The Challenges of Big Data - Contributions in the Field of Data Quality and Artificial Intelligence Applications

Dissertation zur Erlangung des Grades
eines Doktors der Wirtschaftswissenschaft

eingereicht an der Fakultät
für Wirtschaftswissenschaften
der Universität Regensburg

vorgelegt von Michael Szubartowicz

Berichterstatter:
Prof. Dr. Bernd Heinrich
Prof. Dr. Mathias Klier

Tag der Disputation: 13. Juni 2022

Contents

1 Introduction

The term "big data" was made popular in the 1990s by John Mashey and his colleagues at Silicon Graphics Inc. It was used to describe the technical challenges and hardware limitations of that time regarding processing, storage and networking (Diebold, 2021; Mashey, 1998). Subsequently, in 2001, Doug Laney described upcoming data management challenges along the three dimensions volume, velocity and variety, also referred to as the three "Vs" of big data (Laney, 2001). This description (together with various extensions by additional "Vs") has become the most common way of defining big data (de Mauro et al., 2015). Computing technology has progressed notably during the last decades (Roser & Ritchie, 2013) and the challenges and opportunities associated with big data are more relevant than ever:

- E-commerce platforms are able to create product offerings which go far beyond what would be feasible to put in store shelves. However, it is no trivial task to store, process and analyze the overall **volume** of data that is accompanied by such offerings (Akter & Wamba, 2016; Zheng et al., 2020).

- While the **velocity** of data streaming from various cameras and sensors enables the development of self-driving cars or highly automated manufacturing, timely processing and decision-making is needed as delayed reactions might lead to dangerous and costly failures (Badue et al., 2021; Wang et al., 2021).

- Many web portals and online services (e.g., Yelp, YouTube, Twitter, Instagram) have created business models around enabling users to upload user generated content (e.g., comments, reviews, pictures, videos, files). Understanding and analyzing this **variety** of types of data requires the development and use of multiple approaches, each specialized for a specific data type (e.g., network analysis, natural language processing, video content analysis) or domain (Bazzaz Abkenar et al., 2021).

Solving these challenges requires research effort aimed towards developing both hardware and software that fits the needs of big data. To this end, artificial intelligence (AI) has been a major driving force on the software side, providing approaches to process large amounts of data, different types of data as well as streaming data (Chowdhary, 2020; Russell et al., 2016). Thus, developing and applying AI approaches within the field of big data is the first focal point of this dissertation.

The second focal point of this dissertation deals with one of the most common additional "Vs": veracity (Abbasi et al., 2016; Saha & Srivastava, 2014). Veracity (sometimes called uncertainty or accuracy) refers to data quality (DQ) issues related to big data which can be caused, for instance, by errors during data transfer, malfunctioning sensors or human error when entering data (Taleb et al., 2018). Many researchers acknowledge the importance of DQ in big data as it heavily impacts the quality of decisions derived from the data (Batini et al., 2015; Ghasemaghaei & Calic, 2019; Saha & Srivastava, 2014). However, existing approaches to improve DQ have not been designed with big data in mind, making them impossible or too costly to apply (Khayyat et al., 2015; Ridzuan & Wan Zainon, 2019). Thus, managing DQ by defining DQ improvement measures and assessing their effectiveness is the second focal point of this dissertation.

The following sections contain a discussion of the focal points and research questions addressed by this dissertation.

1.1 Applying AI approaches to Big Data

As mentioned above, many social networks and e-commerce platforms store user generated content, which includes unstructured or semi-structured textual data such as tweets and product reviews, which contain rich information expressed in ratings, opinions and emotions (Chau & Xu, 2012). Extracting customer needs and criticism from these texts is beneficial for businesses in order to, for instance, improve existing products or create new products (Siering & Janze, 2019). Regarding online costumer reviews, this can be achieved by understanding why a customer gave a specific star rating based on the associated review text.

Within the field of AI, natural language processing (NLP) techniques can be used to analyze the syntax and semantics of written texts (Chowdhary, 2020). Thereby, many approaches aim to discover topics discussed in texts or determine sentiments with regard to predefined features. However, topics generated by current approaches can be challenging to interpret (Vallurupalli & Bose, 2020) and most feature extraction methods such as BERT (Devlin et al., 2019; Naseem et al., 2021) are applied from a narrow perspective (e.g., focusing on item aspects without taking into account user contexts or personality). Extracting easy-to-interpret features from multiple perspectives could substantially improve the ability to explain star ratings. Thus, the first research question of this focal point is:
RQ1: To what extent can features of different feature perspectives explain star ratings in online consumer reviews and how much does each individual feature perspective contribute to the explanatory power of such an unified model?

With the surge of big data, many users needed tools to navigate online stores offering a seemingly endless stream of products, music and videos. In order to narrow down the selection of goods and services to a manageable amount, recommender systems use numerous AI approaches (k-Nearest Neighbor, Support Vector Machines, Neural Networks, cf. also Ricci et al., 2015) to generate highly personalized recommendations that fit the user's interests or needs. Factorization machines stand out as one of the most versatile approaches and constitute the state-of-the-art in context-aware recommender systems (Lahlou et al., 2017).

Technical advances lead to new use cases and possibilities for recommender systems. For instance, the development of smart personal devices enables a group of friends to control their smart TV via their smartphones. At the same time, the connection between these devices enables the presentation of content recommendations to the group while trying to satisfy each group member's personal preferences (retrieved by their personal devices). Such examples underline the utility and relevance of group recommender systems. However, existing literature has not applied factorization machines to a group recommendation scenario. Therefore, the second research question is as follows:
RQ2: How can factorization machines be utilized to enable group recommendations?

While data velocity is often understood as streams of incoming data that need technical equipment for fast-paced transfer and storage, many use cases also require near real-time processing and analysis (Chardonnens et al., 2013; Laney, 2001). In a similar way, businesses today operate in a highly dynamic environment where business processes need to change continuously (Badakhshan et al., 2019; Reisert et al., 2018). However, increasingly complex business processes and time pressure might lead to errors when designing a business process model (Gambini et al., 2011; Roy et al., 2014). This may lead to business processes not

being correct (i.e., the model contains impossible or wrong execution paths) or complete (i.e., not all execution paths are represented). To alleviate this issue, many automated planning approaches have been proposed which generate new process models (Heinrich et al., 2012; Heinrich et al., 2019; Heinrich & Schön, 2015; Marrella, 2019; Marrella & Chakraborti, 2021). However, it seems promising with regard to time and resource constraints to facilitate the adaptation of existing process models such that correctness and completeness is maintained. With this in mind, the third research question states:
RQ3: How can process models be adapted to needs for change in advance in an automated manner, such that the resulting process models are correct and complete?

1.2 Managing Data Quality in Big Data

Data quality is a multidimensional construct encompassing, inter alia, the dimensions accuracy, currency, consistency and completeness (Pipino et al., 2002). While researchers on DQ in big data investigate issues related to several DQ dimensions (Batini et al., 2015; Liu et al., 2016), one central DQ issue in the field of recommender systems is called the data sparsity problem (Sar Shalom et al., 2015). This completeness issue describes the fact, that real-world recommender data sets contain only a small percentage of all possible ratings. Similarly, item content data might suffer from missing features (e.g., actor or genre information for music) or missing feature values (Picault et al., 2011). To combat this, acquiring item content data sets from an external source (e.g., a competing provider) and integrating the item content data sets to fill up missing feature values or add new features might seem promising, but it is hard to balance needed efforts and benefits. To better assess the benefits of such an approach, research question 4 is as follows:
RQ4: Does the amount of available item features or the amount of filled up missing item feature values influence the prediction accuracy of recommender systems?

While the benefits of increasing item content data completeness might be substantial, the efforts to conduct such an DQ improvement measure have to be estimated as well. However, existing literature does not provide a systematic procedure for the extension of item content data sets. Therefore, the fifth research question of this dissertation is:
RQ5: How can an item content data set be systematically extended with respect to the data quality dimension completeness, aiming to improve recommendation quality?

In the following chapter, the outline of this dissertation is presented. Chapters 3 to 7 contain all paper associated to this dissertation. Where necessary, paper have been included in preprint formatting to follow publisher guidelines. Furthermore, each page is inserted into the page layout with borders in order to indicate both the internal structure of each paper as well as the page numbering of this dissertation.

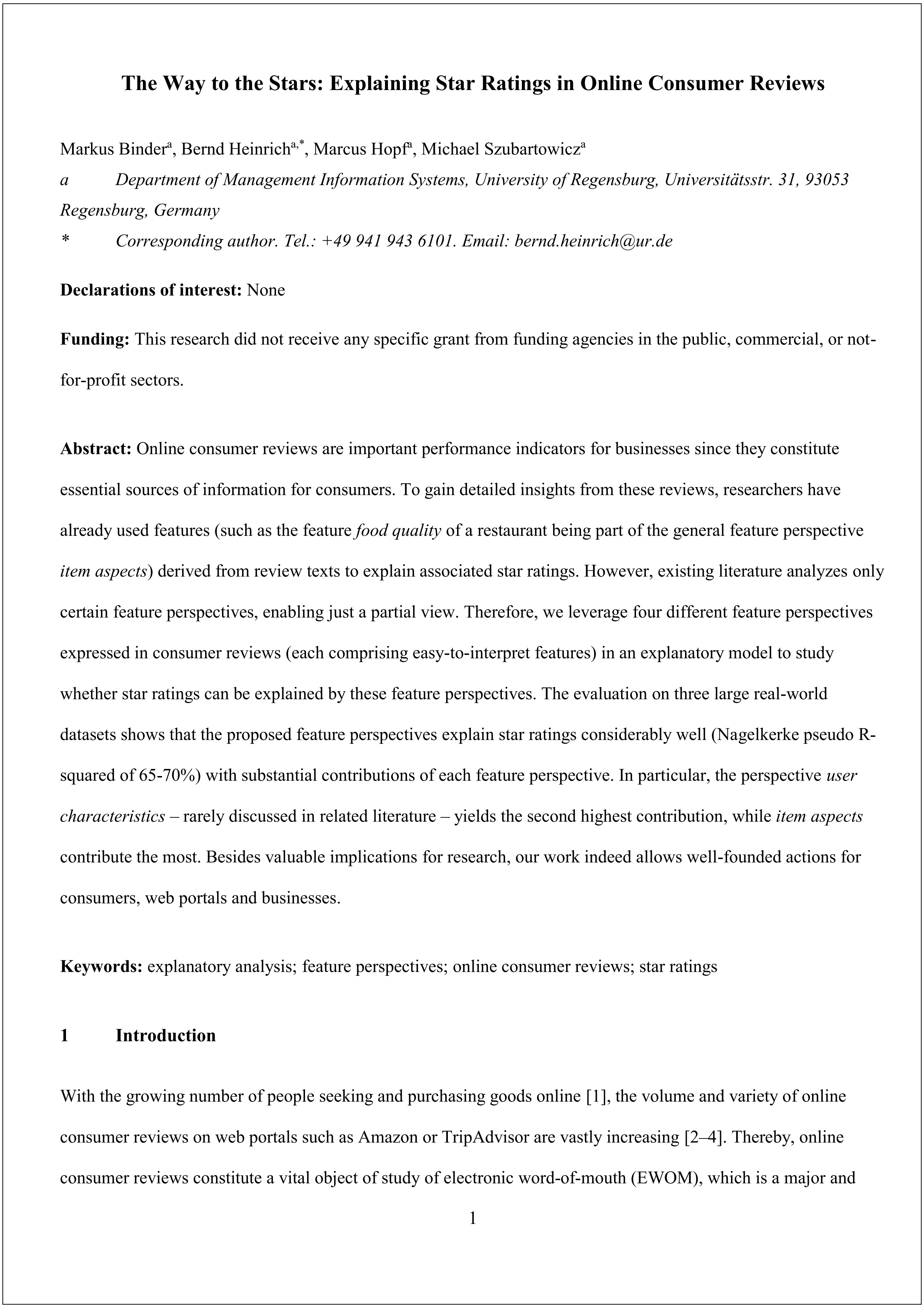
2 Dissertation Outline

This dissertation contains five paper which correspond to the five research questions introduced in the previous chapter. The following table maps each research question to each paper.

Table 2.1: Mapping of Research Questions to Paper of the Dissertation

Research Question	Title	Authors
RQ1	The Way to the Stars: Explaining Star Ratings in Online Consumer Reviews	Markus Binder, Bernd Heinrich, Marcus Hopf, Michael Szubartowicz
RQ2	GroupFM: Enabling Context-Aware Group Recommendations with Factorization Machines	Michael Szubartowicz
RQ3	Adapting Process Models via an Automated Planning Approach	Bernd Heinrich, Alexander Schiller, Dominik Schön, Michael Szubartowicz
RQ4	Data Quality in Recommender Systems: The Impact of Completeness of Item Content Data on Prediction Accuracy of Recommender Systems	Bernd Heinrich, Marcus Hopf, Daniel Lohninger, Alexander Schiller, Michael Szubartowicz
RQ5	Something's Missing? A Procedure for Extending Item Content Data Sets in the Context of Recommender Systems	Bernd Heinrich, Marcus Hopf, Daniel Lohninger, Alexander Schiller, Michael Szubartowicz

3 Paper 1: The Way to the Stars: Explaining Star Ratings in Online Consumer Reviews



The Way to the Stars: Explaining Star Ratings in Online Consumer Reviews

Markus Binder[a], Bernd Heinrich[a,*], Marcus Hopf[a], Michael Szubartowicz[a]

a Department of Management Information Systems, University of Regensburg, Universitätsstr. 31, 93053 Regensburg, Germany

** Corresponding author. Tel.: +49 941 943 6101. Email: bernd.heinrich@ur.de*

Declarations of interest: None

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Abstract: Online consumer reviews are important performance indicators for businesses since they constitute essential sources of information for consumers. To gain detailed insights from these reviews, researchers have already used features (such as the feature *food quality* of a restaurant being part of the general feature perspective *item aspects*) derived from review texts to explain associated star ratings. However, existing literature analyzes only certain feature perspectives, enabling just a partial view. Therefore, we leverage four different feature perspectives expressed in consumer reviews (each comprising easy-to-interpret features) in an explanatory model to study whether star ratings can be explained by these feature perspectives. The evaluation on three large real-world datasets shows that the proposed feature perspectives explain star ratings considerably well (Nagelkerke pseudo R-squared of 65-70%) with substantial contributions of each feature perspective. In particular, the perspective *user characteristics* – rarely discussed in related literature – yields the second highest contribution, while *item aspects* contribute the most. Besides valuable implications for research, our work indeed allows well-founded actions for consumers, web portals and businesses.

Keywords: explanatory analysis; feature perspectives; online consumer reviews; star ratings

1 Introduction

With the growing number of people seeking and purchasing goods online [1], the volume and variety of online consumer reviews on web portals such as Amazon or TripAdvisor are vastly increasing [2–4]. Thereby, online consumer reviews constitute a vital object of study of electronic word-of-mouth (EWOM), which is a major and highly attractive research topic in the field of information systems [5]. Further, it is widely recognized that comprehensible and trustworthy product reviews are a major purchase influence factor [6–8]. Thus, online consumer reviews regarding items (e.g., laptops or restaurant visits) are important instruments for users of web portals to overcome information asymmetries about these items [9]. In addition, online consumer reviews (as part of EWOM) are important performance indicators for businesses and web portal providers [5]. For instance, careful improvements of products as well as the creation of ideas for new products based on users' preferences and valenced review statements are possible [10,11]. This is, such reviews comprise rich information [12–14] and typically consist of a star rating (e.g., 1 to 5 stars) representing the overall user assessment and a textual part. Besides the frequently analyzed EWOM-dimensions valence and volume, recent research started to investigate the semantic and lexical content of EWOM [5]. Here, a major goal of research is to derive insights from star ratings and the semantic and lexical content of review texts [15,16] in order to understand why users rated an item the way they did. Therefore, it is important to leverage features (e.g., the feature *food quality* of a restaurant) derived from review texts to explain associated star ratings. More precisely, we aim to analyze relationships between several features of different feature perspectives (e.g., the feature perspective *item aspects* including the feature *food quality*) expressed in reviews as independent variables and star ratings as dependent variable, which is enabled by an *explanatory* model [17]. Thereby, it is vital to utilize independent variables representing *easy-to-interpret* features (e.g., features that can be traced back to its semantically related feature terms in the review texts) in such an explanatory model which enables to derive both comprehensible and well-founded insights.

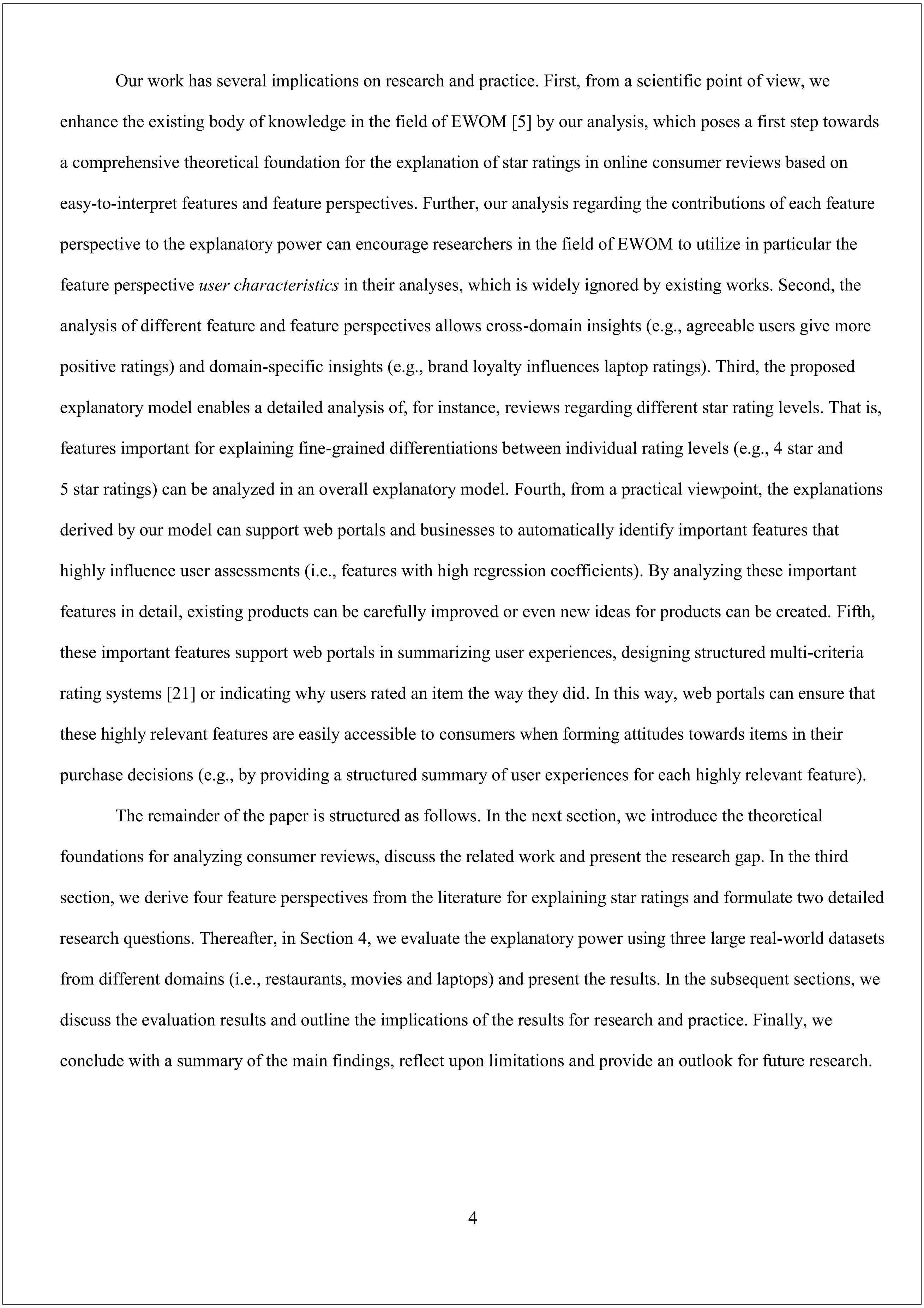
The relevance of such explanatory analyses has been acknowledged by recent works (e.g., [12,18–20]) proposing explanatory models for star ratings based on selected single features. For instance, some works focus on features towards particular *item aspects* in review texts (e.g., food quality of a restaurant) [12,21], while other works aim at specific *user context* features (e.g., dining companions) [22]. Here, existing approaches consider selected features of at most two different feature perspectives in their analyses, enabling only a partial view. For instance, the feature perspective *user characteristics* encompasses personal factors such as user personality or social identity. While it has recently been utilized in the research on personality-based recommender systems (e.g., [18]), it is rarely analyzed in the context of explaining star ratings. In addition, existing works do not investigate the contribution of each individual feature perspective to the explanatory power of their proposed models. This would give important insights, such as that *user characteristics* – rarely discussed in related literature – constitute the feature perspective with the second highest contribution to the explanation of star ratings, which calls for researchers to incorporate this feature perspective in their analysis of online consumer reviews.

In this paper we are the first (A) to integrate the features of more than two feature perspectives into a unified model for explaining star ratings and (B) to analyze the relative importance of each feature perspective for the explanatory power of this model. Therefore, this work could serve as a first step for enabling a thorough understanding of star ratings (cf. discussion in Section 2.1). With this in mind, we focus on the following research questions:

RQ1: *To what extent can features of different feature perspectives explain star ratings in online consumer reviews?*

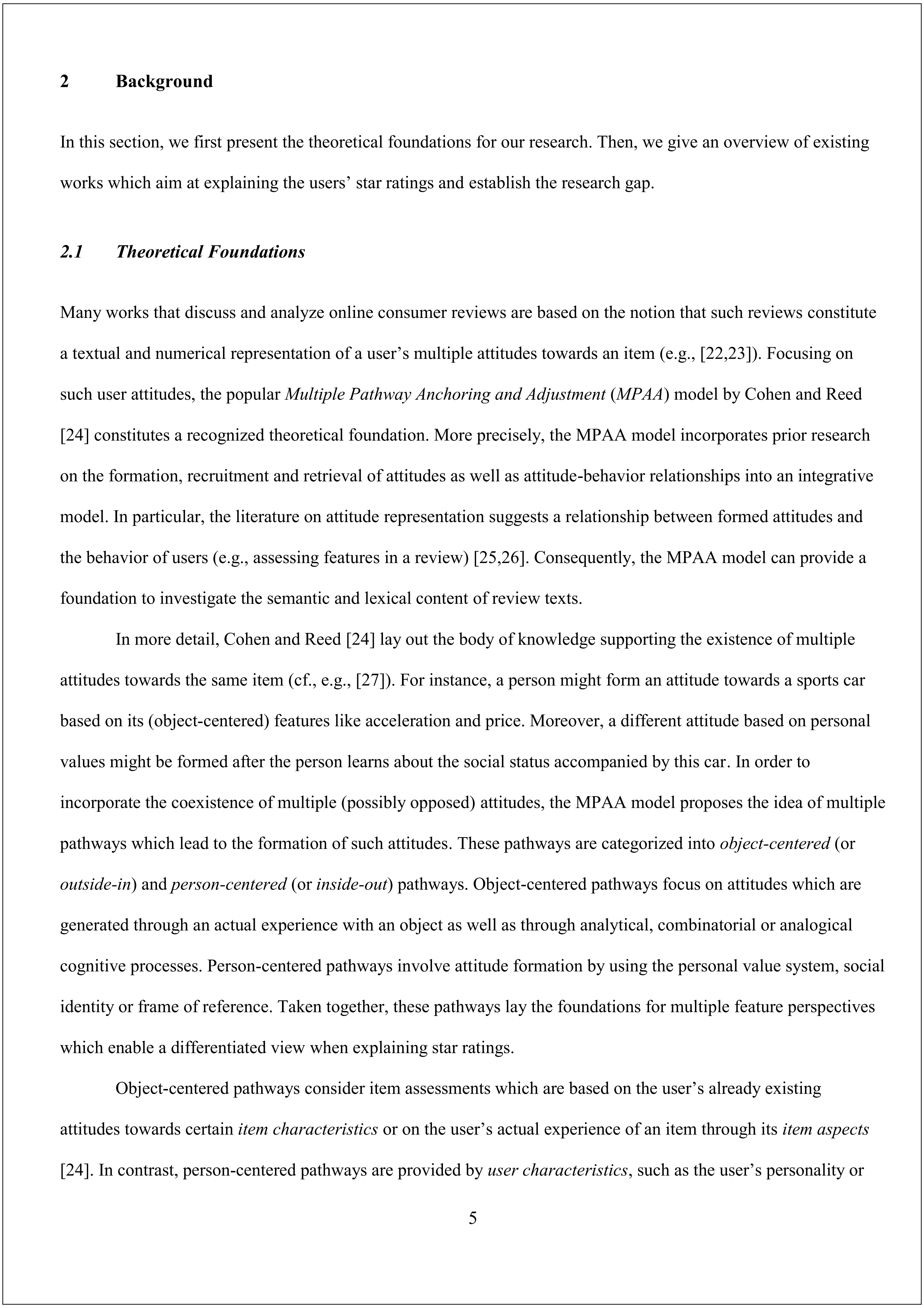
RQ2: *How much does each individual feature perspective contribute to the explanatory power of the unified model?*

To address these questions, we derive the four object and person-centered feature perspectives *item characteristics, item aspects, user characteristics* and *user contexts* from the popular *Multiple Pathway Anchoring and Adjustment* (MPAA) model and unify these feature perspectives into one single model used to explaining star ratings. To extract easy-to-interpret features from a very large number of review texts and thus operationalize the feature perspectives, we apply the state-of-the-art deep learning language model BERT [19]. Given this set of extracted features of different perspectives as independent variables, we evaluate their explanatory power by using the generalized ordered probit regression model (GOPM, cf. [20]). First, we find that the proposed feature perspectives explain star ratings considerably well, which is indicated by Nagelkerke pseudo R-squared values of 65% up to 70%. In comparison, similar works reach Nagelkerke pseudo R-squared values of up to 44% or analyze only 1 star and 5 star ratings. Second, calculating partial R-squared values shows substantial contributions of each feature perspective to the explanatory power of the unified model. In particular, *user characteristics* – rarely discussed in related literature – constitute the feature perspective with the second highest contribution to the explanatory power, while *item aspects* – capturing the user's experience of an item – contribute the most. Additionally, the feature perspective *item characteristics* is able to explain ratings better than *user contexts* for search goods such as laptops while the opposite holds in the restaurant domain.



Our work has several implications on research and practice. First, from a scientific point of view, we enhance the existing body of knowledge in the field of EWOM [5] by our analysis, which poses a first step towards a comprehensive theoretical foundation for the explanation of star ratings in online consumer reviews based on easy-to-interpret features and feature perspectives. Further, our analysis regarding the contributions of each feature perspective to the explanatory power can encourage researchers in the field of EWOM to utilize in particular the feature perspective *user characteristics* in their analyses, which is widely ignored by existing works. Second, the analysis of different feature and feature perspectives allows cross-domain insights (e.g., agreeable users give more positive ratings) and domain-specific insights (e.g., brand loyalty influences laptop ratings). Third, the proposed explanatory model enables a detailed analysis of, for instance, reviews regarding different star rating levels. That is, features important for explaining fine-grained differentiations between individual rating levels (e.g., 4 star and 5 star ratings) can be analyzed in an overall explanatory model. Fourth, from a practical viewpoint, the explanations derived by our model can support web portals and businesses to automatically identify important features that highly influence user assessments (i.e., features with high regression coefficients). By analyzing these important features in detail, existing products can be carefully improved or even new ideas for products can be created. Fifth, these important features support web portals in summarizing user experiences, designing structured multi-criteria rating systems [21] or indicating why users rated an item the way they did. In this way, web portals can ensure that these highly relevant features are easily accessible to consumers when forming attitudes towards items in their purchase decisions (e.g., by providing a structured summary of user experiences for each highly relevant feature).

The remainder of the paper is structured as follows. In the next section, we introduce the theoretical foundations for analyzing consumer reviews, discuss the related work and present the research gap. In the third section, we derive four feature perspectives from the literature for explaining star ratings and formulate two detailed research questions. Thereafter, in Section 4, we evaluate the explanatory power using three large real-world datasets from different domains (i.e., restaurants, movies and laptops) and present the results. In the subsequent sections, we discuss the evaluation results and outline the implications of the results for research and practice. Finally, we conclude with a summary of the main findings, reflect upon limitations and provide an outlook for future research.



2 Background

In this section, we first present the theoretical foundations for our research. Then, we give an overview of existing works which aim at explaining the users' star ratings and establish the research gap.

2.1 Theoretical Foundations

Many works that discuss and analyze online consumer reviews are based on the notion that such reviews constitute a textual and numerical representation of a user's multiple attitudes towards an item (e.g., [22,23]). Focusing on such user attitudes, the popular *Multiple Pathway Anchoring and Adjustment* (*MPAA*) model by Cohen and Reed [24] constitutes a recognized theoretical foundation. More precisely, the MPAA model incorporates prior research on the formation, recruitment and retrieval of attitudes as well as attitude-behavior relationships into an integrative model. In particular, the literature on attitude representation suggests a relationship between formed attitudes and the behavior of users (e.g., assessing features in a review) [25,26]. Consequently, the MPAA model can provide a foundation to investigate the semantic and lexical content of review texts.

In more detail, Cohen and Reed [24] lay out the body of knowledge supporting the existence of multiple attitudes towards the same item (cf., e.g., [27]). For instance, a person might form an attitude towards a sports car based on its (object-centered) features like acceleration and price. Moreover, a different attitude based on personal values might be formed after the person learns about the social status accompanied by this car. In order to incorporate the coexistence of multiple (possibly opposed) attitudes, the MPAA model proposes the idea of multiple pathways which lead to the formation of such attitudes. These pathways are categorized into *object-centered* (or *outside-in*) and *person-centered* (or *inside-out*) pathways. Object-centered pathways focus on attitudes which are generated through an actual experience with an object as well as through analytical, combinatorial or analogical cognitive processes. Person-centered pathways involve attitude formation by using the personal value system, social identity or frame of reference. Taken together, these pathways lay the foundations for multiple feature perspectives which enable a differentiated view when explaining star ratings.

Object-centered pathways consider item assessments which are based on the user's already existing attitudes towards certain *item characteristics* or on the user's actual experience of an item through its *item aspects* [24]. In contrast, person-centered pathways are provided by *user characteristics*, such as the user's personality or social identity, and specific situational *user contexts* that influence the user's assessment of an item [23]. Therefore, the object-centered feature perspectives *item characteristics* and *item aspects* as well as the person-centered feature perspectives *user characteristics* and *user contexts* are described in the following:

Based on *item characteristics* like the *genre* of a movie or the *cuisine* of a restaurant, the user's preliminary attitudes and preferences can be analyzed. In particular, the preferences of a user can be determined based on the characteristics of the items the user liked or disliked in the past [28]. That is, even for items unfamiliar to the user, it is aimed to infer preferences based on familiar items with similar item characteristics [24]. Based on item characteristics, a user explicitly builds her or his preconception for an item ex ante. For example, in the domain of restaurants, a user can have an already existing positive attitude towards the value *Thai food* of the item characteristic *cuisine* of a restaurant. In that case, the user's star rating for this restaurant thereby may be influenced through her or his positive attitude towards *Thai food*. This suggests that item characteristics are related to the users' star ratings [29].

In contrast to item characteristics, *item aspects* (e.g., *service* at a restaurant) and their sentiments (often called aspect-based sentiments) capture attitudes, which are formed after actually experiencing an item and not beforehand (cf. [24]). In particular, a user can determine her or his sentiments towards an item aspect in a very detailed way, as the actual experience enables the user to substantiate or modify her or his existing attitudes towards an item's aspects. This experience may also lead to the formation of attitudes towards hitherto undiscovered item aspects. For instance, a user might expect a pleasant *service* before going to a restaurant. After visiting this restaurant and being served by an impatient waiter, the user would form a negative sentiment towards the experienced *service*. In consequence, a negative sentiment could have a high impact on the assigned star rating for this restaurant. Since this perspective comprises detailed user assessments, it is frequently used to analyze and explain star ratings (e.g., [4,12,22,30]).

In contrast to the object-centered perspectives above, *user characteristics* outline personal factors such as user personality or social identity. By definition, user personality aims to capture psychological traits, which account for individual differences in behavior and experience. Amongst other models such as the Myers-Briggs Type Indicator [31], the Five-Factor Model [32] is the most dominant and widely applied personality model comprising the five factors *openness to experience*, *conscientiousness*, *extraversion*, *agreeableness* and *emotional* *stability*. This model aims to enable a comprehensive, but nonetheless detailed conception of the personality of an individual person. Thereby, the Five-Factor model is referred to as the most comprehensive and parsimonious model of personality [33]. The underlying intuition suggests that the facets of the user's personality allow for a more profound understanding of the user actions, reactions and assessments. Here, studies have shown that the Five-Factor model is particularly useful to examine online behavior in the context of EWOM [33,34]. One reason for that is that (Five-Factor) personality traits effect how individuals attain gratification, for which (the creation of) EWOM is a relevant medium [34]. For example, Hu and Pu [35] discovered that users who score high on *agreeableness* would tend to give higher star ratings. In that line, agreeable users might value harmony and fairness and thus be more inclined not to give extremely negative ratings [24]. Moreover, the analysis in [36] shows, inter alia, that reviews from users scoring high in emotional stability affect similar users while reviews from users with high emotional range do not affect users with similar personality. Analogously to user personality, the relation of a user's social identity to an item can be a significant influence factor for star ratings. According to [24], social identity can be defined by social categories such as demographics, social roles and shared consumption patterns. Here, users with similar demographic background are expected to rate items similarly. For example, user characteristics such as age or gender might influence a user's star rating for a movie. In total, this indicates that user characteristics can be important to explain star ratings of online consumer reviews.

A further person-centered perspective is the *context of a user*, which, in contrast to item aspects, is not directly related to the rated item. Instead, the user contexts refer to the situational circumstances in which a user interacts with the rated item. Contextual features are, for instance, *time*, *location*, *weather and temperature*, *mood*, and *social encounters* [37]. These user contexts already have been discussed as a potential influencing factor for star ratings [4]. For example, Radojevic et al. [29] analyzed that business travelers tend to be more critical in their ratings of items, which might be reasoned with a higher level of stress on business trips. This indicates that the user contexts can influence star ratings.

2.2 Related Work

In this section, we embed our research into the field of EWOM and discuss existing research, which aims at *explaining* star ratings of online consumer reviews. Regarding the framework of existing EWOM literature by [5], our research can be classified as *evaluation of EWOM* focusing on the *investigation of the semantic and lexical content* of online consumer reviews. In contrast to several existing works in this research strand, which investigate the *coherence* between feature assessments and product-level assessments (e.g., [38]), we analyze the *relation* between features of different perspectives and product-level user assessments. Before we discuss the works in our research strand, we outline and delimit related research strands. Existing works, which focus exclusively on a predictive analysis (e.g., recommender systems) such as [18], [39], [40] or [41] and which do not aim to explain or interpret the star ratings, are out of scope for our research. As outlined by [17], prediction and explanation are two different objectives and thus need to be assessed differently. When predicting the relation between different variables, the underlying (theoretical) construct is not focused on. In that line, variables used in predictive approaches such as latent factors are not necessarily interpretable and are not aimed to explain the underlying construct. Similarly, works in literature exist, which rely on research techniques such as consumer surveys or group interviews. A restriction of these works is often the limited size of data used for evaluation (usually well below 1,000 observations). As a result, a more complex explanatory model cannot be evaluated on such a smaller dataset, since the resulting ratio of observations to variables in the model would be too small to obtain reliable results [42]. Furthermore, the observations and data used in these evaluations is often influenced by the fact that interviewed users answer the survey solely based on their imagination and expectations but not on, for instance, real experiences, as they actually did not buy, consume nor use an item in reality. Because of these important differences, these works are also out of the scope for our research.

In accordance with the guidelines of standard approaches to prepare the related work (e.g., [43]), we searched the databases ACM Digital Library, AIS Library, EBSCO Host, IEEE Xplore and ScienceDirect without posing a temporal restriction using the search term *(explain* OR explan* OR understand*) AND ("star ratings" OR "consumer ratings" OR "user ratings" OR "customer ratings") AND review*.* This search led to 305 papers, which were manually screened based on title, abstract and keywords resulting in 14 papers (the vast majority of the 305 papers focused on predictive analyses or analyzed the helpfulness of online consumer reviews for other users). A detailed analysis of these 14 papers led to 10 papers relevant for our research. Additionally, we performed a forward and backward search starting from these 10 relevant papers. After all, 17 papers were identified as relevant for our work at hand and are grouped by their considered feature perspective(s) in Table 1. These works are discussed in the following regarding (A) the considered feature perspectives and (B) the assessment of contributions of the considered feature perspectives (i.e., the relative importance of the feature perspectives) to the explanatory power of the proposed models.

	Ad (A)				**Ad (B)**
	Object-centered Feature Perspectives		*Person-centered Feature Perspectives*		*Assessing the contributions of feature perspectives to the explanatory power of the model*
	Item Characteristics	*Item Aspects*	*User Characteristics*	*User Contexts*	
[12]; [22]; [44]; [20]; [45]; [46]; [47]; [48]	n/a	Selected features such as food and price for restaurants or cleanliness and service	n/a	n/a	n/a
[29]	Selected features such as hotel's star classification and hotel price	n/a	n/a	Only the features trip purpose and date	n/a
[4]; [49]; [50]; [51]; [52]; [53]; [54]	n/a	Selected features such as food, service and price	n/a	Selected features such as trip purpose and travel party	n/a
[55]	n/a	n/a	Selected features such as user personality and metadata	Only the feature trip purpose	n/a

Table 1. Existing Approaches for Explaining the Star Ratings of Online Consumer Reviews

Ad (A): The first set of works contains eight approaches considering only item aspects. Binder et al. [20] aim at a methodological contribution by proposing the GOPM to analyze star ratings and evaluate this model against the common linear regression model. To this end, aspect-based sentiments are only used for demonstration purposes. In their analysis, Jabr et al. [12] focus on 1 star and 5 star ratings aiming to retrieve unambiguous sentiment data, which concentrates their results on explaining the basic rating tendency. Moreover, Chatterjee [22], Chen et al. [44], Guo et al. [47], Linshi [45] and Liu et al. [48] also aim to explain star ratings based on aspect-based sentiments, but do not provide a detailed analysis regarding different steps of the rating scale, which may be interesting in their research. Lastly, Lacic et al. [46] analyze star ratings by determining correlation coefficients between these ratings and individual aspect-based sentiments rather than establishing an explanatory model.

The second set of works comprises only the work of Radojevic et al. [29], which consider single features being part of the perspectives item characteristics and user contexts to explain star ratings. However, these features are extracted only from structured data, excluding the information contained in review texts.

Indeed, there also exist seven approaches that analyze the impact of item aspects combined with user contexts on star ratings. The two consecutive works of [51] and [50] analyze star ratings to determine how much these ratings vary between reviews for different items and within the same item in their model. Thereby, a different set of coefficients for each item is used, which limits the reliability of the results for items not having a considerably high number of available reviews. Ye et al. [52] aim to explain the sub-ratings for service quality and value for money rather than the overall star rating. The work of [53] provides a detailed explanatory analysis, in particular, of the coefficients in their regression model, but with a special emphasis on the traveling domain. Further, Luo and Tang [49] and Xiang et al. [4] aim to examine the influence of aspects and contexts on the star rating. Another recent work analyzes the impact of the feature perspectives item aspects and user contexts on star ratings in the domain of airline traveling [54]. The authors also utilize user features on a cultural-level, that is, these features are derived solely based on the citizenship of a user. However, the authors state that "people within a same culture can have different types of personality traits, which [...] cannot be measured" by these features and recommend that "future researchers could thereby choose more suitable or alternative measures" for the feature perspective user characteristics. In particular, this means that their considered country-related features are hardly suitable for a review-level analysis of star ratings since all users from the same country have the exact same feature values for this feature perspective. Hence, only suitable features from only two perspectives are considered in this work. In addition, none of these seven works investigates whether their explanatory models can explain different steps of the rating scale (e.g., why users rated an item with 4 or 5 stars).

Finally, there also exists a recent work that analyzes the impact of user characteristics on star ratings [55]. In particular, this work focuses on the impact of the Five-Factor user personality traits on star ratings. Additionally, they analyze one feature ('travel type') as user context in the hotel domain (i.e., business trips vs. leisure trips).

Ad (B): Since the first set of works focuses solely on one feature perspective, an analysis and comparative assessment regarding the contributions of different feature perspectives to the explanatory power of the models is not possible. The sets of works considering two feature perspectives also lack such an analysis, which would give important insights, even though only two feature perspectives are considered. While the work of [54] investigates interdependent moderator effects between the considered feature perspectives, the contributions of the two feature perspectives to the explanatory power of the model are not assessed in this work either.

To conclude, there are already interesting works that aim to explain the overall star ratings of online consumer reviews based on different features and feature perspectives. However, these contributions (A) only consider selected suitable features (often only one single feature) of at most two feature perspectives in their analysis. Further, (B) none of the existing works assesses the contribution of each feature perspective in terms of explanatory power which could allow for valuable insights on the relative importance of different feature perspectives (and their features) on user assessments in online reviews. In this paper, we aim at filling the identified research gap by (A) leveraging more than two feature perspectives for explaining star ratings and by (B) analyzing the relative importance of each feature perspective for the explanatory power of the proposed model.

3 Explaining Star Ratings in Online Consumer Reviews

To address the identified research gap, we derive four object and person-centered feature perspectives from the MPAA model and unify these feature perspectives into one single model to explain star ratings in online consumer reviews (cf. Figure 1).

Regarding *object-centered feature perspectives*, Cohen and Reed [24] discuss that a user's attitudes depend on both initial knowledge and actual experience. To be more precise, *item characteristics* like the genre of a movie are usually known before experiencing an item and thus can be used to form an initial attitude by evaluating these (known) item characteristics. This corresponds to the pathway *Analytical Attitude Construction* of the MPAA model. Thereby, star ratings may be influenced in a positive or negative way depending on the (expected) peculiarity or importance of item characteristics. For instance, focusing only on item characteristics, a user might form positive attitudes towards a movie because she or he likes the genre and director of the movie, but might establish also some negative attitudes because she or he has no sympathy for the main actor. In addition, actual exposure to an item may influence its assessment. In this case, an attitude is formed based on a user's directly captured perception of the item, which can be structured by *item aspects*. While most online consumer reviews are composed after a direct experience, the corresponding pathway *Direct/Imagined Experience with the Object* of the MPAA model also encompasses attitudes formed by a simulated experience. Such experiences, however, can be structured according to item aspects as well. Therefore, the detailed analysis regarding item aspects can give further insights into how the user's overall star rating can be explained.

With regard to *person-centered feature perspectives*, Cohen and Reed [24] argue that a user might generate an attitude by relating and evaluating an item to her or his characteristics such as personal traits or social identity. This corresponds to the pathway *Social Identity-Based Attitude Generation* of the MPAA model. As such, *user* *characteristics* like agreeableness can be examined to point out similarities and differences between individuals which can be reflected in similar or dissimilar star ratings. Additionally, an important factor of the MPAA model is the (temporal) change of attitudes due to contextual variations. Depending on a certain context, different subsets of personal beliefs and values might be used to form an attitude. For instance, stress situations and time constraints might lead to favoring junk food over healthier options while other contexts might have the reverse effect. In the MPAA model these contextual variations are reflected in the pathway *Value-Driven Attitudes*. As a consequence, user contexts may play an important role for the explanation of star ratings.

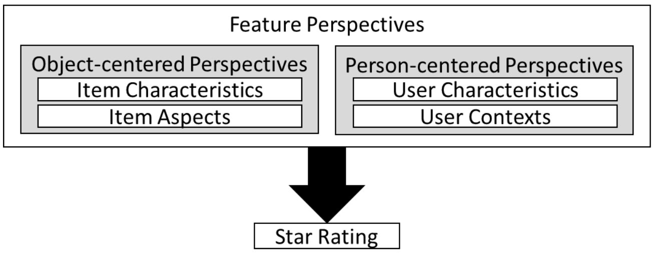


Figure 1. Research Model for Explaining Star Ratings in Online Consumer Reviews

As noted in the previous section, related work focuses on at most two feature perspectives. We argue that a broader view should be established to explain star ratings in online consumer reviews and hence, we pursue RQ1. While we analyze the overall explanatory power in RQ1, it is of high relevance to analyze how much each individual feature perspective (and their features) contributes to this overall explanatory power. This enables to investigate if one individual feature perspective surpasses all others regarding its contribution to the overall explanatory power or whether the combination of the object-centered and person-centered feature perspectives is of additional value. This would substantiate the theoretical grounding given by the MPAA model and therefore, RQ2 is proposed.

