/k$. Where $k = 9$ is the number of dimensions, ρ_k is the coefficient 1 and D_k is k dimension score. The visualization score was obtained by averaging the worker scores.

In the second part of our evaluation, we performed an offline experiment to estimate the quality of personal preferences for visualization recommendations. To this end, we used the preferences collected from the Amazon study as input data to train our recommender. For the CF-RS, we maintain a list of items (visualizations), each having the information about user and provided rating. Similarly, CB-RS uses the tags per item. Finally, for the hybrid approach, we combine the results of both recommender techniques. Following the method described in Trattner et al. [2015], we split the preference model into two distinct sets, one for training the recommender (training set) and another one for testing (test set). The test set acts here as a reference value that, in an ideal case, has to be fully predicted for the given training set. From each of the datasets in the preference model, we randomly selected a certain percentage (more details are given in Section 5.3) of user-rated or user-tagged mapping combinations (visualizations) and entered them into the training set performing fivefold cross validation. The recommendations produced from the training set are further used to evaluate the performance of VizRec. The performance of VizRec depends generally on how good it predicts the test set. We compared the generated recommendations (prediction set) and the test set by applying a variety of well-known evaluation metrics in information retrieval [Herlocker et al. 2004]: Recall (R), Precision (P), F-Measure (F), Mean Average Precision (MAP), and the Normalized Discounted Cumulative Gain ($nDCG$). The first three metrics basically express the quantity of relevant recommended results, whereas MAP and $nDCG$ quantify the concrete ordering of the results (i.e., penalizing results that are not on the top but are relevant for the user). Concretely, the metrics are defined as follows:

Recall (R@k) is calculated as the number of correctly recommended visualizations divided by the number of relevant visualizations, where r_u^k denotes the top k recommended visualizations and R_u the list of relevant visualizations of a user u in the set of all users U. Recall is given by Rijsbergen [1974]:

$$R@k = \frac{1}{|U|} \sum_{u \in U} \left(\frac{|r_u^k \cap R_u|}{|R_u|} \right). \tag{16}$$

Precision (P@k) is calculated as the number of correctly recommended visualizations divided by the number of recommended visualizations k. Precision is defined as

Rijsbergen [1974]:

$$P@k = \frac{1}{|U|} \sum_{u \in U} \left(\frac{|r_u^k \cap R_u|}{k} \right). \tag{17}$$

F1-score (F1) combines precision and recall into one score [Rijsbergen 1974]:

$$F1@k = 2 \cdot \frac{P@k \cdot R@k}{P@k + R@k}. \tag{18}$$

Mean average precision (MAP) is an extension of the precision metric that additionally looks at the ranking of recommended visualizations. MAP is described in the subsequent equation, where B_j is 1 if the recommended visualization at position j is among the relevant visualizations and 0 otherwise [Rawashdeh et al. 2013]:

$$MAP@k = \frac{1}{|U|} \sum_{u \in U} \frac{1}{k} \sum_{j=1}^{k} B_j \cdot P@j. \tag{19}$$

Normalized Discounted Cumulative Gain (nDCG@k) is a ranking-dependent metric that not only measures how many visualizations can be correctly predicted but also takes the position of the visualizations in the recommended list with length k into account. The nDCG metric is based on the *Discounted Cumulative Gain (DCG@k)* that is given by Parra and Sahebi [2013]:

$$DCG@k = \sum_{k=1}^{|r_u^k|} \left(\frac{2^{B(k)} - 1}{log_2(1+k)} \right), \tag{20}$$

where $B(k)$ is a function that returns 1 if the recommended product at position i in the recommended list is relevant. nDCG@k is calculated as DCG@k divided by the ideal DCG value iDCG@k, which is the highest possible DCG value that can be achieved if all the relevant visualizations would be recommended in the correct order. Taken together, it is given by the following equation [Parra and Sahebi 2013]:

$$nDCG@k = \frac{1}{|U|} \sum_{u \in U} \left(\frac{DCG@k}{iDCG@k} \right). \tag{21}$$

5. RESULTS

5.1. Participants

Each HIT was completed by 10 workers. For 92 visualizations, 8,280 ratings across 9 dimensions and 4,483 tags were collected from 70 participants. Participants completed on average 4.7 HITs. The experiment started on November 26, 2014, and ended on December 3, 2014. The allotted working time per HIT was 900s and the average working time of workers was 570s per HIT. Table V summarizes the details about the study. As an example the Figure 7 presents the three (in average) highest rated visualizations in each dataset whereby the Figure 8 the three (in average) lowest rated visualizations in each dataset. Finally, Table VI lists the top-10 stemmed tags in each of the three datasets.

5.2. Visual Quality

The heatmap in Figure 9 shows the mean rating for every dimension for each visualization. The results confirm a clear understanding of the opposing dimensions. Negative dimensions in the lower case received opposite scores to corresponding positive ones (UN-co, OR-cl, EX-bo, in Figure 9 top). The aggregated score for each visualization

Table V. Basic Statistics of the Three Rating and Tag Datasets Collected Via the Crowd-Sourced Experiment

	Movies	EU	Books
#visualizations	55	30	7
#users	36	19	15
#ratings	4950	2700	630
#tags	2731	1403	349
#unique tags	292	166	87
Avg. #tags per visualization	49.65	46.76	49.86
Avg. #unique tags per visualization	23.09	22.27	23.43
Avg. #tags per user per visualization	3.23	3.24	3.23
Avg. #unique tags per user per visualization	2.18	1.74	2.61
Avg. #tags per user	75.86	73.84	23.27
Avg. #unique tags per user	20.97	21.00	12.27
Avg. #users per visualization	10	10	10
Avg. #visualizations rated/tagged	15.27	15.79	4.67

Fig. 7. Three (in average) highest rated visualizations for the datasets *Movies*, *EU*, and *Books* including the five most frequently used tags.

in the bottom row of the heat map (SC) shows that only a handful of visualizations achieved clearly high scores, whereas for each type there were high scoring visualizations. More importantly, the violin plot at the bottom explains these scores: There is a broad variability in scores for most visualization instances. The violin plot shows the density of scores; variability is visible in the different shapes as in the spread of the shapes. The coefficient of variation computed for each chart confirmed this assumption ($M = 0.36$, $SD = 0.12$) the minimum variation was 0.07 and the maximum was 0.64, see Figure 10. A Levene test on scores confirmed significant differences in variances across charts ($F = 1.64$, $p < 0.001$). This supports our assumption that user preferences