To answer these two research questions, we deliberately outline quantitative analyses instead of following a common hypothesis-driven framework. As we aim to evaluate the quantitative extent of the explanatory power from different angles, a solely hypothesis-driven framework would limit the scope of our analyses. We argue that using the GOPM in combination with the Nagelkerke pseudo R-squared (cf. Section 4.2 below) allows for deeper and more differentiated insights of the results of our analyses. Moreover, it has been recognized that focusing on significance results can be misleading when analyzing very large datasets (such as the review datasets analyzed in our evaluation) [56]. As the significance of an effect does not provide any information about the magnitude of the effect, it is even argued that the "notion of statistical significance is not that relevant to big data" [57]. Thus, we focus on quantitative analyses.

4 Analysis and Results

We start this section by introducing the selected datasets and describing their preparation for our evaluation. Thereafter, we outline the methodology for our explanatory analysis. We end the section by presenting the results for RQ1 and RQ2.

4.1 Datasets and Data Preparation

For our analysis, we used three large real-world review datasets from the commonly utilized review domains of restaurants, movies and laptops. Reviews of these three domains are also used for analyses in related research fields such as sentiment analysis or design of EWOM systems [15,58] and allow for a broad view of different types of (reviews for) products and services. For instance, restaurants and movies constitute experience goods, which are goods that have to be mainly experienced by the user to properly assess their quality. In contrast, laptops constitute search goods, which are goods whose quality can be assessed to a greater extent without personal experience [59]. By considering these three multi-faceted datasets for the evaluation, it is possible to derive cross-domain as well as domain-specific insights (e.g., features being important in only one domain). While their properties are typical for online consumer reviews, the three datasets exhibit a higher diversity representing three varying market fields of e-commerce.

In more detail: The restaurant dataset consists of 2.4m reviews for restaurants, bars and cafés in New York City from an established web portal for reviews regarding local businesses. The movie dataset consists of 1.2m reviews for movies and other video content (e.g., documentaries, recorded concerts, etc.), while the laptop dataset consists of 270k reviews for laptops, notebooks and tablet computers, both originating from the Amazon review dataset provided by [60]. Thereby, each review of the above datasets consists of a textual consumer review with an associated star rating on a five-tier scale from 1 star to 5 stars. In order to avoid biases due to specific time frames, the datasets contain reviews from large time periods of ten years or more (time span restaurants: 2008-2017; time span movies: 2000-2018; time span laptops: 2002-2018). Moreover, the datasets cover a broad range of items (e.g., from bistros to gourmet restaurants as well as from economical to high-end laptops). Further, each dataset exhibits the widely recognized "J-shaped" rating distribution (e.g., [61]). To avoid biases due to the skewed rating distribution, we used stratified samples of the datasets with equal rating distributions, similar to [62] and [23]. Thereby, each sample is large enough to analyze various independent variables of different feature perspectives in an explanatory model. That is, the number of events per independent variable (EPV) is higher than 1,000 in our analyses as the smallest sample has 75,000 reviews. This is considered as clearly sufficient in literature (e.g., [63]). An overview of the basic specifications regarding the datasets is given in Table 2.

Dataset	**Restaurants**	**Movies**	**Laptops**
# of reviews	2,396,650	1,167,071	271,883
Rating distribution, i.e., (relative) # of ratings per level of star rating	1 star: 9% (~207k) 2 stars: 9% (~214k) 3 stars: 16% (~389k) 4 stars: 34% (~807k) 5 stars: 33% (~779k)	1 star: 8% (~98k) 2 stars: 6% (~65k) 3 stars: 10% (~117k) 4 stars: 19% (~227k) 5 stars: 57% (~660k)	1 star: 17% (~45k) 2 stars: 7% (~19k) 3 stars: 8% (~23k) 4 stars: 18% (~50k) 5 stars: 49% (~134k)
# of reviews in the sample with equal rating distribution	500,000 [100,000 per rating level]	250,000 [50,000 per rating level]	75,000 [15,000 per rating level]
# of users in the sample # of items in the sample	233,854 10,480	208,787 13,677	69,091 3,441

Table 2. Description of the Datasets

To evaluate our research questions, we operationalized both the person-centered and object-centered feature perspectives (cf. Section 2) as outlined in the following. The features for the perspective *item characteristics* are directly given in each dataset as structured data. In contrast, the features of the feature perspectives *item aspects*, *user characteristics* and *user contexts* are contained in the unstructured review texts. Here, features can be extracted from textual data by using either unsupervised or supervised methods. Unsupervised extraction methods result in abstract representations that are – a priori – independent of any predefined feature or feature perspective. For example, the unsupervised extraction method topic modeling yields topics comprising a specific set of cooccurring words from review texts. Thereby, existing literature (e.g., [64]) argues that it is very challenging to interpret such abstract representations, as it remains unclear what they really mean, and that these abstract representations do not align with predefined features in general. In addition, Vallurupalli et al. [65] argue that the interpretations of topics strongly depend on the human individuals interpreting the topics. Further, they state that the findings obtained from topic modeling is also highly dependent on the used datasets, even if they are of the same domain. Thus, such findings can hardly be generalized to other datasets or domains, as strongly different topics could be identified. In total, these abstract representations require additional work to derive comprehensible insights. Therefore, we decided to choose a supervised feature extraction method based on BERT [19], which is a state-of-the-art deep learning language model. Here, we first selected and analyzed features and the corresponding feature terms in the review texts for each of the three feature perspectives in line with existing works (cf. Table 3). In particular, this initial analysis showed that each such feature can be traced back to semantically related feature terms, which entails a direct interpretation. By individually training the supervised language model BERT on annotated data (i.e., feature terms) for each feature, the extraction of these features is enabled for a large number of reviews in the considered datasets (cf. Table 2). Doing this, for instance, the term "*bartender*" can be extracted as semantically related feature term for the item aspect *service* in the sentence "*The bartender was charming*". Here, the language model BERT is further able to identify semantic and lexical representations of natural language [66] by considering whole sentences for feature extraction, which enables a semantically sensitive feature extraction. For instance, the word "*bartender*" would be extracted for the item aspect *service* in the sentence "*The bartender was charming*", but not in the sentence "*The mojito was listed as bartender's choice"* due to different semantical meanings of the term *"bartender*". To further improve the quality of the feature extraction (cf. Section 5), we used the post-trained BERT models for each specific domain of our considered datasets [67], since the post-trained BERT models have a stronger alignment to the domain-specific use of language. Summing up, by the use of this supervised deep learning language model, we are able to extract and utilize features, which are easy-to-interpret (due to its semantically related feature terms in the review texts).

An overview of the considered features for each feature perspective and each dataset is given in Table 3. We selected five *item characteristics* for each dataset with the lowest pairwise correlations and a sufficient number of occurrences (i.e., assigned to more than 10% of items), such as movie genre or cuisine, which is in line with approaches such as [68]. Furthermore, for each dataset we extracted six item aspects, five user characteristics and five user contexts (cf. Table 3). To capture both the users' situational circumstances and actual experiences of items expressed in the review texts, we extracted the sentiments towards features of the perspectives *user contexts* and *item aspects* [50]. More precisely, we extracted *item aspect*-based sentiments and *user context*-based sentiments from the review texts by firstly conducting aspect term and context term extraction and subsequently conducting the task of term-based sentiment classification. For instance, in the exemplary sentence "*The waiter was very friendly.*"

first the aspect term "*waiter*" was extracted by BERT and assigned to the item aspect *service*. Subsequently, BERT assigned a positive sentiment towards that aspect term based on the term "*very friendly*". Here, all extracted terms which could not be assigned to a specific item aspect or user context were subsumed under the features *miscellaneous item aspects* or *miscellaneous user contexts*.

Before evaluating our model in detail, we analyzed the quality of the preliminary analysis, which means, the aspect term extraction conducted by the deep learning language model. Thereby, F1 scores of 0.78, 0.75 and 0.76 for restaurants, movies and laptops were achieved. Based on this extraction, the aspect term-based sentiment classification yielded F1 scores of 0.84, 0.86 and 0.80, respectively. All F1 scores are comparable to the state-of-the-art [67]. In particular, to extract *user characteristics*, we also trained an individual BERT model for each of the Five-Factor personality traits using a common essay dataset containing personality annotations [69], while structured data regarding the users' social identity (e.g., with respect to demographics) was not available in the datasets. The average accuracy of the resulting BERT personality models was 58%, which coincides with the state-of-the-art validity for Five-Factor personality detection from text on the standard essay benchmark dataset [70]. Moreover, test-retest correlations on the consumer review datasets for successive 6-month intervals were 0.73 on average. Thus, the reliability of the applied BERT personality models is in line with Five-Factor personality detection based on questionnaires (with test-retest correlations typically ranging from 0.65 to 0.85) and similar to existing approaches extracting Five-Factor personality traits from social media texts [71].

Finally, to verify the stability of our explanatory model, multicollinearity between the independent variables was measured by the variance inflation factor (VIF). The maximum VIFs ranged from 1.76 to 2.78 and the average VIFs ranged from 1.23 to 1.38, whereby VIF values less than ten are uncritical regarding model stability [72].

Dataset	**Restaurants**	**Movies**	**Laptops**
Considered item characteristics (independent variables $x_{IC1}, \dots, x_{IC5}$; short x_{IC})	5 characteristics (in line with [73]): cuisine, happy hour specials, noise level, private parking lot, vegetarian food	5 characteristics (in line with [74]): director, price level, genre, languages, cast	5 characteristics (in line with [74]): brand, graphic card, hard drive, processor, memory
Considered item aspects (independent variables $x_{AS1}, \dots, x_{AS6}$; short x_{AS})	(Sentiments towards) 6 aspects (in line with [20]): ambience, food quality, food quantity, price, service, miscellaneous	(Sentiments towards) 6 aspects (in line with [75]): acting, story, cinematography, price, music, miscellaneous	(Sentiments towards) 6 aspects (in line with [76]): battery, performance, price, screen/design, support, miscellaneous
Considered user characteristics (independent variables $x_{UC1}, \dots, x_{UC5}$; short x_{UC})	5 characteristics of the Five-Factor Model (in line with [77]): extraversion, emotional stability, agreeableness, conscientiousness, openness to experience		

Considered user contexts (independent variables $x_{UCxt1}, \dots, x_{UCxt5}$; short x_{UCxt})	5 context variables (in line with [37]): location, time, social, weather, miscellaneous	5 context variables (in line with [78]): purchase type, intended use, social, time, miscellaneous	5 context variables (in line with [78]): intended use, operating system, software, connectivity, miscellaneous
Multicollinearity between the independent variables	Average VIF: 1.28 Maximum VIF: 2.59	Average VIF: 1.38 Maximum VIF: 2.78	Average VIF: 1.23 Maximum VIF: 1.76

Table 3. Features of the Datasets after Data Preparation

4.2 Methodology

As introduced above, our explanatory model comprises the feature perspectives item characteristics (IC), item aspects (IA), user characteristics (UC) and user contexts (UCxt). To explain star ratings and evaluate the explanatory power, we use the GOPM and the Nagelkerke pseudo R-squared [79] both as proposed by [20]. The methodological reasons for this choice are outlined in the following. In general, the GOPM is based on the classical ordered probit model [80]. According to the classical ordered probit model, underlying linear preferences $R_*^i \in \mathbb{R}$ are modelled using the independent variables x_{IC}, x_{IA}, x_{UC} and x_{UCxt} representing the four feature perspectives (cf. Table 3). To ensure reliable results (indicated by a high EPV value, cf. Section 4.1), the same set of coefficients for all reviews is used. This leads to a preference model given by

$$R_*^i = \beta_{IC} x_{IC}^i + \beta_{IA} x_{IA}^i + \beta_{UC} x_{UC}^i + \beta_{UCxt} x_{UCxt}^i + \epsilon, \qquad (1)$$

where $\beta_{IC} (= \beta_{IC1}, \beta_{IC2}, \dots, \beta_{ICn})$, β_{IA} , β_{UC} and β_{UCxt} denote the parameters with respect to the independent variables $x_{IC}^i \left(= x_{IC1}^i, x_{IC2}^i, \dots, x_{ICn}^i\right)$, x_{IA}^i, x_{UC}^i and x_{UCxt}^i in the i-th review, and $\epsilon \sim N(0,1)$ denotes the random error term. Then, a discrete random variable $R^i \in \{1, \dots, 5\}$ and thresholds $\theta_1, .., \theta_4$ are used to estimate the actual star rating $r^i \in \{1, .., 5\}$ in the review i, which means, $R^i = 1$ for $R_*^i \leq \theta_1$, $R^i = 2$ for $\theta_1 < R_*^i \leq \theta_2$, … , $R^i = 5$ for $R_*^i > \theta_4$. That is, the parameters $\beta_{IC}, \beta_{IA}, \beta_{UC}$ and β_{UCxt} as well as the thresholds $\theta_1, \dots, \theta_4$ are determined according to the classical ordered probit model.

In addition to this classical ordered probit model, the GOPM methodically uses different coefficients $\beta_{IC}^1, .., \beta_{IC}^4$ instead of a fixed coefficient β_{IC} (analogous for the other perspectives) to account for varying impacts of the independent variables over the rating scale. This means, for each independent variable v as well as for each rating step between 1 and 5, we can determine a particular coefficient. More precisely, the GOPM for the evaluation is given by

$$R^i \leq j \quad \text{if} \quad \beta_{IC}^j x_{IC}^i + \beta_{IA}^j x_{IA}^i + \beta_{UC}^j x_{UC}^i + \beta_{UCxt}^j x_{UCxt}^i + \epsilon \leq \theta_j \quad \text{for} \quad j = 1,2,3,4. \qquad (2)$$

By assigning preference intervals of different sizes to the star ratings, the GOPM can reflect uneven distances within the rating scale, for instance, in contrast to a common linear regression model. As analyzed by [20], in the restaurant domain a rating level of 4 is far closer to a rating level of 5 with respect to the underlying preference than to a rating level of 3. Further, the GOPM accounts for varying impacts over the rating scale by allowing varying coefficients $\beta_{IC}^1, \beta_{IC}^2, \beta_{IC}^3$ and β_{IC}^4. For instance, an unfriendly waiter in a restaurant (i.e., a negative sentiment towards the aspect *service*) may easily drive a user to assign the lowest star rating, while a pleasant service alone will in general not be sufficient to assign the highest star rating.

To assess the explanatory power of the GOPM for star ratings, we use the Nagelkerke pseudo R-squared. This measure compares the likelihood of the GOPM to a null-model [81], which does not take the independent variables from the four feature perspectives into account. That is, the null-model does not distinguish between different reviews, but still determines the thresholds $\theta_1, .., \theta_4$ according to the rating distribution. In detail, the used comparison of likelihoods is equal to the common R-squared measure in case of a linear regression. However, to account for the transformation on the discrete rating scale (cf. Equation (2)), additionally a rescaling to the range [0,1] is used (as denominator in Equation (3)). Overall, and according to [20], the Nagelkerke pseudo R-squared is given by

$$\mathcal{R}_{Nagelkerke}^2 = \frac{1 - \left[\frac{L_{Null-Model}}{L_{GOPM}}\right]^{2/M}}{1 - L_{Null-Model}^{\ \ 2/M}}, \qquad (3)$$

where L_{GOPM} and $L_{Null-Model}$ denote the value of the likelihood function at the maximum likelihood estimate of the GOPM and the null-model, respectively. Further, M denotes the number of observations in the model. Thereby, the range $[0,1]$ of the Nagelkerke pseudo R-squared measure is in accordance with the common R-squared measure for linear regression models. We denote this overall explanatory model, which comprises the GOPM and the feature perspectives item characteristics, item aspects, user characteristics and user contexts, as unified model in the following.

4.3 Results

In the following, we present the results regarding the research questions RQ1, RQ2 based on the three real-world datasets of restaurant reviews, movie reviews and laptop reviews.

Ad RQ1: Overall, our analysis for explaining star ratings yields a Nagelkerke pseudo R-squared value (cf. Equation 3) of 69.8% on the restaurant dataset, 64.9% on the movie dataset and 65.0% on the laptop dataset. For a more detailed analysis, we evaluated how well the unified model explains the star ratings for different steps of the rating scale. To assess the explanatory power for each rating level, we applied Equation 3 separately for each subset of reviews by the assigned star rating. As the results in Table 4 show, star ratings are best explained for reviews with 1 star or 5 star ratings.

Dataset	**Rating Levels**					**Overall**
	1 Star Reviews	**2 Stars Reviews**	**3 Stars Reviews**	**4 Stars Reviews**	**5 Stars Reviews**	
Restaurants (Nagelkerke Pseudo R^2)	81.3 %	62.9 %	51.6 %	64.0 %	80.0 %	69.8%
Movies (Nagelkerke Pseudo R^2)	75.4 %	54.6 %	43.6 %	61.3 %	78.9 %	64.9%
Laptops (Nagelkerke Pseudo R^2)	72.9 %	49.8 %	40.2 %	63.3 %	83.6 %	65.0%

Table 4. Explanatory Power for Different Rating Levels

Further, we also examined the coefficients for the variables of the four feature perspectives (as introduced in Table 3). In detail, we analyzed the coefficients $\beta_v^1, \beta_v^2, \beta_v^3, \beta_v^4$ for each variable v in the model built on each dataset. Here, the coefficients for the variables *weather* in the restaurant domain and *miscellaneous user contexts* in the laptop domain were statistically significant with $p < 10^{-2}$ and all other variables v were statistically significant with $p < 10^{-9}$ (cf. Table 5). Due to length restrictions, the average coefficients $\overline{\beta_v}$=$(\beta_v^1 + \beta_v^2 + \beta_v^3 + \beta_v^4)/4$ regarding the different rating steps are presented only for selected variables v in Table 5 (different coefficients $\beta_v^1, \beta_v^2, \beta_v^3, \beta_v^4$ always had the same sign). As given in Table 5, for instance, the coefficients for the user characteristic *neuroticism* indicate a negative effect on the star rating of a restaurant, movie or laptop. A positive effect is indicated, for instance, by the coefficient of the user characteristic *agreeableness* across all three domains.

	Independent Variable	Restaurant Coefficient	Movie Coefficient	Laptop Coefficient
Item Aspects	*service*	0.266***		
	support			0.147***
	price	0.055***	0.036***	0.154***
	food quality	0.583***		
	story		0.626***	
	performance			0.327***
Item Characteristics	*vegetarian food*	0.018***		
	language		0.065***	
	brand			0.160***
User Characteristics	*agreeableness*	0.240***	0.459***	0.122***
	neuroticism	-0.234***	-0.232***	-0.344***
	conscientiousness	-0.263***	-0.045***	0.038***
	openness	0.158***	0.055***	-0.022***
	extraversion	0.154***	0.143***	0.021***
User Contexts	*location*	0.215***		
	purchase type		0.130***	
	intended use			0.073***

*** : $\boldsymbol{p < 10^{-9}}$; ** : $\boldsymbol{p < 10^{-5}}$; * : $\boldsymbol{p < 10^{-2}}$

Table 5. Selected Coefficients of Easy-to-interpret Features for the Different Domains

Ad RQ2: To analyze how much each feature perspective contributes to the explanatory power, we evaluated partial R-squared values [82]. That is, we determined how much additional explanatory power is gained by adding a single feature perspective (i.e., by comparing to a model consisting of only the other three perspectives). To directly compare the results to the explanatory power (e.g., Nagelkerke R-squared of 69.8% for the restaurant domain), we assessed the contribution of the each feature perspective by scaling the partial Nagelkerke R-squared values to this benchmark (e.g., cf. [83]).

Dataset	Feature Perspective			
	Item Characteristics	Item Aspects	User Characteristics	User Contexts
Restaurants (69.8% in sum)	1.7%	49.0%	9.8%	9.3%
Movies (64.9% in sum)	5.1%	39.7%	16.0%	4.1%
Laptops (65.0% in sum)	8.9%	44.7%	9.0%	2.4%

Table 6. Contribution of Each Individual Feature Perspective to the Explanatory Power

The results of this analysis are presented in Table 6. When considering individual feature perspectives, item aspects contribute the most to the explanatory power across all domains in our evaluation, followed by user characteristics. For restaurant reviews, user contexts contribute more than item characteristics, whereas for laptop reviews the contribution of item characteristics is higher in comparison.

5 Discussion

In this section, we discuss the above presented results for each research question.

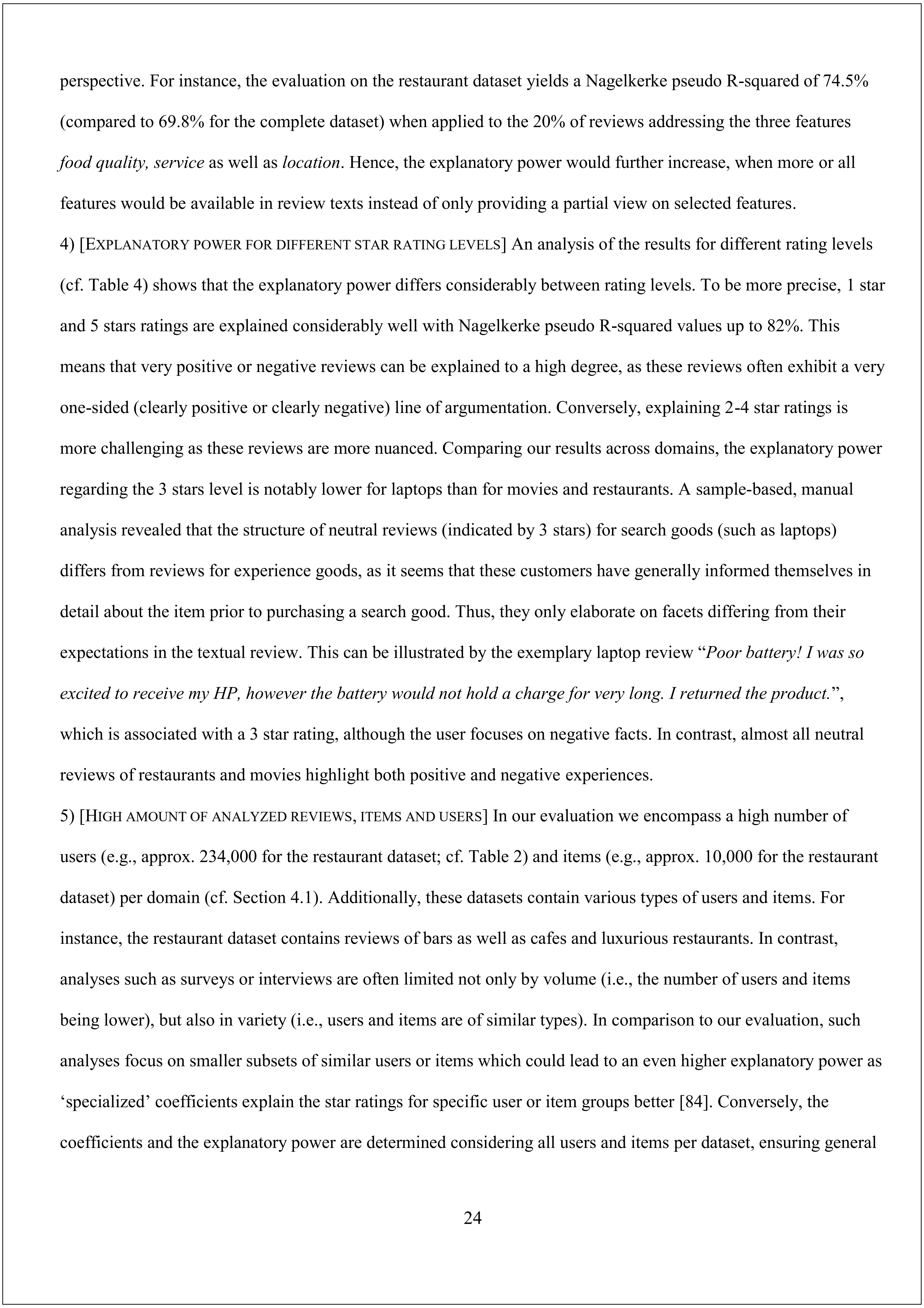
Ad RQ1: There are several reasons indicating that the unified model explains the star ratings of online consumer reviews well across various domains with each domain containing different types of products or services:
1) [HIGHER EXPLANATORY POWER IN RELATION TO SIMILAR WORKS] The explanatory power of the unified model is higher compared to other explanatory models. For instance, the authors of [7], which use the same statistical model (i.e., GOPM), achieve a Nagelkerke pseudo R-squared value of 44% in the restaurant domain with their explanatory model based only on item aspects. Within the same domain, the Nagelkerke pseudo R-squared value of the analysis at hand reaches nearly 70%. Further, the authors of [12] analyze the explanatory power based on datasets restricted to 1 star and 5 star ratings, which contain Amazon reviews for different product categories (e.g., grocery and gourmet food). Using the McFadden pseudo R-squared, Jabr et al. [12] achieved a maximum value of 80% with an average of 64%, whereas our evaluation yields the maximum McFadden pseudo R-squared value of 88% with an average of 86% across our three datasets also evaluated only on 1 star and 5 star ratings. This is in line with our more detailed analysis regarding different rating levels (cf. Table 4), which supports the expectation that star ratings are best explained for 1 star and 5 star ratings as the associated review texts contain words (e.g., sentiments) that clearly indicate an extreme star rating. The reason for the higher explanatory power is that it comprises features of different and additional object- and person-centered feature perspectives of the MPAA model.
2) [EASY-TO-INTERPRET FEATURES EXTRACTED FROM TEXT BY STATE-OF-THE-ART TECHNIQUES] Most of the existing works extract the features from the review texts. The works [12,44,45,47,49,53] use unsupervised topic modeling approaches for generating abstract representations of reviews resulting in aspects that have to be interpreted manually in a time-consuming manner. The works [4,20,22,48,50,51,55] utilize lexicon-based extraction techniques, which achieve much lower validity for the feature extraction compared to the state-of-the-art feature extraction techniques such as BERT ([19]). Different to all of these works, the works [29,46,52,54] do not analyze the vital information contained in the textual parts of online consumer reviews for feature extraction, but use

features that have been queried from the user when making a review. Answering such queries is an additional effort for users. Further, the applicability of analyses regarding such features is limited to specific domains (e.g., airline traveling). In contrast, we used easy-to-interpret features extracted from text by a supervised state-of-the-art deep learning model (cf. Section 4.1). These features ensure that the analyzed feature perspectives are of high validity, directly comprehensible and allow a deeper analysis and meaningful explanations of star ratings, even for different domains. In the following, we analyze and discuss coefficients from the GOPM (Table 5) to illustrate both *cross-domain* and *domain-specific* insights based on these easy-to-interpret features.

2.1) [CROSS-DOMAIN INSIGHTS] Cross-domain insights can be derived from all feature perspectives. For instance, the perspective user characteristics, which has the same features across all domains, enables cross-domain analyses. Here, the results of our evaluation on different domains substantially extends existing insights [55]. First, users with high *agreeableness* tend to give higher star ratings, represented by positive coefficients $\overline{\beta_{aggr.}}$ across all three domains. This indicates that agreeable users behave friendly and generously [35]. In contrast, users with high *neuroticism* might be oversensitive and easily aggravated by items [32], which reasons its negative impact, represented by negative coefficients $\overline{\beta_{neuro.}}$ across all three domains. These observations extend and generalize the findings of [55], which analyzed star ratings for a single domain of experience goods (i.e., hotels). However, the observation of [55] stating that *openness* has a positive effect on star ratings does not hold true in general. Our results show that openness indeed typically has a positive effect on star ratings for experience goods (e.g., restaurants or movies), but has a negative impact for the search good laptops as represented by the coefficients $\overline{\beta_{openn.}}$ in Table 5. As users with high scores on openness like novelty and are enterprising [85], they seek new experiences which can easily be found by testing new foods or movies. Conversely, searching laptops often involves comparing technical details in specifications and data sheets, which is usually not an inspiring experience thus resulting in negative impact. Interestingly, *conscientiousness* has the opposite effect. This might be due to the fact, that conscientious users tend to be well prepared and informed when purchasing an item, which is easier for search goods (e.g., laptops). Finally, *extraversion* consistently has a positive effect, which is plausible, since extraversion also measures a person's tendency to express positive emotions [35].

2.2) [DOMAIN-SPECIFIC INSIGHTS] Domain-specific insights can be derived from all four feature perspectives. For example, we found that the users' current *location* and proximities to restaurants significantly influence their star ratings, which is represented by $\overline{\beta_{location}} = 0.215$ for user contexts in the restaurant domain. Presumably, users might favor conveniently located restaurants to avoid the time and organization effort to travel to and from the restaurant. Moreover, our results show that the item characteristic *brand* considerably influences a user's star ratings in the laptop domain (e.g., $\overline{\beta_{brand}} = 0.160$). This can be reasoned by the relatively high brand loyalty associated to electronic devices like laptops [86]. When shopping for a laptop, users might use brands to infer the performance or quality of a product. In particular, this indicates the high potential of the proposed explanatory analysis enabling differentiated insights for varying types of services and products (e.g., popular products vs. niche products). For instance, we found in a (first) product-differentiated analysis that the item characteristic *brand* is important for laptops of different vendors showing robust results. In particular, the importance is even (slightly) higher for vendors like *apple*. Further, the item aspects *food quality* ($\overline{\beta_{food\ qual.}} = 0.583$) and *service* ($\overline{\beta_{service}} = 0.266$), which are experienced at a restaurant, are even of higher importance. That is, if a user does not enjoy the food and the service in a restaurant, she or he will typically assign a lower star rating and vice versa. In contrast, the price range of a restaurant is often known or at least anticipated prior to the visit, which may lead to the comparably lower importance of the item aspect *price* ($\overline{\beta_{price}} = 0.055$). Moreover, by using the GOPM (cf. Section 4.2), we are able to inspect four coefficients (cf. Equation (2)) for each feature. For instance, the coefficients regarding the aspect *food quality* in the restaurant domain are given by $\beta^1_{food\ qual.} = 0.373$, $\beta^2_{food\ qual.} = 0.633$, $\beta^3_{food\ qual.} = 0.761$ and $\beta^4_{food\ qual.} = 0.564$. This indicates that the item aspect *food quality* is comparably more important for users to distinguish 3 from 4 stars ratings ($\beta^3_{food\ qual.} = 0.761$) than 1 star from 2 stars ratings ($\beta^1_{food\ qual.} = 0.373$). These findings further emphasize the high sensitivity of the GOPM for star rating explanations.

3) [PARTIAL VIEW PROVIDED BY ONLINE CONSUMER REVIEWS] Users typically do not address all features and all feature perspectives in each single review. As the analysis of unstructured review texts can be seen as an instrument of open-ended surveys, users are not forced to assess each feature (e.g., in contrast to structured closed-ended surveys, cf. [87]). For instance, 80% of reviews in the restaurant dataset lack either a sentiment for *food quality, service* or *location*, which constitute frequent item aspects and user contexts. That is, users do not necessarily describe all aspects and contexts being potentially relevant for the assigned star rating. Additionally, review texts might even be bound by length restrictions. In our evaluation, such unknown sentiments of aspects and contexts have to be implicitly assumed as neutral sentiments, which puts the achieved explanatory power further into



perspective. For instance, the evaluation on the restaurant dataset yields a Nagelkerke pseudo R-squared of 74.5% (compared to 69.8% for the complete dataset) when applied to the 20% of reviews addressing the three features *food quality, service* as well as *location*. Hence, the explanatory power would further increase, when more or all features would be available in review texts instead of only providing a partial view on selected features.

4) [EXPLANATORY POWER FOR DIFFERENT STAR RATING LEVELS] An analysis of the results for different rating levels (cf. Table 4) shows that the explanatory power differs considerably between rating levels. To be more precise, 1 star and 5 stars ratings are explained considerably well with Nagelkerke pseudo R-squared values up to 82%. This means that very positive or negative reviews can be explained to a high degree, as these reviews often exhibit a very one-sided (clearly positive or clearly negative) line of argumentation. Conversely, explaining 2-4 star ratings is more challenging as these reviews are more nuanced. Comparing our results across domains, the explanatory power regarding the 3 stars level is notably lower for laptops than for movies and restaurants. A sample-based, manual analysis revealed that the structure of neutral reviews (indicated by 3 stars) for search goods (such as laptops) differs from reviews for experience goods, as it seems that these customers have generally informed themselves in detail about the item prior to purchasing a search good. Thus, they only elaborate on facets differing from their expectations in the textual review. This can be illustrated by the exemplary laptop review "*Poor battery! I was so excited to receive my HP, however the battery would not hold a charge for very long. I returned the product.*", which is associated with a 3 star rating, although the user focuses on negative facts. In contrast, almost all neutral reviews of restaurants and movies highlight both positive and negative experiences.

5) [HIGH AMOUNT OF ANALYZED REVIEWS, ITEMS AND USERS] In our evaluation we encompass a high number of users (e.g., approx. 234,000 for the restaurant dataset; cf. Table 2) and items (e.g., approx. 10,000 for the restaurant dataset) per domain (cf. Section 4.1). Additionally, these datasets contain various types of users and items. For instance, the restaurant dataset contains reviews of bars as well as cafes and luxurious restaurants. In contrast, analyses such as surveys or interviews are often limited not only by volume (i.e., the number of users and items being lower), but also in variety (i.e., users and items are of similar types). In comparison to our evaluation, such analyses focus on smaller subsets of similar users or items which could lead to an even higher explanatory power as 'specialized' coefficients explain the star ratings for specific user or item groups better [84]. Conversely, the coefficients and the explanatory power are determined considering all users and items per dataset, ensuring general

insights and high validity due to the high number and diversity of reviews, items and users.

Ad RQ2: The results regarding RQ2 show that each feature perspective does indeed contribute to the explanatory power demonstrating the importance of a broader and differentiated view when explaining star ratings. This contributes to our opening question (cf. Section 1), viz., why users rated an item the way they did. Moreover, this finding shows that our research poses a substantial progress compared to existing approaches which address (certain features of) at most two feature perspectives. In particular, we emphasize that the contributions of both object-centered feature perspectives as well as person-centered feature perspectives are remarkable and thus both types of feature perspectives are important to understand star ratings. Consequently, these findings further support the MPAA model [24].

To further substantiate our findings, we tested whether the contribution of each feature perspective is statistically significant. To this end, we compared the unified model (containing all four feature perspectives) with restricted (nested) models (containing only the three feature perspectives) by means of the Bayesian information criterion (BIC) and the likelihood ratio test (LRT) [88,89]. The BIC is particularly suited for the large datasets used in our analyses as it takes into account both the number of independent variables as well as the sample size. Here, the unified model yielded a decrease in the BIC value of at least 978 compared to the restricted model for each feature perspective and all domains, whereby a BIC decrease of 10 indicates ‘strong evidence’ that the model with lower BIC value is preferred [90]. In that line, the comparisons with the LRT yielded that the unified model is a better model compared to all four restricted models with $p < 10^{-9}$ on all considered domains. Thus, each feature perspective contributes significantly to explaining star ratings, but by analyzing partial Nagelkerke R-squared values in RQ2, we could additionally obtain valuable quantitative results with respect to these feature perspectives as discussed in the following.

The contributions of each feature perspective are comparable across all three domains. Item aspects contribute the most to the explanatory power (e.g., 49.0% for the restaurant domain) since this perspective captures the user’s direct perception and the actual experience with an item. For perspectives regarding user and item characteristics, user characteristics contribute more to the explanation of star ratings than item characteristics for the experience goods restaurants and movies (e.g., 9.8% vs. 1.7% for the restaurant domain). For the search good laptops, item characteristics and user characteristics have an almost equally high contribution in the dataset (8.9%

vs. 9.0%). These results can be directly attributed to search goods being more clearly defined by their characteristics (e.g., the capacity of the working memory in the domain of laptops), while experience goods can only be actually assessed after experiencing the item [59]. The particularly low contribution of item characteristics in the restaurant domain might also be due to the fact that users mainly attend the type of restaurants they typically enjoy. For instance, users who dislike Italian food usually will not visit an Italian restaurant. To be more precise, while the item characteristic 'Italian' would be relevant for those users when choosing a restaurant, it rarely comes into effect when writing a review. This indicates that not all features being relevant for purchase decisions are necessarily relevant when explaining star ratings. Additionally, the results show that the feature perspective user contexts has less contribution regarding explanatory power in the domain of laptops compared to movies and restaurants. Since laptops constitute a search good, purchase decisions are arguably more rational and planned in advance. Consequently, it is reasonable that (situational) user contexts do not strongly influence star ratings in this domain.

6 Implications for Research and Practice

Overall, the results and discussions of our evaluation show that (a) the proposed feature perspectives are able to explain star ratings considerably well opening the way for a comprehensive understanding of star ratings and (b) each individual feature perspective contributes to the explanatory power. These findings have implications for both scientific research and practical applications, which are outlined in the following.

6.1 Implications for Research

From a theoretical point of view, our research has the following implications.

1) [SYSTEMATIC AND COMPREHENSIBLE EXPLANATIONS OF STAR RATINGS] Based on the MPAA model, we systematically derived different feature perspectives each consisting of easy-to-interpret features, which were extracted from review texts. Studying such easy-to-interpret features of different perspectives is a key driver to comprehensibly explaining star ratings which is affirmed by the high overall explanatory power and the substantial contribution of each feature perspective. That is, each feature perspective significantly improves the explanatory power compared to a model restricted to the other three considered feature perspectives. Hence, this work poses a

first important step to enable a theoretical foundation – starting from the MPAA model – further research can enhance and use for systematic and comprehensible explanations of star ratings.

2) [ANALYSIS OF DIFFERENT STEPS OF THE RATING SCALE] Moreover, our approach is capable of determining the importance of the features regarding different steps of the rating scale. While prior research has mainly focused on the explanation of 1 star and 5 star ratings [12], our analysis instead also reveals to what degree the features influence star ratings for each rating level within the rating scale. Recent research has initially acknowledged this consideration by separately analyzing reviews with specific star ratings [62]. By aligning to our approach, researchers are now able to examine which features are important for explaining fine-grained differentiations between rating levels (e.g., 4 star and 5 star ratings) in an overall explanatory model.

3) [INCORPORATING CROSS-DOMAIN AND DOMAIN-SPECIFIC INSIGHTS] Our evaluation results show that the different feature perspectives enable to derive cross-domain and domain-specific insights regarding star ratings of online consumer reviews. This can be vital for researchers focusing on cross-domain marketing or domain-independent analysis of user preferences. In particular, our results suggest, that user characteristics such as agreeableness have a positive impact on star ratings in all three analyzed domains while the impact of a user's openness on star ratings varies depending on the particular domain. Hence, these researchers might benefit from incorporating user characteristics in their analyses to explore such relationships and to better understand users. Similarly, research focusing on specific application domains (e.g., hospitality and tourism management) might benefit from domain-specific insights.

4) [UTILIZING MULTIPLE FEATURE PERSPECTIVES FOR OTHER RESEARCH STRANDS] Further, the promising results when utilizing multiple feature perspectives could inspire other research strands analyzing user assessments. For instance, research in the field of recommender systems mainly use individual feature perspectives to generate personalized item recommendations. That is, content-based recommender systems are largely based on item characteristics [28] while context-aware recommender systems focus on context features [78]. Although predictive and explanatory analysis have different objectives [17], we are confident that our findings encourage researchers to incorporate multiple feature perspectives in recommender systems and other research strands.

5) [USING DIFFERENT FEATURE PERSPECTIVES TO EXPLAIN RECOMMENDATIONS] In addition, works that aim to explain recommendations to the users have gained higher attention in the past years (e.g., [91]). Nevertheless, as popular

and widely applied recommender systems infer recommendations based on latent factors (e.g., matrix factorization), it is not straightforward to present meaningful and comprehensible explanations to a user for provided recommendations. Thereby, existing literature tries to explain these recommendations by inferring similarities based on latent factors or by examining item statistics (e.g., movies being popular in a certain region) [92]. With this in mind, our findings could inspire researchers in such fields by analyzing review texts and leveraging easy-to-interpret features and feature perspectives to explain recommended items in a comprehensible manner.

6.2 Implications for Practice

Analyzing star ratings based on different feature perspectives enables consumers, web portals and businesses to leverage the versatile information from online consumer reviews. This allows well-founded and advantageous actions in practical business applications.

1) [GENERATING MEANINGFUL ITEM SUMMARIZATIONS IN WEB PORTALS] By means of our explanatory analysis comprising features of multiple feature perspectives, web portal providers can use the unified model to detect meaningful features and feature perspectives which are important for explaining star ratings (i.e., features with very high or low coefficients). In this line, these meaningful features can be used for summarizing review texts and are particularly relevant for users when forming attitudes about items (e.g., cf. [9]). Consumers might benefit from both individual review summaries as well as structured summaries encompassing many user reviews of an item. Similarly, and in line with [93], the analysis of different feature perspectives can be used to identify and highlight "informative or representative" review texts, based on the detected meaningful features, which are able to explain star ratings especially well. In that way, users might be more satisfied with the (summarized or selected) information provided by a web portal.

2) [REDUCING USER QUERIES FOR MULTI-CRITERIA RATING SYSTEMS] Furthermore, the derived explanations are valuable for web portal providers that maintain multi-criteria rating systems. Such systems are based on explicit user queries where users are asked to rate specific features after experiencing an item. While a plethora of queries with different features would be possible, answering such queries is an additional effort for users. In order to not discourage users, it is thus only feasible to ask (very) few queries. As the unified model comprises various features of versatile perspectives, the derived explanations enable to identify the most important features (i.e., features with

high or low regression coefficients) and perspectives for the users' star ratings. Therefore, web portal providers could focus on important features (e.g., regarding a domain or a group of items) and thereby improve the return of each user query.

3) [IMPROVING ITEMS THROUGH IDENTIFIED USER CRITICISM] In addition, applying the unified model enables businesses to identify features with (very) high and low regression coefficients. This includes features which are often subject to criticism and have a high negative impact on star ratings as well as features which exhibit a positive impact on star ratings. With regard to the increasing volume of user generated content of EWOM and in particular online consumer reviews, aligning to our approach enables businesses to assess these critical features in an automated manner, based on a large review basis and with the possibility to distinguish fine-grained differentiations between rating levels. By analyzing the online consumer reviews regarding these critical features, precise and substantial criticism (e.g., which is expressed in several reviews) can be identified. Consequently and in line with works in the field of EWOM such as [11], businesses are then able to systematically and selectively address the identified criticism, evolve their items, create ideas for new items or new business models, and prospectively improve the user experience and thus users' item assessments (e.g., star ratings).

4) [USER CHARACTERISTICS ALLOW FOR BETTER CONSUMER UNDERSTANDING] Finally, our results indicate that the feature perspective user characteristics, which has been hardly considered in prior research explaining star ratings, is a key factor in online item assessments and therefore, exhibit high potential for practitioners. Our findings yield strong relations between users' star ratings and the users' personality traits, substantiating and significantly extending the basic findings of [55] on other domains. Therefore, web portal providers, which focus on recommending relevant items to users, as well as businesses providing services or products could benefit from more accurate and comprehensive analyses of consumer behavior by considering the feature perspective user characteristics. For instance, marketing campaigns could target consumers with user characteristics positively influencing star ratings. This could increase the average star rating and thus the revenue of a business [15].

7 Conclusion, Limitations and Future Work

Many web portals such as Amazon or TripAdvisor provide user assessments in form of star ratings and textual reviews. Both research and practice have acknowledged the importance of explaining star ratings. Based upon the existing body of knowledge on this topic, our work is the first to leverage the four feature perspectives item characteristics, item aspects, user characteristics and user contexts in a unified model to explain star ratings in online consumer reviews. We evaluated this model on three large real-world review datasets from the domains of restaurants, laptops and movies using the GOPM. Our results show that these feature perspectives are indeed able to explain star ratings considerably well. Moreover, the evaluation shows that the feature perspectives *item aspects* and *user characteristics* have the highest importance in terms of explanatory power on the star ratings.

Nevertheless, the work at hand has some limitations which could be a starting point for further research. Firstly, the evaluation was conducted on three large datasets from different domains. However, evaluating the unified model on other domains could substantiate and broaden our findings. Secondly, we operationalized three feature perspectives using a state-of-the-art deep learning language model for analyzing review texts. Nevertheless, other ways of operationalizing feature perspectives (e.g., social identity as user characteristics) could be used to analyze and extend the findings. Thirdly, analyzing interdependent moderator effects between the considered feature perspectives (e.g., do user characteristics influence the effect of item characteristics on star ratings) would be interesting and could complement our findings. Lastly, further analyses of different item or user groups might enable additional insights on the explanation of star ratings and further strengthen the findings of this research.

References

[1] eMarketers, Digital Buyers Worldwide, 2016-2021 (Billions, % Change and % of Internet Users), 2018. https://www.emarketer.com/Chart/Digital-Buyers-Worldwide-2016-2021-billions-change-of-internet-users/215140 (accessed 31 July 2021).

[2] S.M. Mudambi, D. Schuff, What Makes a Helpful Online Review? A Study of Customer Reviews on Amazon.com, MIS Quarterly 34 (2010) 185. https://doi.org/10.2307/20721420.

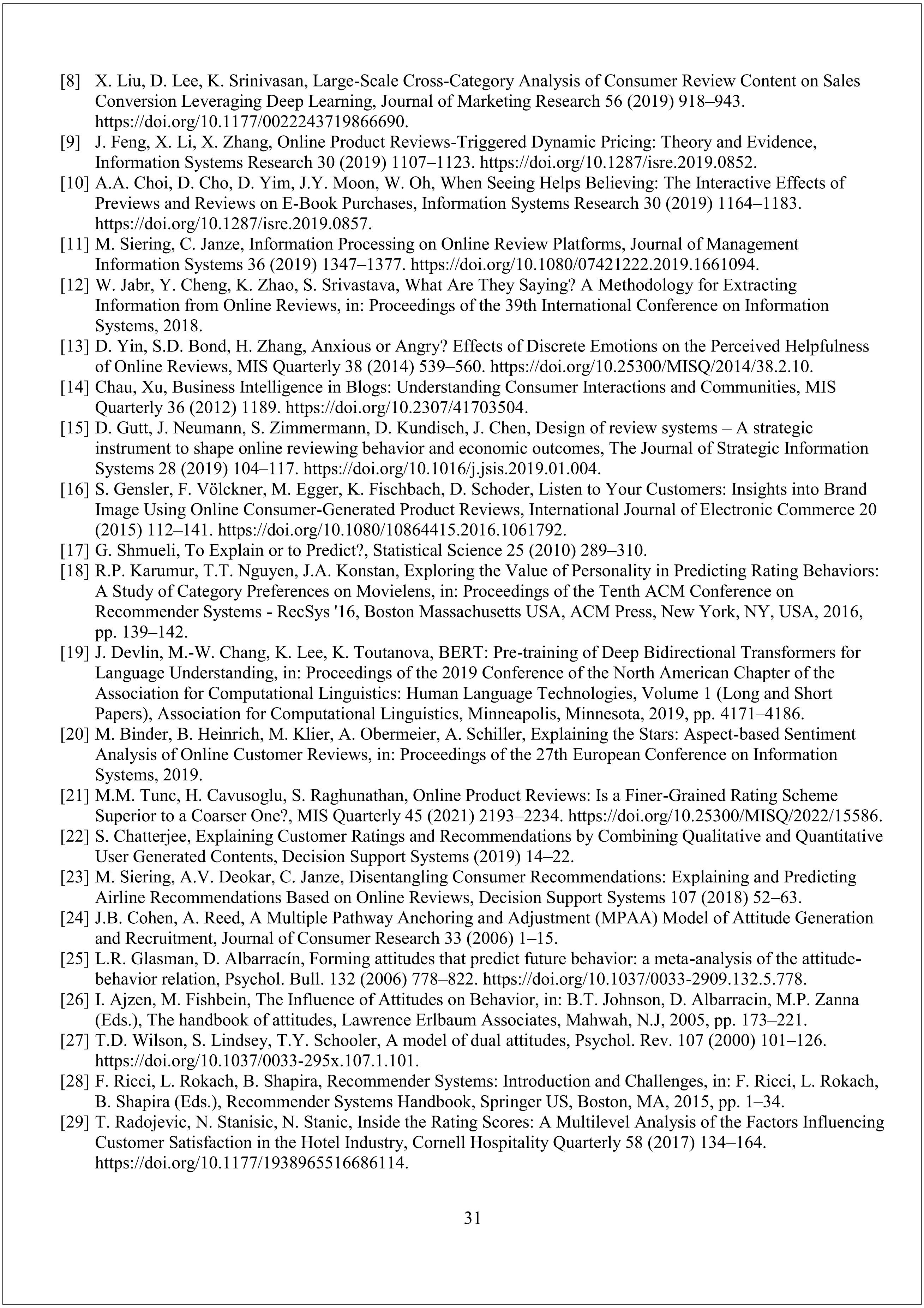
[3] X. Li, Revealing or Non-Revealing: The Impact of Review Disclosure Policy on Firm Profitability, MIS Quarterly 41 (2017) 1335–1345. https://doi.org/10.25300/MISQ/2017/41.4.14.

[4] Z. Xiang, Z. Schwartz, J.H. Gerdes, M. Uysal, What Can Big Data and Text Analytics Tell Us about Hotel Guest Experience and Satisfaction?, International Journal of Hospitality Management 44 (2015) 120–130. https://doi.org/10.1016/j.ijhm.2014.10.013.

[5] W. Jabr, B. Liu, D. Yin, H. Zhang, MIS Quarterly Research Curation on Online Word-of-Mouth Research Curation Team (2020).

[6] B. von Helversen, K. Abramczuk, W. Kopeć, R. Nielek, Influence of consumer reviews on online purchasing decisions in older and younger adults, Decision Support Systems 113 (2018) 1–10. https://doi.org/10.1016/j.dss.2018.05.006.

[7] C. Yi, Z. Jiang, X. Li, X. Lu, Leveraging User-Generated Content for Product Promotion: The Effects of Firm-Highlighted Reviews, Information Systems Research 30 (2019) 711–725. https://doi.org/10.1287/isre.2018.0807.



[8] X. Liu, D. Lee, K. Srinivasan, Large-Scale Cross-Category Analysis of Consumer Review Content on Sales Conversion Leveraging Deep Learning, Journal of Marketing Research 56 (2019) 918–943. https://doi.org/10.1177/0022243719866690.

[9] J. Feng, X. Li, X. Zhang, Online Product Reviews-Triggered Dynamic Pricing: Theory and Evidence, Information Systems Research 30 (2019) 1107–1123. https://doi.org/10.1287/isre.2019.0852.

[10] A.A. Choi, D. Cho, D. Yim, J.Y. Moon, W. Oh, When Seeing Helps Believing: The Interactive Effects of Previews and Reviews on E-Book Purchases, Information Systems Research 30 (2019) 1164–1183. https://doi.org/10.1287/isre.2019.0857.

[11] M. Siering, C. Janze, Information Processing on Online Review Platforms, Journal of Management Information Systems 36 (2019) 1347–1377. https://doi.org/10.1080/07421222.2019.1661094.

[12] W. Jabr, Y. Cheng, K. Zhao, S. Srivastava, What Are They Saying? A Methodology for Extracting Information from Online Reviews, in: Proceedings of the 39th International Conference on Information Systems, 2018.

[13] D. Yin, S.D. Bond, H. Zhang, Anxious or Angry? Effects of Discrete Emotions on the Perceived Helpfulness of Online Reviews, MIS Quarterly 38 (2014) 539–560. https://doi.org/10.25300/MISQ/2014/38.2.10.

[14] Chau, Xu, Business Intelligence in Blogs: Understanding Consumer Interactions and Communities, MIS Quarterly 36 (2012) 1189. https://doi.org/10.2307/41703504.

[15] D. Gutt, J. Neumann, S. Zimmermann, D. Kundisch, J. Chen, Design of review systems – A strategic instrument to shape online reviewing behavior and economic outcomes, The Journal of Strategic Information Systems 28 (2019) 104–117. https://doi.org/10.1016/j.jsis.2019.01.004.

[16] S. Gensler, F. Völckner, M. Egger, K. Fischbach, D. Schoder, Listen to Your Customers: Insights into Brand Image Using Online Consumer-Generated Product Reviews, International Journal of Electronic Commerce 20 (2015) 112–141. https://doi.org/10.1080/10864415.2016.1061792.

[17] G. Shmueli, To Explain or to Predict?, Statistical Science 25 (2010) 289–310.

[18] R.P. Karumur, T.T. Nguyen, J.A. Konstan, Exploring the Value of Personality in Predicting Rating Behaviors: A Study of Category Preferences on Movielens, in: Proceedings of the Tenth ACM Conference on Recommender Systems - RecSys '16, Boston Massachusetts USA, ACM Press, New York, NY, USA, 2016, pp. 139–142.

[19] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186.

[20] M. Binder, B. Heinrich, M. Klier, A. Obermeier, A. Schiller, Explaining the Stars: Aspect-based Sentiment Analysis of Online Customer Reviews, in: Proceedings of the 27th European Conference on Information Systems, 2019.

[21] M.M. Tunc, H. Cavusoglu, S. Raghunathan, Online Product Reviews: Is a Finer-Grained Rating Scheme Superior to a Coarser One?, MIS Quarterly 45 (2021) 2193–2234. https://doi.org/10.25300/MISQ/2022/15586.

[22] S. Chatterjee, Explaining Customer Ratings and Recommendations by Combining Qualitative and Quantitative User Generated Contents, Decision Support Systems (2019) 14–22.

[23] M. Siering, A.V. Deokar, C. Janze, Disentangling Consumer Recommendations: Explaining and Predicting Airline Recommendations Based on Online Reviews, Decision Support Systems 107 (2018) 52–63.

[24] J.B. Cohen, A. Reed, A Multiple Pathway Anchoring and Adjustment (MPAA) Model of Attitude Generation and Recruitment, Journal of Consumer Research 33 (2006) 1–15.

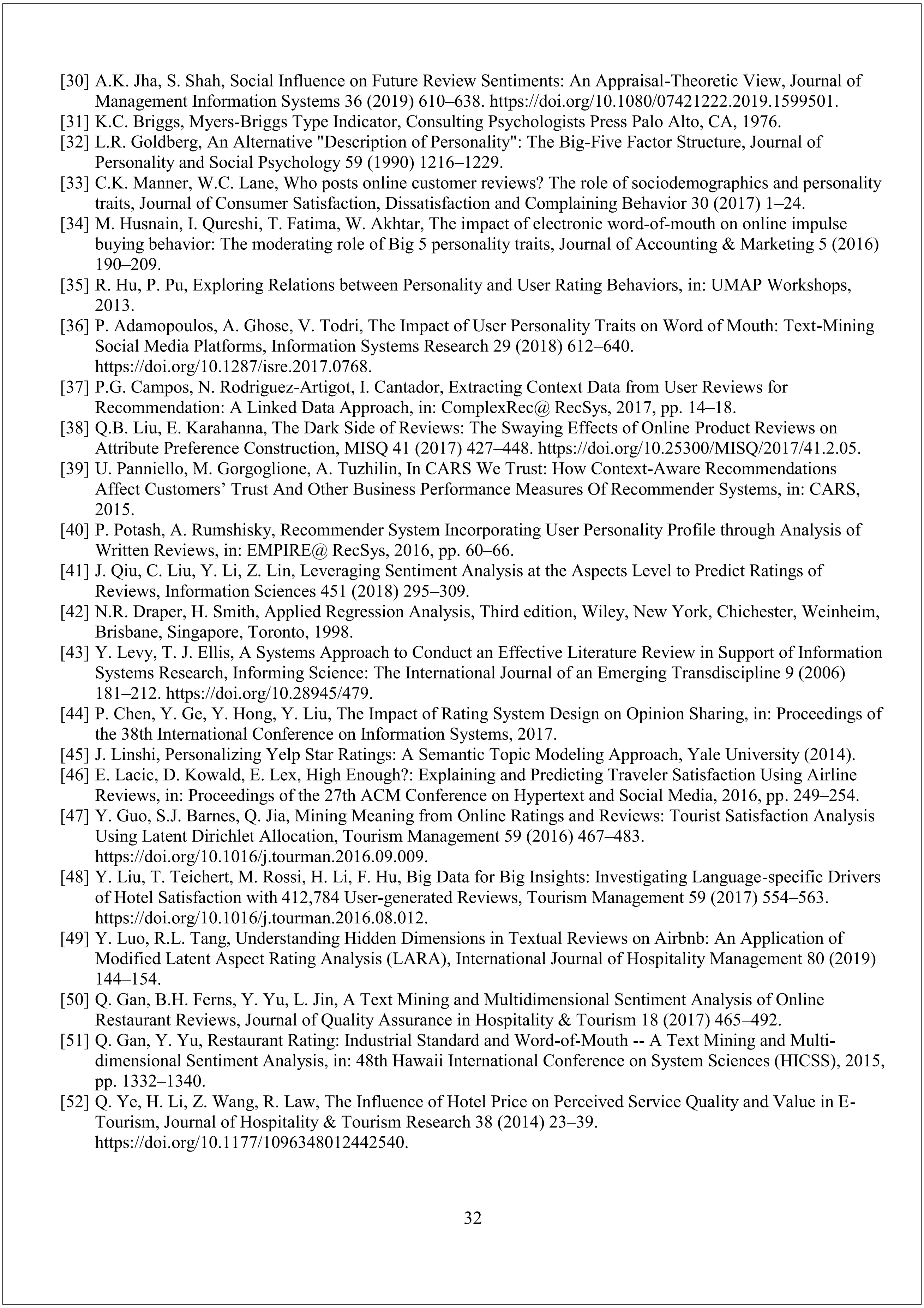
[25] L.R. Glasman, D. Albarracín, Forming attitudes that predict future behavior: a meta-analysis of the attitude-behavior relation, Psychol. Bull. 132 (2006) 778–822. https://doi.org/10.1037/0033-2909.132.5.778.

[26] I. Ajzen, M. Fishbein, The Influence of Attitudes on Behavior, in: B.T. Johnson, D. Albarracin, M.P. Zanna (Eds.), The handbook of attitudes, Lawrence Erlbaum Associates, Mahwah, N.J, 2005, pp. 173–221.

[27] T.D. Wilson, S. Lindsey, T.Y. Schooler, A model of dual attitudes, Psychol. Rev. 107 (2000) 101–126. https://doi.org/10.1037/0033-295x.107.1.101.

[28] F. Ricci, L. Rokach, B. Shapira, Recommender Systems: Introduction and Challenges, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, Springer US, Boston, MA, 2015, pp. 1–34.

[29] T. Radojevic, N. Stanisic, N. Stanic, Inside the Rating Scores: A Multilevel Analysis of the Factors Influencing Customer Satisfaction in the Hotel Industry, Cornell Hospitality Quarterly 58 (2017) 134–164. https://doi.org/10.1177/1938965516686114.



[30] A.K. Jha, S. Shah, Social Influence on Future Review Sentiments: An Appraisal-Theoretic View, Journal of Management Information Systems 36 (2019) 610–638. https://doi.org/10.1080/07421222.2019.1599501.

[31] K.C. Briggs, Myers-Briggs Type Indicator, Consulting Psychologists Press Palo Alto, CA, 1976.

[32] L.R. Goldberg, An Alternative "Description of Personality": The Big-Five Factor Structure, Journal of Personality and Social Psychology 59 (1990) 1216–1229.

[33] C.K. Manner, W.C. Lane, Who posts online customer reviews? The role of sociodemographics and personality traits, Journal of Consumer Satisfaction, Dissatisfaction and Complaining Behavior 30 (2017) 1–24.

[34] M. Husnain, I. Qureshi, T. Fatima, W. Akhtar, The impact of electronic word-of-mouth on online impulse buying behavior: The moderating role of Big 5 personality traits, Journal of Accounting & Marketing 5 (2016) 190–209.

[35] R. Hu, P. Pu, Exploring Relations between Personality and User Rating Behaviors, in: UMAP Workshops, 2013.

[36] P. Adamopoulos, A. Ghose, V. Todri, The Impact of User Personality Traits on Word of Mouth: Text-Mining Social Media Platforms, Information Systems Research 29 (2018) 612–640. https://doi.org/10.1287/isre.2017.0768.

[37] P.G. Campos, N. Rodriguez-Artigot, I. Cantador, Extracting Context Data from User Reviews for Recommendation: A Linked Data Approach, in: ComplexRec@ RecSys, 2017, pp. 14–18.

[38] Q.B. Liu, E. Karahanna, The Dark Side of Reviews: The Swaying Effects of Online Product Reviews on Attribute Preference Construction, MISQ 41 (2017) 427–448. https://doi.org/10.25300/MISQ/2017/41.2.05.

[39] U. Panniello, M. Gorgoglione, A. Tuzhilin, In CARS We Trust: How Context-Aware Recommendations Affect Customers' Trust And Other Business Performance Measures Of Recommender Systems, in: CARS, 2015.

[40] P. Potash, A. Rumshisky, Recommender System Incorporating User Personality Profile through Analysis of Written Reviews, in: EMPIRE@ RecSys, 2016, pp. 60–66.

[41] J. Qiu, C. Liu, Y. Li, Z. Lin, Leveraging Sentiment Analysis at the Aspects Level to Predict Ratings of Reviews, Information Sciences 451 (2018) 295–309.

[42] N.R. Draper, H. Smith, Applied Regression Analysis, Third edition, Wiley, New York, Chichester, Weinheim, Brisbane, Singapore, Toronto, 1998.

[43] Y. Levy, T. J. Ellis, A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research, Informing Science: The International Journal of an Emerging Transdiscipline 9 (2006) 181–212. https://doi.org/10.28945/479.

[44] P. Chen, Y. Ge, Y. Hong, Y. Liu, The Impact of Rating System Design on Opinion Sharing, in: Proceedings of the 38th International Conference on Information Systems, 2017.

[45] J. Linshi, Personalizing Yelp Star Ratings: A Semantic Topic Modeling Approach, Yale University (2014).

[46] E. Lacic, D. Kowald, E. Lex, High Enough?: Explaining and Predicting Traveler Satisfaction Using Airline Reviews, in: Proceedings of the 27th ACM Conference on Hypertext and Social Media, 2016, pp. 249–254.

[47] Y. Guo, S.J. Barnes, Q. Jia, Mining Meaning from Online Ratings and Reviews: Tourist Satisfaction Analysis Using Latent Dirichlet Allocation, Tourism Management 59 (2016) 467–483. https://doi.org/10.1016/j.tourman.2016.09.009.

[48] Y. Liu, T. Teichert, M. Rossi, H. Li, F. Hu, Big Data for Big Insights: Investigating Language-specific Drivers of Hotel Satisfaction with 412,784 User-generated Reviews, Tourism Management 59 (2017) 554–563. https://doi.org/10.1016/j.tourman.2016.08.012.

[49] Y. Luo, R.L. Tang, Understanding Hidden Dimensions in Textual Reviews on Airbnb: An Application of Modified Latent Aspect Rating Analysis (LARA), International Journal of Hospitality Management 80 (2019) 144–154.

[50] Q. Gan, B.H. Ferns, Y. Yu, L. Jin, A Text Mining and Multidimensional Sentiment Analysis of Online Restaurant Reviews, Journal of Quality Assurance in Hospitality & Tourism 18 (2017) 465–492.

[51] Q. Gan, Y. Yu, Restaurant Rating: Industrial Standard and Word-of-Mouth -- A Text Mining and Multi-dimensional Sentiment Analysis, in: 48th Hawaii International Conference on System Sciences (HICSS), 2015, pp. 1332–1340.

[52] Q. Ye, H. Li, Z. Wang, R. Law, The Influence of Hotel Price on Perceived Service Quality and Value in E-Tourism, Journal of Hospitality & Tourism Research 38 (2014) 23–39. https://doi.org/10.1177/1096348012442540.

[53] X. Xu, Does Traveler Satisfaction Differ in Various Travel Group Compositions? Evidence from Online Reviews, International Journal of Contemporary Hospitality Management 30 (2018) 1663–1685.

[54] S. Chatterjee, P. Mandal, Traveler preferences from online reviews: Role of travel goals, class and culture, Tourism Management 80 (2020) 104108.

[55] M. Han, Examining the effect of reviewer expertise and personality on reviewer satisfaction: An empirical study of TripAdvisor, Computers in Human behavior (2020) 106567.

[56] B.B. McShane, D. Gal, Statistical significance and the dichotomization of evidence, Journal of the American Statistical Association 112 (2017) 885–895.

[57] A. Gandomi, M. Haider, Beyond the hype: Big data concepts, methods, and analytics, International Journal of Information Management 35 (2015) 137–144.

[58] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, SemEval-2014 Task 4: Aspect Based Sentiment Analysis, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Association for Computational Linguistics, Dublin, Ireland, 2014, pp. 27–35.

[59] M. Schmalz, M. Carter, J.H. Lee, It's Not You, It's Me: Identity, Self–Verification, and Amazon Reviews, The DATA BASE for Advances in Information Systems 49 (2018).

[60] J. Ni, J. Li, J. McAuley, Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 188–197.

[61] S. Debortoli, O. Müller, I. Junglas, J. Vom Brocke, Text Mining for Information Systems Researchers: An Annotated Topic Modeling Tutorial, CAIS 39 (2016) 110–135. https://doi.org/10.17705/1CAIS.03907.

[62] D. Keller, M. Kostromitina, Characterizing non-chain restaurants' Yelp star-ratings: Generalizable findings from a representative sample of Yelp reviews, International Journal of Hospitality Management 86 (2020) 102440.

[63] G. Heinze, C. Wallisch, D. Dunkler, Variable selection-a review and recommendations for the practicing statistician, Biometrical Journal 60 (2018) 431–449.

[64] D. Ramage, C.D. Manning, S. Dumais, Partially labeled topic models for interpretable text mining, in: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, NY, 2011.

[65] V. Vallurupalli, I. Bose, Exploring thematic composition of online reviews: A topic modeling approach, Electron Markets 30 (2020) 791–804. https://doi.org/10.1007/s12525-020-00397-5.

[66] I. Tenney, D. Das, E. Pavlick, BERT Rediscovers the Classical NLP Pipeline, Association for Computational Linguistics, 2019.

[67] H. Xu, B. Liu, L. Shu, P. Yu, BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 2324–2335.

[68] S.-T. Park, W. Chu, Pairwise Preference Regression for Cold-Start Recommendation, in: Proceedings of the Third ACM Conference on Recommender Systems, 2009, pp. 21–28.

[69] J.W. Pennebaker, L.A. King, Linguistic styles: Language use as an individual difference, Journal of Personality and Social Psychology 77 (1999) 1296–1312. https://doi.org/10.1037/0022-3514.77.6.1296.

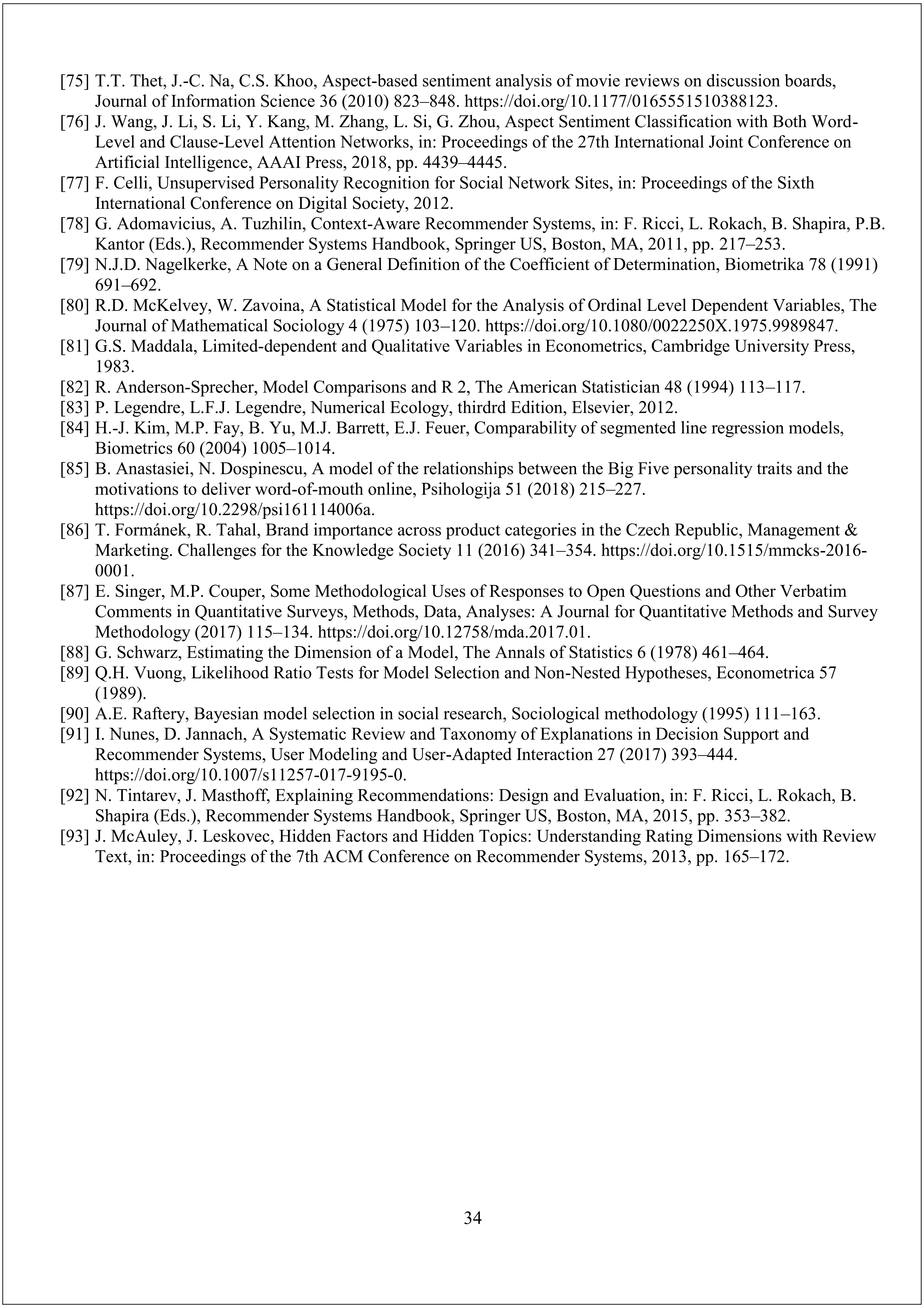
[70] Y. Mehta, N. Majumder, A. Gelbukh, E. Cambria, Recent trends in deep learning based personality detection, Artificial Intelligence Review 53 (2020) 2313–2339.

[71] G. Park, H.A. Schwartz, J.C. Eichstaedt, M.L. Kern, M. Kosinski, D.J. Stillwell, L.H. Ungar, M.E.P. Seligman, Automatic personality assessment through social media language, Journal of Personality and Social Psychology 108 (2015) 934.

[72] R.M. O'brien, A Caution Regarding Rules of Thumb for Variance Inflation Factors, Quality & Quantity 41 (2007) 673–690. https://doi.org/10.1007/s11135-006-9018-6.

[73] L. Yu, J. Huang, G. Zhou, C. Liu, Z.-K. Zhang, TIIREC: A tensor approach for tag-driven item recommendation with sparse user generated content, Information Sciences 411 (2017) 122–135. https://doi.org/10.1016/j.ins.2017.05.025.

[74] M. de Gemmis, P. Lops, C. Musto, F. Narducci, G. Semeraro, Semantics-Aware Content-Based Recommender Systems, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, Springer US, Boston, MA, 2015, pp. 119–159.



[75] T.T. Thet, J.-C. Na, C.S. Khoo, Aspect-based sentiment analysis of movie reviews on discussion boards, Journal of Information Science 36 (2010) 823–848. https://doi.org/10.1177/0165551510388123.

[76] J. Wang, J. Li, S. Li, Y. Kang, M. Zhang, L. Si, G. Zhou, Aspect Sentiment Classification with Both Word-Level and Clause-Level Attention Networks, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, AAAI Press, 2018, pp. 4439–4445.

[77] F. Celli, Unsupervised Personality Recognition for Social Network Sites, in: Proceedings of the Sixth International Conference on Digital Society, 2012.

[78] G. Adomavicius, A. Tuzhilin, Context-Aware Recommender Systems, in: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), Recommender Systems Handbook, Springer US, Boston, MA, 2011, pp. 217–253.

[79] N.J.D. Nagelkerke, A Note on a General Definition of the Coefficient of Determination, Biometrika 78 (1991) 691–692.

[80] R.D. McKelvey, W. Zavoina, A Statistical Model for the Analysis of Ordinal Level Dependent Variables, The Journal of Mathematical Sociology 4 (1975) 103–120. https://doi.org/10.1080/0022250X.1975.9989847.

[81] G.S. Maddala, Limited-dependent and Qualitative Variables in Econometrics, Cambridge University Press, 1983.

[82] R. Anderson-Sprecher, Model Comparisons and R 2, The American Statistician 48 (1994) 113–117.

[83] P. Legendre, L.F.J. Legendre, Numerical Ecology, thirdrd Edition, Elsevier, 2012.

[84] H.-J. Kim, M.P. Fay, B. Yu, M.J. Barrett, E.J. Feuer, Comparability of segmented line regression models, Biometrics 60 (2004) 1005–1014.

[85] B. Anastasiei, N. Dospinescu, A model of the relationships between the Big Five personality traits and the motivations to deliver word-of-mouth online, Psihologija 51 (2018) 215–227. https://doi.org/10.2298/psi161114006a.

[86] T. Formánek, R. Tahal, Brand importance across product categories in the Czech Republic, Management & Marketing. Challenges for the Knowledge Society 11 (2016) 341–354. https://doi.org/10.1515/mmcks-2016-0001.

[87] E. Singer, M.P. Couper, Some Methodological Uses of Responses to Open Questions and Other Verbatim Comments in Quantitative Surveys, Methods, Data, Analyses: A Journal for Quantitative Methods and Survey Methodology (2017) 115–134. https://doi.org/10.12758/mda.2017.01.

[88] G. Schwarz, Estimating the Dimension of a Model, The Annals of Statistics 6 (1978) 461–464.

[89] Q.H. Vuong, Likelihood Ratio Tests for Model Selection and Non-Nested Hypotheses, Econometrica 57 (1989).

[90] A.E. Raftery, Bayesian model selection in social research, Sociological methodology (1995) 111–163.

[91] I. Nunes, D. Jannach, A Systematic Review and Taxonomy of Explanations in Decision Support and Recommender Systems, User Modeling and User-Adapted Interaction 27 (2017) 393–444. https://doi.org/10.1007/s11257-017-9195-0.

[92] N. Tintarev, J. Masthoff, Explaining Recommendations: Design and Evaluation, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, Springer US, Boston, MA, 2015, pp. 353–382.

[93] J. McAuley, J. Leskovec, Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text, in: Proceedings of the 7th ACM Conference on Recommender Systems, 2013, pp. 165–172.

4 Paper 2: GroupFM: Enabling Context-Aware Group Recommendations with Factorization Machines

GroupFM: Enabling Context-Aware Group Recommendations with Factorization Machines

Completed Research Paper

Michael Szubartowicz
University of Regensburg
Universitätsstr. 31
93053 Regensburg, Germany
Michael.Szubartowicz@ur.de

Abstract

During the last decade, factorization machines have proven to be a highly versatile recommendation technique, capable of incorporating additional information such as the rating context. This versatility makes factorization machines a highly promising choice for capturing group dynamics in a group recommender scenario. Therefore, we present with GroupFM three independent approaches (PseudoU, MultiU and GC) to enable group recommendations with factorization machines. To evaluate GroupFM, we used the CAMRa2011 data set as well as a newly collected data set. The new data set contains contextually annotated ratings from both single users and groups and is made public to support the evaluation of future context-aware group recommender systems. Based on our results on both data sets, each approach was able to outperform existing group recommender systems given sufficient group ratings. Therefore, we discuss the individual benefits and characteristics of each approach depending on the number of available group ratings.

Keywords: Factorization Machines, Context-Aware Recommender Systems, Group Recommender Systems

Introduction

Recommender systems have become an essential tool to navigate the digital world offering a virtually limitless number of products to buy or experience goods to consume (Aggarwal 2016; Ricci et al. 2015). Highly personalized recommendations are needed to support users in their decision making. With this in mind, research has shown that user ratings are highly dependent on contextual factors (Adomavicius and Tuzhilin 2011). This has been acknowledged by context-aware recommender systems which were introduced by Herlocker and Konstan (2001) and Adomavicius and Tuzhilin (2001). In their survey on context-aware recommender systems, Adomavicius and Tuzhilin (2011) distinguish the three paradigms contextual pre-filtering, contextual post-filtering and contextual modeling. While contextual pre-filtering and post-filtering approaches employ existing recommendation strategies by modifying their input or output to match a given context, contextual modeling approaches integrate the contextual information into the recommendation process. In 2011, Adomavicius and Tuzhilin observed that "the field of context-aware recommender systems (CARS) is a relatively new and underexplored area of research" which had proposed a small number of contextual modeling approaches. However, this research field experienced a major increase in interest with the use of factorization approaches for contextual modeling (Lahlou et al. 2017). A recent study about deep learning in recommender systems by Dacrema et al. (2019) shows that factorization approaches are still widely used in the field of recommender systems and may serve as a reference when developing novel approaches. Accordingly, context-aware factorization machines (FM) introduced by Rendle et al. (2011) are regarded as the current state-of-the-art context-aware recommendation method outperforming other context-aware recommenders such as Multiverse (Karatzoglou et al. 2010) in terms of predictive accuracy and recommendation time (Hong et al. 2015; Lahlou et al. 2017). This may be attributed to the use of feature vectors which are highly versatile and designed with data sparsity in mind as well as to the development of efficient learning algorithms which are capable of computing factorization machines in linear time.

While the main goals of recommender systems are increasing cross-selling, acquiring customers and building customer loyalty (Schafer et al. 2001), Jannach and Adomavicius (2016) found many other underexplored goals and purposes, one of which being to establish group consensus. In fact, recent technological developments have enabled an increasing number of devices such as phones, TVs and cars to be connected, which provides plentiful opportunities to leverage recommendation techniques to support groups choosing their group activities. This is underlined by the surge of remote collaboration and socializing. The main application areas of existing group recommender systems are tourism, movies, TV and music (Baltrunas et al. 2011; Bobadilla et al. 2013; Lu et al. 2015; McCarthy et al. 2006; O'Connor et al. 2002) with the predominant method to generate group recommendations being the use of aggregation strategies before or after the application of existing recommender systems (preference aggregation vs. prediction aggregation). However, these strategies are not able to fully capture group preferences and dynamics (Delic et al. 2016; Sacharidis 2017) and only few researchers have explored the use of factorization strategies in this research field. In particular, the versatility of factorization machines has not been leveraged to capture the dynamics of group decision processes. Therefore, we present the following research question:

How can factorization machines be utilized to enable group recommendations?

To address this research question, we present GroupFM, an implementation containing three independent approaches PseudoU, MultiU and GC to establish feature vectors capable of representing both single user ratings and group ratings. More precisely, each of the aforementioned approaches provide a single model which can be learned by and predict both single user ratings and group ratings. While PseudoU models each group as a separate pseudo user, MultiU models group ratings by assigning a weight to each group member and GC introduces group context features to indicate the presence of group members. To the best of our knowledge, this is the first work utilizing the versatility and performance of factorization machines for group recommendations. With this in mind, GroupFM is evaluated on the CAMRa2011 data set (Said et al. 2011) as well as a new data set containing real group ratings, single user ratings, and assignments of contextual effects. This data set is made publicly available and can be retrieved from https://github.com/michaelszubartowicz/groupfm.

The remainder of this article is structured as follows: In the following section, we define the problem setting and discuss the related work. GroupFM and its approaches PseudoU, MultiU and GC are introduced in the third section. In the following section, we describe the used data sets as well as our evaluation methodology and discuss the evaluation results. The last section contains a summary, limitations and directions for future work.

Background

Problem setting

Given a set of users U, a set of user groups $G \subseteq \wp(U)$, a set of items I, a nonempty set $R_U = \{r_{ui} | ui \in U \times I\}$ of user ratings, a set $R_G = \{r_{gi} | gi \in G \times I\}$ of group ratings the task of generating group recommendations is to predict group ratings r_{gi} for a group g in order recommend those items which satisfy the group the most (i.e., with the highest predicted group ratings). Typically, only items which have not been rated by the group are considered for recommendation (i.e., $r_{gi} \notin R_G$). We extend the group recommendation problem by assigning each rating from R_U and R_G with contextual information. This contextual information can be explicitly defined by a set of contextual variables and their values such as *{weather: sunny, time: evening}* (Adomavicius and Tuzhilin 2011; Dourish 2004). With this in mind, the task of context-aware group recommendation is to predict a rating score given a specific group g, an item i and with respect to a contextual situation (i.e., contextual values $c_3, \dots, c_m$)[1].

Related work

In this section, we give a brief overview of the literature on factorization machines and discuss relevant works presenting context-aware group recommender systems as well as approaches incorporating factorization techniques into the group recommendation process (i.e., factorization is not used as a "black box" in conjunction with aggregation strategies).

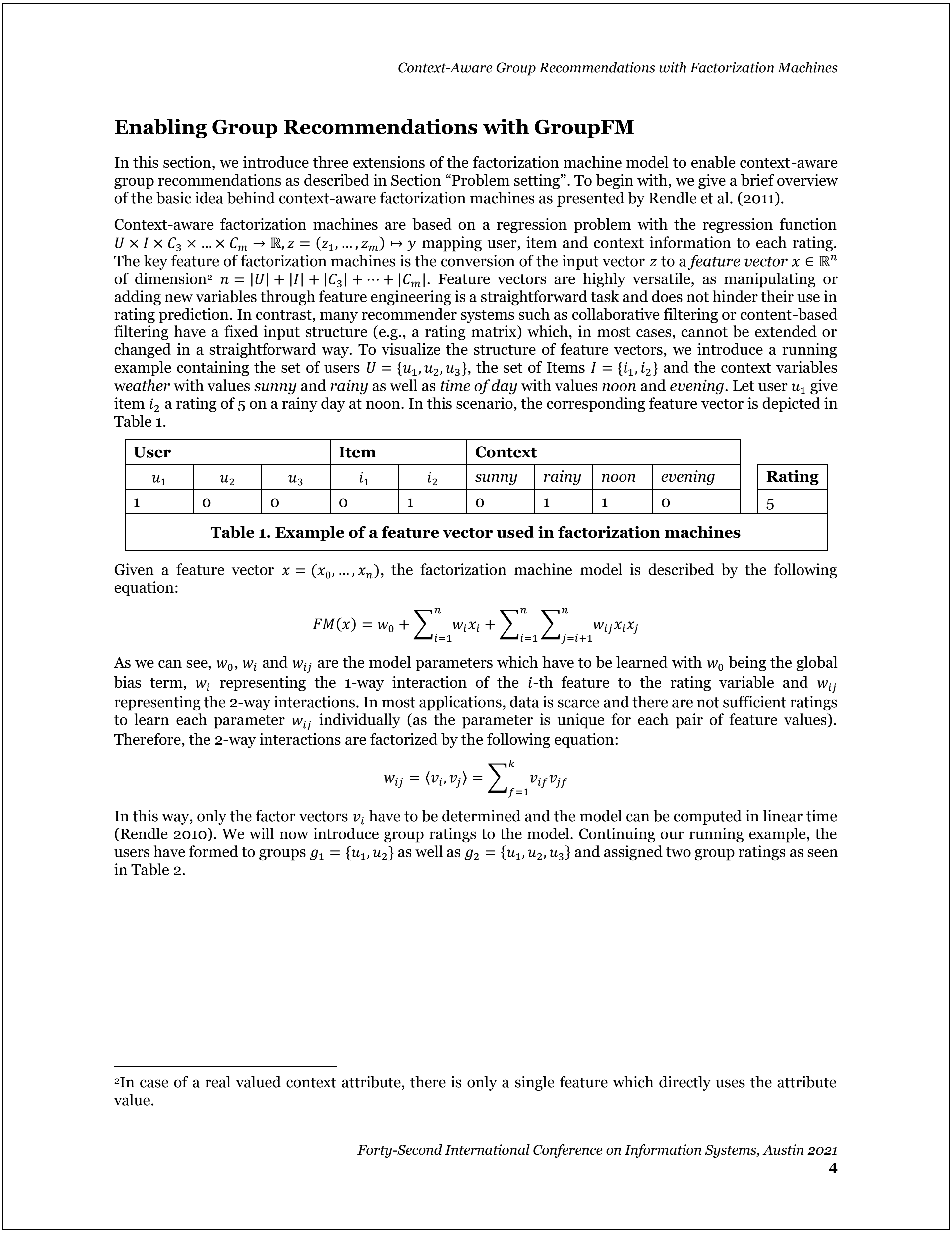
Factorization machines were introduced by Rendle (2010) and have since seen a wide adoption in many application scenarios (Chu and Huang 2017; Huang et al. 2019; Juan et al. 2016; Pan et al. 2015). They are regarded as a highly versatile general model which can be applied to a multitude of use cases through feature engineering (i.e., choosing the right data representation) without the need for any special adaptation (Lahlou et al. 2017). This has led to the development of approaches incorporating different types of additional data into factorization machines. Most notably, the addition of contextual variables to factorization machines by Rendle et al. (2011) is regarded as the hitherto best performing context-aware recommender system (Hong et al. 2015; Lahlou et al. 2017). However, the use of factorization machines in a group recommendation scenario was not yet investigated.

Many approaches providing context-aware recommendation to groups have been proposed. Hussein et al. (2014) present a universal software framework which implements voting strategies (aggregation) to enable group recommendations while the base recommendation module can be filled with any context-aware recommender system. Khalid et al. (2014) use the location, current speed, and traffic conditions to filter items based on the expected arrival time of each group member. The recommendation is based on a score which aggregates the product of arrival time, item check-ins and authority score of each user. Similarly, Chang et al. (2015) use coefficients based around social network analysis of likes and posts on Facebook while incorporating a k-nearest neighbor approach, group aggregation and contextual similarity to facilitate contextualized group recommendations. Quintarelli et al. (2016, 2019) compute the weighted average of group member ratings by determining the contextualized user influence based on historical data. Instead of using an aggregation strategy to create a group profile, Stefanidis et al. (2012) use aggregation on contextualized user-user-similarities to calculate the similarity of a user to a group. By using a similarity threshold, the peers of a group are determined (after context relaxation, if needed) and their ratings are aggregated. The work of Baltrunas et al. (2011) extends matrix factorization by item-context bias terms similar to Koren (2009) followed by recommendation aggregation according to Baltrunas et al. (2010). However, none of these approaches utilize the versatility and predictive power of factorization machines.

In the field of group recommender systems, many approaches calculate group recommendations by aggregating user profiles and using them with conventional (single user) recommender systems as a "black box" or, similarly, by aggregating the output of such recommender systems. As such, approaches that use factorization techniques to fill in this "black box" (e.g., Baltrunas et al. 2011; Castro et al. 2018; Christensen and Schiaffino 2013) are not the focus of this paper as the insufficiencies of common aggregation strategies (Delic and Neidhardt 2017; Hu et al. 2014; Sacharidis 2017) are not addressed by the factorization techniques. In contrast, Ortega et al. (2016) present three approaches to compute latent vectors of groups in a matrix factorization scenario and thereby including the grouping strategy into the recommendation algorithm. Similarly, Sacharidis (2017) introduces RESIDUAL, which learns user-group bias terms by a separate matrix factorization approach, and TRIAD, in which user weights are learned together with the latent features of the matrix factorization model. Furthermore, Wang et al. (2016) utilize separable non-negative matrix factorization in order to infer a member contribution score for each user-group combination and Wang et al. (2018) propose a tensor factorization approach which incorporates user and group preferences. While each of these approaches constitutes a valuable contribution to the group recommender systems literature, no approach using factorization machines has been proposed.

Summing up the above, the application of factorization machines in a group recommendation setting seems highly promising, but has not been realized to the best of our knowledge.

[1] Contextual variables start at index 3 in alignment with Rendle et al. (2011).



Enabling Group Recommendations with GroupFM

In this section, we introduce three extensions of the factorization machine model to enable context-aware group recommendations as described in Section “Problem setting”. To begin with, we give a brief overview of the basic idea behind context-aware factorization machines as presented by Rendle et al. (2011).