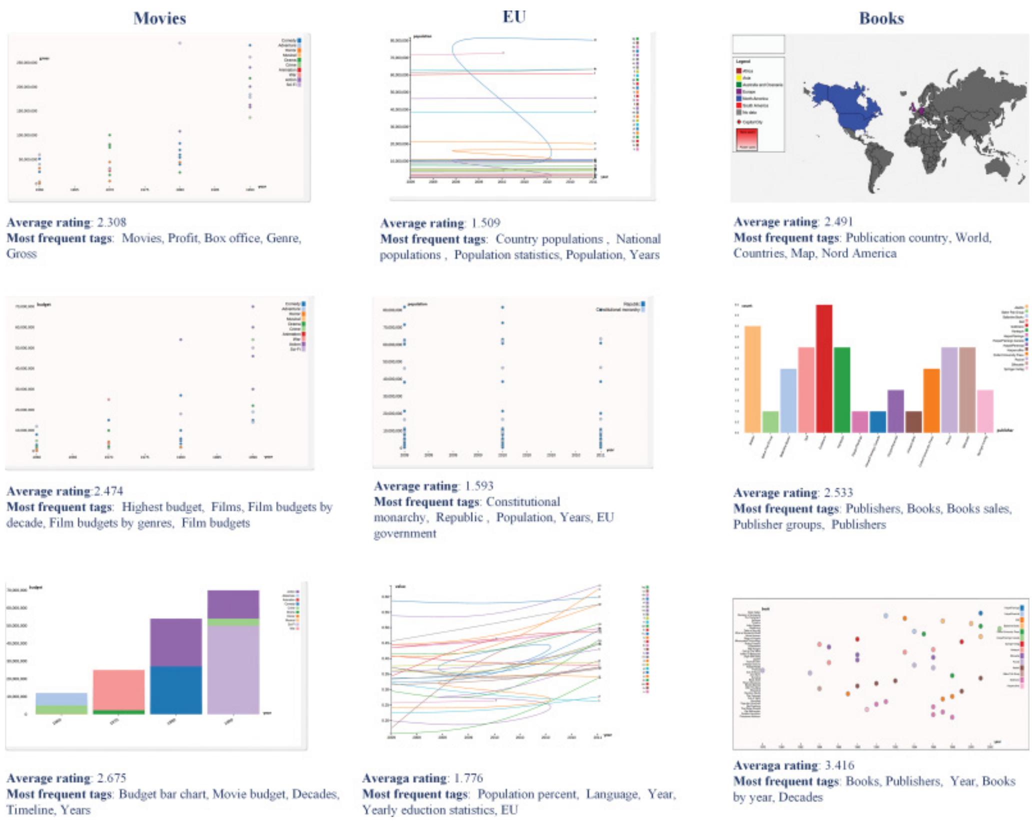


Fig. 8. Three (in average) lowest rated visualizations for the datasets *Movies*, *EU*, and *Books* including the five most frequently used tags.

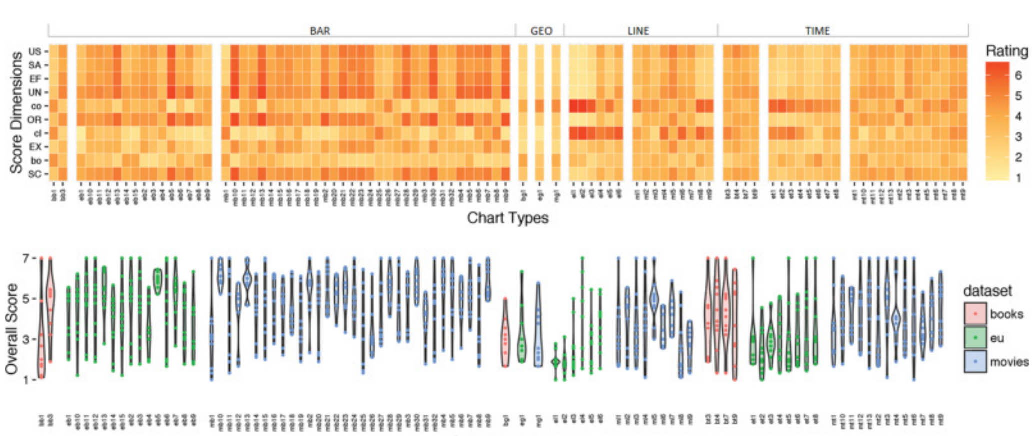


Fig. 9. Mean and variability in scores (rating 1–7, higher is better). The heatmap illustrates the contribution of nine dimensions (US = useful, SA = satisfying, EF = efficient, UN = easy to understand, co = confusing, OR = organized, cl = cluttered, EX = exciting, bo = boring) to the overall score (SC). The violin plot below illustrates the high variability in personal ratings.

Table VI. Top-10 Tags in Each of the Three Datasets. The Tags Were Stemmed (Normalized) as Described in Section 3.3

Rank	tag	#tags	#visuals	#users
Movies				
1	movi	545	55	34
2	genr	252	34	26
3	budget	212	20	22
4	popul	191	32	30
5	gross	130	26	21
6	film	127	48	14
7	year	108	32	17
8	chart	97	46	7
9	titl	90	27	12
10	countri	60	18	16
EU				
1	popul	194	27	18
2	constitut	91	15	13
3	educ	88	16	11
4	republ	86	15	15
5	monarch	79	15	16
6	form	63	19	11
7	countri	62	22	13
8	govern	61	15	10
9	year	60	19	14
10	language	37	8	11
Books				
1	book	93	7	14
2	publish	68	7	15
3	count	35	5	13
4	year	21	4	9
5	titl	12	4	7
6	countri	11	3	7
7	timelin	7	4	3
8	novel	7	6	3
9	famou	6	4	3
10	inform	5	4	1

matter when choosing the right representation. The results confirm that only a very small number of visualizations achieved high scores and the rest were variable.

From the heatmap individual top-scoring visualizations can be identified. To establish differences in the visualization categories and datasets, we performed a factorial ANOVA with the visualization type and dataset as factors (visualization-type: *bar, line, time, geo*, and dataset: *Movies, Books, Eu*). Homogeneity of variance was confirmed by a Levene test. The factorial ANOVA revealed a significant effect of dataset $F(2{,}908) = 21.19$, $p < 0.0001$; a significant effect of visualization type $F(3{,}908) = 38.98$, $p < 0.001$; and significant interaction effect dataset visualization type $F(5{,}908) = 3.81$, $p < 0.01$. TukeyHSD multiple comparisons revealed a significant difference in scores between *Movies* ($M = 4.86$) and *Books* ($M = 3.82$) $p < 0.05$, as well as between *Movies* and *Eu* data ($M = 3.68$), $p < 0.001$. For the visualization type, there was a significant difference in scores between *bar* ($M = 4.60$) and *geo*, ($M = 3.06$) $p < 0.001$; *bar* and *line*, ($M = 3.29$) $p < 0.001$; *bar* and *time*, ($M = 3.72$) $p < 0.001$; as