Context-aware factorization machines are based on a regression problem with the regression function $U \times I \times C_3 \times \ldots \times C_m \to \mathbb{R}, z = (z_1, \ldots, z_m) \mapsto y$ mapping user, item and context information to each rating. The key feature of factorization machines is the conversion of the input vector z to a *feature vector* $x \in \mathbb{R}^n$ of dimension[2] $n = |U| + |I| + |C_3| + \cdots + |C_m|$. Feature vectors are highly versatile, as manipulating or adding new variables through feature engineering is a straightforward task and does not hinder their use in rating prediction. In contrast, many recommender systems such as collaborative filtering or content-based filtering have a fixed input structure (e.g., a rating matrix) which, in most cases, cannot be extended or changed in a straightforward way. To visualize the structure of feature vectors, we introduce a running example containing the set of users $U = \{u_1, u_2, u_3\}$, the set of Items $I = \{i_1, i_2\}$ and the context variables w*eather* with values *sunny* and *rainy* as well as *time of day* with values *noon* and *evening*. Let user u_1 give item i_2 a rating of 5 on a rainy day at noon. In this scenario, the corresponding feature vector is depicted in Table 1.

User			**Item**		**Context**				
u_1	u_2	u_3	i_1	i_2	*sunny*	*rainy*	*noon*	*evening*	**Rating**
1	0	0	0	1	0	1	1	0	5

Table 1. Example of a feature vector used in factorization machines

Given a feature vector $x = (x_0, \ldots, x_n)$, the factorization machine model is described by the following equation:

$$FM(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{ij} x_i x_j$$

As we can see, w_0, w_i and w_{ij} are the model parameters which have to be learned with w_0 being the global bias term, w_i representing the 1-way interaction of the i-th feature to the rating variable and w_{ij} representing the 2-way interactions. In most applications, data is scarce and there are not sufficient ratings to learn each parameter w_{ij} individually (as the parameter is unique for each pair of feature values). Therefore, the 2-way interactions are factorized by the following equation:

$$w_{ij} = \langle v_i, v_j \rangle = \sum_{f=1}^{k} v_{if} v_{jf}$$

In this way, only the factor vectors v_i have to be determined and the model can be computed in linear time (Rendle 2010). We will now introduce group ratings to the model. Continuing our running example, the users have formed to groups $g_1 = \{u_1, u_2\}$ as well as $g_2 = \{u_1, u_2, u_3\}$ and assigned two group ratings as seen in Table 2.

[2] In case of a real valued context attribute, there is only a single feature which directly uses the attribute value.

Group		**Item**		**Context**				
g_1	g_2	i_1	i_2	*sunny*	*rainy*	*noon*	*evening*	**Rating**
1	0	1	0	1	0	0	1	3
0	1	0	1	1	0	1	0	4

Table 2. Example of feature vectors for group ratings

In this table, we replaced all user-related features with features which indicate the group that a rating belongs to. One way to facilitate group recommendations is to apply factorization machines to such feature vectors. However, this means that the feature vectors for group ratings and single user ratings are not compatible and, thus, a separate model for group recommendations is needed. As rating data suffers from a high level of sparsity, it would be preferable to also use single user ratings when learning the model, which would help to optimize model parameters related to items and context variables on a broader data basis.

Therefore, GroupFM contains three approaches to define feature vectors that can describe both single user and group ratings in order to create a factorization machine model for both single user and group recommendation. These approaches are based on the two ideas of introducing group-specific features and using user features to indicate groups. Firstly, based on the aforementioned idea of using feature vectors as described in Table 2, both user and group-specific features are used in PseudoU. In contrast, MultiU operates on the feature vectors given by Table 1 without introducing new features. Groups are indicated by assigning a weight to the user feature of each group member. Lastly, GC combines these two concepts by proposing a second set of user features with each feature indicating that a user was present as a group member.

PseudoU

The first approach is based on the idea of defining additional features which are specifically used for group ratings. This can be realized directly by treating each group as a pseudo user and extending the number of user features accordingly (O'Connor et al. 2002). Looking at the running example, the feature vectors corresponding to PseudoU are presented in Table 3.

User					**Item**		**Context**				
u_1	u_2	u_3	g_1	g_2	i_1	i_2	*sunny*	*rainy*	*noon*	*evening*	**Rating**
0	0	0	1	0	1	0	1	0	0	1	3
0	0	0	0	1	0	1	1	0	1	0	4

Table 3. Example of feature vectors produced by PseudoU

When applying this approach, the interaction parameters for groups and users are learned separately. This means that features are dedicated to capture the dynamics and characteristics of each group exclusively. In that sense, PseudoU might prove beneficial for groups whose rating behavior deviates from the ratings of each individual group member as no user-specific features are involved in the group recommendation process. To give an example, there might be users which do not enjoy going out at night individually, but who will love to engage in group activities at this time. In this scenario, approaches that use captured user preferences to infer group preferences would be at a disadvantage. There is, however, a drawback to PseudoU when recommendations have to be made for new groups. Similar to the new user problem, rating predictions can only be given for groups with existing group ratings as the corresponding features have been established in the feature vectors.

MultiU

Instead of establishing new features, MultiU processes group ratings while using the existing user features. To be more precise, MutliU uses the feature values x_i to assign a weight to each member of a group g in such a way that $\sum_{u_l \in g} x_l = 1$. A straightforward way to do this is by using equal weights (i.e., the feature value for each group member is $1/|g|$). Depending on the application scenario, other approaches such as weighting each user individually based on his or her authority, experience or physical abilities might seem appropriate (Masthoff 2015). Regarding our running example, the feature values computed by MultiU with equal weighting are depicted in Table 4.

User			**Item**		**Context**				
u_1	u_2	u_3	i_1	i_2	*sunny*	*rainy*	*noon*	*evening*	**Rating**
0.5	0.5	0	1	0	1	0	0	1	3
0.3	0.3	0.3	0	1	1	0	1	0	4

Table 4. Example of feature vectors produced by MultiU

In contrast to PseudoU, this approach has the advantage of being able to predict ratings for newly formed groups based on a previously learned factorization machine as long as each user has an individual rating history. Another advantage is the fact, that the 1-way interaction parameters w_i and the 2-way interaction factor vectors v_i used for predicting group ratings have been learned on a broader data basis including group ratings and single user ratings. In addition, MultiU does not introduce new features which could speed up model training time which is linear in k and n.

Given a feature vector x for a group g with a fixed weighting of users $\sum_{u_i \in g} x_i = 1$, the predicted group rating $FM(x)$ utilizing MultiU is equivalent to computing the weighted sum of single user rating predictions for each group member (while using the same model) and adding the summation of the user-user interactions. This can be inferred from a simple reformulation of $FM(x)$: Let $L = \{l_1, \dots, l_{|g|}\}$ be the feature indices corresponding to the group members (i.e., $g = \bigcup_{l \in L}\{u_l\}$ and $\sum_{l \in L} x_l = 1$). Then, the group rating prediction related to x can be written as:

$$
\begin{aligned}
FM(x) &= w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{ij} x_i x_j \\
&= \sum_{l \in L} x_l\, w_0 + \sum_{i=1, i \notin L}^{n} \sum_{l \in L} x_l\, w_i x_i + \sum_{l \in L} w_l x_l + \sum_{i=1, i \notin L}^{n} \sum_{j=i+1, j \notin L}^{n} \sum_{l \in L} x_l\, w_{ij} x_i x_j \\
&+ \sum_{i=1, i \notin L}^{n} \sum_{j \in L} w_{ij} x_i x_j + \sum_{i=l_1}^{l_{|g|}} \sum_{j=i+1}^{l_{|g|}} w_{ij} x_i x_j \\
&= \sum_{l \in L} x_l FM(\hat{x}^{u_l}) \; + \sum_{i=l_1}^{l_{|g|}} \sum_{j=i+1}^{l_{|g|}} w_{ij} x_i x_j
\end{aligned}
$$

Here, $\hat{x}^{u_l}$ describes the feature vector x which was adapted to predict a rating for the user u_l:

$$
\hat{x}_i^{u_l} = \begin{cases} 1, & i = l \\ 0, & i \in L, i \neq l \\ x_i, & \text{else.} \end{cases}
$$

The implication of this relationship is, that the mechanism of capturing and predicting group rating behavior with MultiU can be divided into two parts. The first part of the mechanism is to compute the weighted mean of single user predictions similar to existing group recommender systems. The second part, however, accounts for pairwise interactions between group members. Therefore, MultiU is able to encompass group dynamics which are more complex than averaging individual ratings. Compared to PseudoU, the freedom of capturing group dynamics is smaller as in both parts, the same paramters w_i and v_i are used. Another possible disadvantage is the fact that, given a model which is learned on single user ratings only, the factor vectors v_i are used to compute user-user interactions without optimizing these vectors for such a use.

GC

The third approach contained in GroupFM is based on the idea that the presence of group members can be seen as contextual information influencing single user ratings (Kristoffersen et al. 2018). With this approach, a set of group context features is added, one for each user. If set, the group context feature of a user indicates that s/he is part of a group while a group member (different from the user) is giving a rating.

The group context features can be seen as a generalization of the feature value group named "watched with" from Rendle et al. (2011).

Given a group rating, GC creates multiple feature vectors, namely one for each group member. These feature vectors differ in such a way that one group member is taking the role of the user giving the rating while the other group members are indicated by the group context features. Similar to MultiU, the group context feature values are set as weights if the group has more than two members. Applying GC with equal weighting to the running example provides the feature vectors presented in Table 5.

User			**Group Context**			**Item**		**Context**				
u_1	u_2	u_3	u_1	u_2	u_3	i_1	i_2	*sunny*	*rainy*	*noon*	*evening*	**Rating**
1	0	0	0	1	0	1	0	1	0	0	1	3
0	1	0	1	0	0	1	0	1	0	0	1	3
1	0	0	0	0.5	0.5	0	1	1	0	1	0	4
0	1	0	0.5	0	0.5	0	1	1	0	1	0	4
0	0	1	0.5	0.5	0	0	1	1	0	1	0	4

Table 5. Example of feature vectors produced by GC

Group recommendations are enabled by computing the rating prediction of each group member with the appropriate group context and applying a group aggregation strategy. In that line, a group aggregation strategy is needed which can greatly influence the resulting group recommendations.

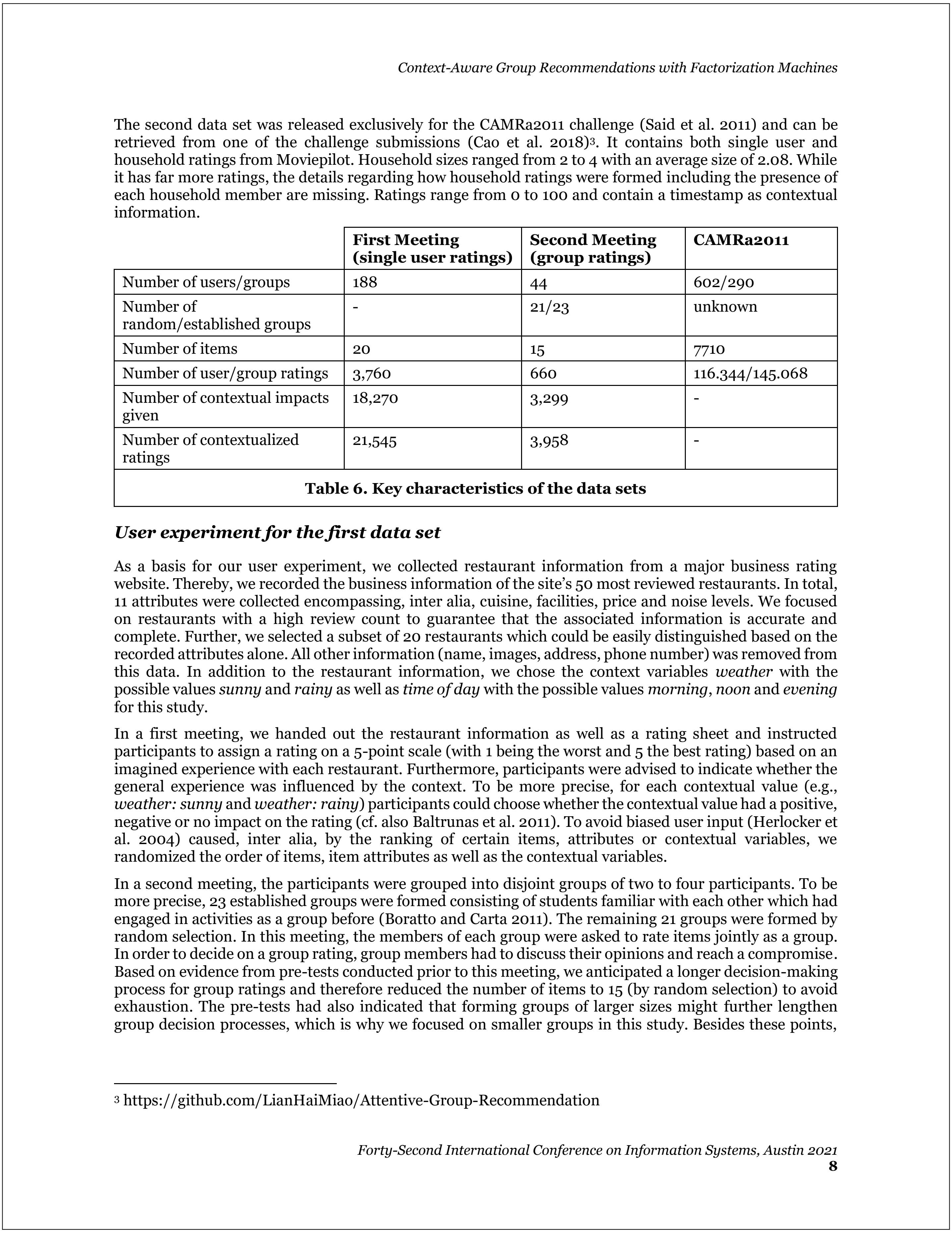
Overall, GC can be seen as a combination of the core principles of PseudoU and MultiU as both new group-specific features are added and the existing user features are used for the representation and prediction of group ratings. Moreover, instead of generating the feature vectors from group ratings, this structure can be used to rate a group activity by each group member individually. Doing so would enable to capture more fine-grained assessments of group experiences as feelings about a movie watched together or a restaurant visited together might differ on an individual basis. However, it should be noted that group members might hesitate to disclose their opinion about a group activity if it could become visible to other group members as it might be hurtful or insulting to disclose that a group activity is not well received by a group member.

Evaluation

Data sets

Existing literature on group recommender systems has raised the issue that most group recommendation systems are evaluated with regard to synthetic group ratings which adds a strong bias towards approaches mimicking the used group rating generation method (Felfernig et al. 2018; Masthoff 2015) or using only implicit group feedback (Quintarelli et al. 2016). Therefore, we used two data sets containing both single user ratings and group ratings in order to evaluate GroupFM. Using group ratings for the evaluation of group recommender systems might not seem appropriate due to the inability to analyze the satisfaction of each individual group member with a recommendation. However, the work of Delic et al. (2016) has shown that the vast majority of group members is satisfied with a group decision, even if it deviates from the personal preference. The characteristics of both data sets are summarized in Table 6.

For the first data set, we conducted a user experiment with Master's degree students in the field of information systems (for similar studies cf. Christensen and Schiaffino 2013; Delic and Neidhardt 2017). In addition to user and group ratings, we collected the users' estimates with respect to the impact of contextual variables on their rating, which enabled us to generate contextualized ratings. The data set is released under the GPL-3.0 license at the link https://github.com/michaelszubartowicz/groupfm. We encourage researchers to use this data set for their own evaluation purposes. In the following section, we will describe the design and procedure of the user experiment conducted to collect the first data set.



The second data set was released exclusively for the CAMRa2011 challenge (Said et al. 2011) and can be retrieved from one of the challenge submissions (Cao et al. 2018)[3]. It contains both single user and household ratings from Moviepilot. Household sizes ranged from 2 to 4 with an average size of 2.08. While it has far more ratings, the details regarding how household ratings were formed including the presence of each household member are missing. Ratings range from 0 to 100 and contain a timestamp as contextual information.

	First Meeting (single user ratings)	**Second Meeting (group ratings)**	**CAMRa2011**
Number of users/groups	188	44	602/290
Number of random/established groups	-	21/23	unknown
Number of items	20	15	7710
Number of user/group ratings	3,760	660	116.344/145.068
Number of contextual impacts given	18,270	3,299	-
Number of contextualized ratings	21,545	3,958	-

Table 6. Key characteristics of the data sets

User experiment for the first data set

As a basis for our user experiment, we collected restaurant information from a major business rating website. Thereby, we recorded the business information of the site's 50 most reviewed restaurants. In total, 11 attributes were collected encompassing, inter alia, cuisine, facilities, price and noise levels. We focused on restaurants with a high review count to guarantee that the associated information is accurate and complete. Further, we selected a subset of 20 restaurants which could be easily distinguished based on the recorded attributes alone. All other information (name, images, address, phone number) was removed from this data. In addition to the restaurant information, we chose the context variables *weather* with the possible values *sunny* and *rainy* as well as *time of day* with the possible values *morning*, *noon* and *evening* for this study.

In a first meeting, we handed out the restaurant information as well as a rating sheet and instructed participants to assign a rating on a 5-point scale (with 1 being the worst and 5 the best rating) based on an imagined experience with each restaurant. Furthermore, participants were advised to indicate whether the general experience was influenced by the context. To be more precise, for each contextual value (e.g., *weather: sunny* and *weather: rainy*) participants could choose whether the contextual value had a positive, negative or no impact on the rating (cf. also Baltrunas et al. 2011). To avoid biased user input (Herlocker et al. 2004) caused, inter alia, by the ranking of certain items, attributes or contextual variables, we randomized the order of items, item attributes as well as the contextual variables.

In a second meeting, the participants were grouped into disjoint groups of two to four participants. To be more precise, 23 established groups were formed consisting of students familiar with each other which had engaged in activities as a group before (Boratto and Carta 2011). The remaining 21 groups were formed by random selection. In this meeting, the members of each group were asked to rate items jointly as a group. In order to decide on a group rating, group members had to discuss their opinions and reach a compromise. Based on evidence from pre-tests conducted prior to this meeting, we anticipated a longer decision-making process for group ratings and therefore reduced the number of items to 15 (by random selection) to avoid exhaustion. The pre-tests had also indicated that forming groups of larger sizes might further lengthen group decision processes, which is why we focused on smaller groups in this study. Besides these points,

[3] https://github.com/LianHaiMiao/Attentive-Group-Recommendation

the rating procedure was conducted analogously to the first meeting. Note that not all participants attended both meetings, which is why more students gave single user ratings than group ratings.

Based on this experiment, we created contextualized ratings for our evaluation. The procedure was adapted from Karatzoglou et al. (2010) who have previously generated semi-synthetic contextualized ratings by introducing randomly created contextual variables and adding (resp. subtracting) a fixed contextual effect to context-free ratings for each context situation. One key advantage of this approach is the ability to compute a multitude of contextualized ratings without conducting lengthy and exhaustive interviews. Accordingly, each recorded rating was divided into multiple contextualized ratings by the following procedure: For each combination of contextual values we added (resp. subtracted) one rating point whenever a user or group had assigned a positive (resp. negative) impact to one of the contextual values, while making sure to not leave the boundaries of the rating scale. This contextualization led to a total of 21,545 single user ratings as well as 3,958 group ratings with an average number of over 114 contextualized ratings per user as well as 89 contextualized ratings per group. Compared to Karatzoglou et al. (2010), our data set has the advantages of using context preferences directly generated from both single users and user groups, which was also achieved by Baltrunas et al. (2011).

Methodology

Our evaluation has two objectives. First, we want to compare the prediction accuracy of the presented approaches to existing group recommender systems. Second, we want to analyze how the prediction accuracy of each approach changes depending on the number of available group ratings.

In order to evaluate GroupFM, we extracted feature vectors corresponding to each approach PseudoU, MultiU and GC from our data set as outlined above and applied the factorization machine implementation libFM by Rendle (2012). For the evaluation of GC, we chose the group aggregation strategies *average* and *least misery*, which are the most popular group aggregation strategies in the field of group recommender systems (Masthoff 2015). These two approaches are denoted by GC_LM and GC_AVG. For our baseline approaches we derived group rating predictions from a factorization machine trained and evaluated on contextualized single user ratings followed by the aforementioned group aggregation strategies. We denote these approaches by FM_LM and FM_AVG, respectively. With these two baselines we can compare the use of factorization machines with popular aggregation strategies to our approaches which directly incorporate group information into the feature vectors. In addition, we chose the group recommender system RESIDUAL (Sacharidis 2017) as a third baseline. RESIDUAL was chosen as it also employs factorization techniques to predict group ratings, is able to model group dynamics by user-group interactions and performs well on data sets containing real group assessments. In order to incorporate contextual information with the RESIDUAL approach, we extended the original approach by using factorized contextual user ratings as well as contextual group ratings (training data, cf. below) to calculate the $U \times G$ matrix of residuals and to generate contextual group ratings.

The prediction accuracy of each approach was measured by the root mean squared error (RMSE). To this end, we split the group ratings into different sets of training and test data while varying the *train ratio* at which group ratings were drawn randomly for the training data set. From the remaining group ratings a fixed set of 250 group ratings was used as test data.[4] That is, each model is trained based on all single user ratings and the group ratings used for training while the predictions for 250 group ratings are used for the calculation of the RMSE. This was done to examine how the prediction accuracy of each approach varies based on the number of available group ratings while ensuring comparable RMSE values across all *train ratios*. Furthermore, the procedure of randomly drawing ratings and calculating the RMSE was repeated a total of 1000 times (CAMRa2011 data set: 100 times) for each approach and each selected value for *train ratio*. This way, effects which might be caused by a particular random drawing of training and test data could be eliminated.

[4] To give an example, a *train ratio* of 0.4 means that 40 percent of group ratings were assigned to the training set and 250 of the remaining 60 percent of group ratings were assigned to the test set.

Results

The prediction accuracy of each approach is depicted in Figure 1 and Figure 2. Each point indicates the average RMSE across the aforementioned 1000 runs on the vertical axis. To further improve readability, the starting point of each vertical axis has been set individually for each figure and is different from zero. To improve readability, the two outliers in Figure 2 were not included. The missing values at *train ratio* of 0 are 46.6 for MultiU and 65.5 for RESIDUAL.

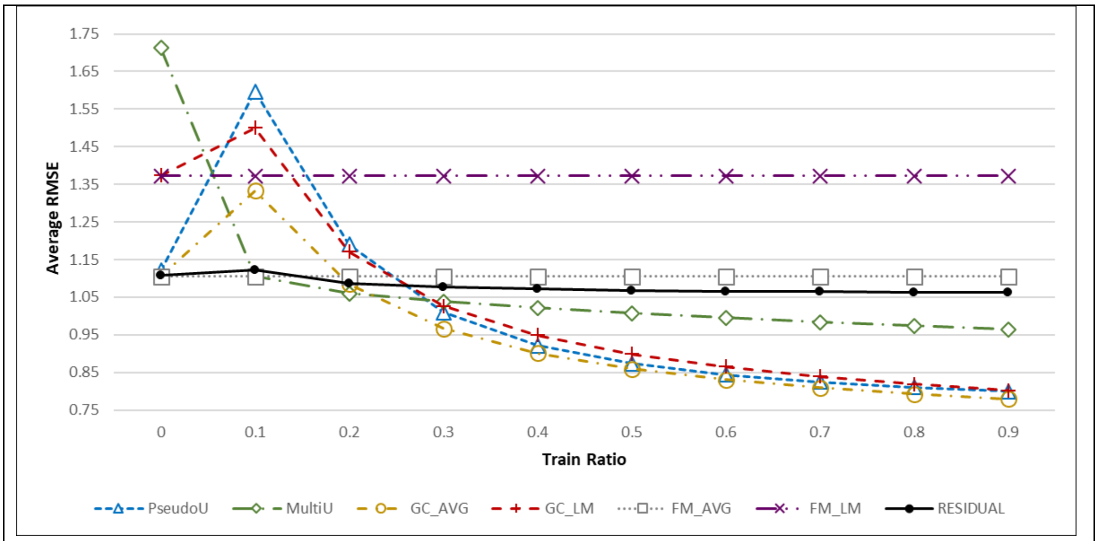


Figure 1. Prediction accuracy for the GroupFM data set

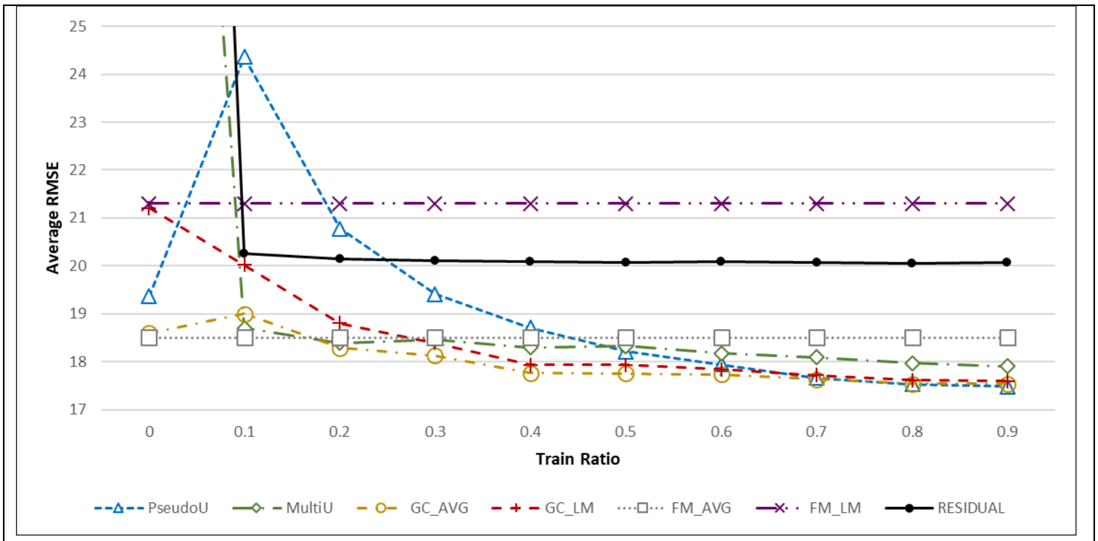


Figure 2. Prediction accuracy for the CAMRa2011 data set

Discussion

General performance comparison. Given a sufficient number of group ratings, all approaches presented in this paper are capable of giving more accurate group recommendations than the baseline approaches. This indicates that each presented approach is able to incorporate the dynamics of group ratings into factorization machines and thereby outperform existing group recommendation strategies in terms of prediction accuracy. However, if the amount of training data is low there is not enough data to train the factorization machines properly. In the following, we give a more detailed analysis of each presented approach and discuss some effects.

MultiU. We first focus on MultiU. Here, for *train ratio* at or above 10% prediction accuracy is near or higher than all baseline approaches. As we pointed out in Section “MulitU”, the only difference between MultiU and FM_AVG is the use of the factor vectors v_{if} to model 2-way interactions between the group members. Thus, the results show that it is possible to utilize these vectors to model group dynamics. If, however, there are too few group ratings available (cf. results for *train ratio* below 0.1), the factor vectors v_{if} are initialized and learned (almost) exclusively for use in a single user recommendation task including user-item interactions as well as user-context interactions. Hence, their use for user-user interactions leads to a significant loss in prediction accuracy compared to computing the average of rating predictions without user-user interactions. Nevertheless, MutliU is able to adapt to group ratings the fastest as evidenced by the particularly high prediction accuracies for *train ratio* between 0.1 and 0.2. This might be due to the fact that there are no group-exclusive features that have to be learned.

Similarities between PseudoU and GC. Considering all data points with *train ratio* between 0.5 and 0.9, the approaches PseudoU, GC_AVG and GC_LM perform very similar and show the lowest RMSE values in that range. This may be attributed to the fact that these approaches define new features which are exclusively dedicated to group ratings and representing group dynamics. However, with fewer group ratings to learn these features from, the prediction accuracies decrease even more drastically as in the case of MultiU. Interestingly, this trend seems to stop and invert when decreasing *train ratio* from 0.1 to 0 (i.e., assigning no group ratings for training purposes) which is why we conducted a more fine-grained analysis with regard to this range.

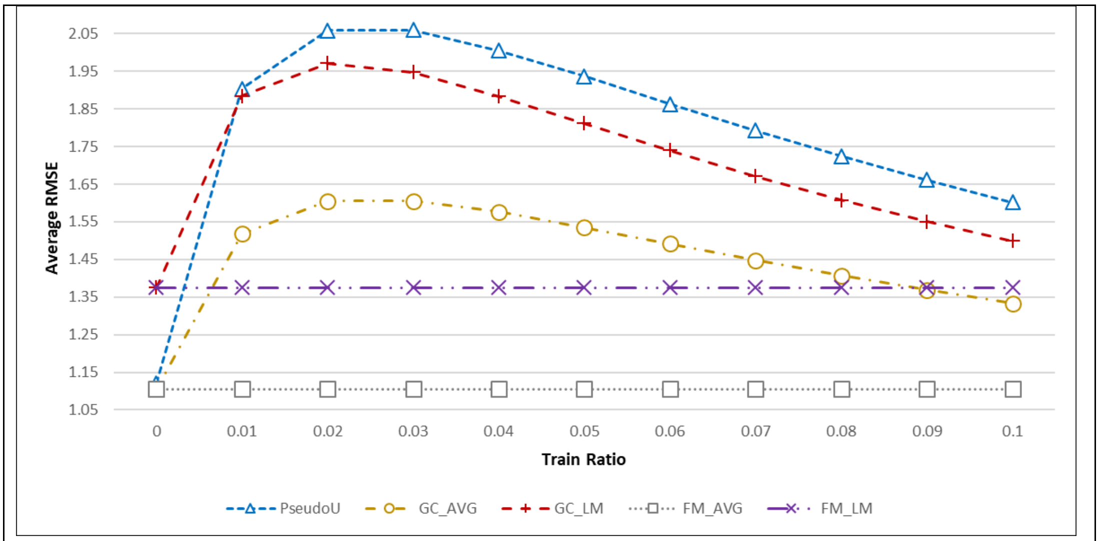


Figure 3. Detailed comparison of PseudoU, GC_AVG, GC_LM, FM_AVG and FM_LM for a low values of *train ratio* (GroupFM data set)

As can be seen in Figure 3, PseudoU and GC_AVG (GC_LM) are very close to FM_AVG (FM_LM) in terms of prediction accuracy if there are no group ratings to learn from. In this situation, all feature values x_i

dedicated to each group as a user or context variable, respectively, are zero throughout the model learning phase. This means that the corresponding 1-way interactions w_i as well as the factorization parameters v_{if} are initialized with 0 or with 0-centered random numbers which are then set to 0 in the learning phase (Rendle et al. 2011). Based on this, these features take no effect within the model and can be removed, meaning that the factorization machine for PseudoU, GC_AVG and GC_LM is equivalent to a factorization machine trained on feature vectors with similar structure as seen in Table 1.

This has differing implications for PseudoU and GC_AVG/GC_LM. For PseudoU, this means that each group prediction is group-agnostic (i.e., the prediction is the same as if no user or group was given) as the feature corresponding to each group as a pseudo user is irrelevant and only the terms related to the global bias, item and context biases and the item-context interactions are used to determine the prediction. To validate this, we set up an approach which eliminates all user or group-specific values from each feature vector in the test data set and applied the same model to this data. In both cases, the RMSE was at 1.12 (GroupFM data set) or 19.4 (CAMRa2011 data set). For GC_AVG and GC_LM, where group ratings are modeled as individual ratings in a group context and aggregated, the features describing the group as context variables take no effect and can be discarded, but the user features are present, which is why these approaches are equivalent to FM_AVG and FM_LM, respectively. Again, this claim was substantiated by matching RMSE values of 1.11 and 1.37 (GroupFM data set) or 18.5 and 21.3 (CAMRa2011 data set).

If group ratings are very sparse (*train ratio* below 0.1, but above 0), GC_AVG, GC_LM and, in particular, PseudoU show a considerably high RMSE. This may be attributed to the fact, that with fewer group ratings, the 1-way interactions w_i as well as the factorization parameters v_{if} regarding the group-specific features (which contributed to the high prediction accuracy in Observation 4) have fewer reference points for learning and thus fewer optimization iterations.

Impact of contextual information. To analyze the impact of the contextual information in the GroupFM data set, we repeated our evaluation without incorporating contextual impact values and compared the resulting RMSE values. The comparison is depicted in Figure 4. Across all approaches and train ratios, incorporating contextual information reduces RMSE values in average by 16%. Furthermore, the impact is stronger for higher values of *train ratio*. This indicates that group recommender systems are able to benefit greatly from using contextual information.

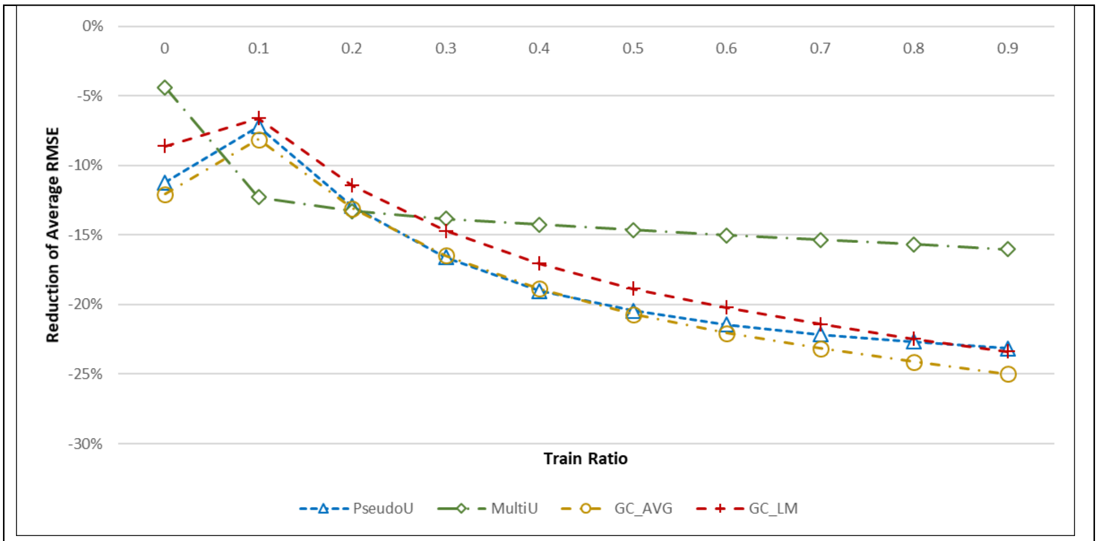


Figure 4. Reduction of average RMSE values through the incorporation of contextual information (GroupFM data set)

Influence of group type. Another interesting aspect is the analysis of different group types within the GroupFM data set. Figure 5 shows the difference in RMSE values between familiar and random groups. In most cases, prediction accuracy is higher for familiar groups. At a train ratio of 0, however, random groups achieve higher prediction accuracy. This might be linked to the fact, that the approaches fall back to FM_AVG, FM_LM or a group-agnostic approach which are not able to capture the stronger group dynamics present in familiar groups. Our analysis showed no preference of a specific approach for one group type, which indicates, that each approach is suitable for both random and familiar groups.

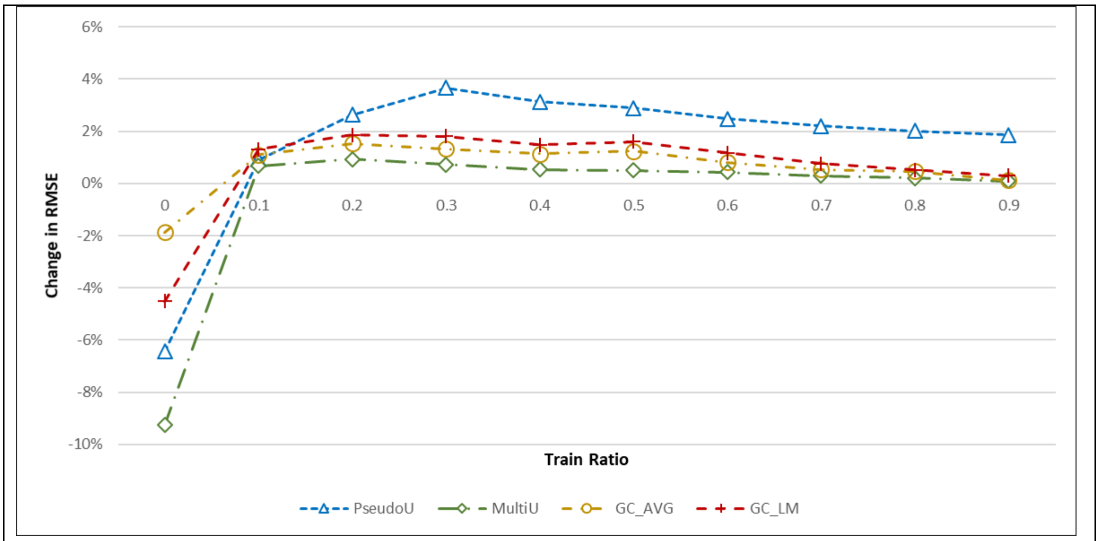
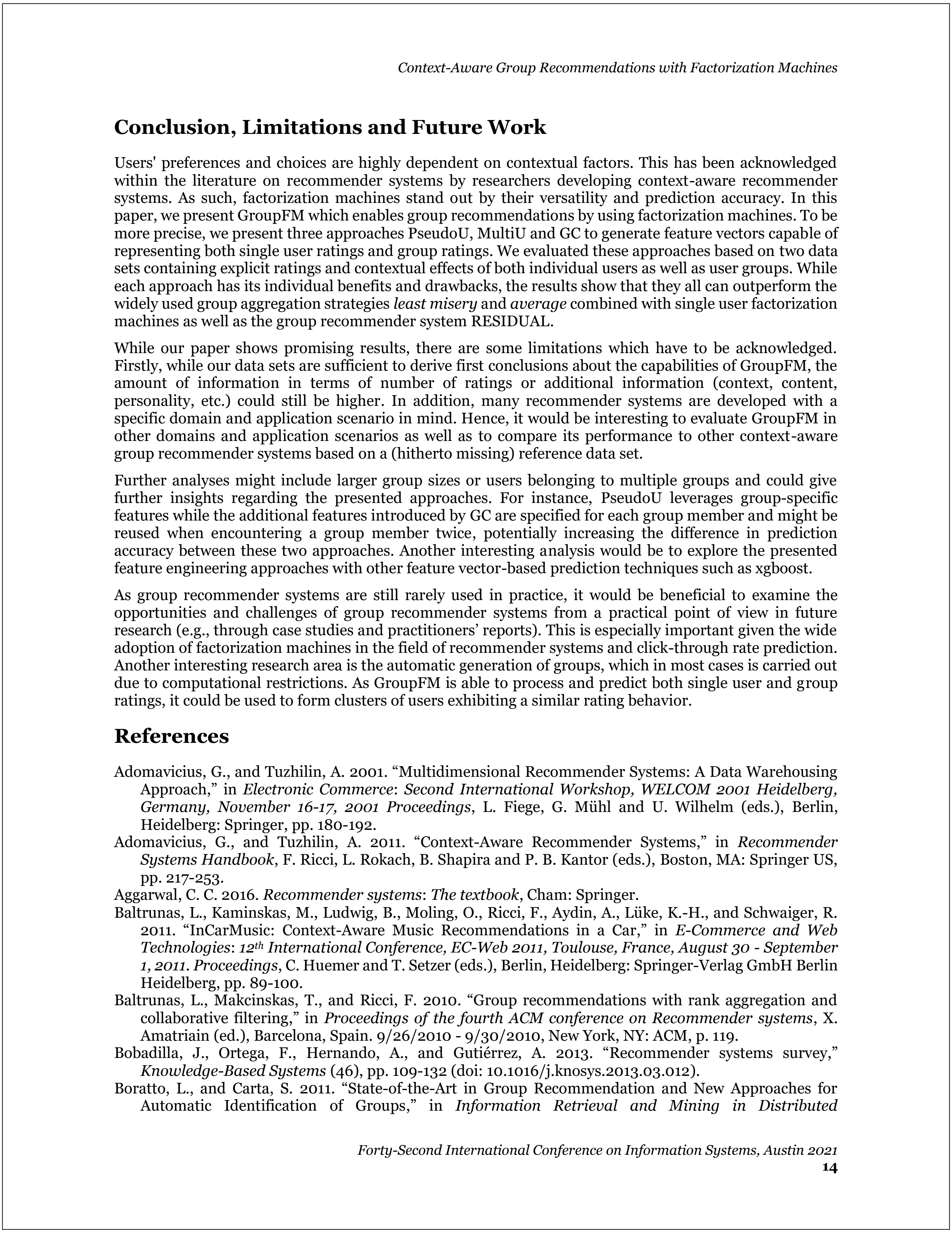


Figure 5. Change in RMSE for random groups compared to familiar groups (GroupFM data set)

Implications for research and practice. With the increasing popularity of remote collaboration and interaction, our research could support the development of new products and services based on remote group activities such as remote viewing or gaming. Having users make their decisions and interactions in a digital world opens the possibility to incorporate a multitude of information such as context, user personality or demographic information into the recommendation process. Our work shows that factorization machines can handle and even benefit from such information in a group recommendation context. Furthermore, our work could motivate researchers to develop group recommender systems which capture group dynamics directly. Furthermore, factorization machines could be analyzed in even more application scenarios.

Summing up, our evaluation shows that it is possible utilize factorization machines to model group dynamics directly (PseudoU and GC) or to use existing knowledge to model group dynamics (MultiU). Note, that none of the presented approaches is able to deliver the most accurate predictions consistently. Instead, the approach should be chosen based on the number of available group ratings or the expected lifespan of groups. Groups formed to watch movies or visit restaurants might accumulate a rich group rating history which the presented approaches can benefit from. Conversely, groups formed in the travel or music domain are often short-lived. When generating recommendations for such groups or newly formed groups common aggregation strategies might be an adequate choice. Groups which already have expressed a few ratings might benefit from the addition of user-user interactions and fast increase in prediction accuracy when using MultiU. If given a rich group rating history (in our data set, this was observed for 30 or more group ratings), more sophisticated and tailored approaches such as PseudoU or GC will have the best recommendation quality with PseudoU being the more straightforward option which requires less configuration. Obviously, the aforementioned thresholds are not set in stone but should act as a starting point to determine domain- or data set-specific thresholds for each application scenario.



Conclusion, Limitations and Future Work

Users' preferences and choices are highly dependent on contextual factors. This has been acknowledged within the literature on recommender systems by researchers developing context-aware recommender systems. As such, factorization machines stand out by their versatility and prediction accuracy. In this paper, we present GroupFM which enables group recommendations by using factorization machines. To be more precise, we present three approaches PseudoU, MultiU and GC to generate feature vectors capable of representing both single user ratings and group ratings. We evaluated these approaches based on two data sets containing explicit ratings and contextual effects of both individual users as well as user groups. While each approach has its individual benefits and drawbacks, the results show that they all can outperform the widely used group aggregation strategies *least misery* and *average* combined with single user factorization machines as well as the group recommender system RESIDUAL.

While our paper shows promising results, there are some limitations which have to be acknowledged. Firstly, while our data sets are sufficient to derive first conclusions about the capabilities of GroupFM, the amount of information in terms of number of ratings or additional information (context, content, personality, etc.) could still be higher. In addition, many recommender systems are developed with a specific domain and application scenario in mind. Hence, it would be interesting to evaluate GroupFM in other domains and application scenarios as well as to compare its performance to other context-aware group recommender systems based on a (hitherto missing) reference data set.

Further analyses might include larger group sizes or users belonging to multiple groups and could give further insights regarding the presented approaches. For instance, PseudoU leverages group-specific features while the additional features introduced by GC are specified for each group member and might be reused when encountering a group member twice, potentially increasing the difference in prediction accuracy between these two approaches. Another interesting analysis would be to explore the presented feature engineering approaches with other feature vector-based prediction techniques such as xgboost.