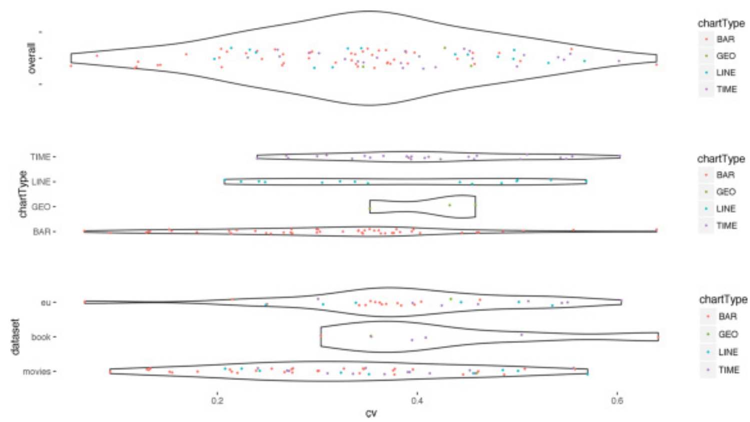


Fig. 10. Coefficient of variation. Overall variation for all charts (top), variation broken down by chart type (middle) and by dataset (bottom). Note that the coefficient of variation is the ratio of SD/Mean. The density in the violin chart shows where the broad variation of scores across charts.

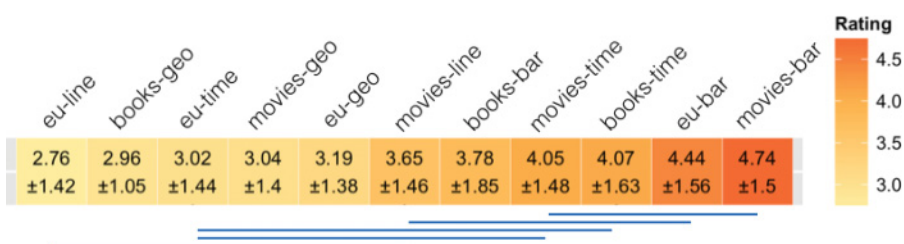


Fig. 11. Significant Interactions Visualization Type/dataset. The heat-map illustrates the mean score and standard deviation for each combination of *dataset-visualization type* (1 = completely disagree, 7 = totally agree). The lines below show where differences begin to be significant. Note that, due to its high variability, *books-bar* is not significantly better than *eu-line*, whereas *movies-line* is.

well as between *time* and *line*, $p < 0.02$. The significant effects of multiple comparisons for interaction are shown in Figure 11.

The main outcomes are the information about user preferences and the clear differences among them. The interaction effects illustrate several differences amongst visualization type. For instance, the majority of the users preferred bar chart, probably since it is familiar to most people. Another reason may be that it is easier to compare the values of several numbers at once using bar chart. Yet, these results merely indicate that there are varied preferences. Looking at each dataset, visualization, and visualization type in the heatmap of Figure 9, it is clear that while a small number of visualizations are generally preferred, in most cases the ratings vary widely and a personalized approach would accommodate those user preferences better.

5.3. Recommendation Quality

In this section, we summarize the results of the offline evaluation. As defined in the protocol, first, we show how VizRec performs with regard to user preferences collected in the Amazon Turk experiment. We analyze here how the recommender performs using individual rating dimensions compared to the performance with the aggregated ratings (overall score). Second, we show what kind of user feedback, rating values, or annotations via tags would be more adequate for recommending visualizations. In addition, we compare the performance of the collaborative filtering-based recommender

Table VII. Quality Metrics Values Estimated for the Three Example Datasets Using VizRec's CF-RS. The Values Are Calculated First for the Ratings Taken from One of the Nine Dimensions (BO = boring, CL = cluttered, CO = confusing, EF = efficient, EX = exciting, OR = organized, SA = satisfying, UN = Easy to Understand, US = useful) and Than for the Overall Rating Score (O.a). Note: That We Inverted the Ratings of the Negative Dimensions (N) Boring, Cluttered and Confusing Using Their Opposites (P) Exciting, Organized, and Easy to Understand According to the Equation $\frac{(P-N)+7}{2}$. For This Test We Used a Fivefold Cross Validation whereby Each Iteration Used 80% of User's Data as Training Set and 20% as Test Set

		Metric				
Dataset	**Dimensions**	**R@3**	**P@3**	**F@3**	**MAP@3**	**nDCG@3**
Movies	**¬bo**	0.0814	0.1425	0.0854	0.0598	0.0924
	¬cl	0.0551	0.1389	0.0757	0.0419	0.0761
	¬co	0.0868	0.1481	0.0891	0.0548	0.0920
	EF	0.0905	0.1629	0.0972	0.0638	0.1027
	EX	0.0993	0.1592	0.1003	0.0687	0.1071
	OR	0.0692	0.1463	0.0866	0.0584	0.0872
	SA	0.0834	0.1481	0.0895	0.0696	0.1022
	UN	0.0470	0.1185	0.0632	0.0345	0.0642
	US	0.0983	0.1537	0.0970	0.0620	0.1028
	O.a	0.1320	0.1685	0.1137	0.1011	0.1362
EU	**¬bo**	0.2080	0.3473	0.1286	0.1800	0.2488
	¬bl	0.2592	0.3649	0.2754	0.2061	0.2833
	¬co	0.2259	0.3789	0.2745	0.1785	0.2540
	EF	0.2471	0.3754	0.2783	0.2005	0.2768
	EX	0.2203	0.3684	0.2687	0.1814	0.2555
	OR	0.2107	0.3614	0.2588	0.1764	0.2511
	SA	0.1884	0.3403	0.2392	0.1691	0.2348
	UN	0.2080	0.3614	0.2589	0.1859	0.2551
	US	0.2270	0.3649	0.2640	0.1945	0.2615
	O.a	0.2701	0.3684	0.2801	0.2199	0.2954
Books	**¬bo**	0.5888	0.4259	0.4677	0.4629	0.4949
	¬cl	0.6666	0.5155	0.5573	0.5711	0.5980
	¬co	0.6066	0.4888	0.5182	0.5333	0.5513
	EF	0.5222	0.3518	0.3955	0.3833	0.4186
	EX	0.5466	0.4906	0.4955	0.1814	0.5110
	OR	0.6133	0.4488	0.4920	0.4544	0.4944
	SA	0.5466	0.3822	0.4266	0.4377	0.4675
	UN	0.6400	0.4844	0.5266	0.5522	0.5753
	US	0.5444	0.4074	0.4422	0.4592	0.4812
	O.a	0.6933	0.4400	0.5626	0.5966	0.6220

and content-based recommender techniques with their hybridized version (hybrid recommender).

5.3.1. Using Each Rating Dimension Separately. This part of the experiment was intended to compare VizRec's recommender performance when using a single rating vs. using a multidimensional scale. To do so, we estimated the quality metric values R, P, F, MAP, and the Normalized for each of the individual ratings and for the overall score. Table VII summarizes the results.

The results show that recommendations generated with O.a (overall score) are more accurate than those obtained with either of the nine dimensions separately. For instance, when comparing by dataset *Movies* the recommendation accuracy (F@3) for dimension UN with the value for the overall score, the dependent t-test reveals that VizRec's CF performs, on average, significantly better for the overall rating ($M = 0.1137, SE = 0.0077$) than for the dimension UN ($M = 0.0632, SE = 0.0036$), $t(35) = 2.5204$, $p < 0.01, r = 0.400$. Subsequently, MAP@3 ascertains that when using the overall rating ($M = 0.1011, SE = 0.0063$) VizRec can sort individual recommendations according to their relevance to the user significantly better than, for example, using the dimension UN ($M = 0.0345, SE = 0.0038$), $t(35) = 2.6759$, $p < 0.01, r = 0.41$. Note, the effect size estimate (r) indicates that the difference in performance is a large, and therefore a substantive, effect (just below 0.5)—all effects are reported at a 0.05 level of significance.

The results support our assumption that considering different aspects to rate visualizations improves recommendation quality. This has a root in the fact that individual dimensions are potential source of errors, as a user may understand and interpret them in different ways. In addition, when providing rating values, there is often a need for a reference value, based on which such absolute ratings can be made (e.g., when just taking the subjective judgment on "useful" for the first time). On the contrary, different aspects may provide such a reference value, as user get insight on what else may be required that eventually stays in relation with other dimensions (e.g., easy to understand and confusing). Furthermore, it is more likely that additional dimensions will compensate for mistakes on individual dimensions, like being unable to evaluate it objectively.

Another finding here is that there is no pattern across the nine dimensions implying a dependence of the recommendation accuracy on negative (boring, cluttered, etc.) or on positive (effective, exciting, etc.) dimensions. For instance, for the *Movies* dataset the $F-Measure$ for the positive dimension *easy to understand* is $F@3_{UN} = 0.0632$ whereby for its opposite dimension *confusing* $F@3_{co} = 0.0891$. A dependent t-test reveals that the recommendation accuracy for *confusing (¬co)* ($M = 0.0891, SE = 0.0067$) is, on average, not significantly higher than for *easy to understand (UN)* ($M = 0.0632, SE = 0.0036$), $t(35) = 1.4146$, $p > 0.01, r = 0.2325$. The effect size estimate indicates that the difference in recommendation accuracy given by negative dimensions ¬co is a small and therefore unsubstantial effect. The same effect is present for the positive dimension *exciting (EX)* ($M = 0.1003, SE = 0.0130$) and its opposite dimension *boring (¬bo)* ($M = 0.0854, SE = 0.0079$) $t(35) = 0.7042$, $p > 0.01, r = 0.1181$. These results indicate that no dimension dominates the others and thus has a special impact on the overall rating. In summary, negative ratings are as valuable input as the positive ratings [Schafer et al. 1999] but as many recommender systems, VizRec performs better using both positive and negative ratings.

5.3.2. Using Overall Scores. To measure the improvements in terms of recommender quality (=accuracy), we compared the VizRec CF with the baseline filtering algorithms Most Popular (MP) and Random (RD). The RD simulates the recommender behavior providing an arbitrary order of visualizations, that is, it can be compared with having only the first two units in the VizRec pipeline from Figure 1. The MP, in contrast, generates the results sorted according to global ratings, in our case accumulated from ratings of individual users. Considering RD and MP, baseline algorithms should unveil whether the recommender systems can in general help with providing useful visualizations and whether the personalized approach improves the quality of the results, respectively.

Table VIII summarizes the results of the evaluation. VizRec CF outperforms both baseline algorithms in all three datasets. The first three quality metrics clearly indicate that the results are more accurate using *VizRec* CF than simply generating arbitrary visualizations (cf. $F@3(CF) = 0.1137$ and $F@3(RD) = 0.0055$ for *Movies*). Concretely, the dependent t-test reveals that, on average, the performance of the CF ($M = 0.1137, SE = 0.0077$) is significantly higher that of the baseline algorithm RD ($M = 0.0055, SE = 0.0149$), $t(35) = 3.0375$, $p < 0.01, r = 0.4567$. The effect size estimate (r) indicates that the difference in performance is a large, and therefore a substan-

Table VIII. Quality Metrics Values R@3, P@3, F@3, MAP@3, nDCG@3 Estimated for the Three Datasets Using the Baseline Algorithms MP and RD (k = 3). Note That for This Test We Executed a Fivefold Cross Validation whereby Each Iteration Used 80% of User's Data as Training Set and 20% as Test Set

		Metric				
Dataset	**Alg.**	**R@3**	**P@3**	**F@3**	**MAP@3**	**nDCG@3**
Movies	**CF**	0.1320	0.1685	0.1137	0.1011	0.1362
	MP	0.0488	0.0926	0.0591	0.0163	0.0419
	RD	0.0039	0.0093	0.0055	0.0020	0.0048
EU	**CF**	0.2701	0.3684	0.2801	0.2199	0.2954
	MP	0.0263	0.0175	0.0211	0.0088	0.0161
	RD	0.0132	0.0175	0.0150	0.0044	0.0103
Books	**CF**	0.6933	0.4400	0.5626	0.5966	0.6220
	MP	0.1333	0.0444	0.0667	0.0444	0.0667
	RD	0.0667	0.0222	0.0333	0.0333	0.0420