As group recommender systems are still rarely used in practice, it would be beneficial to examine the opportunities and challenges of group recommender systems from a practical point of view in future research (e.g., through case studies and practitioners' reports). This is especially important given the wide adoption of factorization machines in the field of recommender systems and click-through rate prediction. Another interesting research area is the automatic generation of groups, which in most cases is carried out due to computational restrictions. As GroupFM is able to process and predict both single user and group ratings, it could be used to form clusters of users exhibiting a similar rating behavior.

References

Adomavicius, G., and Tuzhilin, A. 2001. “Multidimensional Recommender Systems: A Data Warehousing Approach,” in *Electronic Commerce*: *Second International Workshop, WELCOM 2001 Heidelberg, Germany, November 16-17, 2001 Proceedings*, L. Fiege, G. Mühl and U. Wilhelm (eds.), Berlin, Heidelberg: Springer, pp. 180-192.

Adomavicius, G., and Tuzhilin, A. 2011. “Context-Aware Recommender Systems,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira and P. B. Kantor (eds.), Boston, MA: Springer US, pp. 217-253.

Aggarwal, C. C. 2016. *Recommender systems*: *The textbook*, Cham: Springer.

Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Aydin, A., Lüke, K.-H., and Schwaiger, R. 2011. “InCarMusic: Context-Aware Music Recommendations in a Car,” in *E-Commerce and Web Technologies*: *12th International Conference, EC-Web 2011, Toulouse, France, August 30 - September 1, 2011. Proceedings*, C. Huemer and T. Setzer (eds.), Berlin, Heidelberg: Springer-Verlag GmbH Berlin Heidelberg, pp. 89-100.

Baltrunas, L., Makcinskas, T., and Ricci, F. 2010. “Group recommendations with rank aggregation and collaborative filtering,” in *Proceedings of the fourth ACM conference on Recommender systems*, X. Amatriain (ed.), Barcelona, Spain. 9/26/2010 - 9/30/2010, New York, NY: ACM, p. 119.

Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. 2013. “Recommender systems survey,” *Knowledge-Based Systems* (46), pp. 109-132 (doi: 10.1016/j.knosys.2013.03.012).

Boratto, L., and Carta, S. 2011. “State-of-the-Art in Group Recommendation and New Approaches for Automatic Identification of Groups,” in *Information Retrieval and Mining in Distributed*

Environments, A. Soro, E. Vargiu, G. Armano and G. Paddeu (eds.), Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, pp. 1-20.

Cao, D., He, X., Miao, L., An, Y., Yang, C., and Hong, R. 2018. "Attentive Group Recommendation," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, K. Collins-Thompson, Q. Mei, B. Davison, Y. Liu and E. Yilmaz (eds.), Ann Arbor MI USA, New York, NY, USA: ACM, pp. 645-654.

Castro, J., Lu, J., Zhang, G., Dong, Y., and Martinez, L. 2018. "Opinion Dynamics-Based Group Recommender Systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (48:12), pp. 2394-2406 (doi: 10.1109/TSMC.2017.2695158).

Chang, A., Hwang, T.-K., Li, Y.-M., and Lin, L.-F. 2015. "A Contextual Group Recommender Mechanism for Location-based Service," in *AMCIS*.

Christensen, I., and Schiaffino, S. 2013. "Matrix Factorization in Social Group Recommender Systems," in *2013 12th Mexican International Conference on Artificial Intelligence,* México, Mexico, IEEE, pp. 10-16.

Chu, W.-T., and Huang, W.-H. 2017. "Cultural difference and visual information on hotel rating prediction," *World Wide Web* (20:4), pp. 595-619 (doi: 10.1007/s11280-016-0404-2).

Dacrema, M. F., Cremonesi, P., and Jannach, D. 2019. "Are we really making much progress?" in *Proceedings of the 13th ACM Conference on Recommender Systems - RecSys '19*, T. Bogers, A. Said, P. Brusilovsky and D. Tikk (eds.), Copenhagen, Denmark. 16.09.2019 - 20.09.2019, New York, New York, USA: ACM Press, pp. 101-109.

Delic, A., and Neidhardt, J. 2017. "A Comprehensive Approach to Group Recommendations in the Travel and Tourism Domain," in *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization - UMAP '17*, M. Bielikova, E. Herder, F. Cena and M. Desmarais (eds.), Bratislava, Slovakia. 09.07.2017 - 12.07.2017, New York, New York, USA: ACM Press, pp. 11-16.

Delic, A., Neidhardt, J., Nguyen, T. N., Ricci, F., Rook, L., Werthner, H., and Zanker, M. 2016. "Observing Group Decision Making Processes," in *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16*, S. Sen, W. Geyer, J. Freyne and P. Castells (eds.), Boston, Massachusetts, USA. 15.09.2016 - 19.09.2016, New York, New York, USA: ACM Press, pp. 147-150.

Dourish, P. 2004. "What we talk about when we talk about context," *Personal and Ubiquitous Computing* (8:1), pp. 19-30 (doi: 10.1007/s00779-003-0253-8).

Felfernig, A., Boratto, L., Stettinger, M., and Tkalčič, M. 2018. "Evaluating Group Recommender Systems," in *Group Recommender Systems*: *An Introduction*, A. Felfernig, L. Boratto, M. Stettinger and M. Tkalčič (eds.), Cham: Springer International Publishing, pp. 59-71.

Herlocker, J. L., and Konstan, J. A. 2001. "Content-independent task-focused recommendation," *IEEE Internet Computing* (5:6), pp. 40-47 (doi: 10.1109/4236.968830).

Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. 2004. "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems* (22:1), pp. 5-53 (doi: 10.1145/963770.963772).

Hong, L., Zou, L., Zeng, C., Zhang, L., Wang, J., and Tian, J. 2015. "Context-Aware Recommendation Using Role-Based Trust Network," *ACM Transactions on Knowledge Discovery from Data* (10:2), pp. 1-25 (doi: 10.1145/2751562).

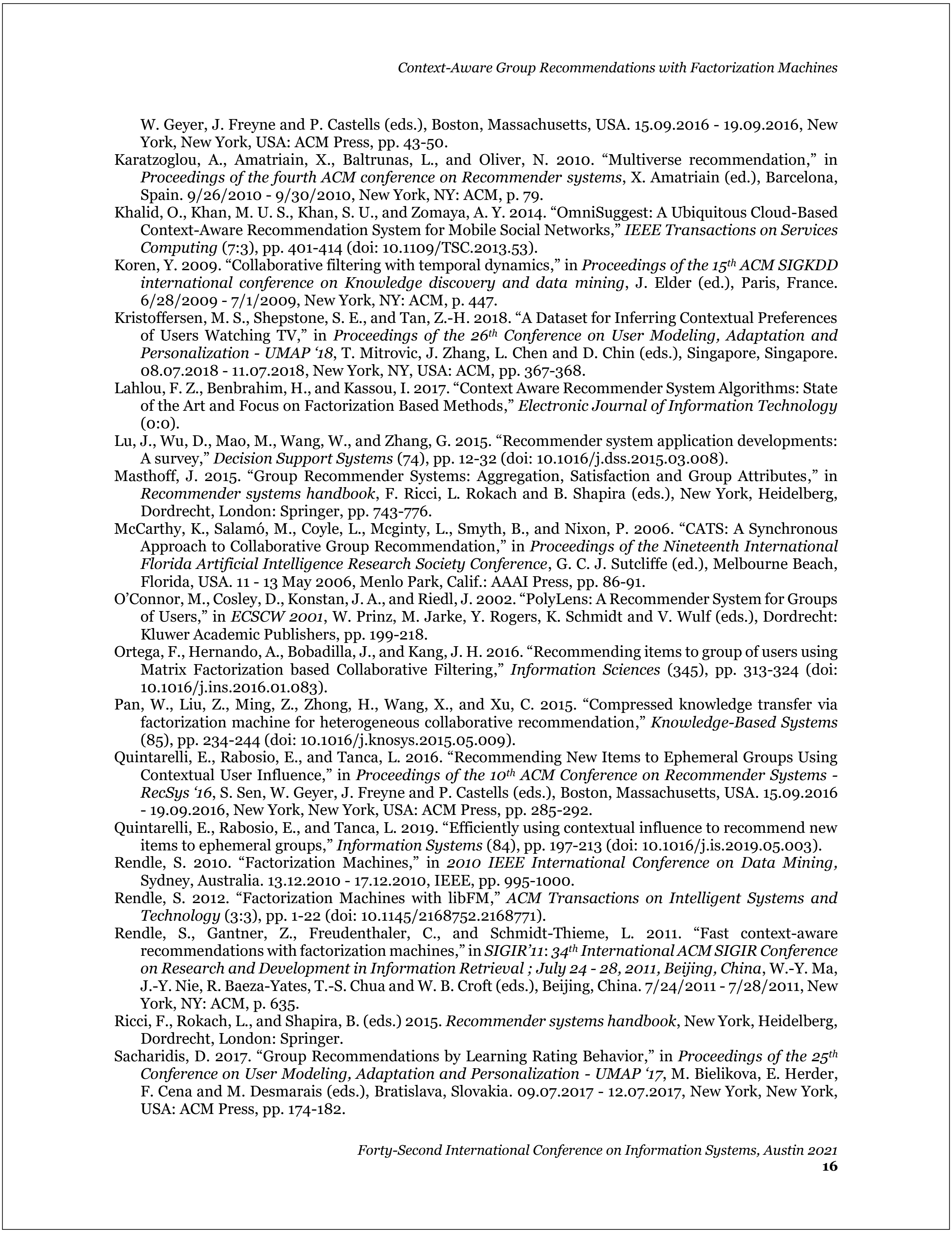
Hu, L., Cao, J., Xu, G., Cao, L., Gu, Z., and Cao, W. 2014. "Deep modeling of group preferences for group-based recommendation," *Proceedings of the National Conference on Artificial Intelligence* (3), pp. 1861-1867.

Huang, T., Zhang, Z., and Zhang, J. 2019. "FiBiNET," in *Proceedings of the 13th ACM Conference on Recommender Systems - RecSys '19*, T. Bogers, A. Said, P. Brusilovsky and D. Tikk (eds.), Copenhagen, Denmark. 16.09.2019 - 20.09.2019, New York, New York, USA: ACM Press, pp. 169-177.

Hussein, T., Linder, T., Gaulke, W., and Ziegler, J. 2014. "Hybreed: A software framework for developing context-aware hybrid recommender systems," *User Modeling and User-Adapted Interaction* (24:1-2), pp. 121-174 (doi: 10.1007/s11257-012-9134-z).

Jannach, D., and Adomavicius, G. 2016. "Recommendations with a Purpose," in *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16*, S. Sen, W. Geyer, J. Freyne and P. Castells (eds.), Boston, Massachusetts, USA. 15.09.2016 - 19.09.2016, New York, New York, USA: ACM Press, pp. 7-10.

Juan, Y., Zhuang, Y., Chin, W.-S., and Lin, C.-J. 2016. "Field-aware Factorization Machines for CTR Prediction," in *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16*, S. Sen,



W. Geyer, J. Freyne and P. Castells (eds.), Boston, Massachusetts, USA. 15.09.2016 - 19.09.2016, New York, New York, USA: ACM Press, pp. 43-50.

Karatzoglou, A., Amatriain, X., Baltrunas, L., and Oliver, N. 2010. "Multiverse recommendation," in *Proceedings of the fourth ACM conference on Recommender systems*, X. Amatriain (ed.), Barcelona, Spain. 9/26/2010 - 9/30/2010, New York, NY: ACM, p. 79.

Khalid, O., Khan, M. U. S., Khan, S. U., and Zomaya, A. Y. 2014. "OmniSuggest: A Ubiquitous Cloud-Based Context-Aware Recommendation System for Mobile Social Networks," *IEEE Transactions on Services Computing* (7:3), pp. 401-414 (doi: 10.1109/TSC.2013.53).

Koren, Y. 2009. "Collaborative filtering with temporal dynamics," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, J. Elder (ed.), Paris, France. 6/28/2009 - 7/1/2009, New York, NY: ACM, p. 447.

Kristoffersen, M. S., Shepstone, S. E., and Tan, Z.-H. 2018. "A Dataset for Inferring Contextual Preferences of Users Watching TV," in *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization - UMAP '18*, T. Mitrovic, J. Zhang, L. Chen and D. Chin (eds.), Singapore, Singapore. 08.07.2018 - 11.07.2018, New York, NY, USA: ACM, pp. 367-368.

Lahlou, F. Z., Benbrahim, H., and Kassou, I. 2017. "Context Aware Recommender System Algorithms: State of the Art and Focus on Factorization Based Methods," *Electronic Journal of Information Technology* (0:0).

Lu, J., Wu, D., Mao, M., Wang, W., and Zhang, G. 2015. "Recommender system application developments: A survey," *Decision Support Systems* (74), pp. 12-32 (doi: 10.1016/j.dss.2015.03.008).

Masthoff, J. 2015. "Group Recommender Systems: Aggregation, Satisfaction and Group Attributes," in *Recommender systems handbook*, F. Ricci, L. Rokach and B. Shapira (eds.), New York, Heidelberg, Dordrecht, London: Springer, pp. 743-776.

McCarthy, K., Salamó, M., Coyle, L., Mcginty, L., Smyth, B., and Nixon, P. 2006. "CATS: A Synchronous Approach to Collaborative Group Recommendation," in *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference*, G. C. J. Sutcliffe (ed.), Melbourne Beach, Florida, USA. 11 - 13 May 2006, Menlo Park, Calif.: AAAI Press, pp. 86-91.

O'Connor, M., Cosley, D., Konstan, J. A., and Riedl, J. 2002. "PolyLens: A Recommender System for Groups of Users," in *ECSCW 2001*, W. Prinz, M. Jarke, Y. Rogers, K. Schmidt and V. Wulf (eds.), Dordrecht: Kluwer Academic Publishers, pp. 199-218.

Ortega, F., Hernando, A., Bobadilla, J., and Kang, J. H. 2016. "Recommending items to group of users using Matrix Factorization based Collaborative Filtering," *Information Sciences* (345), pp. 313-324 (doi: 10.1016/j.ins.2016.01.083).

Pan, W., Liu, Z., Ming, Z., Zhong, H., Wang, X., and Xu, C. 2015. "Compressed knowledge transfer via factorization machine for heterogeneous collaborative recommendation," *Knowledge-Based Systems* (85), pp. 234-244 (doi: 10.1016/j.knosys.2015.05.009).

Quintarelli, E., Rabosio, E., and Tanca, L. 2016. "Recommending New Items to Ephemeral Groups Using Contextual User Influence," in *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16*, S. Sen, W. Geyer, J. Freyne and P. Castells (eds.), Boston, Massachusetts, USA. 15.09.2016 - 19.09.2016, New York, New York, USA: ACM Press, pp. 285-292.

Quintarelli, E., Rabosio, E., and Tanca, L. 2019. "Efficiently using contextual influence to recommend new items to ephemeral groups," *Information Systems* (84), pp. 197-213 (doi: 10.1016/j.is.2019.05.003).

Rendle, S. 2010. "Factorization Machines," in *2010 IEEE International Conference on Data Mining,* Sydney, Australia. 13.12.2010 - 17.12.2010, IEEE, pp. 995-1000.

Rendle, S. 2012. "Factorization Machines with libFM," *ACM Transactions on Intelligent Systems and Technology* (3:3), pp. 1-22 (doi: 10.1145/2168752.2168771).

Rendle, S., Gantner, Z., Freudenthaler, C., and Schmidt-Thieme, L. 2011. "Fast context-aware recommendations with factorization machines," in *SIGIR'11: 34th International ACM SIGIR Conference on Research and Development in Information Retrieval ; July 24 - 28, 2011, Beijing, China*, W.-Y. Ma, J.-Y. Nie, R. Baeza-Yates, T.-S. Chua and W. B. Croft (eds.), Beijing, China. 7/24/2011 - 7/28/2011, New York, NY: ACM, p. 635.

Ricci, F., Rokach, L., and Shapira, B. (eds.) 2015. *Recommender systems handbook*, New York, Heidelberg, Dordrecht, London: Springer.

Sacharidis, D. 2017. "Group Recommendations by Learning Rating Behavior," in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization - UMAP '17*, M. Bielikova, E. Herder, F. Cena and M. Desmarais (eds.), Bratislava, Slovakia. 09.07.2017 - 12.07.2017, New York, New York, USA: ACM Press, pp. 174-182.

Said, A., Berkovsky, S., Luca, E. W. de, and Hermanns, J. 2011. "Challenge on context-aware movie recommendation," in *Proceedings of the fifth ACM conference on Recommender systems - RecSys '11*, B. Mobasher, R. Burke, D. Jannach and G. Adomavicius (eds.), Chicago, Illinois, USA. 23.10.2011 - 27.10.2011, New York, New York, USA: ACM Press, p. 385.

Schafer, J. B., Konstan, J. A., and Riedl, J. 2001. "E-Commerce Recommendation Applications," *Data Mining and Knowledge Discovery* (5:1/2), pp. 115-153 (doi: 10.1023/A:1009804230409).

Stefanidis, K., Shabib, N., Nørvåg, K., and Krogstie, J. 2012. "Contextual Recommendations for Groups," in *Advances in Conceptual Modeling*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, S. Castano, P. Vassiliadis, L. V. Lakshmanan and M. L. Lee (eds.), Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 89-97.

Wang, J., Jiang, Y., Sun, J., Liu, Y., and Liu, X. 2018. "Group recommendation based on a bidirectional tensor factorization model," *World Wide Web* (21:4), pp. 961-984 (doi: 10.1007/s11280-017-0493-6).

Wang, W., Zhang, G., and Lu, J. 2016. "Member contribution-based group recommender system," *Decision Support Systems* (87), pp. 80-93 (doi: 10.1016/j.dss.2016.05.002).

5 Paper 3: Adapting Process Models via an Automated Planning Approach

Adapting Process Models via an Automated Planning Approach

BERND HEINRICH, University of Regensburg[1]
ALEXANDER SCHILLER, University of Regensburg[2]
DOMINIK SCHÖN, University of Regensburg[3]
MICHAEL SZUBARTOWICZ, University of Regensburg[4]

Today's fast-paced business world poses many challenges to companies. Amongst them is the necessity to quickly react to needs for change due to shifts in their competitive environment. Hence, a high flexibility of business processes while maintaining their quality has become a crucial success factor. We address this issue by proposing an automated planning approach that is capable of adapting existing process models to upcoming needs for change. This means that the needs for change have not yet been implemented and the adapted process models have so far not yet been realized. Our work identifies and addresses possible changes to existing process models. Further, it provides adapted process models, which are complete and correct. More precisely, the process models resulting from the presented approach contain all feasible and no infeasible paths. To enable an automated adaptation, the approach is based on enhanced methods from automated planning. We evaluate our approach by means of mathematical proofs, a prototypical implementation and an application in a real-world situation. Further, by means of a computational complexity analysis and a simulation experiment, its runtime is benchmarked against planning process models from scratch.

KEYWORDS

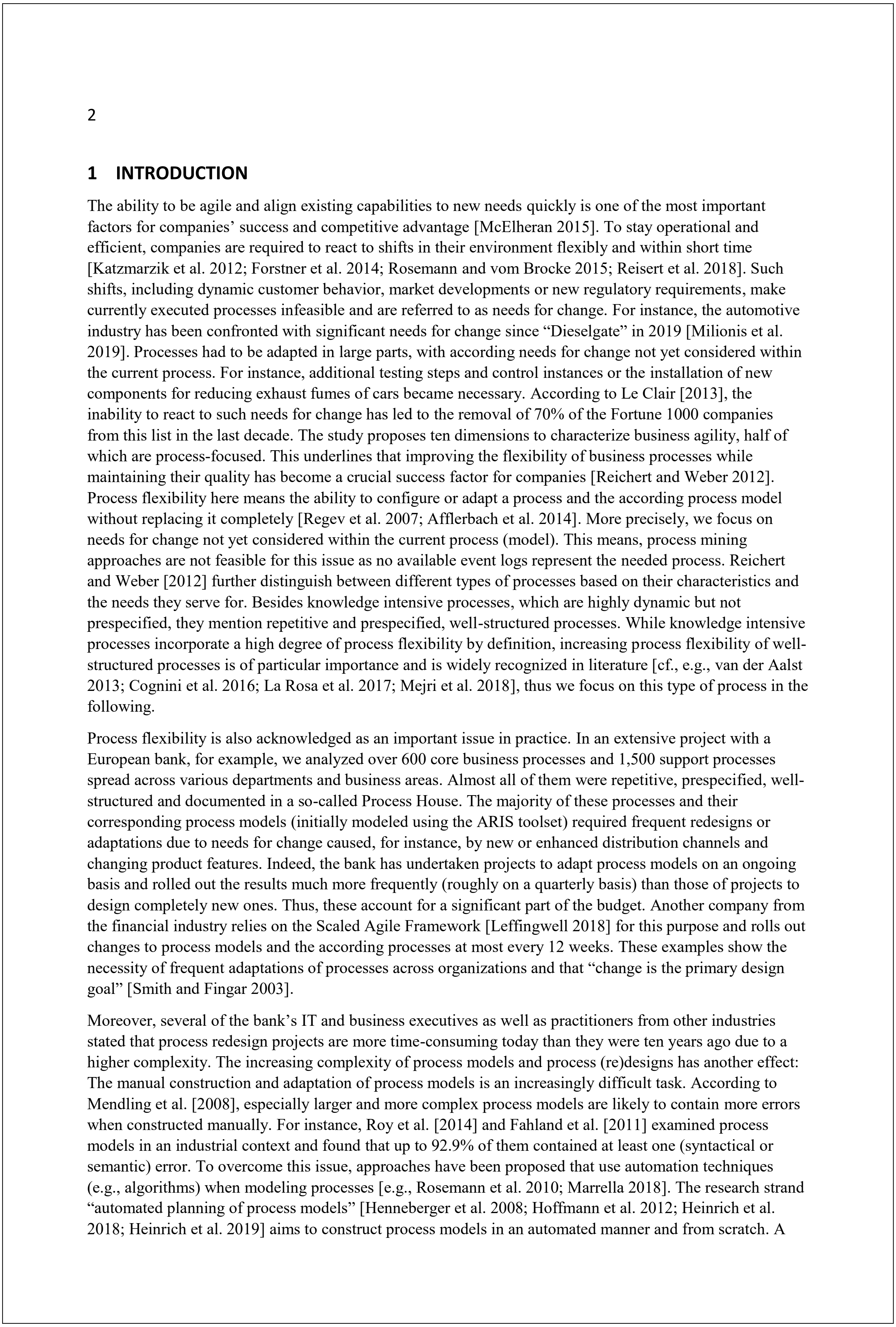
Process Flexibility, Process Changes, Process Models, Business Process Management

[1] Corresponding Author; Faculty of Business, Economics and Management Information Systems, Chair for Information Systems II, Universitätsstraße 31, 93053 Regensburg, Germany, Bernd.Heinrich@ur.de, Tel.: +49 (0)941 943-6101, Fax: +49 (0)941 943-6120

[2] Faculty of Business, Economics and Management Information Systems, Chair for Information Systems II, Universitätsstraße 31, 93053 Regensburg, Germany, Alexander.Schiller@ur.de

[3] Faculty of Business, Economics and Management Information Systems, Chair for Information Systems II, Universitätsstraße 31, 93053 Regensburg, Germany, Dominik.Schoen@ur.de

[4] Faculty of Business, Economics and Management Information Systems, Chair for Information Systems II, Universitätsstraße 31, 93053 Regensburg, Germany, Michael.Szubartowicz@ur.de



1 INTRODUCTION

The ability to be agile and align existing capabilities to new needs quickly is one of the most important factors for companies' success and competitive advantage [McElheran 2015]. To stay operational and efficient, companies are required to react to shifts in their environment flexibly and within short time [Katzmarzik et al. 2012; Forstner et al. 2014; Rosemann and vom Brocke 2015; Reisert et al. 2018]. Such shifts, including dynamic customer behavior, market developments or new regulatory requirements, make currently executed processes infeasible and are referred to as needs for change. For instance, the automotive industry has been confronted with significant needs for change since "Dieselgate" in 2019 [Milionis et al. 2019]. Processes had to be adapted in large parts, with according needs for change not yet considered within the current process. For instance, additional testing steps and control instances or the installation of new components for reducing exhaust fumes of cars became necessary. According to Le Clair [2013], the inability to react to such needs for change has led to the removal of 70% of the Fortune 1000 companies from this list in the last decade. The study proposes ten dimensions to characterize business agility, half of which are process-focused. This underlines that improving the flexibility of business processes while maintaining their quality has become a crucial success factor for companies [Reichert and Weber 2012]. Process flexibility here means the ability to configure or adapt a process and the according process model without replacing it completely [Regev et al. 2007; Afflerbach et al. 2014]. More precisely, we focus on needs for change not yet considered within the current process (model). This means, process mining approaches are not feasible for this issue as no available event logs represent the needed process. Reichert and Weber [2012] further distinguish between different types of processes based on their characteristics and the needs they serve for. Besides knowledge intensive processes, which are highly dynamic but not prespecified, they mention repetitive and prespecified, well-structured processes. While knowledge intensive processes incorporate a high degree of process flexibility by definition, increasing process flexibility of well-structured processes is of particular importance and is widely recognized in literature [cf., e.g., van der Aalst 2013; Cognini et al. 2016; La Rosa et al. 2017; Mejri et al. 2018], thus we focus on this type of process in the following.

Process flexibility is also acknowledged as an important issue in practice. In an extensive project with a European bank, for example, we analyzed over 600 core business processes and 1,500 support processes spread across various departments and business areas. Almost all of them were repetitive, prespecified, well-structured and documented in a so-called Process House. The majority of these processes and their corresponding process models (initially modeled using the ARIS toolset) required frequent redesigns or adaptations due to needs for change caused, for instance, by new or enhanced distribution channels and changing product features. Indeed, the bank has undertaken projects to adapt process models on an ongoing basis and rolled out the results much more frequently (roughly on a quarterly basis) than those of projects to design completely new ones. Thus, these account for a significant part of the budget. Another company from the financial industry relies on the Scaled Agile Framework [Leffingwell 2018] for this purpose and rolls out changes to process models and the according processes at most every 12 weeks. These examples show the necessity of frequent adaptations of processes across organizations and that "change is the primary design goal" [Smith and Fingar 2003].

Moreover, several of the bank's IT and business executives as well as practitioners from other industries stated that process redesign projects are more time-consuming today than they were ten years ago due to a higher complexity. The increasing complexity of process models and process (re)designs has another effect: The manual construction and adaptation of process models is an increasingly difficult task. According to Mendling et al. [2008], especially larger and more complex process models are likely to contain more errors when constructed manually. For instance, Roy et al. [2014] and Fahland et al. [2011] examined process models in an industrial context and found that up to 92.9% of them contained at least one (syntactical or semantic) error. To overcome this issue, approaches have been proposed that use automation techniques (e.g., algorithms) when modeling processes [e.g., Rosemann et al. 2010; Marrella 2018]. The research strand "automated planning of process models" [Henneberger et al. 2008; Hoffmann et al. 2012; Heinrich et al. 2018; Heinrich et al. 2019] aims to construct process models in an automated manner and from scratch. A study by Schön [2019] shows that using such an approach for planning process models in an automated way enables process modelers to construct process models containing less syntactical and even semantical errors. Here, a process model, based on states, actions, and control flow patterns, is planned by means of algorithms. Our approach for an automated adaptation of process models enhances methods of automated planning and thus contributes to this research strand.

Adapting process models is twofold. On the one hand, process models have to be adapted to changes that already have become effective in conducted processes. Such changes include deviations from process models or so called concept drifts that have already been implemented [Bose et al. 2014; Seeliger et al. 2017]. Hence as-is processes [cf. van der Aalst 2013] are considered. In this case, which is not the scope of this paper, the changed process can be modeled by adapting or reconstructing an existing process model regarding new records of event logs (cf. process enhancement; [Kalenkova et al. 2017]). On the other hand, we aim at modeling to-be processes [cf. van der Aalst 2013]. This means that needs for change have not yet become effective and the desired process models have so far not yet been realized. Thus, there is no possibility to mine the changed process model as instances of it do not yet exist. For example, the car industry had to adapt processes in large parts in 2019 due to Dieselgate, as mentioned. Outdated NEDC emissions tests were replaced by WLTP tests and real-driving emissions tests became necessary as part of the so-called type-approval testing. Additionally, new checks by third parties of vehicles in circulation as well as (newly required) conformity checks, verifying whether a type-approved car remains compliant during lifetime, became mandatory. This led to processes that had to be adapted *in advance* as the previously executed processes became infeasible due to this additional regulatory requirement. In this paper, we focus on this latter perspective and aim to adapt existing process models to needs for change *in advance* and to construct models of desired processes, leading to the following research question:

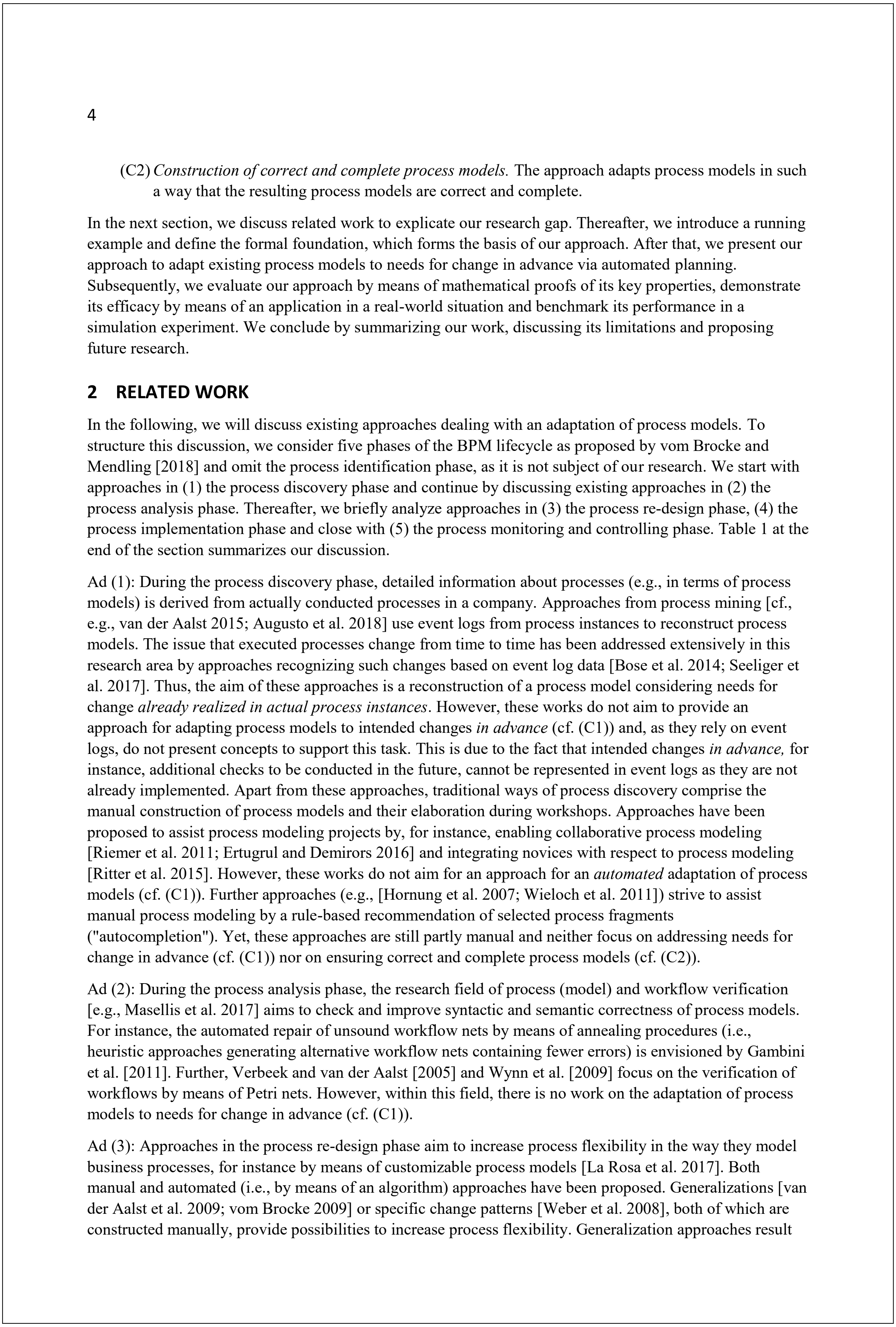
> *How can process models be adapted to needs for change in advance in an automated manner?*

In literature, many existing approaches for the adaptation of process models aim to "repair" process models locally when considering changes [e.g., Eisenbarth et al. 2011; Alférez et al. 2014]. However, both process models and process (re)designs are becoming increasingly large and complex [cf. Hornung et al. 2007 and the discussion above] and local repairs or changes to just some components of process models are not sufficient. Instead, the challenging task of providing adapted process models, which are *correct* and *complete*, has to be addressed. Correct means that the adapted process models contain *only* feasible paths and no infeasible paths, while complete means that the adapted process models contain *all* feasible paths. Correct and complete process models are important to, for instance, increase "flexibility by definition", which is "the ability to incorporate alternative execution paths within a process definition at design time such that selection of the most appropriate execution path can be made at runtime for each process instance" [van der Aalst 2013, p.25]. A correct and complete process model enhances the decision-making aspect of process models by offering the flexibility to assess feasible paths based on economic and resource criteria and constraints. Subsequently the most beneficial feasible path can be selected for process execution (e.g., based on economic criteria). For instance, based on an optimization model, a feasible path with favorable (optimized) execution time may be chosen, when necessary [Heinrich and Lewerenz 2015]. Hence, we discuss the following research question:

> *How can process models be adapted such that the resulting process models are correct and complete?*

The main contributions of this paper are thus as follows:

(C1) *Adaptation to needs for change in advance in an automated manner.* The approach adapts existing process models to needs for change in advance (i.e., no reconstruction of existing process models, e.g., to new records of event logs). To this end, it enhances methods especially from automated planning of process models.



(C2) *Construction of correct and complete process models.* The approach adapts process models in such a way that the resulting process models are correct and complete.

In the next section, we discuss related work to explicate our research gap. Thereafter, we introduce a running example and define the formal foundation, which forms the basis of our approach. After that, we present our approach to adapt existing process models to needs for change in advance via automated planning. Subsequently, we evaluate our approach by means of mathematical proofs of its key properties, demonstrate its efficacy by means of an application in a real-world situation and benchmark its performance in a simulation experiment. We conclude by summarizing our work, discussing its limitations and proposing future research.

2 RELATED WORK

In the following, we will discuss existing approaches dealing with an adaptation of process models. To structure this discussion, we consider five phases of the BPM lifecycle as proposed by vom Brocke and Mendling [2018] and omit the process identification phase, as it is not subject of our research. We start with approaches in (1) the process discovery phase and continue by discussing existing approaches in (2) the process analysis phase. Thereafter, we briefly analyze approaches in (3) the process re-design phase, (4) the process implementation phase and close with (5) the process monitoring and controlling phase. Table 1 at the end of the section summarizes our discussion.

Ad (1): During the process discovery phase, detailed information about processes (e.g., in terms of process models) is derived from actually conducted processes in a company. Approaches from process mining [cf., e.g., van der Aalst 2015; Augusto et al. 2018] use event logs from process instances to reconstruct process models. The issue that executed processes change from time to time has been addressed extensively in this research area by approaches recognizing such changes based on event log data [Bose et al. 2014; Seeliger et al. 2017]. Thus, the aim of these approaches is a reconstruction of a process model considering needs for change *already realized in actual process instances*. However, these works do not aim to provide an approach for adapting process models to intended changes *in advance* (cf. (C1)) and, as they rely on event logs, do not present concepts to support this task. This is due to the fact that intended changes *in advance,* for instance, additional checks to be conducted in the future, cannot be represented in event logs as they are not already implemented. Apart from these approaches, traditional ways of process discovery comprise the manual construction of process models and their elaboration during workshops. Approaches have been proposed to assist process modeling projects by, for instance, enabling collaborative process modeling [Riemer et al. 2011; Ertugrul and Demirors 2016] and integrating novices with respect to process modeling [Ritter et al. 2015]. However, these works do not aim for an approach for an *automated* adaptation of process models (cf. (C1)). Further approaches (e.g., [Hornung et al. 2007; Wieloch et al. 2011]) strive to assist manual process modeling by a rule-based recommendation of selected process fragments ("autocompletion"). Yet, these approaches are still partly manual and neither focus on addressing needs for change in advance (cf. (C1)) nor on ensuring correct and complete process models (cf. (C2)).

Ad (2): During the process analysis phase, the research field of process (model) and workflow verification [e.g., Masellis et al. 2017] aims to check and improve syntactic and semantic correctness of process models. For instance, the automated repair of unsound workflow nets by means of annealing procedures (i.e., heuristic approaches generating alternative workflow nets containing fewer errors) is envisioned by Gambini et al. [2011]. Further, Verbeek and van der Aalst [2005] and Wynn et al. [2009] focus on the verification of workflows by means of Petri nets. However, within this field, there is no work on the adaptation of process models to needs for change in advance (cf. (C1)).

Ad (3): Approaches in the process re-design phase aim to increase process flexibility in the way they model business processes, for instance by means of customizable process models [La Rosa et al. 2017]. Both manual and automated (i.e., by means of an algorithm) approaches have been proposed. Generalizations [van der Aalst et al. 2009; vom Brocke 2009] or specific change patterns [Weber et al. 2008], both of which are constructed manually, provide possibilities to increase process flexibility. Generalization approaches result in less specific process models, for instance, by combining several specific actions to one abstract, general action. Specific change patterns, in contrast, allow the replacement of parts of a process model by different, predesigned parts. The purpose of those approaches is different from ours, since they do not aim to provide an approach for the *automated* adaptation of process models (cf. (C1)). Further, research on adaptive planning is based on a (re)planning problem which is solved by providing an abstraction (also called generalization or template) of a plan and choosing a specification (also called variant or interpretation) at plan execution [Krause et al. 2004]. Consequently, instead of providing complete process models (cf. (C2)), these approaches generate process instances on an ad-hoc basis, in some instances requiring manual interaction [Hulpuș et al. 2010] (cf. (C1)). Moreover, only anticipated adaptations (e.g., changes in supply or demand within a manufacturing process, cf. [Ivanov 2010; Feld and Hoffmann 2014]) are considered. Automated planning of process models, a second research strand in the process re-design phase, strives to increase process flexibility [cf., e.g., Heinrich et al. 2018]. Several approaches in this strand address the issue of adapting process models to needs for change in advance [cf. Lautenbacher et al. 2009; Eisenbarth et al. 2011] by adapting parts of a process model due to (a few) changed actions. They identify so called single-entry-single-exit fragments surrounding an action to be changed. Based on such an identified fragment, "quasi-" initial and goal states (for the considered fragment) are to be determined. Thereafter, the existing fragment is replaced with a new fragment, constructed by regular process planning. Changed actions, however, can affect the whole process model (e.g., when a changed action results in several new feasible paths), so that the process models adapted by these approaches are usually not complete. Further, these approaches do not ensure that the whole process model is correct because of adapting only fragments. Additionally, the need for adapting a process model may not only arise from actions to be changed but also from changed initial and goal states, which is not covered by this research. To sum up, these works do not aim to provide adapted process models which are complete and correct and are "interested in adapting only parts of a model" [Eisenbarth et al. 2011], in contrast to (C2). In addition to that, further approaches in the process re-design phase aim at business process improvement (BPI) or process re-design itself. As Vanwersch et al. [2016] describe, such an improvement or re-design may realize substantial potential in terms of efficiency . For instance, Johannsen and Fill [2017] propose a BPI roadmap as an approach for systematically performing BPI projects. Page [2016] gives a broad overview of steps to improve business processes. Seethamraju and Marjanovic [2009] identify and document the issues, strategies, and practices related to influence of process knowledge possessed by individual participants in a case study. Despite these important contributions, this research field does not aim to propose automated approaches (cf. (C1)) and does not focus on ensuring correct and complete process models (cf. (C2)).

Ad (4): During the process implementation phase, constructed process models are implemented in the according execution systems. For instance, (web) services are composed with the aim of aggregating existing functionality into new functionality. To do so, graph structures consisting of services and states (similar to actions and states in process models) are constructed. Within the research field of (web) service composition, issues similar to the adaptation of process models are discussed as "network configurations and [Quality-of-Service] offerings may change, new service providers and business relationships may emerge and existing ones may be modified or terminated" [Chafle et al. 2006]. Research focuses on replacing (web) services (or small combinations of services) by other, functionally equivalent (small combinations of) services [cf., e.g., Bucchiarone et al. 2011]. Some authors use so called variability models, which are very similar to the change patterns mentioned above, to adapt service compositions [Alférez et al. 2014; La Rosa et al. 2017]. In contrast to these approaches, however, our considered changes regarding process models are not limited to exchanging (a few) actions. We rather aim to adapt whole process models in such a way that the resulting process models are correct and complete (cf. (C2)).

Ad (5): In the process monitoring and controlling phase, several works aim for error handling procedures to resolve process executions interrupted by, for example, external events. They rely on so called continuous planning for the recovery of failed process executions [cf., e.g., Marrella et al. 2012; Linden et al. 2014; van

Beest et al. 2014; Tax et al. 2017]. Other works support users by providing change operations to address ad-hoc deviations from pre-modeled task sequences within a workflow [Reichert and Dadam 1997, 1998; Rinderle et al. 2004]. However, they do not propose an approach for the adaptation of process models to needs for change in advance (cf. (C1)). Further, they aim to address particular process instances, so they do not strive to provide complete process models (cf. (C2)). Another kind of approaches [cf., e.g., Scala et al. 2015; Marrella et al. 2017; Nunes et al. 2018] deals with the issue of adapting a process model due to discrepancies occurred during the conduction, using planning algorithms. They aim to find a sequence of actions that will resolve the misalignment between the modeled and the actual reality [Marrella et al. 2017]. Similarly, Kambhampati [1997] introduces the concept of refinement planning as "the process of starting with the set of all action sequences and gradually narrowing it down to reach the set of all solutions". Here, so called candidates (i.e., parts of a plan consistent with certain constraints) are combined to subsequently construct a feasible complete plan. Further, Gerevini and Serina [2000], propose a fast plan adaptation by identifying delimited parts of the plan that are inconsistent and then replanning the subgraph for these delimited parts. However, these approaches do not aim to adapt process models to needs for change in advance (cf. (C1)) and do not strive to ensure complete process models (cf. (C2)). Further, Nebel and Koehler [1995] suggest a so called "planning from second principles", consisting of two steps: The identification of an appropriate plan candidate from a plan library and its modification so that it solves a new problem instance. However, they do not aim to adapt process models and their characteristics to needs for change in advance (cf. (C1)) and do not strive to construct complete process models (cf. (C2)).

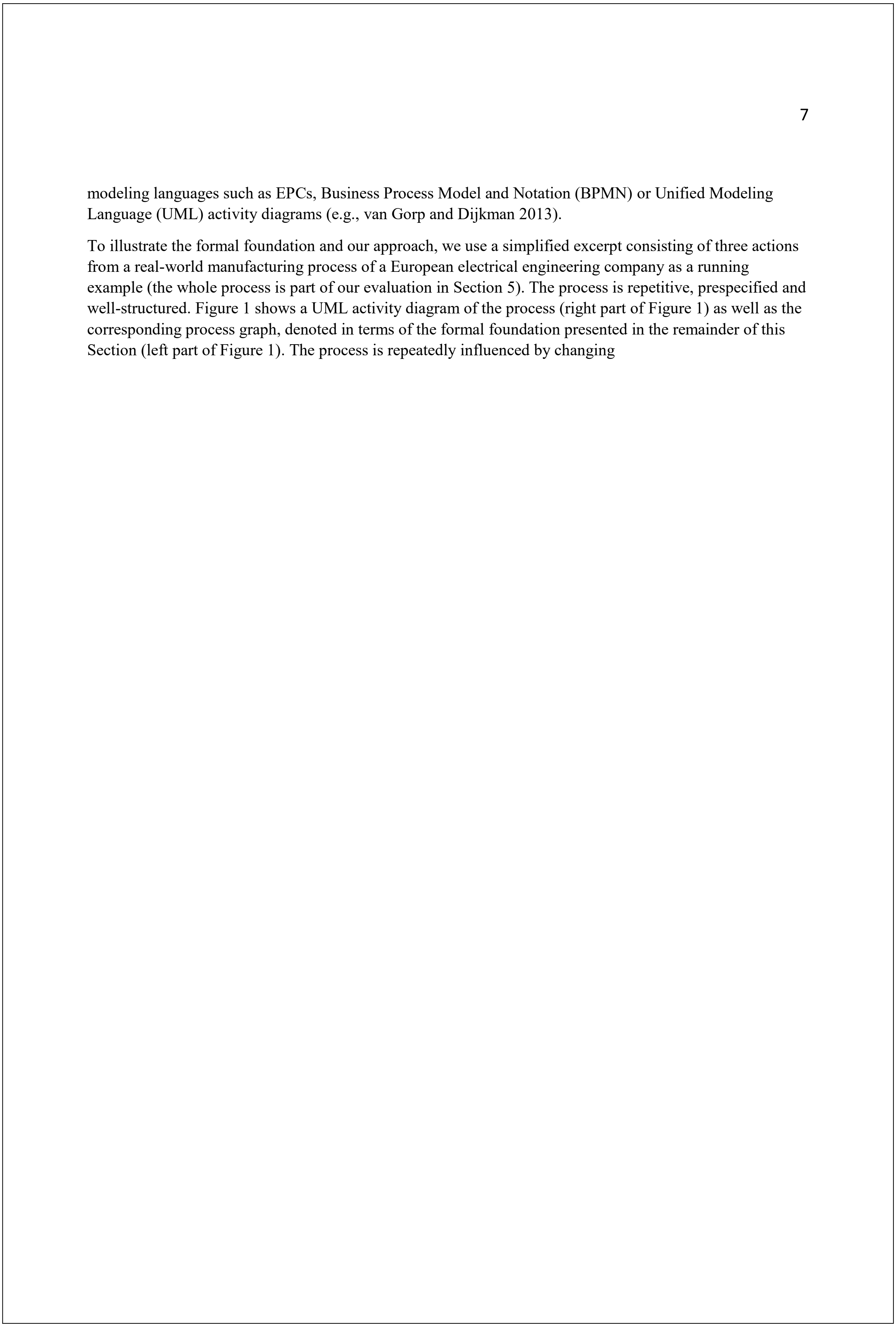
Declarative process models are an alternative to the (imperative) process models addressed in this paper, specifying what should be done in a process, not how [Pesic and van der Aalst 2006; van der Aalst et al. 2009]. They tend to address *momentary* changes, whereas we aim to address both momentary and *evolutionary* changes [van der Aalst and Jablonski 2000]. For declarative process models, it is further proposed to generate so called "optimized enactment plans" that could be understood as a planning problem [cf., e.g., Barba et al. 2013a]. In this context, a replanning approach is envisioned by Barba et al. [2013b], in case the actually conducted process deviates from the generated optimized enactment plan. However, they aim at "optimizing performance goals like minimizing the overall completion time" in contrast to (C2) and do not adapt to needs for change in advance (cf. (C1)).

Finally, the research field of process mining comprises the areas of conformance checking and process enhancement [van der Aalst 2015; Leemans et al. 2018] that are also part of the process monitoring and controlling phase. Conformance checking is used to detect differences between the traces of a process execution (e.g., found in event logs) and a given process model [Garcia-Bañuelos et al. 2017; Leoni and Marrella 2017]. In process enhancement (which deals with tasks such as "model extension" or "model repair"), the goal is to change or extend an already existing process model by taking information about the process instances from event logs into account [cf., e.g., Fahland and van der Aalst 2012]. The focus of process enhancement, analyzing an *existing*, already instantiated and enacted process with respect to deviations from an *existing* process model, however, is different to ours. Therefore, these works do not aim to provide an approach for adapting process models to so far unconsidered changes such as new requirements in advance as they rely on event logs. In contrast, we aim to model a desired process which is not yet realized and thus to adapt to needs for change in advance (cf. (C1)).

In sum, to the best of our knowledge no existing approach adapts process models to needs for change in advance in an automated manner (cf. (C1)) and constructs correct and complete process models (cf. (C2)).

3 FORMAL FOUNDATION & RUNNING EXAMPLE

In our research, we aim for a representation of process models independent of a particular process modeling language. More precisely, in contrast to relying on one single process modeling language such as Event-driven Process Chains (EPC), we use a formal foundation that provides a broader application scope for our approach. Our formal foundation includes so called process graphs, which are also referred to as planning graphs in the research field of automated planning of process models [e.g., Henneberger et al. 2008; Lin et al. 2012; Heinrich et al. 2015]. Process graphs utilize similar concepts as existing well-known process



modeling languages such as EPCs, Business Process Model and Notation (BPMN) or Unified Modeling Language (UML) activity diagrams (e.g., van Gorp and Dijkman 2013).

To illustrate the formal foundation and our approach, we use a simplified excerpt consisting of three actions from a real-world manufacturing process of a European electrical engineering company as a running example (the whole process is part of our evaluation in Section 5). The process is repetitive, prespecified and well-structured. Figure 1 shows a UML activity diagram of the process (right part of Figure 1) as well as the corresponding process graph, denoted in terms of the formal foundation presented in the remainder of this Section (left part of Figure 1). The process is repeatedly influenced by changing

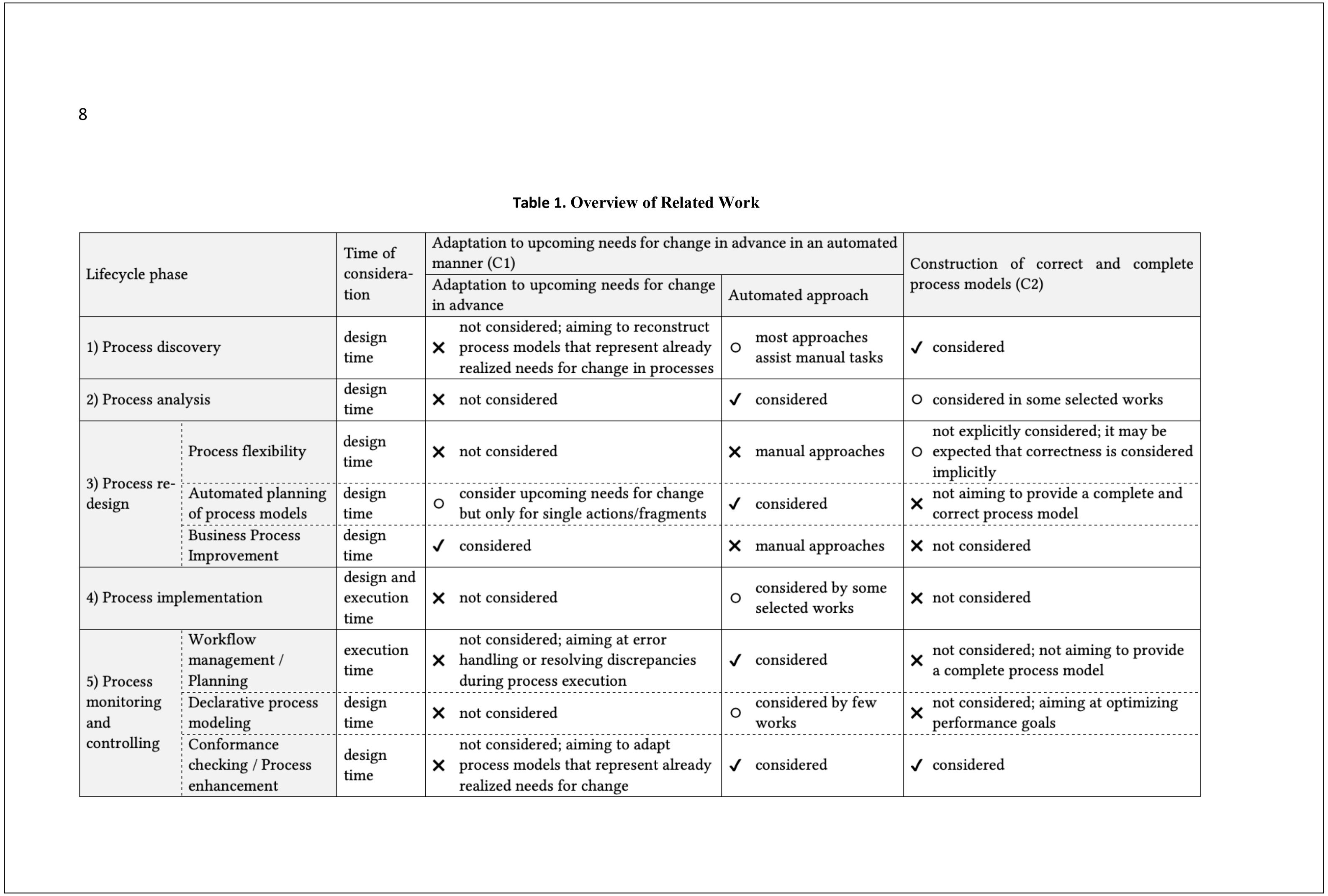


Table 1. Overview of Related Work

Lifecycle phase		Time of considera-tion	Adaptation to upcoming needs for change in advance in an automated manner (C1)		Construction of correct and complete process models (C2)
			Adaptation to upcoming needs for change in advance	Automated approach	
1) Process discovery		design time	✕ not considered; aiming to reconstruct process models that represent already realized needs for change in processes	○ most approaches assist manual tasks	✓ considered
2) Process analysis		design time	✕ not considered	✓ considered	○ considered in some selected works
3) Process re-design	Process flexibility	design time	✕ not considered	✕ manual approaches	○ not explicitly considered; it may be expected that correctness is considered implicitly
	Automated planning of process models	design time	○ consider upcoming needs for change but only for single actions/fragments	✓ considered	✕ not aiming to provide a complete and correct process model
	Business Process Improvement	design time	✓ considered	✕ manual approaches	✕ not considered
4) Process implementation		design and execution time	✕ not considered	○ considered by some selected works	✕ not considered
5) Process monitoring and controlling	Workflow management / Planning	execution time	✕ not considered; aiming at error handling or resolving discrepancies during process execution	✓ considered	✕ not considered; not aiming to provide a complete process model
	Declarative process modeling	design time	✕ not considered	○ considered by few works	✕ not considered; aiming at optimizing performance goals
	Conformance checking / Process enhancement	design time	✕ not considered; aiming to adapt process models that represent already realized needs for change	✓ considered	✓ considered

requirements and new legal regulations and therefore needs to be adapted frequently. In a first step, the required material needs to be ordered (action "Order material") if there is no material in stock. Thereafter, a circuit board is prefabricated (action "Prefabricate circuit board"). Here, basically, the circuit board goes through the actions of developing, etching and stripping. In order to produce a complete product, the prefabricated circuit board subsequently needs to be assembled with other parts such as microchips and resistors (action "Assemble product"). Finally, the product is ready for sale and the process terminates.

The process starts at an initial belief state (short: initial state). *Belief states* are denoted by tables (e.g., in Figure 1, the first belief state at the top, annotated with "Initial state"). They comprise multiple pieces of information, so called *belief state tuples* which are represented by the rows in the according tables. For instance, within our running example, the belief state tuple *(product, not manufactured)* in the upmost table of the process graph in the left part of Figure 1 (annotated with "Initial state") expresses that at the beginning of the process, the product is not yet manufactured. *Actions* which lead from one belief state to another are denoted by rounded rectangles (e.g., the action "Order material"). Actions contain *preconditions* (denoted by *pre(a)*) and *effects* (denoted by *eff(a)*). Preconditions (including inputs) denote everything an action needs to be applied, whereas *effects* (including outputs) denote everything an action provides, deallocates or alters after it was applied. The process ends at one to possibly many defined belief states meeting a goal state (i.e., the goal of the process is achieved). For example, in the belief state at the very bottom in the left part of Figure 1, a belief state meeting a goal state is reached because the product is manufactured which represents the defined goal state *(product, manufactured)*, denoted in italics.

The essential notions are presented formally in the following Definitions 1 to 5. Hereby, we follow common ways to represent a planning domain [Ghallab et al. 2004, 2016] within automated planning [Bertoli et al. 2006; Henneberger et al. 2008; Heinrich and Schön 2015, 2016]. We thus ensure compatibility with existing works.

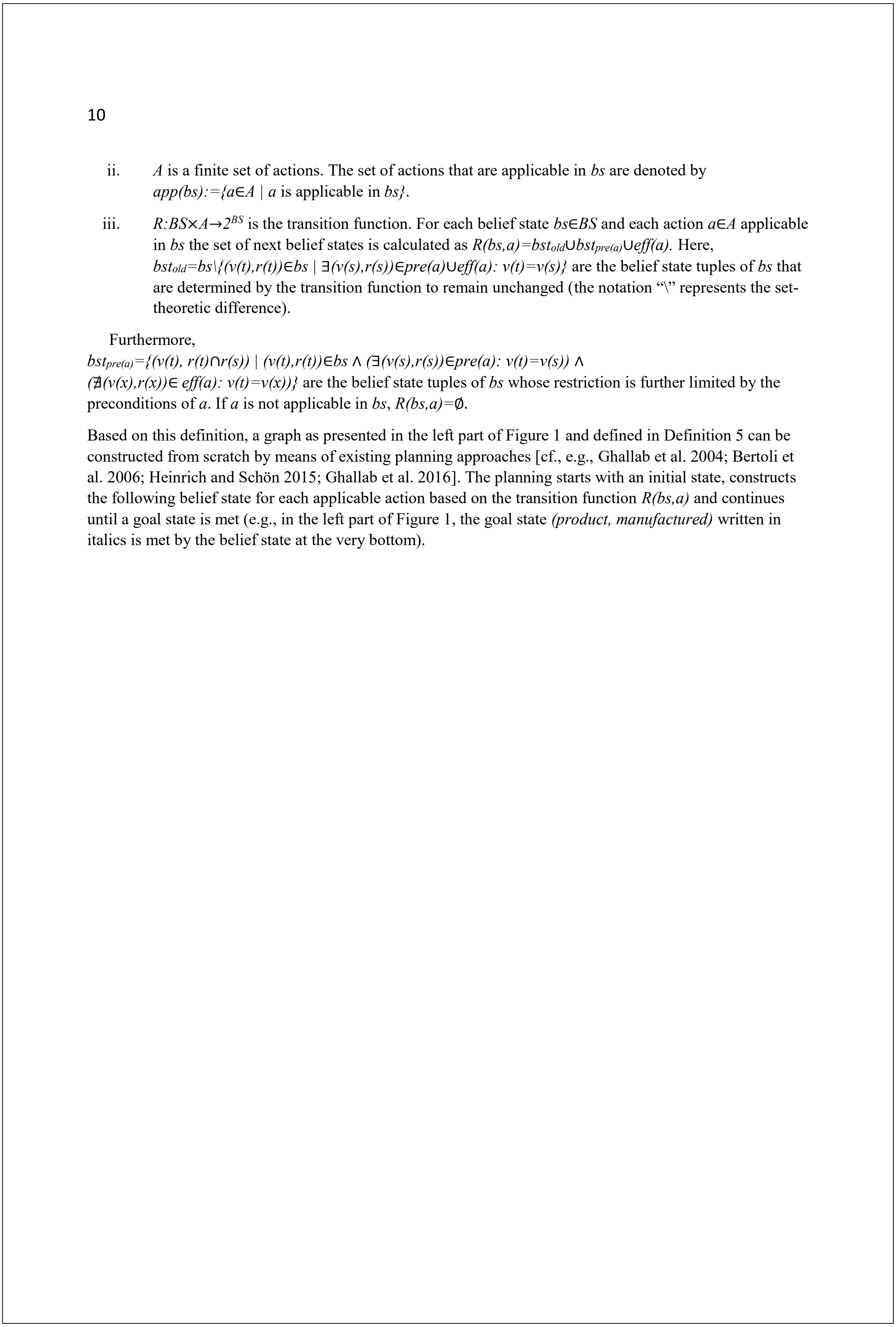
Definition 1. (belief state tuple). A *belief state tuple p* is a tuple consisting of a belief state variable $v(p)$ and a subset $r(p)$ of its domain $dom(p)$, which we will write as $p:=(v(p),r(p))$. The domain $dom(p)$ specifies which values can generally be assigned to $v(p)$ and can for instance represent a data type such as integer or a finite set. The set $r(p)\subseteq dom(p)$ is called the restriction of $v(p)$ and contains the values that can be assigned to $v(p)$ in this specific belief state tuple p. Let $BST=\{p_1,\dots, p_n\}$ be a finite set of belief state tuples.

Definition 2. (*action*). An *action a* is a triple consisting of the action name and two sets, which we write as $a:=(name(a),pre(a),eff(a))$. The set $pre(a)\subseteq BST$ are the *preconditions* of a and the set $eff(a)\subseteq BST$ are the *effects* of a. An action a is *applicable* in a belief state bs iff $\forall w\in pre(a)\ \exists u\in bs\colon v(w)=v(u) \land r(w)\cap r(u) \neq \emptyset$. In other words, a is applicable in bs iff all belief state variables in $pre(a)$ also exist in bs and the respective restrictions of the belief state variables intersect.

Belief state tuples and actions are used in the definition of a nondeterministic belief state-transition system presented in the following. The graph in the left part of Figure 1 is based on such an underlying nondeterministic belief-state transition system. Here, the initial state contains the two belief state tuples *(material, {not in stock})* and *(product, {not manufactured})* with *material* and *product* being the belief state variables and *{not in stock}* and *{not manufactured}* being their restrictions. A nondeterministic belief state-transition system is defined in terms of its belief states, its actions and a transition function that describes how an action leads from one belief state to possibly many belief states [Ghallab et al. 2004; Bertoli et al. 2006; Ghallab et al. 2016].

Definition 3. (nondeterministic belief state-transition system). A nondeterministic belief state-transition system is a tuple $\sum=(BS,A,R)$, where

i. $BS\subseteq 2^{BST}$ is a finite set of *belief states*. A *belief state* $bs\in BS$ is a subset of BST, containing every belief state variable one time at the most.



ii. A is a finite set of actions. The set of actions that are applicable in bs are denoted by $app(bs){:=}\{a{\in}A \mid a$ is applicable in $bs\}$.

iii. $R{:}BS{\times}A{\rightarrow}2^{BS}$ is the transition function. For each belief state $bs{\in}BS$ and each action $a{\in}A$ applicable in bs the set of next belief states is calculated as $R(bs,a){=}bst_{old}{\cup}bst_{pre(a)}{\cup}eff(a)$. Here, $bst_{old}{=}bs\backslash\{(v(t),r(t)){\in}bs \mid \exists(v(s),r(s)){\in}pre(a){\cup}eff(a){:}\ v(t){=}v(s)\}$ are the belief state tuples of bs that are determined by the transition function to remain unchanged (the notation "\" represents the set-theoretic difference).

Furthermore, $bst_{pre(a)}{=}\{(v(t),\ r(t){\cap}r(s)) \mid (v(t),r(t)){\in}bs \wedge (\exists(v(s),r(s)){\in}pre(a){:}\ v(t){=}v(s)) \wedge (\nexists(v(x),r(x)){\in}\ eff(a){:}\ v(t){=}v(x))\}$ are the belief state tuples of bs whose restriction is further limited by the preconditions of a. If a is not applicable in bs, $R(bs,a){=}\emptyset$.

Based on this definition, a graph as presented in the left part of Figure 1 and defined in Definition 5 can be constructed from scratch by means of existing planning approaches [cf., e.g., Ghallab et al. 2004; Bertoli et al. 2006; Heinrich and Schön 2015; Ghallab et al. 2016]. The planning starts with an initial state, constructs the following belief state for each applicable action based on the transition function $R(bs,a)$ and continues until a goal state is met (e.g., in the left part of Figure 1, the goal state *(product, manufactured)* written in italics is met by the belief state at the very bottom).

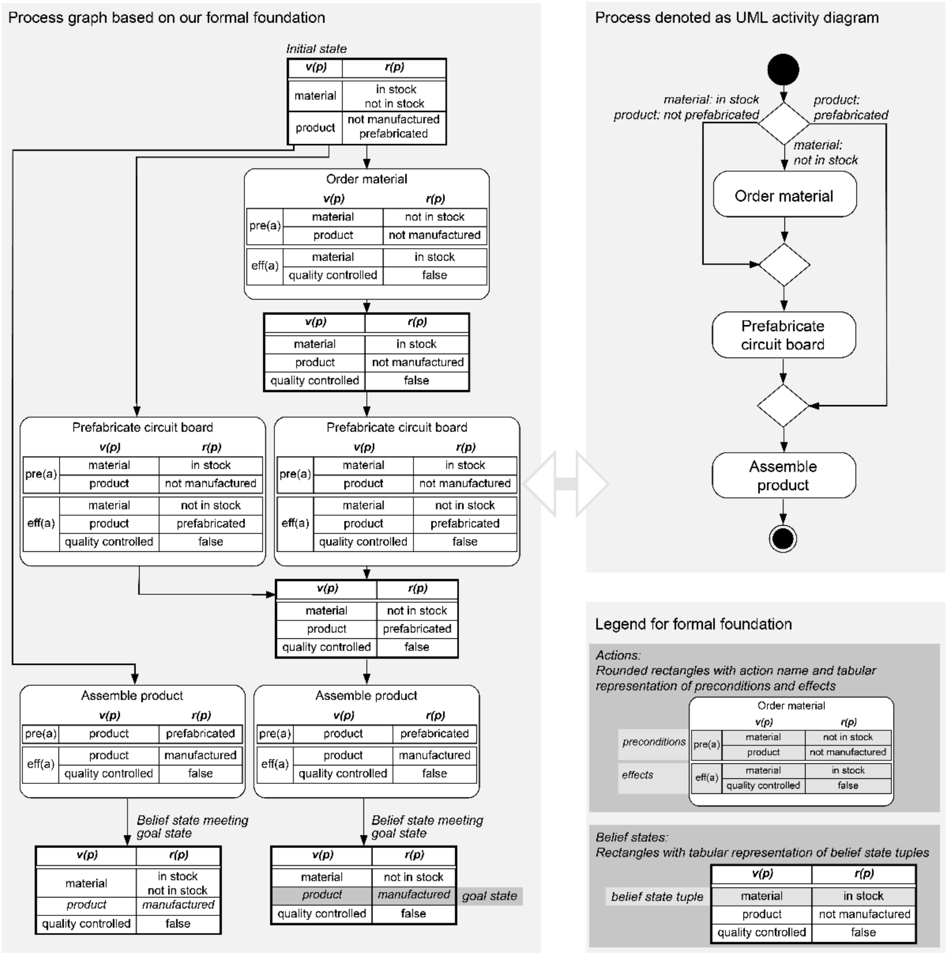

Process graph based on our formal foundation
Initial state
v(p)
r(p)
material
in stock
not in stock
product
not manufactured
prefabricated
Order material
pre(a)
eff(a)
not manufactured
quality controlled
false
Prefabricate circuit board
prefabricated
Assemble product
manufactured
Belief state meeting goal state
goal state
Process denoted as UML activity diagram
material: in stock
product: not prefabricated
product: prefabricated
material: not in stock
Order material
Prefabricate circuit board
Assemble product
Legend for formal foundation
Actions:
Rounded rectangles with action name and tabular representation of preconditions and effects
preconditions
effects
Belief states:
Rectangles with tabular representation of belief state tuples
belief state tuple

Fig. 1. Process Graph and UML Activity Diagram of the Simplified Manufacturing Process.

Definition 4. (*goal state*). A *goal state* is a subset of *BST*, containing every belief state variable one time at the most, which represents a termination criterion for the process. If a belief state *bs* fulfills the termination criterion represented by a goal state *goal* (i.e., $\forall p \in goal\colon \exists p' \in bs,\ v(p)=v(p'),\ r(p') \subset r(p)$), we denote *bs* as *meeting goal*.

Definition 5. (*process graph*). A *process graph* is a bipartite, directed, finite graph $G=(N,E)$ with the set of nodes *N* and the set of edges *E*. The set of nodes *N* consists of two partitions: The set of action nodes *A* and the set of belief state nodes *BS*. Each node $bs \in BS$ represents one distinct belief state in the process graph. Each action node $a \in A$ represents an action in the process graph. The process graph starts with one initial state $bs_{init} \in BS$ and ends with one to possibly many belief states $bs_{goal,j} \in BS$ meeting a goal state. A (finite) sequence of states and actions $(bs_{init}, a_1, bs_2, \ldots, bs_k)$ starting with the initial state is called a *path*. A path $(bs_{init}, a_1, bs_2, \ldots, bs_k)$ is called a *feasible path* if the following three additional conditions apply:

i. bs_k meets a goal state

ii. $bs_{init},\ldots,bs_{k-1}$ do not meet any goal state

iii. $a_1 \in app(bs_{init}), bs_2=R(bs_{init},a_1),\ldots, a_{k-1} \in app(bs_{k-1}), bs_k=R(bs_{k-1},a_{k-1})$.

Within this paper, we present an approach to adapt process graphs as described in Definition 5. The result of this adaptation is again a process graph based on the definitions presented above. Hence, existing works for the automated construction of control flow patterns [van der Aalst et al. 2003] such as exclusive choice based on process graphs [e.g., Meyer and Weske 2006; Heinrich and Schön 2016; Heinrich et al. 2019] can be used as usual to construct process models containing control flow patterns. Thus, it is not necessary to address how to consider control flow patterns in this paper. In addition, for reasons of readability and conciseness, we do not explicitly state how to cope with process graphs containing cycles in the following. However, process graphs containing arbitrary cycles can be adapted as follows: First, (sub) paths within cycles are identified. Then, each of the (sub) paths within cycles and outside of the cycles is treated using the approach presented in Section 4, setting the first state of the (sub) path as initial state and its last state as goal state. The adapted graphs are then combined by merging in each case the identical initial states respective goal states and result in an adapted process graph containing cycles.

Our approach primarily contributes to the research strand of automated planning of process models. Here, belief state-transition systems including transitions and actions with preconditions and effects equaling the formal definitions used in this paper are widespread [e.g., Heinrich et al. 2018; Heinrich et al. 2019]. However, these or similar concepts are known in related fields as well. BPM toolsets such as the Camunda Platform (comprising Camunda Modeler) heavily rely on so called input and output variables of actions due to their focus on workflow and process automation. Additionally, this information could (at least partly) be extracted from traditional process modeling tools such as the ARIS toolset. ARIS in particular offers an XML interface that allows to export actions, including their inputs and outputs. These exported actions can be used for the set of actions *A* together with newly defined actions (if necessary). Additionally, well-known process modeling languages incorporate concepts that are similar to belief state variables. For instance, data objects defined by the process modeling language BPMN or events in EPC are such similar representations (e.g., the creation of a particular document “check list” is transferred to the belief state variable “check list” being set to “created”; and if necessary contents of this document can also be represented by several (sub) belief state variables). Even more, transitions and places being part of process modeling languages such as Petri Nets can be transferred to actions and belief state tuples as used in our planning domain as well. A place containing tokens with certain features (e.g., token color) could be seen as belief state variables with particular values. A transition can fire if each of its input places contains a token. This is very similar to an action being applicable (cf. Definition 3) as each belief state tuple of the preconditions of an action must be fulfilled. Our formal foundations are also similar to such used in other research areas like the conceptualization of (web) services or notations and language elements used in process mining [Bashari et al. 2018; Fan et al. 2018; Noura and Gaedke 2019]. Here, many approaches incorporate AI planning and strongly rely on states and actions containing preconditions and effects, while others have developed approaches to derive this information [Kindler et al. 2006; van der Aalst et al. 2010; Mannhardt et al. 2018].

To sum up, the concepts and foundations used across different research strands to represent processes are similar and applicable to process graphs as discussed above. This underlines the aforementioned goal of representing process models independent of a particular process modeling language which enables a broader applicability and transferability compared to focusing on a single modeling language.

4 DESIGN OF OUR APPROACH

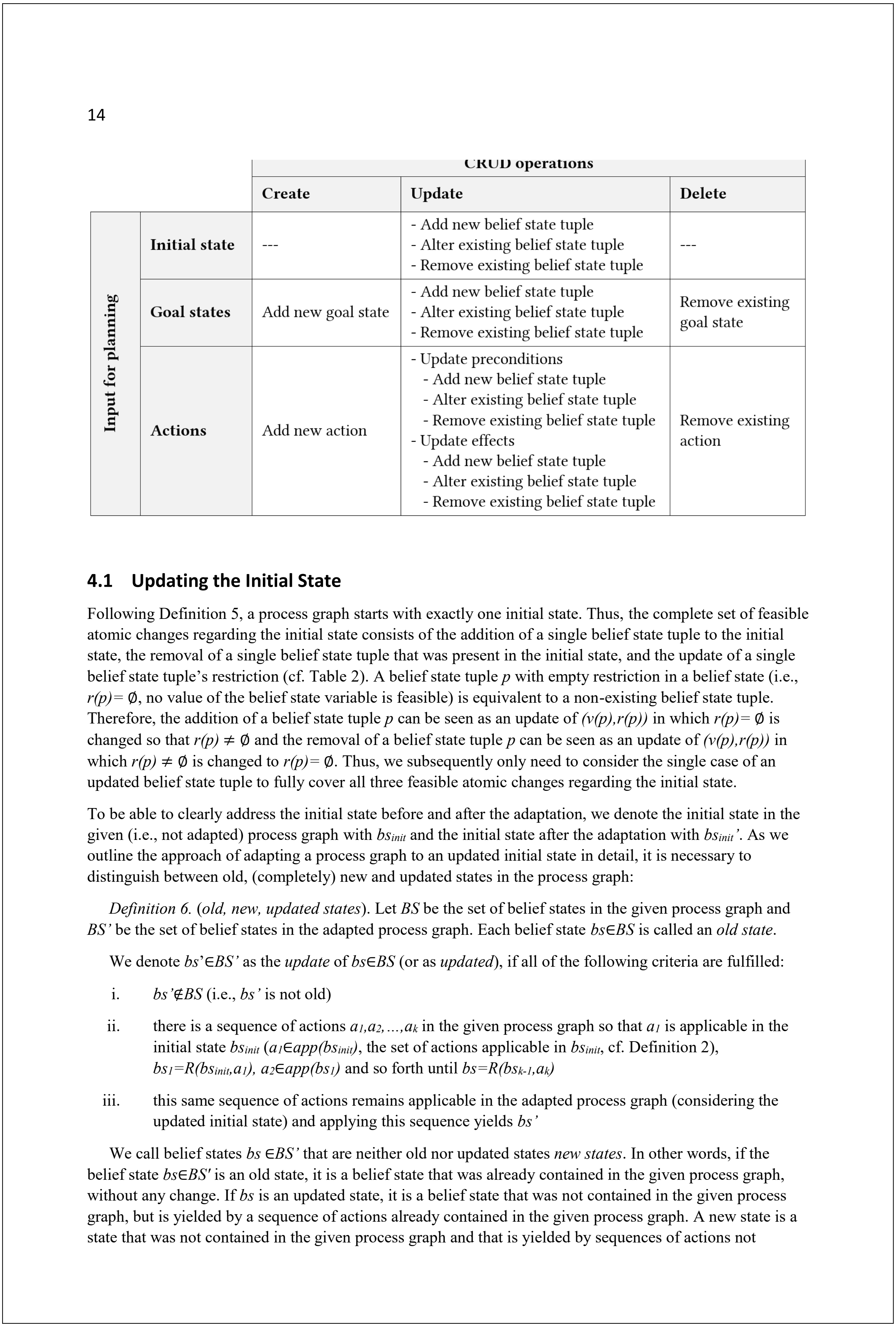
The characteristics of needs for change are manifold. They may vary not only by their quantity and scale (e.g., one simple vs. many complex needs for changes) but also by their consequences for the process graph (e.g., a simple need for change requires the adaptation of the entire process graph due to many dependencies between actions). To cope with this challenge of inhomogeneous needs for change, we at first propose to define so-called *atomic changes* (in the sense of most basic forms like building blocks), which then can be composed to address (probably complex) needs for change and their consequences. More precisely, identifying a well-founded set of *all feasible* atomic changes allows to design an approach for the adaptation of process models to various needs for change by representing them as a well-founded composition of these predefined atomic changes. Thereby the set of atomic changes has to be complete (i.e., all possible needs for change can be addressed) and has to contain only changes which cannot be decomposed further (i.e., atomic changes). These atomic changes are feasible because adapting to them transforms an existing (correct and complete) process graph into a new, adapted process graph that is correct and complete. Furthermore, an arbitrary composition of adaptations to feasible atomic changes results in a feasible adaptation as well, which means, any composition leads to a correct and complete adapted process graph. The idea behind atomic changes (building blocks) can be traced back to the paradigm of composition / decomposition, for instance, in the field of service orientation [Parnas 1972; Bucchiarone et al. 2020; Megargel et al. 2020] as well as to the idea of well-founded transaction concepts in database management systems. In particular, to identify atomic changes, we align our research to the well-known CRUD operations and apply the underlying thought to our problem setting. CRUD operations [cf. Martin 1983] consist of the four elemental, low-level (atomic) operations “create”, “read”, “update” and “delete” that cover all possible ways of accessing and altering data. Similar to our discussion, they cannot be decomposed further, provide a complete set of basic data manipulation operations and can be composed in a well-founded manner to address complex data manipulations.

As given in Definition 5, a process graph consists of belief states (amongst them one initial state and one to possibly many belief states meeting a goal state) and actions. When constructing a process graph by an existing automated planning approach, the initial state, goal states and the set of actions are used as input. All other belief states of the process graph are constructed during planning. Thus, it is not feasible that these belief states are *directly* adapted due to needs for change without creating inconsistencies. Therefore, by using our formal foundation, every need for change to a process graph is reflected in a change to this input. Consequently, only atomic changes to this input need to be considered for the adaptation of process graphs. The initial state, goal states and preconditions and effects of actions are all represented by sets of belief state tuples. When changing these sets, we will consider atomic changes (which cannot be decomposed further) to single belief state tuples in the following as changes to multiple belief state tuples can be represented by a sequence of atomic changes.

Combining the aforementioned CRUD operations with the input discussed above results in the complete set of feasible atomic changes that cannot be decomposed further, as presented in Table 2. As “read” does not represent a change in our context of adapting process models, this operation is not taken into account in Table 2. Since exactly one initial state is used as input for planning, each change to the initial state can be represented by an update of it. For goal states and actions as well as for each belief state tuple, however, the operations “create”, “update” and “delete” can be applied.

As already mentioned, a composition of these predefined, feasible atomic changes can be used to address any (complex) adaptation of a process graph. A more detailed discussion of this fact is presented in Section 4.4.

In the following we propose an approach that copes with every feasible atomic change. To do so, we identify potential consequences for the process graph (e.g., actions becoming applicable in an updated initial state of the graph) resulting from each feasible atomic change. Thereby, we do not merely reduce each atomic change to a planning problem solvable by existing techniques for the automated planning of process models [e.g., Henneberger et al. 2008; Hoffmann et al. 2012; Lin et al. 2012; Heinrich et al. 2015]. Instead, we enhance these techniques to address each individual atomic change compared to “planning from scratch”. To do this, we determine where parts of the existing process graph can be reused or where new belief states and actions have to be planned. In addition, we incorporate knowledge about the applicability of actions in the existing process graph to reduce the effort of verifying the applicability of these actions to changed belief states. A pseudocode of our approach was created and is provided at https://epub.uni-regensburg.de/43489/1/Adaption_Online_Material.pdf.



		CRUD operations		
		Create	**Update**	**Delete**
Input for planning	**Initial state**	---	- Add new belief state tuple - Alter existing belief state tuple - Remove existing belief state tuple	---
	Goal states	Add new goal state	- Add new belief state tuple - Alter existing belief state tuple - Remove existing belief state tuple	Remove existing goal state
	Actions	Add new action	- Update preconditions - Add new belief state tuple - Alter existing belief state tuple - Remove existing belief state tuple - Update effects - Add new belief state tuple - Alter existing belief state tuple - Remove existing belief state tuple	Remove existing action

4.1 Updating the Initial State

Following Definition 5, a process graph starts with exactly one initial state. Thus, the complete set of feasible atomic changes regarding the initial state consists of the addition of a single belief state tuple to the initial state, the removal of a single belief state tuple that was present in the initial state, and the update of a single belief state tuple's restriction (cf. Table 2). A belief state tuple *p* with empty restriction in a belief state (i.e., $r(p) = \emptyset$, no value of the belief state variable is feasible) is equivalent to a non-existing belief state tuple. Therefore, the addition of a belief state tuple *p* can be seen as an update of $(v(p),r(p))$ in which $r(p) = \emptyset$ is changed so that $r(p) \neq \emptyset$ and the removal of a belief state tuple *p* can be seen as an update of $(v(p),r(p))$ in which $r(p) \neq \emptyset$ is changed to $r(p) = \emptyset$. Thus, we subsequently only need to consider the single case of an updated belief state tuple to fully cover all three feasible atomic changes regarding the initial state.

To be able to clearly address the initial state before and after the adaptation, we denote the initial state in the given (i.e., not adapted) process graph with bs_{init} and the initial state after the adaptation with bs_{init}'. As we outline the approach of adapting a process graph to an updated initial state in detail, it is necessary to distinguish between old, (completely) new and updated states in the process graph:

Definition 6. (*old, new, updated states*). Let *BS* be the set of belief states in the given process graph and *BS'* be the set of belief states in the adapted process graph. Each belief state $bs \in BS$ is called an *old state*.

We denote $bs' \in BS'$ as the *update* of $bs \in BS$ (or as *updated*), if all of the following criteria are fulfilled:

i. $bs' \notin BS$ (i.e., *bs'* is not old)

ii. there is a sequence of actions $a_1, a_2, \ldots, a_k$ in the given process graph so that a_1 is applicable in the initial state bs_{init} ($a_1 \in app(bs_{init})$, the set of actions applicable in bs_{init}, cf. Definition 2), $bs_1 = R(bs_{init}, a_1)$, $a_2 \in app(bs_1)$ and so forth until $bs = R(bs_{k-1}, a_k)$

iii. this same sequence of actions remains applicable in the adapted process graph (considering the updated initial state) and applying this sequence yields *bs'*

We call belief states $bs \in BS'$ that are neither old nor updated states *new states*. In other words, if the belief state $bs \in BS'$ is an old state, it is a belief state that was already contained in the given process graph, without any change. If *bs* is an updated state, it is a belief state that was not contained in the given process graph, but is yielded by a sequence of actions already contained in the given process graph. A new state is a state that was not contained in the given process graph and that is yielded by sequences of actions not contained in the given process graph.

Now we will identify potential consequences resulting from updating the initial state bs_{init}. Updating a belief state tuple p of bs_{init} can impact whether an action a is applicable in bs_{init}' if a belief state tuple p' is contained in the preconditions of a such that $v(p')=v(p)$ (cf. Definition 2). Otherwise, the belief state tuple p is not relevant in order to determine whether a is applicable. Hence, the sets $app(bs_{init})$ and $app(bs_{init}')$ can only differ in actions containing a belief state tuple p' with $v(p)=v(p')$ in their preconditions. Thus, for the set of actions $\{a \in app(bs_{init}) | \nexists p' \in pre(a): v(p')=v(p)\}$, the applicability regarding bs_{init}' does not need to be checked as these actions are unaffected and thus remain applicable. Actions in the set $\{a \in A | \exists p' \in pre(a): v(p')=v(p)\}$, however, need to be checked for potential applicability in bs_{init}'. The actions not contained in $app(bs_{init}')$ are not planned at this point in the adapted process graph.

For each action a that is applicable in both bs_{init} and bs_{init}' (i.e., $a \in app(bs_{init}') \cap app(bs_{init})$) and hence "retained" its applicability we can use $bs=R(bs_{init},a)$ from the given graph, which helps us to determine $bs'=R(bs_{init}',a)$ as we only need to apply the transition function R (cf. Definition 3) with respect to p and transfer these effects to bs. If bs' was contained in the given process graph and thus is an old state, we can retain the whole subgraph starting with bs' as the actions that can be applied in this belief state are known from the given process graph and do not differ since both the belief state and the actions did not change. This is in accordance to existing techniques for the automated planning of process models where the traversal of a previously known state terminates planning. Otherwise (i.e., if bs' is not an old state) bs' is the update of bs (cf. Definition 6). In this case, the updated belief state can now either meet a goal state, which completes the path, or we need to continue by treating bs' as we currently handle bs_{init}'.

The set $app(bs_{init}')$, however, can also contain actions that were not applicable in bs_{init}. For each such action $a \in app(bs_{init}')$ with $a \notin app(bs_{init})$, the transition function R needs to be applied entirely (i.e., not only with respect to the updated belief state tuple p) in order to obtain the belief state $bs=R(bs_{init}',a)$. If bs meets a goal state, the path is completed, else bs is either old, updated (from a hitherto feasible path) or new. In the first case, we retain the whole subgraph starting with bs from the given process graph. If, however, bs is a new state, we have to apply the transition function R entirely: We compute $app(bs)$ and, for each $a \in app(bs)$, the belief state $R(bs,a)$ following bs. Again, these belief states have to be checked in regard to being old, updated or new. Updated states are handled in the same way as bs_{init}'. We proceed iteratively in this manner with every upcoming state depending on its classification regarding Definition 6.

Within the example (cf. Figure 2; parts influenced by the adaptation are black, not influenced parts are grey), a new external supplier that meets the service level requirements is acquired as a business partner. This external supplier is able to provide prefabricated circuit boards. Hence, the fact that now an appropriate external supplier is available is denoted in terms of the belief state tuple *(external supplier, {available})*, which therefore is added in the initial state (bold). By means of this change, the action "Order prefabricated circuit board" (retrieved from the set of actions A, cf. Definition 3), which requires this particular belief state tuple, becomes applicable and thus is planned in the adapted initial state. After this action, a new belief state is created in which the action "Assemble product" is applicable, which in turn leads to the goal state. Thus, as result of the adaptation, a new feasible path (denoted by means of bold arrows and bold-bordered actions and belief states) is constructed.

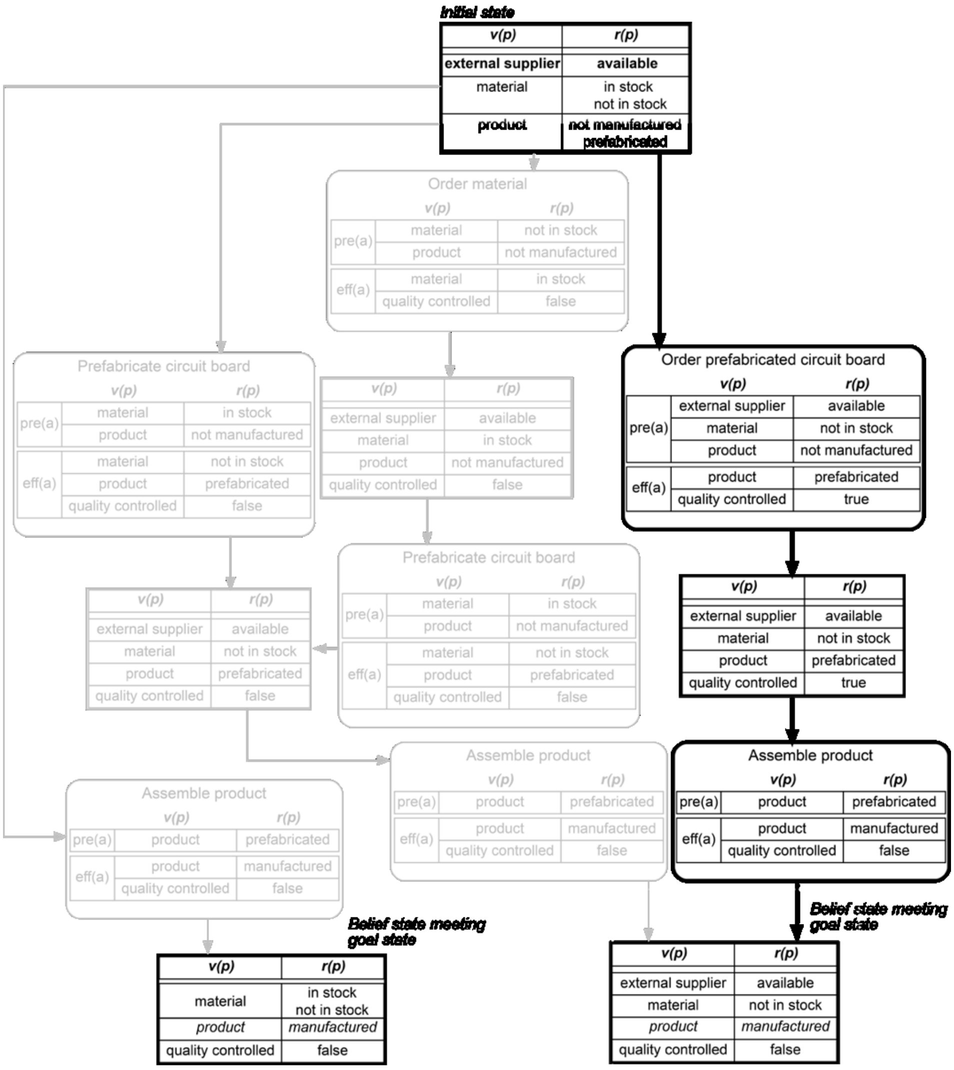


Fig. 2. Process Graph after the Adaptation resulting from updating the Initial State.