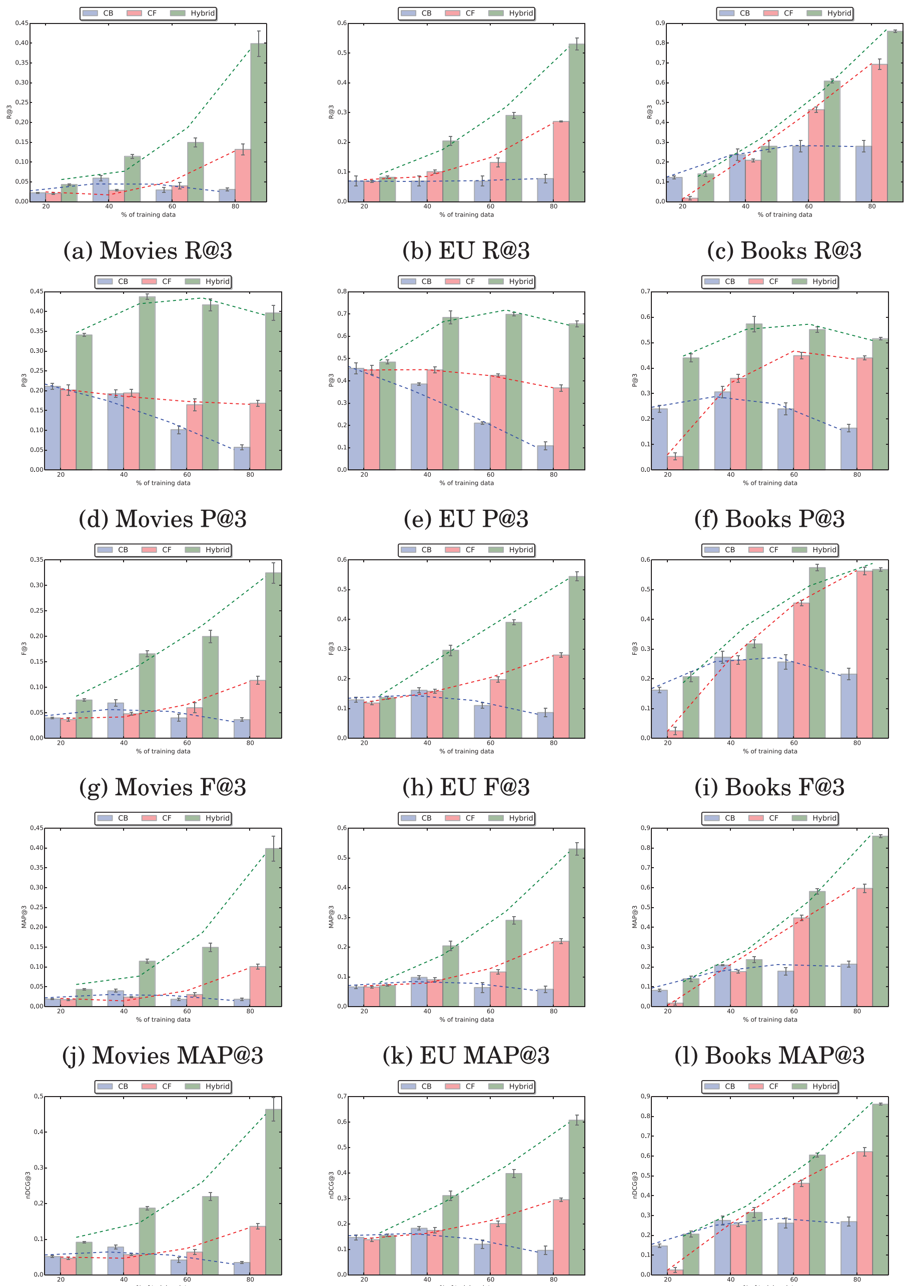
tive, effect. Additionally, *MAP@3* and *nDCG@3* reveal that *VizRec* CF is significantly better at sorting individual visualizations according to their relevance to the user. For example, the results for *MAP@3* show that CF ($M = 0.1011, SE = 0.0063$) significantly outperforms RD ($M = 0.0020, SE = 0.0053$), $t(35) = 3.9771, p < 0.01, r = 0.0558$. The effect size estimate indicates also a large difference in performance and therefore a substantive effect—all effects are reported at a 0.05 level of significance.

Note that the difference between individual metrics among the datasets is to a large extent influenced by the considerable difference in size of the three datasets (e.g., Books has only seven different visualizations—$F@3(CF) = 0.5626$, whereas *Movies* has 55—$F@3(CF) = 0.1137$, see Figure 9).

Another interesting finding is that the recommender strategy based on global ratings (MP) generated less accurate results than *VizRec* CF, with regard to both providing relevant visualizations and their ranking order. This supports our main assumption that, in terms of the wide variability in user preference ratings, the personalized approach performs better recommendations.

5.3.3. Collaborative Filtering vs. Content-Based Recommendations. For the sake of comparing both VizRec recommenders, we extended the comparison using the same quality metrics to rating-based (CF) and tag-based (CB) recommender techniques. The estimation was performed in five runs using random splitting of training/test data, as described in the procedure of the experiment. Each iteration uses 80%, 60%, 40%, and 20% of user's data as training and the rest as a test set. Table IX summarizes the results, and Figure 12 illustrates them. The first observation reveals that both approaches show relatively comparable performance in all three datasets, particularly for smaller training data (i.e., cases with 20% and 40% of the users' data). For the majority of the quality measures in these two cases, the CB outperforms the rating-based (CF) recommender technique in both providing expected recommendations and in sorting. However, in few exceptional cases, mostly for 40% of the users' data, CF shows slightly better results (cf. Figures 12(b), (d), (e), and (f)).

The results also reveal that the relative improvement in recommendation performance depends to some extent on characteristics of the dataset. For example, for *Books* the F-score measure for CB is about 5 times better than CF, while for *Movies* and *EU* datasets the relative improvement in performance lies at about 8% and 9%, respectively. Concretely, when considering the recommendation quality ($F@3$) for the dataset *Movies* with 20% of users' data, the dependent t-test reveals that CB ($M = 0.0399, SE = 0.0018$) does not perform significantly better than CF ($M = 0.0368, SE = 0.0069$), $t(35) = 0.5879, p > 0.01, r = 0.0988$. For the dataset EU, the test reveals simi-



(a) Movies R@3 (b) EU R@3 (c) Books R@3

(d) Movies P@3 (e) EU P@3 (f) Books P@3

(g) Movies F@3 (h) EU F@3 (i) Books F@3

(j) Movies MAP@3 (k) EU MAP@3 (l) Books MAP@3

(m) Movies nDCG@3 (n) EU nDCG@3 (o) Books nDCG@3

Fig. 12. Performance plots for collaborative filtering (CF), content based (CB), and hybrid approaches for different sizes of training sets (20%–80%) considering the first three recommendations in the result list ($k = 3$).

Table IX. Comparing VizRec Hybrid Approach with Rating-Based and Tag-Based Approaches: Quality Metric Values Considering the First Three Recommendations in the List ($k = 3$). The Results Are Distributed over Four Sets (from 20% to 80%), Each Containing Different Number of Items (Visualizations) in the Training Set ts. Note That for This Test We Used a Fivefold Cross Validation

		Metric									
		R@3		**P@3**		**F@3**		**MAP@3**		**nDCG@3**	
Dataset	**Alg.**	ts_{20}	ts_{40}	ts_{20}	ts_{40}	ts_{20}	ts_{40}	ts_{20}	ts_{40}	ts_{20}	ts_{40}
	CB	0.0226	0.0595	0.2111	0.1925	0.0399	0.0693	0.0189	0.0399	0.0514	0.0782
Movies	**CF**	0.0207	0.0289	0.2018	0.1944	0.0368	0.0482	0.0175	0.0241	0.0467	0.0571
	Hybrid	0.0434	0.1145	0.3407	0.4370	0.0751	0.1658	0.0434	0.1145	0.0916	0.1873
	CB	0.0694	0.0694	0.4561	0.3859	0.1294	0.1617	0.0663	0.0983	0.1472	0.1832
EU	**CF**	0.0687	0.1012	0.4491	0.4491	0.1188	0.1584	0.0658	0.0906	0.1382	0.1746
	Hybrid	0.0821	0.2043	0.4842	0.6842	0.1372	0.2961	0.0714	0.2043	0.1550	0.3113
	CB	0.1226	0.2371	0.2400	0.3066	0.1622	0.2730	0.0822	0.2093	0.1462	0.2750
Books	**CF**	0.0177	0.2093	0.0533	0.3600	0.0253	0.2633	0.0177	0.1768	0.0252	0.2536
	Hybrid	0.1408	0.2800	0.4400	0.5733	0.2074	0.3177	0.1408	0.2371	0.2060	0.3154
		ts_{60}	ts_{80}	ts_{60}	ts_{80}	ts_{60}	ts_{80}	ts_{60}	ts_{80}	ts_{60}	ts_{80}
	CB	0.0296	0.0310	0.1018	0.0574	0.0400	0.0367	0.0185	0.0186	0.0418	0.0348
Movies	**CF**	0.0401	0.1320	0.1684	0.1685	0.0600	0.1137	0.0305	0.1011	0.0641	0.1362
	Hybrid	0.1495	0.3988	0.4166	0.3962	0.1998	0.3246	0.1495	0.3988	0.2196	0.4640
	CB	0.0694	0.0778	0.2105	0.1087	0.1104	0.0867	0.0643	0.0583	0.1206	0.0965
EU	**CF**	0.1322	0.2701	0.4245	0.3684	0.1975	0.2801	0.1167	0.2199	0.2016	0.2954
	Hybrid	0.2903	0.5303	0.6982	0.6561	0.3901	0.5444	0.2903	0.5303	0.3979	0.6073
	CB	0.2800	0.2800	0.2400	0.1644	0.2568	0.2160	0.1788	0.2144	0.2612	0.2690
Books	**CF**	0.4822	0.6933	0.4488	0.4400	0.4551	0.5626	0.4477	0.5966	0.4623	0.6220
	Hybrid	0.6088	0.8600	0.5511	0.5155	0.5742	0.5680	0.5811	0.8600	0.6048	0.8615

lar results—CB ($M = 0.1294, SE = 0.0082$) does not perform significantly better than CF ($M = 0.1188, SE = 0.0061$), $t(18) = 0.5800, p > 0.01, r = 0.1354$. On the other hand, for the the dataset *Books* we can observe that the performance of CB ($M = 0.1622, SE = 0.0091$) is significantly better than CF ($M = 0.0253, SE = 0.0125$), $t(14) = 2.8789, p < 0.01, r = 0.5965$. The effect size estimate (r) indicates that the difference in performance is a large, and therefore a substantive, effect.

Conversely, the last case of the experiment—with the 80% of the users' data—the dependent t-test for *Movies* reveals that the recommendation quality ($F@3$) with CF ($M = 0.1137, SE = 0.0077$) is, on average, significantly higher than with CB ($M = 0.0367, SE = 0.0033$), $t(35) = 3.1604, p < 0.01, r = 0.4711$. The effect size estimate (r) indicates that the difference in performance is a large, and therefore a substantive, effect. For this training-set configuration, the user has more data (i.e., rated or tagged items), and specific preferences are of more importance than in previous cases with 20% and 40% of the training data. Since the CF looks at other user to find recommendations, more specific results could be observed compared to CB.

In a nutshell, an important finding in this study was that both algorithms behave differently in response to size of user preferences/profiles. With small user profiles/preferences, the tag-based recommender performs better recommendation quality than the rating-based filtering approach, where the results remain stable for almost all quality metrics. With smaller training sets, the tag-based filtering seems to be a method of choice.

5.3.4. Hybrid Recommendations. To evaluate the performance of our hybrid recommender, we use the same quality metrics and compare the results with those previously estimated for CF and CB. Again, we run the recommender using a fivefold cross validation. The results are summarized in Table IX and illustrated in Figure 12.

Considering recommendation accuracy for all four training/test sets and three datasets, the hybrid recommender outperforms both CF and tag-based CB. For instance, for $Movies@t_{80}$, the dependent t-test between the Hybrid and rating-based (CF) recommenders reveals that, on average, the recommendation accuracy for Hybrid ($M = 0.3988, SE = 0.0202$) is significantly higher than for CF ($M = 0.1137, SE = 0.0077$), $t(35) = 6.6380, p < 0.01, r = 0.7465$. The effect size estimate (r) indicates that the difference in performance is a large, and therefore a substantive, effect ($r > 0.5$). Furthermore, the dependent t-test between the Hybrid and tag-based (CB) recommenders for $Movies@t_{80}$ delivers similar results. Concretely, the recommendation accuracy for Hybrid ($M = 0.3988, SE = 0.0202$) is significantly higher than for CB ($M = 0.0367, SE = 0.0033$), $t(35) = 9.6200, p < 0.01, r = 0.8518$. The effect size estimate (r) also indicates a large difference in performance, and therefore a substantive, effect ($r > 0.5$).

For the sake of evaluating the performance of a Hybrid recommender for more than one dataset, we consider now the recommendation accuracy for the dataset $Eu@t_{80}$. The dependent t-test between the Hybrid and rating-based (CF) recommenders reveals that, on average, the recommendation accuracy for Hybrid ($M = 0.5444$, $SE = 0.0151$) is significantly higher than for CF ($M = 0.2801, SE = 0.0076$), $t(18) = 3.1960, p < 0.01, r = 0.6016$. The effect size estimate (r) for this test indicates that the difference in performance is a large, and therefore a substantive, effect ($r > 0.5$). Furthermore, the dependent t-test between the Hybrid and tag-based (CB) recommenders for $Movies@t_{80}$ delivers similar results. Concretely, the recommendation accuracy for Hybrid ($M = 0.5444, SE = 0.0151$) is significantly higher than for CB ($M = 0.0867, SE = 0.0141$), $t(18) = 8.1385, p < 0.01, r = 0.8867$. Similarly to the previous test, the effect size estimate (r) indicates that the difference in performance is a large, and therefore a substantive, effect ($r > 0.5$).

Subsequently, $MAP@3$ and $nDCG@3$ values $@ts_{80}$ ascertain that the Hybrid recommender can sort individual recommendations according to their relevance to the user better. Concretely, the dependent t-test for $Movies@t_{80}$ between Hybrid and CF (on $nDCG@3$ values) reveals that the Hybrid ($M = 0.4640, SE = 0.0333$) recommender performs significantly better than CF ($M = 0.1362, SE = 0.0081$), $t(35) = 7.7978, p < 0.01, r = 0.7966$. Finally, the effect size estimate (r) indicates that the difference in performance is a large, and therefore a substantive, effect ($r > 0.5$).