4.2 Changing (the Set of) Goal States

A process graph contains one to possibly many goal states (cf. Definition 5). In alignment with the CRUD functions, the complete set of feasible atomic changes regarding the set of goal states comprises "adding a goal state", "removing a goal state" and "updating a goal state" (cf. Table 2).

4.2.1 Adding a Goal State. Denoting the set of all goal states in the given process graph with *GOALS*, the addition of a new goal state $goal \notin GOALS$ with $GOALS' = GOALS \cup \{goal\}$ could, on the one hand, result in new feasible paths, which lead to this new goal state. Such new feasible paths have not been feasible in the given process graph and thus need to be newly constructed. On the other hand, as goal states serve as termination criteria, this new goal state could imply feasible paths in the given process graph being "shortened" so that for a given path $bs_{init}, a_1, bs_1, a_2, \ldots, bs_k$ there exists $j<k$ with bs_j meeting *goal*.

To determine these consequences, we traverse the paths of the given process graph and their belief states (except for the belief states meeting a goal state from *GOALS* at the end of each such feasible path), starting with the initial state. For each belief state *bs*, we need to check whether *bs* meets the new goal state (first case) or whether actions applicable in *bs* lead to the new goal state subsequently (second case). If, in the first case, the currently considered belief state *bs* meets the new goal state, we abort the traversal of this path as it ends here. In the second case, if *bs* does not meet the new goal state, we have to take into account every possible new belief state that can follow right after *bs* and start planning from each of these new belief states in order to (possibly) retrieve new feasible paths that lead to *goal*. With this in mind, we first determine all actions $a \in app(bs)$ (retrieved from the set of actions *A*, cf. Definition 3) which were not planned in *bs* in the given process graph. For each of these actions we then determine the belief state $bs'=R(bs,a)$ and continue planning from *bs'*. If, during this planning, no belief state that meets *goal* is retrieved or no further action is applicable, the planning of the current path is aborted.

4.2.2 Removing a Goal State. Removing a goal state $goal \in GOALS$ (i.e., $GOALS'=GOALS \setminus \{goal\}$) implies that a termination criterion for the process is deleted. Therefore, each path in the given process graph that ends at a belief state meeting *goal* needs to be checked whether it can be extended by an existing planning technique so that it leads to one of the remaining goal states. If no goal state can be reached from its last belief state (which formerly had met the now removed goal state *goal*), it is not considered in the adapted process graph. No other paths are affected by this change.

We therefore take into account each belief state *bs* of the given process graph that meets *goal*. Starting with each such *bs*, we try to reach one of the remaining goal states by applying the transition function *R* and computing all applicable actions and the resulting belief states. Thus, we first check each belief state *bs* that meets *goal* for the criteria of the remaining goal states. If *bs* meets the criteria of a remaining goal state, it is ensured that the paths which had ended at *goal* remain feasible in the adapted process graph. Else, the next planning step is executed: We determine all applicable actions in *bs* and construct the according resulting belief states by applying *R*. Note that as soon as there are no actions applicable in the examined belief state and thus the planning step fails, the paths which had ended at goal cannot be extended to a feasible path and are therefore not considered in the adapted process graph.

4.2.3 Updating a Goal State. We separate the case of updating a goal state *goal* into two subcases. Since goal states serve as termination criteria, we distinguish between a strengthening update (i.e., making the conditions for meeting *goal* more severe) and a weakening update (i.e., making the conditions less severe). The updated goal state will be denoted by *goal'* (and thus $GOALS'=(GOALS \setminus \{goal\}) \cup \{goal'\}$). Any other feasible change to a goal state that is not included in the following two cases can be represented as the composition of a weakening update followed by a strengthening update.

Strengthening update. Strengthening the conditions of a goal state *goal* includes the addition of a belief state tuple to *goal* as well as changes to a belief state tuple $p \in goal$ limiting its restriction, formally replacing *p* by *p'* with $v(p)=v(p'),\ r(p) \neq \emptyset \neq r(p'),\ r(p) \neq r(p')$ and $r(p') \subset r(p)$ so that $goal'=(goal \setminus \{p\}) \cup \{p'\}$. When strengthening the conditions of *goal*, the set of (world) states that meet *goal'* is a proper subset of the set of states that meet *goal*, as these criteria are more severe. Thus, we proceed in a similar way to the case of removing a goal state (cf. Section 4.2.2): We start planning for each belief state *bs* meeting *goal* and each action that can be applied in *bs*, trying to reach one of the goal states from *GOALS'*.

Looking at the running example, a new compliance directive has come into force, requiring the company to integrate quality management as a documented and controlled task in the manufacturing process. Due to the new directive, it is required that the quality assurance is documented as an inherent part of the process. Therefore, the belief state tuple *(quality controlled, {true})* is added to the goal state (bold and in italics). Thus, as seen in Figure 3, an action “External quality assurance” is now planned in the belief state meeting the old goal state in order to meet the new, adapted goal state including the new belief state tuple.

Weakening update. This case covers removing a belief state tuple from *goal* as well as changes that extend the restriction of a belief state tuple $p \in goal$ (i.e., replacing *p* by *p'* with $v(p)=v(p')$, $r(p) \neq \emptyset \neq r(p')$, $r(p) \neq r(p')$ and $r(p) \subset r(p')$). Belief states meeting the goal state *goal* canonically meet *goal'*. Additionally, there are possibly further belief states meeting *goal'* which do not meet *goal*. Therefore, we align the approach to the case of adding the goal state *goal'* (cf. Section 4.2.1): We traverse all belief states in the process graph, check whether a belief state meets *goal'*, and try to retrieve new feasible paths to *goal'* by checking whether actions applicable in the belief states lead to *goal'* subsequently. In this way, feasible paths in the existing process graph may be shortened and new feasible paths may be constructed.

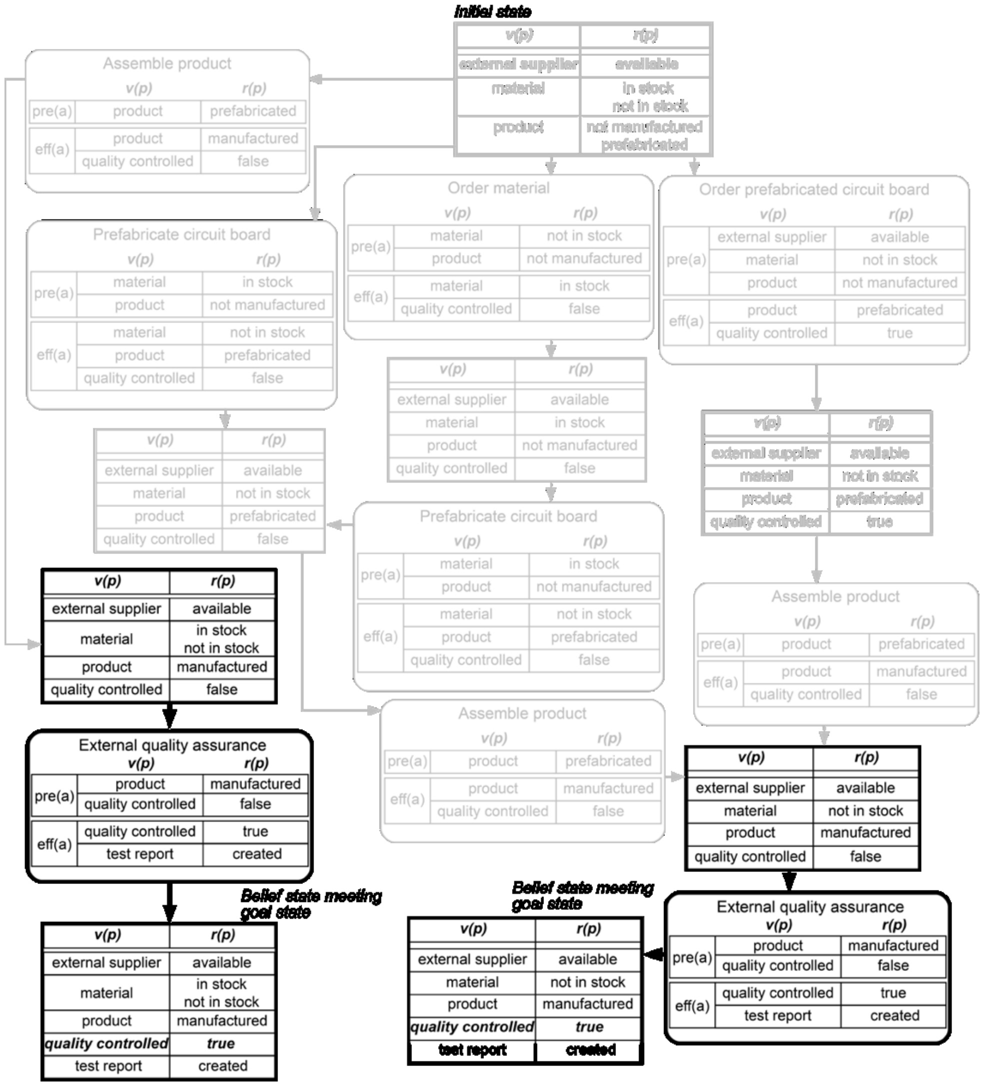

Initial state
v(p) r(p)
external supplier available
material in stock, not in stock
product not manufactured, prefabricated
Assemble product
pre(a) product prefabricated
eff(a) product manufactured; quality controlled false
Order material
pre(a) material not in stock; product not manufactured
eff(a) material in stock; quality controlled false
Order prefabricated circuit board
pre(a) external supplier available; material not in stock; product not manufactured
eff(a) product prefabricated; quality controlled true
Prefabricate circuit board
pre(a) material in stock; product not manufactured
eff(a) material not in stock; product prefabricated; quality controlled false
External quality assurance
pre(a) product manufactured; quality controlled false
eff(a) quality controlled true; test report created
Belief state meeting goal state
external supplier available
material in stock, not in stock / not in stock
product manufactured
quality controlled true
test report created

Fig. 3. Process Graph after the Adaptation due to a strengthening Update of the Goal State.

4.3 Changing (the Set of) Actions

As described in Definition 2, actions are triples consisting of the action name, the preconditions of the action and the effects of the action. According to CRUD, the complete set of feasible atomic changes regarding the set of actions *A* comprises the addition of an action to *A*, the removal of an action from *A* or the update of the preconditions or effects of an action in *A* (cf. Table 2).

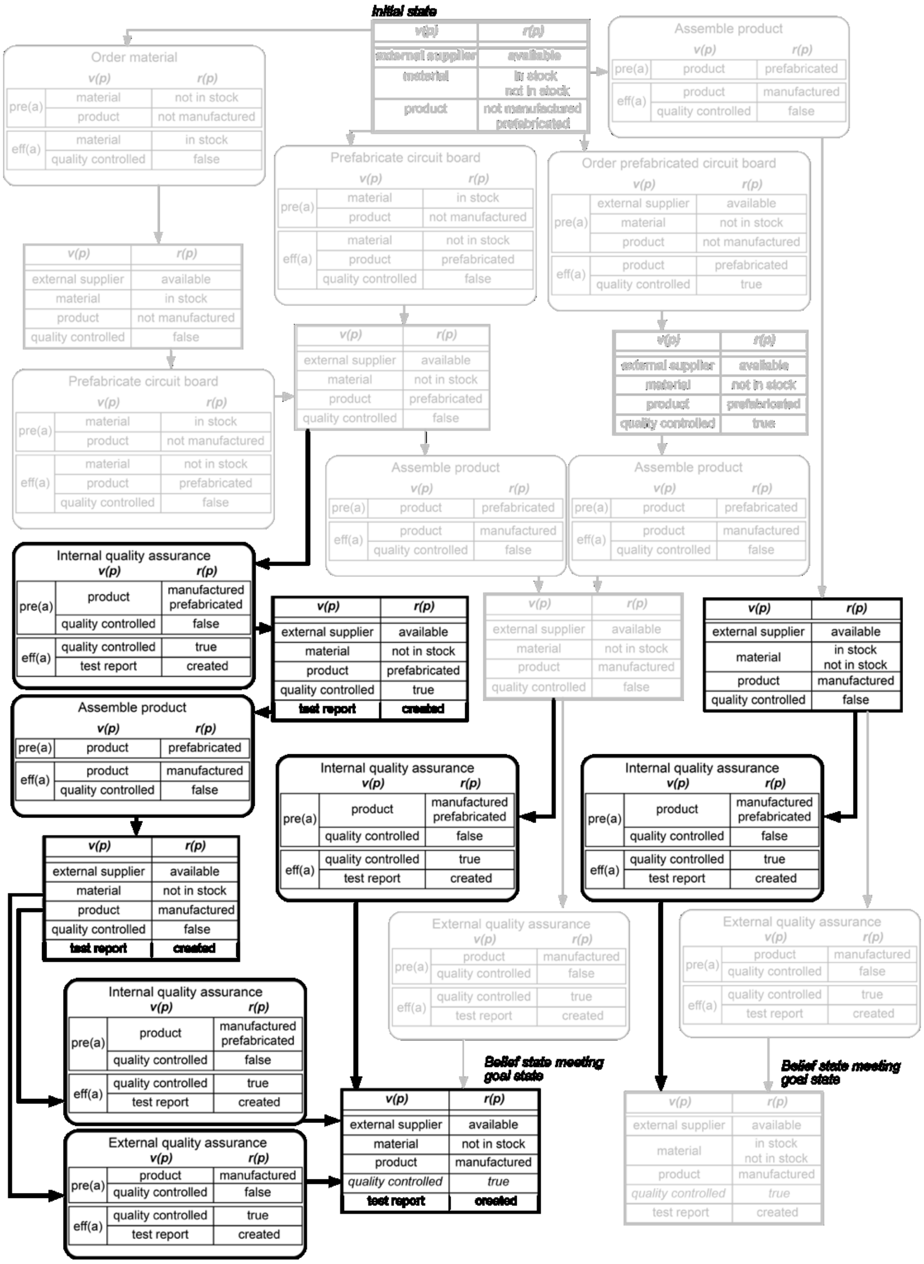


Fig. 4. Process Graph after the Adaptation due to an Added Action.

4.3.1 Adding an Action. Let a be a new action so that $A'=A\cup\{a\}$. As a might be applicable in the given process graph, we need to check whether there exists a belief state bs in the given process graph such that a is applicable in bs. In such belief states we recursively apply the transition function $R(bs,a)$. Further, there may exist paths $(bs_{init},a_1,\dots,bs_k)$ with $a_1,\dots, a_{k-1}\in A$ that have not been feasible paths in the given process graph and with a being applicable in bs_k. In such belief states we also start planning by applying the transition function $R(bs_k,a)$. Thereby, we possibly retrieve new feasible paths leading to a goal state.

Within the running example, the company decides to establish an own, internal quality assurance. This, in difference to the external quality assurance contractor, is able to check the assembled product as well as (optionally) the internally prefabricated circuit board. As we see in Figure 4, an action "Internal quality assurance" (bold) is added to the process graph appropriately throughout the whole process.

4.3.2 Removing an Action. When removing an action a from A so that $A'=A\backslash\{a\}$, there can be no new feasible paths leading to a goal state. Further, each path in the given process graph containing a is not feasible in the adapted process graph and hence not retained. The paths not containing a are not affected at all and are retained.

4.3.3 Updating an Action. When updating an action a to a' we need to consider the case of updating the preconditions as well as the case of updating the effects of a. Updating the preconditions can be separated into the two subcases of strengthening and weakening updates since any update that is not covered by one of these two cases can be treated as performing an weakening update, followed by a strengthening update.

Strengthening update of the preconditions. When strengthening the preconditions of an action a (i.e., adding a belief state tuple p to $pre(a)$ or updating p to p' so that $v(p)=v(p'), r(p)\neq \emptyset \neq r(p'), r(p)\neq r(p'), r(p')\subset r(p)$ and $pre(a')=(pre(a)\backslash\{p\})\cup\{p'\})$, only a subset of the belief states of the given process graph in which a was applicable also fulfills the requirements for the applicability of a'. Hence, we need to check for each belief state bs in which a was applicable whether a' is still applicable. If this is not the case, we do not consider the paths containing a' in the adapted process graph (cf. case of removing an action, Section 4.3.2). On the other hand, if a' is still applicable in bs, the belief state bs_1 that results from $R(bs,a)$ may differ from the belief state bs_1' resulting from $R(bs,a')$ (cf. Definition 2). In this case, it is possible that the sets $app(bs_1)$ and $app(bs_1')$ do not coincide. We then proceed analogously as we did when treating the case of updating the initial state (cf. Section 4.1) with bs_1' taking the role of the updated state to bs_1.

Weakening update of the preconditions. When weakening the preconditions of an action a (i.e., removing a belief state tuple from $pre(a)$ or updating p to p' so that $v(p)=v(p'), r(p)\neq \emptyset \neq r(p'), r(p)\neq r(p'), r(p)\subset r(p')$ and $pre(a')=(pre(a)\backslash\{p\})\cup\{p'\})$ it is possible that a' becomes applicable in additional belief states in which a has not been applicable. We therefore check each belief state bs of the given process graph with $a\notin app(bs)$ in regard to $a'\in app(bs)$. If, indeed, $a'\in app(bs)$ holds, we apply a planning approach in accordance to the case of adding a new action (cf. Section 4.3.1). Further, there may exist paths $(bs_{init},a_1,\dots,bs_k)$ with $a_1,\dots, a_{k-1}\in A$ that have not been feasible paths in the given process graph and with $a\notin app(bs_k)$, but $a'\in app(bs_k)$. In such belief states we also start planning by applying the transition function $R(bs_k,a')$. Thereby, we may retrieve new feasible paths leading to a goal state. Additionally, the same situation as in the preceding paragraph ($R(bs,a)\neq R(bs,a')$) can arise and is handled in the same manner as above (cf. Section 4.1).

Updating the effects. Finally, when updating the effects of an action a with respect to a single belief state tuple, we consider each belief state bs of the given process graph in which a is applicable. Due to the changed effects, once again, we may encounter the situation in which $R(bs,a)\neq R(bs,a')$ holds, which is handled as above (cf. Section 4.1).

4.4 Summary of the Approach

In the Sections 4.1-4.3 it was shown how to adapt a process graph to each feasible atomic change (cf. Table 2). Table 3 summarizes the main enhancements with regard to existing methods from automated planning which do not reuse any results from previous planning runs.

Table 3. Enhancements over Existing Planning Approaches

		Main enhancements		
		Reuse of applicability information	Reuse of existing subgraphs	Planning based on existing process graph
Type of atomic change (cf. Table 2)	Update initial state	✓	✓	✕
	Add goal state	✕	✕	✓
	Update goal state	✕	✕	✓
	Remove goal state	✕	✕	✓
	Add action	✕	✕	✓
	Update action	✓	✓	✓
	Remove action	✕	✕	✓

As all adaptations can be realized as a sequence of feasible atomic changes, a full-featured approach for the adaptation of process models has thus been developed. We discuss this by means of our running example:

In order to enter new markets, a new manufacturing facility is built by the company. In this new facility the manufacturing process (cf. Figure 4) is planned to be applied, but needs to be adapted. To reach a broad market coverage, a second production line for the prefabrication of circuit boards consisting of two machines has to be added. Additionally, analyses show that a new packaging is needed for this market and hence, product packing is planned to be included into the process. As the external quality assurance contractor does not operate in this market, it is planned to exclusively handle quality assurance at the facility. Lastly, local regulation requires mandatory quality assurance for prefabricated circuit boards.

The atomic changes can be directly specified based on this description. First, a second production line is incorporated by adding the actions “Prefabricate circuit board on machine 1” and “Prefabricate circuit board on machine 2”. These actions have preconditions and effects similar to the action “Prefabricate circuit board” with the only difference being the belief state tuple *(product, {in prefabrication}),* which is needed as these actions have to be put in sequence. Second, product packing is enabled by adding the action “Packing product” and updating the goal state to contain the belief state tuple *(product, {packed})*. With these feasible atomic changes, the ability as well as the necessity for a product to be packed is given. Third, to meet the business changes regarding quality assurance, the action “External quality assurance” is deleted. Additionally, the belief state tuple *(quality controlled, {true})* is added to the preconditions of the action “Assemble product” to comply with legal requirements (i.e., prefabricated circuit boards cannot be processed without having their quality checked). We first adapted the process graph using our approach. Thereafter, a process model comprising control flow patterns has been constructed and is shown in Figure 5 in terms of an UML activity diagram. Please note that the first XOR after the initial node, for instance, allows to decide whether to fabricate the circuit board internally or to order it externally, based on Quality-of-Service parameters at runtime. Thus, factors such as resource allocation of machines, delivery times or current prices of externally ordered circuit boards can be taken into consideration.

Furthermore, we did not explicitly address the abstraction / granularity level of process models [Milani et al.

2016], as this is a matter of defining the planning domain and, in particular, the specified belief state tuples and actions. Sub-processes, which are used to split large processes into smaller parts, start at a corresponding sub-initial state and end at sub-goal states. Process graphs depicting such sub-processes may thus be adapted by means of our approach. On the other hand, abstract / granular process graphs [Turetken et al. 2020] comprising processes in their entirety may be adapted as well. Depending on the desired abstraction / granularity, belief state tuples and actions may be defined more or less granular. For instance, within our running example the action "prefabricate circuit board" actually consists of several actions such as "equipping the circuit board" and "soldering" in reality. However, for reasons for simplicity and understandability, they are only considered in an abstract manner in this paper.

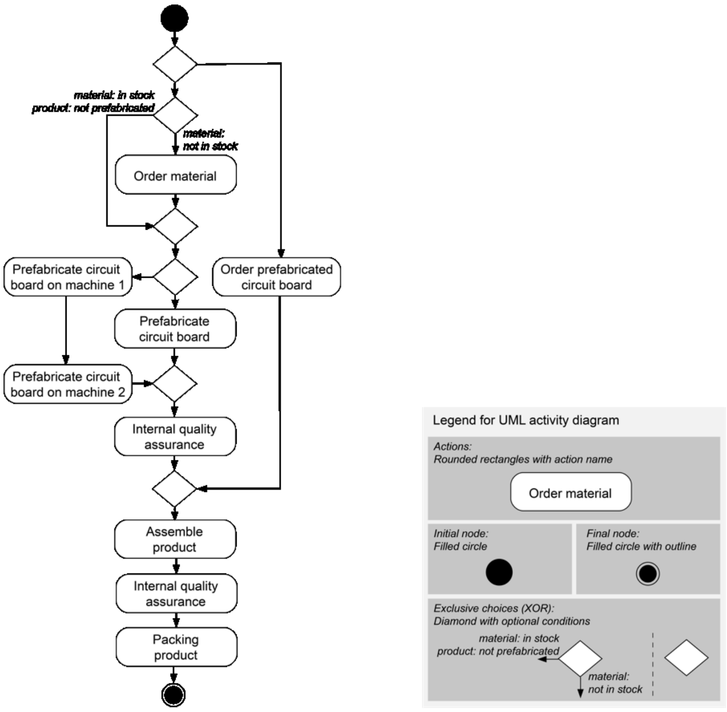

material: in stock
product: not prefabricated
material: not in stock
Order material
Prefabricate circuit board on machine 1
Order prefabricated circuit board
Prefabricate circuit board
Prefabricate circuit board on machine 2
Internal quality assurance
Assemble product
Internal quality assurance
Packing product
Legend for UML activity diagram
Actions:
Rounded rectangles with action name
Order material
Initial node:
Filled circle
Final node:
Filled circle with outline
Exclusive choices (XOR):
Diamond with optional conditions
material: in stock
product: not prefabricated
material:
not in stock

Fig. 5. Process Graph after the Adaptation due to *multiple feasible Atomic Changes*.

5 EVALUATION

We assessed our approach based on evaluation criteria stated in literature [Prat et al. 2015]. More precisely, in the next section, we pursue mathematical proofs for the evaluation of correctness, completeness and termination (E1) of our approach to support contribution (C2). Thereafter, we outline the evaluation of the technical feasibility (E2) of our approach based on a prototypical implementation. In the subsequent section, the operational feasibility (E3) of the approach is evaluated by means of a real-world scenario for the adaptation of existing process models to needs for change in advance in an automated manner (cf. (C1)). Lastly, to evaluate the performance (E4) of our approach, we conduct a complexity analysis of our algorithm as well as a simulation experiment. Please note that we focus on core aspects here due to length restrictions, but more detailed discussions are available in the appendix.

5.1 Evaluation of Correctness, Completeness, and Termination (E1)

To evaluate the key properties correctness, completeness, and termination, we rely on mathematical proofs. To conduct this evaluation in a systematic manner, each part of our proposed approach for addressing a particular feasible atomic change is evaluated with respect to these three properties. In the following, we give an overview over the essential points that substantiate our argumentation.

(E1.1) Correctness: It has to be shown that only feasible paths are contained in an adapted process graph. To confirm correctness, we proved that conditions i. to iii. of Definition 5 hold for each path contained in process graphs adapted by means of our approach.

(E1.2) Completeness: It has to be shown that each path, which is feasible, is contained in the adapted process graph. To this end, we distinguished and proved two cases. For each feasible path which consists of actions and belief states from the original process graph, we showed that it is a (shortened) path from the original process graph, remains unchanged and is retained in complete by our approach. In case of a feasible path which includes actions or belief states not present in the original process graph, we showed that it is created by conducting planning steps starting from each (changed) belief state. The first and second case were proven to be complete as the approaches that constructed the original process graph as well as the aforementioned planning steps were proven to be complete.

(E1.3) Termination: Each part of our presented approach to adapt a process graph to one particular feasible atomic change mainly consists of two procedures. The first procedure consists of checking applicability criteria or goal state criteria within the original process graph, which has a finite number of actions and belief states. The second procedure consists of conducting planning steps, which terminates as there is a finite number of distinct belief states which are reachable from any given belief state. Hence, both procedures were proved and confirmed to terminate, thus the presented approach terminates.

5.2 Evaluation of Technical Feasibility (E2)

We implemented our approach in a software prototype. An existing Java implementation of a planning technique [cf. Bertoli et al. 2006; Heinrich and Schön 2015] for nondeterministic belief state-transition systems able to construct process graphs (cf. Definition 5) served as a basis. The existing implementation allows to specify planning problems by means of XML files. We added the functionality to also specify needs for change via XML files. In particular, using a pair of XML files (one file specifying the initial regular planning problem and another file specifying needs for change), both a process graph and one to possibly many subsequent feasible atomic changes (cf. Table 2) can be specified. We further integrated the presented approach (cf. Sections 4.1 to 4.3) in the already implemented planning algorithm. Persons other than the programmers validated the source code via structured walkthroughs. Moreover, the validity of this extended prototype was ensured by carrying out structured tests using the JUnit framework. This supports the technical feasibility (E2) of our approach. Further, a pseudocode of our approach was created and is provided at https://epub.uni-regensburg.de/43489/1/Adaption_Online_Material.pdf.

5.3 Evaluation of Operational Feasibility (E3)

In order to evaluate whether the proposed approach is able to adapt process models to needs for change in advance (contribution (C1)) and is operationally feasible, we conducted a field experiment by applying our approach to a manufacturing process of a European electrical engineering company. This process had been subject to several complex needs for change in recent history. The initial process graph (before any adaptation occurred) as well as the needs for change (six major changes in the course of approx. 24 months) were derived from several interviews with the manager and staff. Please note that, while the needs for change occurred in the past, our approach still adapted to needs for change in advance in this setting because the initial process graph and the needs for change were used as input. The actually planned process models after addressing the needs for change (i.e., the adaptations in the real world) just served for comparing our adaptation with the real-world reference. To assess the operational feasibility of our approach, we analyzed the following three questions necessary for a valid application in this real-world scenario:

(E3.1) Is our approach able to adapt the process graph to the needs for change stated by the manager?

After determining the composition of atomic changes to address each need for change stated by the manager, we specified them in terms of the aforementioned XML files. Then the XML files were used to adapt the process graph by means of our prototype in the order the changes occurred in reality. All needs for change stated by the manager could be represented by means of a composition of atomic changes and subsequently, the process graph could be adapted to all needs for change.

(E3.2) Do the adapted process graphs represent the actually used process models?

To compare the adapted process graphs with the respective used process models of the company after each need for change, we visually simplified the adapted process graphs before presenting them to the staff. In detail, we removed the belief states and omitted the preconditions and effects of the actions depicted in the graphs so that their layout was similar to UML activity diagrams. We discussed these graphs with the manager and employees of the manufacturing department. In particular, for each need for change, we elaborated on the differences between the graph before adaptation and the adapted graph in detail. Thereafter, we asked the staff whether the adapted graphs represent the process models actually used in the company once the according change took place. The staff confirmed this for every case.

(E3.3) Are the adapted process graphs assessed as correct and complete by the staff?

In a further discussion with the staff, all paths of the adapted process graphs were assessed to be correct. We also asked whether the adapted graphs neglected any feasible paths. Here, the staff validated that the graphs contained all paths actually used in the company and that no feasible paths not represented in the adapted process graphs were known. While the number of paths to check was high in some cases, most paths just contained the same actions in different order, making a manual verification possible.

To conclude, the operational feasibility (E3) of our approach was supported in this real-world scenario. Our approach was able to adapt process models to needs for change in advance in an automated manner (cf. (C1)). We did observe that the resulting process graphs are not yet easy to comprehend, but this issue may be solved by using approaches to construct control flow patterns based on our adapted process graph [Heinrich et al. 2015; Heinrich and Schön 2016; Heinrich et al. 2019]. Yet, to assess (E3), it was sufficient to visually simplify the process graphs and discuss the changes of these graphs in detail with the staff.

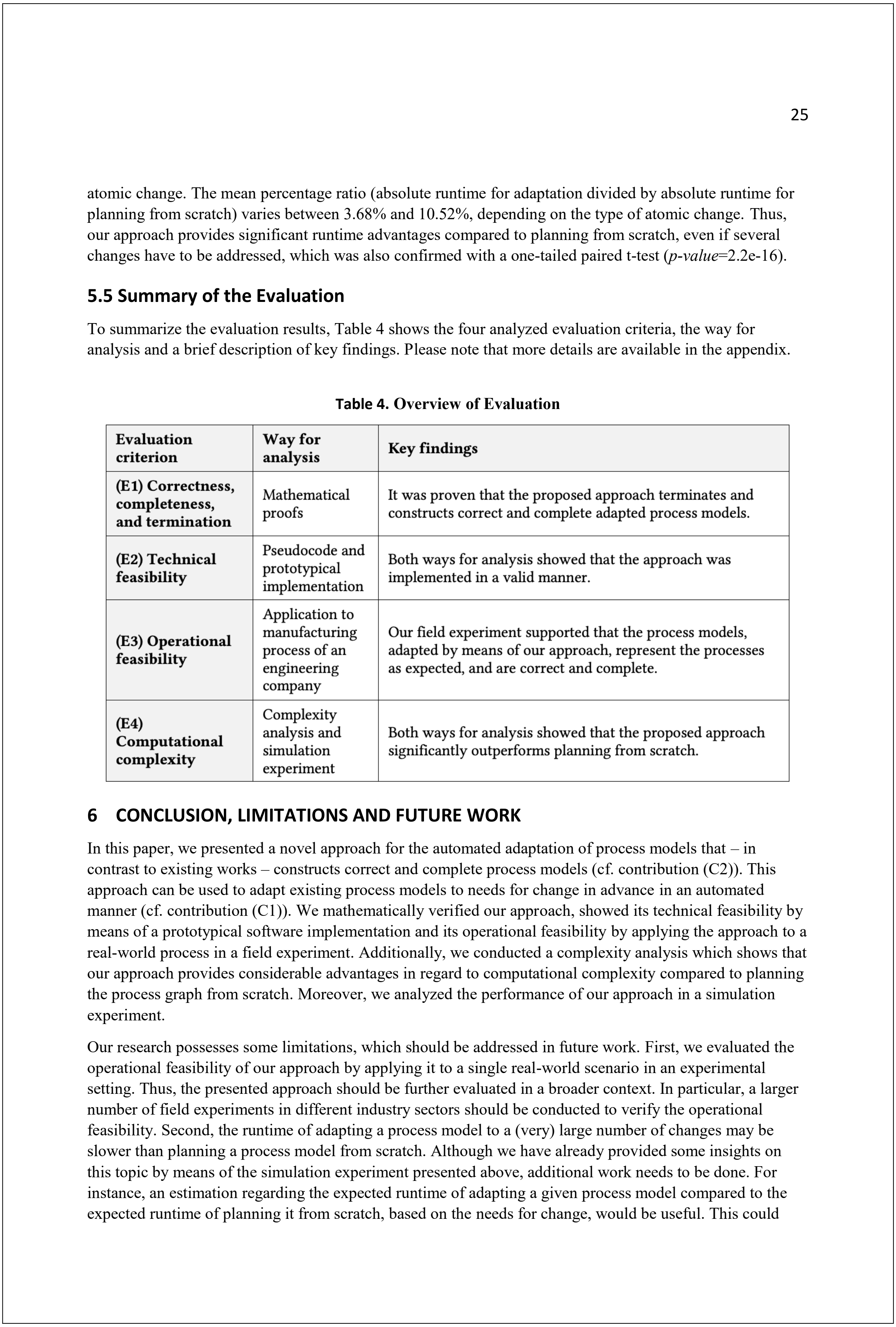
5.4 Evaluation of Performance (E4)

In order to analyze the difference in performance of our approach compared to planning from scratch, we conducted an analysis to evaluate its computational complexity (E4.1) and a simulation experiment (E4.2)[5].

(E4.1) Evaluation of computational complexity: As our approach as well as approaches for planning process graphs from scratch heavily rely on comparisons rather than on arithmetic operations, we calculated complexity in terms of comparisons needed to apply each approach. Thereby, we took the number of belief states, actions, belief state variables, as well as goal states into consideration. It was shown that the number of comparisons is polynomial when applying our approach in contrast to factorial in the number of actions when planning from scratch, which is a considerable improvement.

(E4.2) Simulation experiment: To conduct this experiment in a systematic manner, we focused on the complete set of feasible atomic changes. For our analysis, we used adaptations based upon 12 existing real-world process graphs of different companies from different application contexts. These process graphs consist of 17 to 8,267 actions and 15 to 2,693 belief states. The corresponding belief state variables comprise numeric domains as well as discrete domains. We observed that the required runtime for adapting the existing process graphs with the prototype is lower than for planning the graphs from scratch for each type of

[5] We executed the prototype on a Dell Latitude Notebook with an Intel Core i7-2640M, 2.80 GHz, 8GB RAM running on Windows 8.1 (Version 6.3.9600) 64 bit and Java 1.8.0 (build 1.8.0.-b132) 64 bit.



atomic change. The mean percentage ratio (absolute runtime for adaptation divided by absolute runtime for planning from scratch) varies between 3.68% and 10.52%, depending on the type of atomic change. Thus, our approach provides significant runtime advantages compared to planning from scratch, even if several changes have to be addressed, which was also confirmed with a one-tailed paired t-test (*p-value*=2.2e-16).

5.5 Summary of the Evaluation

To summarize the evaluation results, Table 4 shows the four analyzed evaluation criteria, the way for analysis and a brief description of key findings. Please note that more details are available in the appendix.

Table 4. Overview of Evaluation

Evaluation criterion	Way for analysis	Key findings
(E1) Correctness, completeness, and termination	Mathematical proofs	It was proven that the proposed approach terminates and constructs correct and complete adapted process models.
(E2) Technical feasibility	Pseudocode and prototypical implementation	Both ways for analysis showed that the approach was implemented in a valid manner.
(E3) Operational feasibility	Application to manufacturing process of an engineering company	Our field experiment supported that the process models, adapted by means of our approach, represent the processes as expected, and are correct and complete.
(E4) Computational complexity	Complexity analysis and simulation experiment	Both ways for analysis showed that the proposed approach significantly outperforms planning from scratch.

6 CONCLUSION, LIMITATIONS AND FUTURE WORK

In this paper, we presented a novel approach for the automated adaptation of process models that – in contrast to existing works – constructs correct and complete process models (cf. contribution (C2)). This approach can be used to adapt existing process models to needs for change in advance in an automated manner (cf. contribution (C1)). We mathematically verified our approach, showed its technical feasibility by means of a prototypical software implementation and its operational feasibility by applying the approach to a real-world process in a field experiment. Additionally, we conducted a complexity analysis which shows that our approach provides considerable advantages in regard to computational complexity compared to planning the process graph from scratch. Moreover, we analyzed the performance of our approach in a simulation experiment.

Our research possesses some limitations, which should be addressed in future work. First, we evaluated the operational feasibility of our approach by applying it to a single real-world scenario in an experimental setting. Thus, the presented approach should be further evaluated in a broader context. In particular, a larger number of field experiments in different industry sectors should be conducted to verify the operational feasibility. Second, the runtime of adapting a process model to a (very) large number of changes may be slower than planning a process model from scratch. Although we have already provided some insights on this topic by means of the simulation experiment presented above, additional work needs to be done. For instance, an estimation regarding the expected runtime of adapting a given process model compared to the expected runtime of planning it from scratch, based on the needs for change, would be useful. This could provide fruitful insights for deciding whether to use our approach or to plan the process model from scratch, given a large number of needs for change. Third, we aim to construct complete adapted graphs (cf. contribution (C2)) and thus do not focus on a heuristic approach in this paper. This is of particular interest with respect to decision-making. The ability to construct complete adapted process models (including all decision alternatives) allows selecting a beneficial feasible path based on Quality-of-Service parameters and economic criteria and hence offers valuable decision support. However, the runtime for adapting process graphs may additionally be reduced further by means of a heuristic approach in case a process model with all feasible paths is not necessary.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

AFFLERBACH, P., KASTNER, G., KRAUSE, F., AND RÖGLINGER, M. 2014. The Business Value of Process Flexibility. *Business & Information Systems Engineering 6*, 4, 203–214.

ALFÉREZ, G.H., PELECHANO, V., MAZO, R., SALINESI, C., AND DIAZ, D. 2014. Dynamic adaptation of service compositions with variability models. *Journal of Systems and Software 91*, 24–47.

AUGUSTO, A., CONFORTI, R., DUMAS, M., LA ROSA, M., MAGGI, F.M., MARRELLA, A., MECELLA, M., AND SOO, A. 2018. Automated discovery of process models from event logs: Review and benchmark. *IEEE Transactions on Knowledge and Data Engineering 31*, 4.

BARBA, I., DEL VALLE, C., WEBER, B., AND JIMÉNEZ-RAMÍREZ, A. 2013a. Automatic generation of optimized business process models from constraint-based specifications. *International Journal of Cooperative Information Systems 22*.

BARBA, I., WEBER, B., DEL VALLE, C., AND JIMÉNEZ-RAMÍREZ, A. 2013b. User recommendations for the optimized execution of business processes. *Data & Knowledge Engineering 86*, 61–84.

BASHARI, M., BAGHERI, E., AND DU, W. 2018. Automated composition and optimization of services for variability-intensive domains. *Journal of Systems and Software 146*, 356–376.

BERTOLI, P., CIMATTI, A., ROVERI, M., AND TRAVERSO, P. 2006. Strong planning under partial observability. *Artificial Intelligence 170*, 4–5, 337–384.

BOSE, R.P.J.C., VAN DER AALST, W.M.P., ZLIOBAITE, I., AND PECHENIZKIY, M. 2014. Dealing with concept drifts in process mining. *IEEE transactions on neural networks and learning systems 25*, 1, 154–171.

BUCCHIARONE, A., DRAGONI, N., AND DUSTDAR, S. 2020. *Microservices. Science and Engineering*.

BUCCHIARONE, A., PISTORE, M., RAIK, H., AND KAZHAMIAKIN, R. 2011. Adaptation of service-based business processes by context-aware replanning. In *Service-Oriented Computing and Applications (SOCA), 2011 IEEE International Conference on*, 1–8.

CHAFLE, G., DASGUPTA, K., KUMAR, A., MITTAL, S., AND SRIVASTAVA, B. 2006. Adaptation in Web Service Composition and Execution. *Proceedings of the 2006 IEEE International Conference on Web Services (ICWS'06)*, 549–557.

COGNINI, R., CORRADINI, F., GNESI, S., POLINI, A., AND RE, B. 2016. Business process flexibility - a systematic literature review with a software systems perspective. *Information Systems Frontiers*.

EISENBARTH, T., LAUTENBACHER, F., AND BAUER, B. 2011. Adaptation of Process Models – A Semantic-based Approach. *Journal of Research and Practice in Information Technology 43*, 1, 5–23.

ERTUGRUL, A.M., AND DEMIRORS, O. 2016. A Method for Modeling Business Processes in a Role-based and Decentralized Way. In *Proceedings of the 8th International Conference on Subject-oriented Business Process Management - S-BPM '16*, 1–4.

Fahland, D., Favre, C., Koehler, J., Lohmann, N., Völzer, H., and Wolf, K. 2011. Analysis on demand. Instantaneous soundness checking of industrial business process models. *Data & Knowledge Engineering 70*, 5, 448–466.

Fahland, D., and van der Aalst, W.M.P. 2012. Repairing Process Models to Reflect Reality. *Business Process Management 7481*, 229–245.

Fan, S.-L., Yang, Y.-B., and Wang, X.-X. 2018. Efficient Web Service Composition via Knapsack-Variant Algorithm. In *Proceedings of the International Conference on Services Computing*, 51–66.

Feld, T., and Hoffmann, M. 2014. Process on Demand: Planning and Control of Adaptive Business Processes. In *Future Business Software*, G. Brunetti, T. Feld, L. Heuser, J. Schnitter and C. Webel, Eds. Springer International Publishing, Cham, 55–66.

Forstner, E., Kamprath, N., and Röglinger, M. 2014. Capability development with process maturity models – Decision framework and economic analysis. *Journal of Decision Systems (JDS) 23*, 2, 127–150.

Gambini, M., La Rosa, M., Migliorini, S., and ter Hofstede, A.H.M. 2011. Automated error correction of business process models. In *Business Process Management*, 148–165.

Garcia-Bañuelos, L., van Beest, N.R., Dumas, M., La Rosa, M., and Mertens, W. 2017. Complete and interpretable conformance checking of business processes. *IEEE Transactions on Software Engineering 44*, 3, 262–290.

Gerevini, A.E., and Serina, I. 2000. Fast Plan Adaptation through Planning Graphs: Local and Systematic Search Techniques. *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems*, 112–121.

Ghallab, M., Nau, D.S., and Traverso, P. 2004. *Automated Planning: Theory & Practice*. Morgan Kaufmann, San Francisco.

Ghallab, M., Nau, D.S., and Traverso, P. 2016. *Automated Planning and Acting*. Cambridge University Press, New York, NY.

Heinrich, B., Klier, M., and Zimmermann, S. 2015. Automated planning of process models: Design of a novel approach to construct exclusive choices. *Decision Support Systems 78*, 1–14.

Heinrich, B., Krause, F., and Schiller, A. 2019. Automated planning of process models: The construction of parallel splits and synchronizations. *Decision Support Systems 125*.

Heinrich, B., and Lewerenz, L. 2015. Decision support for the usage of mobile information services: A context-aware service selection approach that considers the effects of context interdependencies. *Journal of Decision Systems (JDS) 24*, 4, 406–432.