An interesting finding here is that the more data exist about the user, the better the Hybrid recommender performs (cf., $Movies@ts_{80}$: $F@3_{Hybrid} = 0.3246$, $nDCG@3_{Hybrid} = 0.4640$, $Movies@ts_{20}$:$F@3_{Hybrid} = 0.0751$, $nDCG@3_{Hybrid} = 0.0916$). As already shown in the previous study, this finding does hold for the CB. On the other side, the CF recommender behaves similarly to the Hybrid recommender but not in the same volume (cf., $Movies@ts_{80}$: $F@3_{Rate} = 0.1137$, $F@3_{Hybrid} = 0.3246$).

In summary, our studies reveal that combining users' tags and ratings improves the quality of recommendations significantly. The ratings and tags are user specific, that is, ratings present user's general tastes and tags user's topic of interest. When combining both, a recommender system has more detailed information stored in user preferences and can respond more accurately. Concretely, the system can consider larger diversity of item types for defining the prediction, which in turn increases the likelihood that user will be recommend items that match her preferences the best. When now considering our second evaluation goal—to investigate what kind of feedback is more useful, in terms of recommendation quality—we can finally say that ratings and tags together help to build a more accurate recommender system for personalized visualizations.

6. CONCLUSIONS

Creating and proposing just the relevant visualizations requires appropriate filtering and recommendation strategies. Our investigations build on the premise that preference of a visual representation for data is a personal matter. We set out to investigate which information lets us anticipate the choice of chart and how to represent such information and use it for recommendation. Through a crowd-based experiment, we collected empirical evidence supporting the assumption that preferences vary widely for visual representations generated automatically. Beyond visual perception guidelines, there are other reasons that lead people to choose a particular representation. It may be habituation, as a user may be comfortable with using a particular representation for data analysis, though this has to be validated in future studies. Our research tries to unveil which aspects have to be present to recommend relevant visualizations. We outlined a rating scale comprising nine dimensions built on established usability factors. Scores for charts obtained in each of these dimensions were used to train a CF-RS and compared to an aggregated score averaging all nine dimensions. The overall score performed better recommendations compared to our rule-based approach and to a most popular method, confirming our assumptions that personalization is important.

These metrics are only based on assessed quality of a chart. However, the choice of representation is also tied to the task or question the user seeks to answer. To represent these aspects, we used tags elicited through the crowd-based experiment to train a CB-RS. Comparing both CF and CB leads to the conclusion that tags are good descriptors when there is little knowledge about user preferences. Yet, as tags contribute a great deal of knowledge about the user, we combined both pieces of information in a hybrid recommender approach. The studies revealed that the hybrid approach significantly outperformed both CF and CB in most occasions.

A major contribution of our work is that it is based on the empirical evidence collected following methodical studies involving the general public. Our approach to generating and suggesting visualizations, the process of elicitation of users' preferences, and the insights described in this article are, to the best of our knowledge, novel. The rule-based recommender was developed for the web-based tool, *VisWizard*, to automatically suggest appropriate visualizations for Linked open data.[10] Furthermore, the Recommendation dashboard,[11] a tool that organizes recommended items for visual analysis, benefits from the mapping algorithm to generate appropriate visualizations for the recommendations.

Our research did not concentrate on whether users are willing to provide information (tags/ratings) for visualizations. This is a valid research question for future work. Relevant works [Viegas et al. 2007; Wright et al. 2006] reveal the benefit of annotating visualizations in the context of information retrieval. When annotating, the user provides her insights and her interpretation on the data being visualized. Hence, they serve as analysis finding records and personal reminder for later data discovering and analysis tasks [Elias and Bezerianos 2012]. In the current work, we simply used known information such as the user query and the dataset fields as part of the tag vector describing the user's needs and visualization respectively, but, clearly, better interfaces are needed to make sure that the needed information is there for the recommendation strategy to work. In the future, we will investigate interfaces to elicit such information with minimal effort, making it part of the analysis process whenever possible.

[10]http://codev.know-center.tugraz.at/vis.
[11]https://github.com/EEXCESS/hackathon.

REFERENCES

Jae-wook Ahn and Peter Brusilovsky. 2009. Adaptive visualization of search results: Bringing user models to visual analytics. *Inform. Vis.* 8, 3 (June 2009), 167–179.

Jacques Bertin. 1983. *Semiology of Graphics*. University of Wisconsin Press.

Toine Bogers and Antal Van den Bosch. 2009. Collaborative and content-based filtering for item recommendation on social bookmarking websites. In *Proceedings of the ACM Recommender Systems Workshop on Recommender Systems and the Social Web*, Vol. 9. 9–16.

Michelle A. Borkin, Azalea A. Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. 2013. What makes a visualization memorable? *IEEE Trans. Vis. Comput. Graph.* 19, 12 (2013), 2306–2315.

Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact.* 12, 4 (Nov. 2002), 331–370.

Mike Cammarano, Xin (Luna) Dong, Bryan Chan, Jeff Klingner, Justin Talbot, Alon Halevey, and Pat Hanrahan. 2007. Visualization of heterogeneous data. *IEEE Trans. Vis. Comput. Graph.* 13, 6 (2007), 1200–1207.

M. S. T. Carpendale. 2003. *Considering Visual Variables as a Basis for Information Visualisation*. Technical Report. University of Calgary, Calgary, AB.

Min Chen and Heike Jänicke. 2010. An information-theoretic framework for visualization. *IEEE Trans. Vis. Comput. Graph.* 16, 6 (Nov. 2010), 1206–1215.

Micheline Elias and Anastasia Bezerianos. 2012. Annotating BI visualization dashboards: Needs & challenges. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'12)*. ACM, New York, NY, 1641–1650.

O. Gilson, N. Silva, P. W. Grant, and M. Chen. 2008. From web data to visualization via ontology mapping. *Comput. Graph. Forum* 27, 3 (2008), 959–966.

Nathaniel Good, J. Ben Schafer, Joseph A. Konstan, Al Borchers, Badrul Sarwar, Jon Herlocker, and John Riedl. 1999. Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the 16th National Conference on Artificial Intelligence and the 11th Innovative Application of Artificial Intelligence Conference (AAAI'99/IAAI'99)*. American Association for Artificial Intelligence, Menlo Park, CA, 439–446.

Jeffrey Heer and Michael Bostock. 2010. Crowdsourcing graphical perception: Using mechanical turk to assess visualization design. In *ACM Human Factors in Computing Systems (CHI)*. 203–212.

Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inform. Syst.* 22, 1 (Jan. 2004), 5–53.

Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. *Recommender Systems*. Cambridge University Press. Cambridge Books Online.

Wahiba Ben Abdessalem Karaa and Nidhal Gribâa. 2013. Information retrieval with porter stemmer: A new version for english. In *Advances in Computational Science, Engineering and Information Technology*. Advances in Intelligent Systems and Computing, Vol. 225. Springer International Publishing, 243–254.

Aniket Kittur, Ed H. Chi, and Bongwon Suh. 2008. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'08)*. ACM, New York, NY, 453–456.

Santiago Larrain, Christoph Trattner, Denis Parra, Eduardo Graells-Garrido, and Kjetil Nørvåg. 2015. Good times bad times: A study on recency effects in collaborative filtering for social tagging. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys'15)*. ACM, New York, NY, 269–272.

Sharon Lin, Julie Fortuna, Chinmay Kulkarni, Maureen Stone, and Jeffrey Heer. 2013. Selecting semantically-resonant colors for data visualization. In *Proceedings of the 15th Eurographics Conference on Visualization (EuroVis'13)*. 401–410.

Yi-Ling Lin, Christoph Trattner, Peter Brusilovsky, and Daqing He. 2015. The impact of image descriptions on user tagging behavior: A study of the nature and functionality of crowdsourced tags. *J. Assoc. Inform. Sci. Technol.* 66, 9 (2015), 1785–1798.

Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*. Springer, 73–105.

Jock Mackinlay. 1986. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.* 5, 2 (April 1986), 110–141.

Jock Mackinlay, Pat Hanrahan, and Chris Stolte. 2007. Show me: Automatic presentation for visual analysis. *IEEE Trans. Vis. Comput. Graph.* 13, 6 (Nov 2007), 1137–1144.

Tamara Munzner. 2014. *Visualization Analysis and Design*. Peters/CRC Press.

Belgin Mutlu, Patrick Hoefler, Gerwald Tschinkel, Eduardo Veas, Vedran Sabol, Florian Stegmaier, and Michael Granitzer. 2014. Suggesting visualizations for published data. In *Proceedings of the 5th International Conference on Information Visualization Theory and Applications (IVAPP'14)*. SCITEPRESS, 41–47.

Kawa Nazemi, Reimond Retz, Jürgen Bernard, Jörn Kohlhammer, and Dieter Fellner. 2013. Adaptive semantic visualization for bibliographic entries. In *Advances in Visual Computing*. Lecture Notes in Computer Science, Vol. 8034. Springer, Berlin, 13–24.

Thomas Orgel, Martin Höffernig, Werner Bailer, and Silvia Russegger. 2015. A metadata model and mapping approach for facilitating access to heterogeneous cultural heritage assets. *Int. J. Dig. Libr.* 15, 2–4 (2015), 189–207.

Denis Parra and Shaghayegh Sahebi. 2013. Recommender systems: Sources of knowledge and evaluation metrics. In *Advanced Techniques in Web Intelligence-2*. Studies in Computational Intelligence, Vol. 452. Springer, Berlin, 149–175.

Erhard Rahm and Philip A. Bernstein. 2001. A survey of approaches to automatic schema matching. *VLDB J,* 10, 4 (Dec. 2001), 334–350.

Majdi Rawashdeh, Heung-Nam Kim, JihadMohamad Alja'am, and Abdulmotaleb El Saddik. 2013. Folksonomy link prediction based on a tripartite graph for tag recommendation. *J, Intell, Inform, Syst,* 40, 2 (2013), 307–325.

C. J. Van Rijsbergen. 1974. Foundation of evaluation. *J, Doc,* 30, 4 (1974), 365–373.

Vedran Sabol, Gerwald Tschinkel, Eduardo Veas, Patrick Hoefler, Belgin Mutlu, and Michael Granitzer. 2014. Discovery and visual analysis of linked data for humans. In *The Semantic Web—ISWC 2014*. Lecture Notes in Computer Science, Vol. 8796. Springer International Publishing, 309–324.

J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. *The Adaptive Web*. Springer-Verlag, Berlin, 291–324.

J. Ben Schafer, Joseph Konstan, and John Riedl. 1999. Recommender systems in e-commerce. In *Proceedings of the 1st ACM Conference on Electronic Commerce (EC'99)*. ACM, New York, NY, 158–166.

Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02)*. ACM, New York, NY, 253–260.

Ahmed Seffah, Mohammad Donyaee, Rex B. Kline, and Harkirat K. Padda. 2006. Usability measurement and metrics: A consolidated model. *Softw. Qual. Contr.* 14, 2 (June 2006), 159–178.

Chris Stolte and Pat Hanrahan. 2002. Polaris: A system for query, analysis and visualization of multidimensional relational databases. *IEEE Trans. Vis. Comput. Graph.* 8 (2002), 52–65.

Card Stuart and Mackinlay Jock. 1997. The structure of the information visualization design space. In *Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis'97)*. IEEE Computer Society, Washington, DC, 92.

Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Adv. Artif. Intell.* 2009 (Jan 2009), 4:2–4:2.

Christoph Trattner, Dominik Kowald, and Emanuel Lacic. 2015. TagRec: Towards a toolkit for reproducible evaluation and development of tag-based recommender algorithms. *ACM Special Interest Group on Hypertext and the Web, SIGWEB Newsl.* Winter, Article 3 (Feb. 2015), 3:1–3:10 pages.

Manasi Vartak, Samuel Madden, Aditya Parameswaran, and Neoklis Polyzotis. 2014. SeeDB: Automatically generating query visualizations. *VLDB Proc. Endow.* 7, 13 (Aug. 2014), 1581–1584.

Fernanda B. Viegas, Martin Wattenberg, Frank van Ham, Jesse Kriss, and Matt McKeon. 2007. ManyEyes: A site for visualization at internet scale. *IEEE Trans. Vis. Comput. Graph.* 13, 6 (Nov. 2007), 1121–1128.

Martin Voigt, Martin Franke, and Klaus Meißner. 2013a. Capturing and reusing empirical visualization knowledge. In *Proceedings of the 1st International Workshop on User-Adaptive Visualization*.

Martin Voigt, Martin Franke, and Klaus Meissner. 2013b. Using expert and empirical knowledge for context-aware recommendation of visualization components. *IARIA Int. J. Adv. Life Sci.* 5, 1 (2013), 27–41.

Jörg von Engelhardt. 2002. *The Language of Graphics: A Framework for the Analysis of Syntax and Meaning in Maps, Charts and Diagrams*. Yuri Engelhardt.

Colin Ware. 2012. *Information Visualization: Perception for Design* (3rd ed.). Morgan Kaufmann.

Hadley Wickham. 2009. *ggplot2: Elegant Graphics for Data Analysis*. Springer, New York.

Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 649–658.

William Wright, David Schroh, Pascale Proulx, Alex Skaburskis, and Brian Cort. 2006. The sandbox for analysis: Concepts and methods. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'06)*. ACM, New York, NY, 801–810.

Xianjun Sam Zheng, James J. W. Lin, Salome Zapf, and Claus Knapheide. 2007. Visualizing user experience through "perceptual maps": Concurrent assessment of perceived usability and subjective appearance in car infotainment systems. In *Proceedings of the 1st International Conference on Digital Human Modeling (ICDHM'07)*. Springer-Verlag, Berlin, 536–545.

Received July 2015; revised January 2016; accepted March 2016