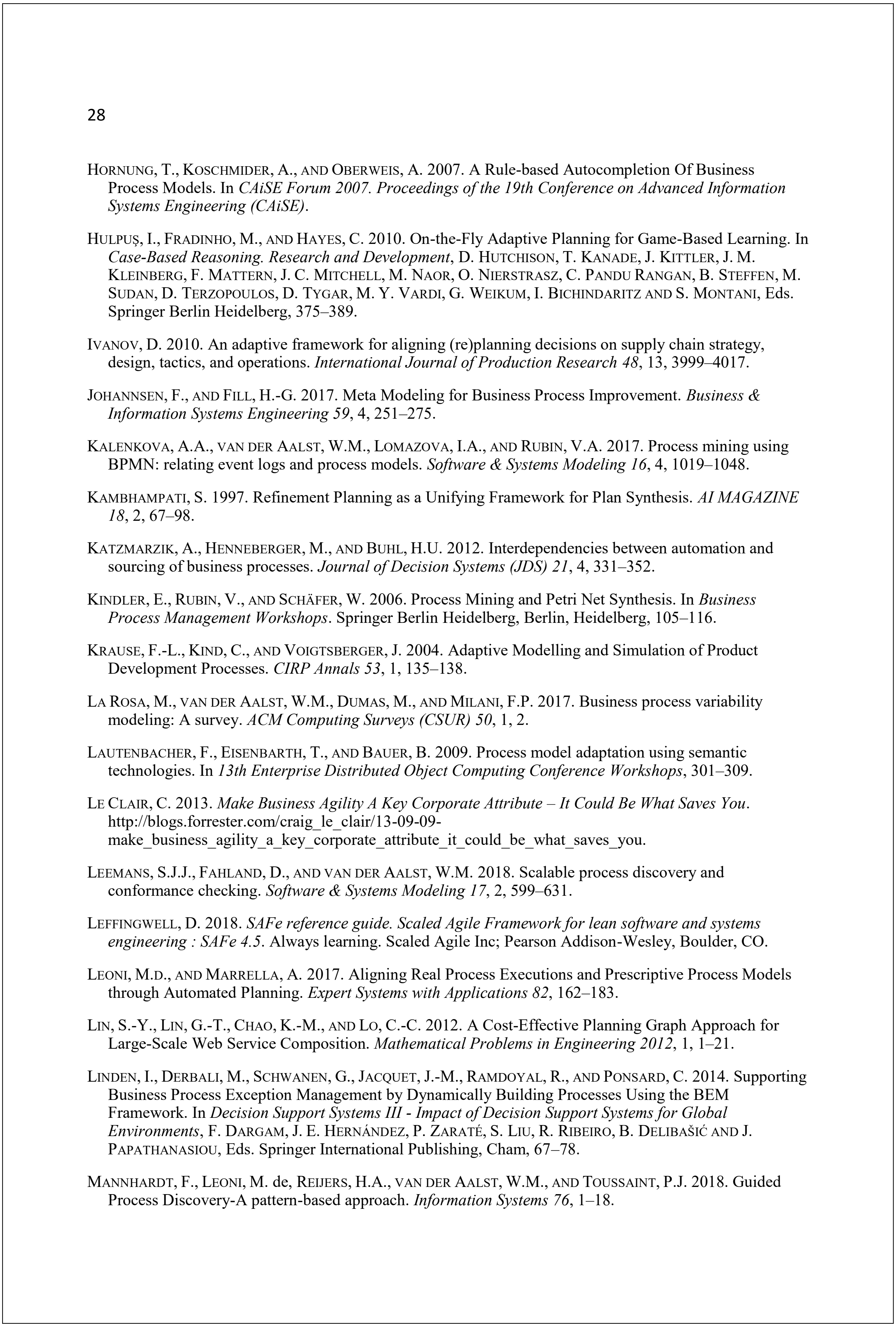
Heinrich, B., Schiller, A., and Schön, D. 2018. The cooperation of multiple actors within process models: an automated planning approach. *Journal of Decision Systems (JDS) 27*, 4, 238–274.

Heinrich, B., and Schön, D. 2015. Automated Planning of context-aware Process Models. In *Proceedings of the 23rd European Conference on Information Systems (ECIS)*, Münster, Germany.

Heinrich, B., and Schön, D. 2016. Automated Planning of Process Models: The Construction of Simple Merges. In *Proceedings of the 24rd European Conference on Information Systems (ECIS)*, Istanbul, Turkey.

Henneberger, M., Heinrich, B., Lautenbacher, F., and Bauer, B. 2008. Semantic-Based Planning of Process Models. In *Multikonferenz Wirtschaftsinformatik (MKWI)*, 1677–1689.

Hoffmann, J., Weber, I., and Kraft, F.M. 2012. SAP Speaks PDDL: Exploiting a Software-Engineering Model for Planning in Business Process Management. *Journal of Artificial Intelligence Research 44*, 1, 587–632.



Hornung, T., Koschmider, A., and Oberweis, A. 2007. A Rule-based Autocompletion Of Business Process Models. In *CAiSE Forum 2007. Proceedings of the 19th Conference on Advanced Information Systems Engineering (CAiSE)*.

Hulpuş, I., Fradinho, M., and Hayes, C. 2010. On-the-Fly Adaptive Planning for Game-Based Learning. In *Case-Based Reasoning. Research and Development*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, I. Bichindaritz and S. Montani, Eds. Springer Berlin Heidelberg, 375–389.

Ivanov, D. 2010. An adaptive framework for aligning (re)planning decisions on supply chain strategy, design, tactics, and operations. *International Journal of Production Research 48*, 13, 3999–4017.

Johannsen, F., and Fill, H.-G. 2017. Meta Modeling for Business Process Improvement. *Business & Information Systems Engineering 59*, 4, 251–275.

Kalenkova, A.A., van der Aalst, W.M., Lomazova, I.A., and Rubin, V.A. 2017. Process mining using BPMN: relating event logs and process models. *Software & Systems Modeling 16*, 4, 1019–1048.

Kambhampati, S. 1997. Refinement Planning as a Unifying Framework for Plan Synthesis. *AI MAGAZINE 18*, 2, 67–98.

Katzmarzik, A., Henneberger, M., and Buhl, H.U. 2012. Interdependencies between automation and sourcing of business processes. *Journal of Decision Systems (JDS) 21*, 4, 331–352.

Kindler, E., Rubin, V., and Schäfer, W. 2006. Process Mining and Petri Net Synthesis. In *Business Process Management Workshops*. Springer Berlin Heidelberg, Berlin, Heidelberg, 105–116.

Krause, F.-L., Kind, C., and Voigtsberger, J. 2004. Adaptive Modelling and Simulation of Product Development Processes. *CIRP Annals 53*, 1, 135–138.

La Rosa, M., van der Aalst, W.M., Dumas, M., and Milani, F.P. 2017. Business process variability modeling: A survey. *ACM Computing Surveys (CSUR) 50*, 1, 2.

Lautenbacher, F., Eisenbarth, T., and Bauer, B. 2009. Process model adaptation using semantic technologies. In *13th Enterprise Distributed Object Computing Conference Workshops*, 301–309.

Le Clair, C. 2013. *Make Business Agility A Key Corporate Attribute – It Could Be What Saves You*. http://blogs.forrester.com/craig_le_clair/13-09-09-make_business_agility_a_key_corporate_attribute_it_could_be_what_saves_you.

Leemans, S.J.J., Fahland, D., and van der Aalst, W.M. 2018. Scalable process discovery and conformance checking. *Software & Systems Modeling 17*, 2, 599–631.

Leffingwell, D. 2018. *SAFe reference guide. Scaled Agile Framework for lean software and systems engineering : SAFe 4.5*. Always learning. Scaled Agile Inc; Pearson Addison-Wesley, Boulder, CO.

Leoni, M.d., and Marrella, A. 2017. Aligning Real Process Executions and Prescriptive Process Models through Automated Planning. *Expert Systems with Applications 82*, 162–183.

Lin, S.-Y., Lin, G.-T., Chao, K.-M., and Lo, C.-C. 2012. A Cost-Effective Planning Graph Approach for Large-Scale Web Service Composition. *Mathematical Problems in Engineering 2012*, 1, 1–21.

Linden, I., Derbali, M., Schwanen, G., Jacquet, J.-M., Ramdoyal, R., and Ponsard, C. 2014. Supporting Business Process Exception Management by Dynamically Building Processes Using the BEM Framework. In *Decision Support Systems III - Impact of Decision Support Systems for Global Environments*, F. Dargam, J. E. Hernández, P. Zaraté, S. Liu, R. Ribeiro, B. Delibašić and J. Papathanasiou, Eds. Springer International Publishing, Cham, 67–78.

Mannhardt, F., Leoni, M. de, Reijers, H.A., van der Aalst, W.M., and Toussaint, P.J. 2018. Guided Process Discovery-A pattern-based approach. *Information Systems 76*, 1–18.

Marrella, A. 2018. Automated Planning for Business Process Management. *Journal on Data Semantics*, 1–20.

Marrella, A., Mecella, M., and Sardina, S. 2017. Intelligent Process Adaptation in the SmartPM System. *ACM Transactions on Intelligent Systems and Technology 8*, 2, 1–43.

Marrella, A., Russo, A., and Mecella, M. 2012. Planlets: Automatically Recovering Dynamic Processes in YAWL. In *On the Move to Meaningful Internet Systems: OTM 2012*, 268–286.

Martin, J. 1983. *Managing the data-base environment*. Prentice-Hall, Englewood Cliffs, NJ.

Masellis, R. de, Di Francescomarino, C., Ghidini, C., Montali, M., and Tessaris, S. 2017. Add data into business process verification: Bridging the gap between theory and practice. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

McElheran, K. 2015. Do Market Leaders Lead in Business Process Innovation? The Case(s) of E-business Adoption. *Management Science 61*, 6, 1197–1216.

Megargel, A., Shankararaman, V., and Walker, D.K. 2020. Migrating from Monoliths to Cloud-Based Microservices: A Banking Industry Example. In *Software Engineering in the Era of Cloud Computing*, M. Ramachandran, Ed. Springer International Publishing, Cham, 85–108.

Mejri, A., Ayachi-Ghannouchi, S., and Martinho, R. 2018. A quantitative approach for measuring the degree of flexibility of business process models. *Business Process Managment Journal*.

Mendling, J., Verbeek, H.M.W., van Dongen, B.F., van der Aalst, W.M.P., and Neumann, G. 2008. Detection and prediction of errors in EPCs of the SAP reference model. *Data & Knowledge Engineering 64*, 1, 312–329.

Meyer, H., and Weske, M. 2006. Automated service composition using heuristic search. *Business Process Management*, 81–96.

Milani, F., Dumas, M., Matulevičius, R., Ahmed, N., and Kasela, S. 2016. Criteria and Heuristics for Business Process Model Decomposition. *Business & Information Systems Engineering 58*, 1, 7–17.

Milionis, N., Jereb, S., Henderson, K., Vrabic, J., Bain, M., Dolezal, J., Roessing, E., Dos Santos, João Nuno Coelho, Simeonova, R., and Otto, J. 2019. *The EU's response to the "dieselgate" scandal*. https://www.eca.europa.eu/lists/ecadocuments/brp_vehicle_emissions/brp_vehicle_emissions_en.pdf. Accessed 22 September 2019.

Nebel, B., and Koehler, J. 1995. Plan reuse versus plan generation. A theoretical and empirical analysis. *Artificial Intelligence 76*, 1-2, 427–454.

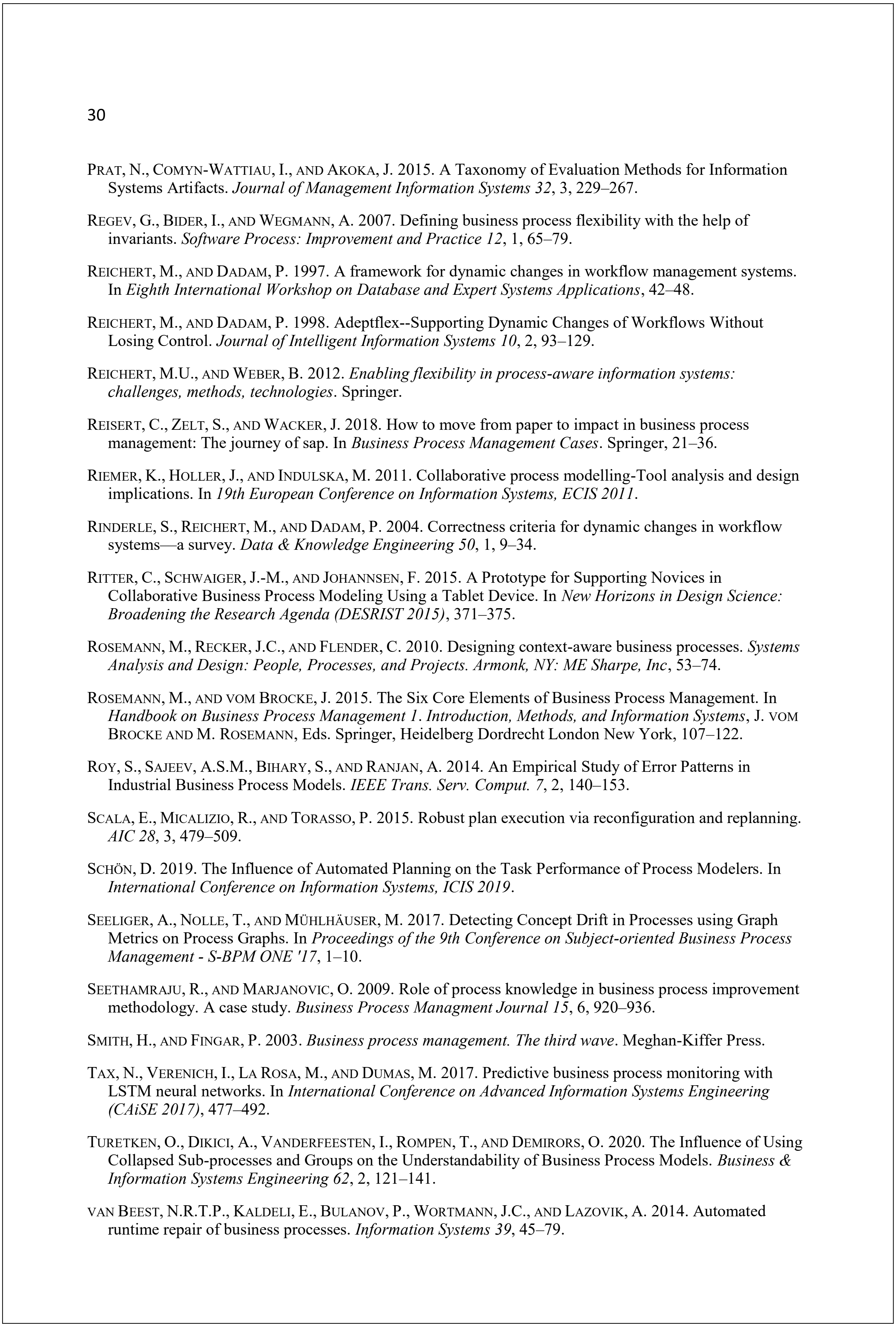
Noura, M., and Gaedke, M. 2019. An Automated Cyclic Planning Framework Based on Plan-Do-Check-Act for Web of Things Composition. In *Proceedings of the 10th ACM Conference on Web Science - WebSci '19*, 205–214.

Nunes, V.T., Santoro, F.M., Werner, C.M.L., and G. Ralha, C. 2018. Real-Time Process Adaptation. A Context-Aware Replanning Approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems 48*, 1, 99–118.

Page, S.A. 2016. *The power of business process improvement. 10 simple steps to increase effectiveness, efficiency, and adaptability*. American Management Association, New York.

Parnas, D.L. 1972. On the criteria to be used in decomposing systems into modules. *Commun. ACM 15*, 12, 1053–1058.

Pesic, M., and van der Aalst, W.M. 2006. A declarative approach for flexible business processes management. In *Business Process Management Workshops*, 169–180.



Prat, N., Comyn-Wattiau, I., and Akoka, J. 2015. A Taxonomy of Evaluation Methods for Information Systems Artifacts. *Journal of Management Information Systems 32*, 3, 229–267.

Regev, G., Bider, I., and Wegmann, A. 2007. Defining business process flexibility with the help of invariants. *Software Process: Improvement and Practice 12*, 1, 65–79.

Reichert, M., and Dadam, P. 1997. A framework for dynamic changes in workflow management systems. In *Eighth International Workshop on Database and Expert Systems Applications*, 42–48.

Reichert, M., and Dadam, P. 1998. Adeptflex--Supporting Dynamic Changes of Workflows Without Losing Control. *Journal of Intelligent Information Systems 10*, 2, 93–129.

Reichert, M.U., and Weber, B. 2012. *Enabling flexibility in process-aware information systems: challenges, methods, technologies*. Springer.

Reisert, C., Zelt, S., and Wacker, J. 2018. How to move from paper to impact in business process management: The journey of sap. In *Business Process Management Cases*. Springer, 21–36.

Riemer, K., Holler, J., and Indulska, M. 2011. Collaborative process modelling-Tool analysis and design implications. In *19th European Conference on Information Systems, ECIS 2011*.

Rinderle, S., Reichert, M., and Dadam, P. 2004. Correctness criteria for dynamic changes in workflow systems—a survey. *Data & Knowledge Engineering 50*, 1, 9–34.

Ritter, C., Schwaiger, J.-M., and Johannsen, F. 2015. A Prototype for Supporting Novices in Collaborative Business Process Modeling Using a Tablet Device. In *New Horizons in Design Science: Broadening the Research Agenda (DESRIST 2015)*, 371–375.

Rosemann, M., Recker, J.C., and Flender, C. 2010. Designing context-aware business processes. *Systems Analysis and Design: People, Processes, and Projects. Armonk, NY: ME Sharpe, Inc*, 53–74.

Rosemann, M., and vom Brocke, J. 2015. The Six Core Elements of Business Process Management. In *Handbook on Business Process Management 1. Introduction, Methods, and Information Systems*, J. vom Brocke and M. Rosemann, Eds. Springer, Heidelberg Dordrecht London New York, 107–122.

Roy, S., Sajeev, A.S.M., Bihary, S., and Ranjan, A. 2014. An Empirical Study of Error Patterns in Industrial Business Process Models. *IEEE Trans. Serv. Comput. 7*, 2, 140–153.

Scala, E., Micalizio, R., and Torasso, P. 2015. Robust plan execution via reconfiguration and replanning. *AIC 28*, 3, 479–509.

Schön, D. 2019. The Influence of Automated Planning on the Task Performance of Process Modelers. In *International Conference on Information Systems, ICIS 2019*.

Seeliger, A., Nolle, T., and Mühlhäuser, M. 2017. Detecting Concept Drift in Processes using Graph Metrics on Process Graphs. In *Proceedings of the 9th Conference on Subject-oriented Business Process Management - S-BPM ONE '17*, 1–10.

Seethamraju, R., and Marjanovic, O. 2009. Role of process knowledge in business process improvement methodology. A case study. *Business Process Managment Journal 15*, 6, 920–936.

Smith, H., and Fingar, P. 2003. *Business process management. The third wave*. Meghan-Kiffer Press.

Tax, N., Verenich, I., La Rosa, M., and Dumas, M. 2017. Predictive business process monitoring with LSTM neural networks. In *International Conference on Advanced Information Systems Engineering (CAiSE 2017)*, 477–492.

Turetken, O., Dikici, A., Vanderfeesten, I., Rompen, T., and Demirors, O. 2020. The Influence of Using Collapsed Sub-processes and Groups on the Understandability of Business Process Models. *Business & Information Systems Engineering 62*, 2, 121–141.

van Beest, N.R.T.P., Kaldeli, E., Bulanov, P., Wortmann, J.C., and Lazovik, A. 2014. Automated runtime repair of business processes. *Information Systems 39*, 45–79.

van der Aalst, W.M.P. 2013. Business Process Management: A Comprehensive Survey. *ISRN Software Engineering 2013*, 1, 1–37.

van der Aalst, W.M.P. 2015. Extracting Event Data from Databases to Unleash Process Mining. In *BPM - Driving Innovation in a Digital World*, J. vom Brocke and T. Schmiedel, Eds. Springer International Publishing, Cham, 105–128.

van der Aalst, W.M.P., and Jablonski, S. 2000. Dealing with workflow change: identification of issues and solutions. *International Journal of Computer Systems Science & Engineering 15*, 5, 267–276.

van der Aalst, W.M.P., Pesic, M., and Schonenberg, H. 2009. Declarative workflows: Balancing between flexibility and support. *Computer Science - Research and Development 23*, 2, 99–113.

van der Aalst, W.M.P., Rubin, V., Verbeek, H.M.W., van Dongen, B.F., Kindler, E., and Günther, C.W. 2010. Process mining: a two-step approach to balance between underfitting and overfitting. *Software & Systems Modeling 9*, 1, 87–111.

van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., and Barros, A.P. 2003. Workflow Patterns. *Distributed and Parallel Databases 14*, 1, 5–51.

van Gorp, P., and Dijkman, R. 2013. A visual token-based formalization of BPMN 2.0 based on in-place transformations. *Information and Software Technology 55*, 2, 365–394.

Vanwersch, R.J.B., Shahzad, K., Vanderfeesten, I., Vanhaecht, K., Grefen, P., Pintelon, L., Mendling, J., van Merode, G.G., and Reijers, H.A. 2016. A Critical Evaluation and Framework of Business Process Improvement Methods. *Business & Information Systems Engineering 58*, 1, 43–53.

Verbeek, H.M.W., and van der Aalst, W.M.P. 2005. Analyzing BPEL processes using Petri nets. In *Proceedings of the Second International Workshop on Applications of Petri Nets to Coordination, Workflow and Business Process Management*, 59–78.

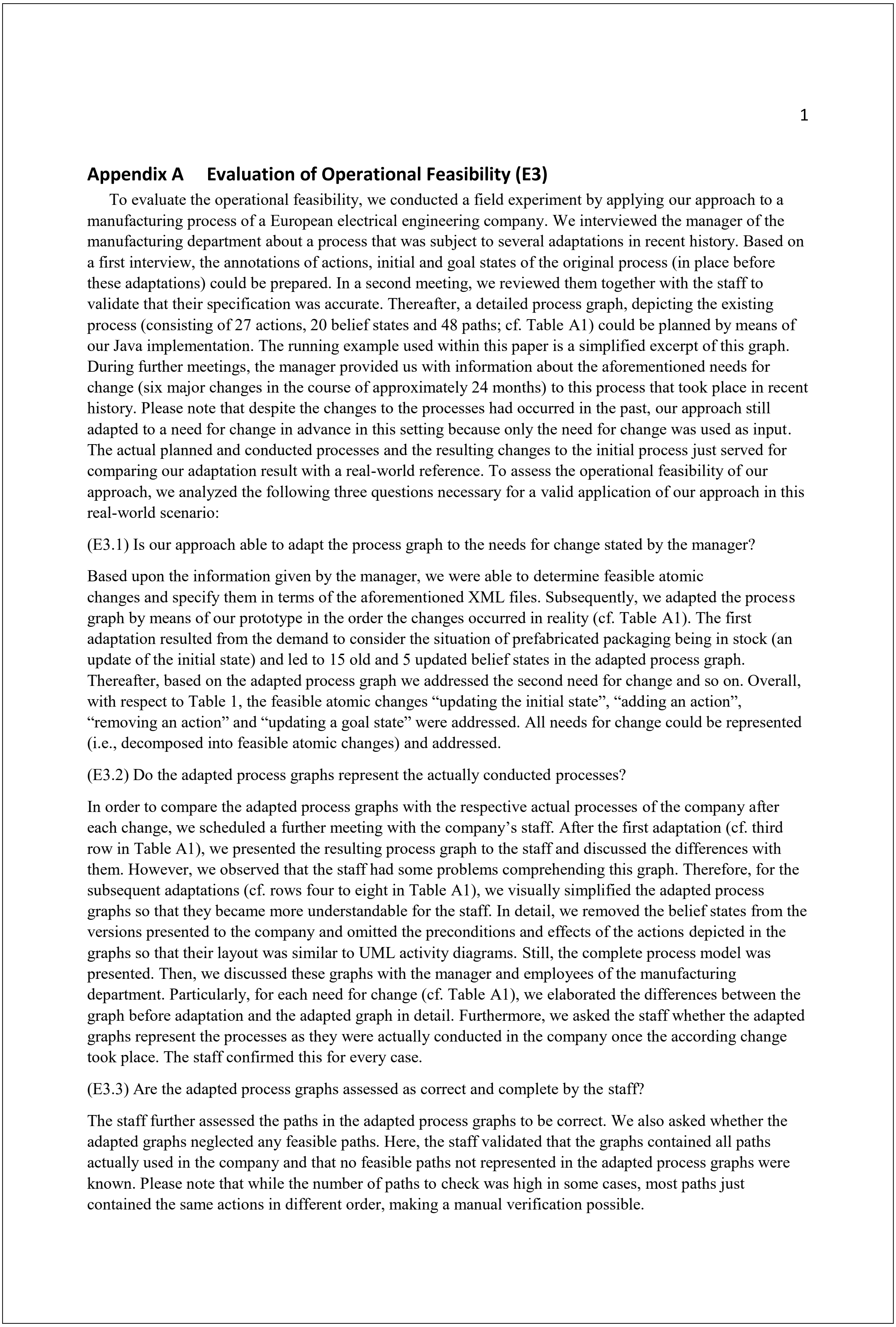
vom Brocke, J. 2009. Design Principles for Reference Modelling. In *Innovations in Information Systems Modeling*, 269–296.

vom Brocke, J., and Mendling, J., Eds. 2018. *Business Process Management Cases. Digital Innovation and Business Transformation in Practice*. Management for Professionals. Springer.

Weber, B., Reichert, M., and Rinderle-Ma, S. 2008. Change patterns and change support features- enhancing flexibility in process-aware information systems. *Data & Knowledge Engineering 66*, 3.

Wieloch, K., Filipowska, A., and Kaczmarek, M. 2011. Autocompletion for Business Process Modelling. In *Business Information Systems Workshops*, 30–40.

Wynn, M.T., Verbeek, H.M.W., van der Aalst, W.M.P., ter Hofstede, A.H.M., and Edmond, D. 2009. Business process verification-finally a reality! *Business Process Managment Journal 15*, 1, 74–92.



Appendix A Evaluation of Operational Feasibility (E3)

To evaluate the operational feasibility, we conducted a field experiment by applying our approach to a manufacturing process of a European electrical engineering company. We interviewed the manager of the manufacturing department about a process that was subject to several adaptations in recent history. Based on a first interview, the annotations of actions, initial and goal states of the original process (in place before these adaptations) could be prepared. In a second meeting, we reviewed them together with the staff to validate that their specification was accurate. Thereafter, a detailed process graph, depicting the existing process (consisting of 27 actions, 20 belief states and 48 paths; cf. Table A1) could be planned by means of our Java implementation. The running example used within this paper is a simplified excerpt of this graph. During further meetings, the manager provided us with information about the aforementioned needs for change (six major changes in the course of approximately 24 months) to this process that took place in recent history. Please note that despite the changes to the processes had occurred in the past, our approach still adapted to a need for change in advance in this setting because only the need for change was used as input. The actual planned and conducted processes and the resulting changes to the initial process just served for comparing our adaptation result with a real-world reference. To assess the operational feasibility of our approach, we analyzed the following three questions necessary for a valid application of our approach in this real-world scenario:

(E3.1) Is our approach able to adapt the process graph to the needs for change stated by the manager?

Based upon the information given by the manager, we were able to determine feasible atomic changes and specify them in terms of the aforementioned XML files. Subsequently, we adapted the process graph by means of our prototype in the order the changes occurred in reality (cf. Table A1). The first adaptation resulted from the demand to consider the situation of prefabricated packaging being in stock (an update of the initial state) and led to 15 old and 5 updated belief states in the adapted process graph. Thereafter, based on the adapted process graph we addressed the second need for change and so on. Overall, with respect to Table 1, the feasible atomic changes “updating the initial state”, “adding an action”, “removing an action” and “updating a goal state” were addressed. All needs for change could be represented (i.e., decomposed into feasible atomic changes) and addressed.

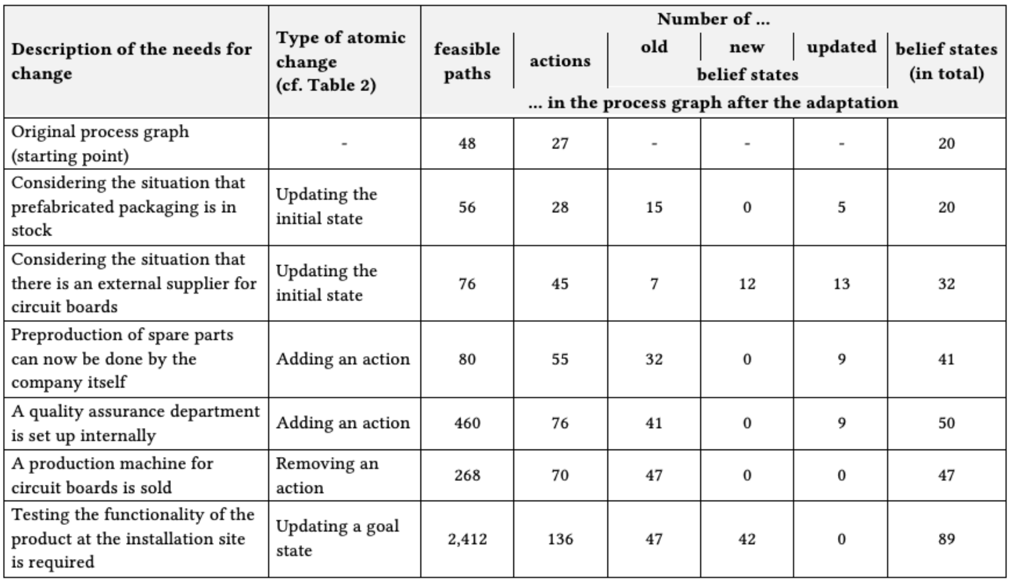
(E3.2) Do the adapted process graphs represent the actually conducted processes?

In order to compare the adapted process graphs with the respective actual processes of the company after each change, we scheduled a further meeting with the company’s staff. After the first adaptation (cf. third row in Table A1), we presented the resulting process graph to the staff and discussed the differences with them. However, we observed that the staff had some problems comprehending this graph. Therefore, for the subsequent adaptations (cf. rows four to eight in Table A1), we visually simplified the adapted process graphs so that they became more understandable for the staff. In detail, we removed the belief states from the versions presented to the company and omitted the preconditions and effects of the actions depicted in the graphs so that their layout was similar to UML activity diagrams. Still, the complete process model was presented. Then, we discussed these graphs with the manager and employees of the manufacturing department. Particularly, for each need for change (cf. Table A1), we elaborated the differences between the graph before adaptation and the adapted graph in detail. Furthermore, we asked the staff whether the adapted graphs represent the processes as they were actually conducted in the company once the according change took place. The staff confirmed this for every case.

(E3.3) Are the adapted process graphs assessed as correct and complete by the staff?

The staff further assessed the paths in the adapted process graphs to be correct. We also asked whether the adapted graphs neglected any feasible paths. Here, the staff validated that the graphs contained all paths actually used in the company and that no feasible paths not represented in the adapted process graphs were known. Please note that while the number of paths to check was high in some cases, most paths just contained the same actions in different order, making a manual verification possible.

Table A1. Adaptations performed in the case of the Engineering Company



Description of the needs for change	Type of atomic change (cf. Table 2)	Number of ... feasible paths	actions	old belief states	new belief states	updated belief states	belief states (in total)
		... in the process graph after the adaptation					
Original process graph (starting point)	-	48	27	-	-	-	20
Considering the situation that prefabricated packaging is in stock	Updating the initial state	56	28	15	0	5	20
Considering the situation that there is an external supplier for circuit boards	Updating the initial state	76	45	7	12	13	32
Preproduction of spare parts can now be done by the company itself	Adding an action	80	55	32	0	9	41
A quality assurance department is set up internally	Adding an action	460	76	41	0	9	50
A production machine for circuit boards is sold	Removing an action	268	70	47	0	0	47
Testing the functionality of the product at the installation site is required	Updating a goal state	2,412	136	47	42	0	89

Appendix B Evaluation of Computational Complexity (E4.1)

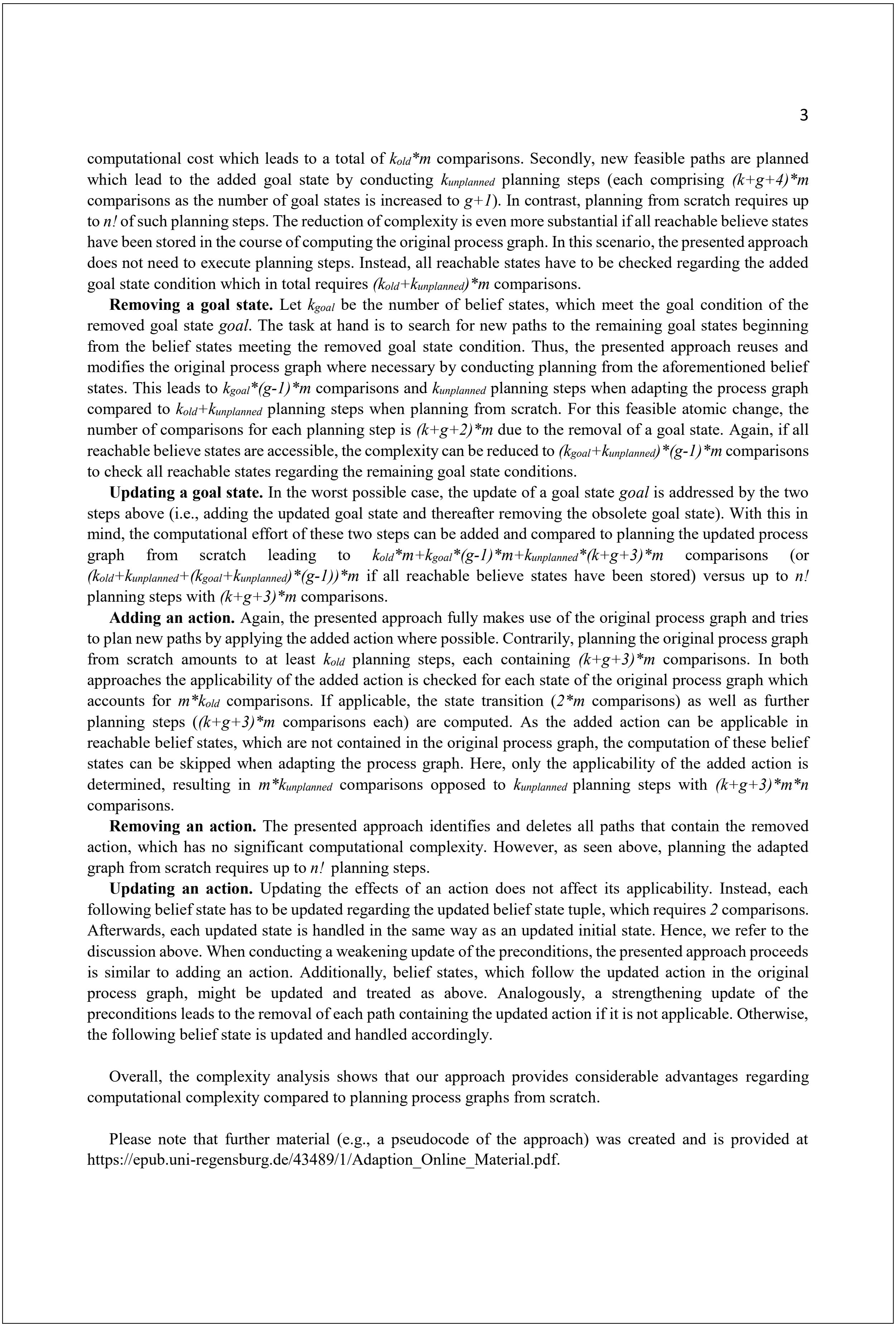
In the following, we outline the differences in complexity between the presented adaptation of a process graph and planning the adapted process graph from scratch. To this end, we use the notation found in Table A2. If necessary, further notation is provided for each adaptation case.

Table A2. Notation

k	Number of belief states that are planned or otherwise known during planning
k_{old}	Number of belief states in the original process graph
$k_{unplanned}$	Number of belief states which are reachable from the initial state, but not planned in the process graph (i.e., they do not lead to a goal state)
k_{goal}	Number of belief states meeting the goal condition of *goal*
n	Number of all actions
n_{old}	Number of actions applicable in bs_{old}
m	Number of all belief state variables
g	Number of goal states

Updating the initial state. Let n_{old}=|$app(bs_{old})$| be the number of actions applicable in an old belief state. Evaluating the applicability in such a belief state can be done just for the updated belief state tuple. The same holds for the application of the transition function. Hence, these two steps require no more than $3*n_{old}$ comparisons using the presented approach versus $3*m*n_{old}$ comparisons when planning from scratch. Having determined the following belief states to each applicable action, the effort of checking whether these states are already planned or meeting a goal state condition is the same for both approaches with up to $(k+g)*m*n_{old}$ comparisons. Please note that for every belief state which is contained in the original process model the entire subgraph is adopted by our adaptation approach which takes (virtually) no effort. In contrast, when planning from scratch in a worst case scenario every combination of actions is feasible and thus $n!$ planning steps are required with each planning step consisting of $(k+g+3)*m$ comparisons.

Adding a goal state. As shown in Section 4.2.1, adding a goal state is addressed in two possible ways. Firstly, paths are shortened by checking each belief state of the existing process graph for the goal condition of the added goal state. Once this check yields true, removing all following edges and nodes is of insignificant



computational cost which leads to a total of $k_{old}*m$ comparisons. Secondly, new feasible paths are planned which lead to the added goal state by conducting $k_{unplanned}$ planning steps (each comprising $(k+g+4)*m$ comparisons as the number of goal states is increased to $g+1$). In contrast, planning from scratch requires up to $n!$ of such planning steps. The reduction of complexity is even more substantial if all reachable believe states have been stored in the course of computing the original process graph. In this scenario, the presented approach does not need to execute planning steps. Instead, all reachable states have to be checked regarding the added goal state condition which in total requires $(k_{old}+k_{unplanned})*m$ comparisons.

Removing a goal state. Let k_{goal} be the number of belief states, which meet the goal condition of the removed goal state *goal*. The task at hand is to search for new paths to the remaining goal states beginning from the belief states meeting the removed goal state condition. Thus, the presented approach reuses and modifies the original process graph where necessary by conducting planning from the aforementioned belief states. This leads to $k_{goal}*(g\text{-}1)*m$ comparisons and $k_{unplanned}$ planning steps when adapting the process graph compared to $k_{old}+k_{unplanned}$ planning steps when planning from scratch. For this feasible atomic change, the number of comparisons for each planning step is $(k+g+2)*m$ due to the removal of a goal state. Again, if all reachable believe states are accessible, the complexity can be reduced to $(k_{goal}+k_{unplanned})*(g\text{-}1)*m$ comparisons to check all reachable states regarding the remaining goal state conditions.

Updating a goal state. In the worst possible case, the update of a goal state *goal* is addressed by the two steps above (i.e., adding the updated goal state and thereafter removing the obsolete goal state). With this in mind, the computational effort of these two steps can be added and compared to planning the updated process graph from scratch leading to $k_{old}*m+k_{goal}*(g\text{-}1)*m+k_{unplanned}*(k+g+3)*m$ comparisons (or $(k_{old}+k_{unplanned}+(k_{goal}+k_{unplanned})*(g\text{-}1))*m$ if all reachable believe states have been stored) versus up to $n!$ planning steps with $(k+g+3)*m$ comparisons.

Adding an action. Again, the presented approach fully makes use of the original process graph and tries to plan new paths by applying the added action where possible. Contrarily, planning the original process graph from scratch amounts to at least k_{old} planning steps, each containing $(k+g+3)*m$ comparisons. In both approaches the applicability of the added action is checked for each state of the original process graph which accounts for $m*k_{old}$ comparisons. If applicable, the state transition ($2*m$ comparisons) as well as further planning steps ($(k+g+3)*m$ comparisons each) are computed. As the added action can be applicable in reachable belief states, which are not contained in the original process graph, the computation of these belief states can be skipped when adapting the process graph. Here, only the applicability of the added action is determined, resulting in $m*k_{unplanned}$ comparisons opposed to $k_{unplanned}$ planning steps with $(k+g+3)*m*n$ comparisons.

Removing an action. The presented approach identifies and deletes all paths that contain the removed action, which has no significant computational complexity. However, as seen above, planning the adapted graph from scratch requires up to $n!$ planning steps.

Updating an action. Updating the effects of an action does not affect its applicability. Instead, each following belief state has to be updated regarding the updated belief state tuple, which requires *2* comparisons. Afterwards, each updated state is handled in the same way as an updated initial state. Hence, we refer to the discussion above. When conducting a weakening update of the preconditions, the presented approach proceeds is similar to adding an action. Additionally, belief states, which follow the updated action in the original process graph, might be updated and treated as above. Analogously, a strengthening update of the preconditions leads to the removal of each path containing the updated action if it is not applicable. Otherwise, the following belief state is updated and handled accordingly.

Overall, the complexity analysis shows that our approach provides considerable advantages regarding computational complexity compared to planning process graphs from scratch.

Please note that further material (e.g., a pseudocode of the approach) was created and is provided at https://epub.uni-regensburg.de/43489/1/Adaption_Online_Material.pdf.

6 Paper 4: Data Quality in Recommender Systems: The Impact of Completeness of Item Content Data on Prediction Accuracy of Recommender Systems

Electronic Markets – The International Journal on Networked Business

Full Title of Article:	Data Quality in Recommender Systems: The Impact of Completeness of Item Content Data on Prediction Accuracy of Recommender Systems
Subtitle (optional):	
Preferred Abbreviated Title for Running Head (maximum of 65 characters including spaces)	Data Quality in Recommender Systems
Key Words (for indexing and abstract services – up to 6 words):	Completeness, Data Quality, Prediction Accuracy, Recommender Systems
JEL classification	C80 Data Collection and Data Estimation Methodology; Computer Programs: General
Word Count	11640
Word Processing Program Name and Version Number:	Microsoft Office Word Professional Plus 2016

Abstract:

Recommender systems strive to guide users, especially in the field of e-commerce, to their individually best choice when a large number of alternatives is available. In general, literature suggests that the quality of data which a recommender system is based on may have important impact on recommendation quality. In this paper, we focus on the data quality dimension completeness of item content data (i.e., features of items and their feature values) and investigate its impact on the prediction accuracy of recommender systems. In particular, we examine the increase in completeness per item, per user and per feature as moderators for this impact. To this end, we present a theoretical model based on the literature and derive ten hypotheses. We test these hypotheses on two real-world data sets, one from two leading web portals for restaurant reviews and another one from a movie review portal. The results strongly support that, in general, the prediction accuracy is positively influenced by increased completeness. However, the results also reveal, contrary to existing literature, that among others

increasing completeness by adding features which differ significantly from already existing features (i.e., a high diversity) does not positively influence the prediction accuracy of recommender systems.

Introduction

Recommender systems strive to guide users to their individually best choice when a large number of alternatives is available. Due to a broad variety of interesting problem settings for scholars and a plethora of practical applications, recommender systems continue to be a topic widely discussed in literature (Adomavicius and Tuzhilin 2005; Bobadilla et al. 2013; Karatzoglou and Hidasi 2017). For example, in recent years, many of these practical applications have been in the field of e-commerce and electronic markets (Li and Karahanna 2015; Lu et al. 2015; Ricci et al. 2011). Thereby, recommender systems “have become one of the most powerful and popular tools” (Ricci et al. 2011), mainly because of the large amount of available data about items (e.g., songs or movies). Here, usually, a choice amongst an abundance of items needs to be made, which has inspired providers such as *Netflix* or *Spotify* to develop elaborate recommender systems (Bell et al. 2007; Gomez-Uribe and Hunt 2016; Song et al. 2013). Similarly, recommender systems can assist users in their choice of which restaurant to visit or in which hotel to stay (Levi et al. 2012; Vargas-Govea et al. 2011). In this context, several works suggest that the quality of the determined recommendations depends on the quality of the data which a recommender system is based on (Adomavicius and Zhang 2012; Felfernig et al. 2007; Konstan and Riedl 2012; Picault et al. 2011; Sar Shalom et al. 2015). As discussed by Jannach et al. (2016), these works mainly investigate the data quality of rating data (e.g., how to achieve the most accurate completion of the user-item matrix with rating predictions) and therefore, propose to leverage additional user data such as the user’s context, the user’s browsing history or the user’s social graph. In contrast to these articles mainly discussing data quality of user or rating data (cf. Section “Background”), this paper focuses on data quality of *item content* data, which means, features of items such as *Genre* or *Actors* of movies and their feature values.

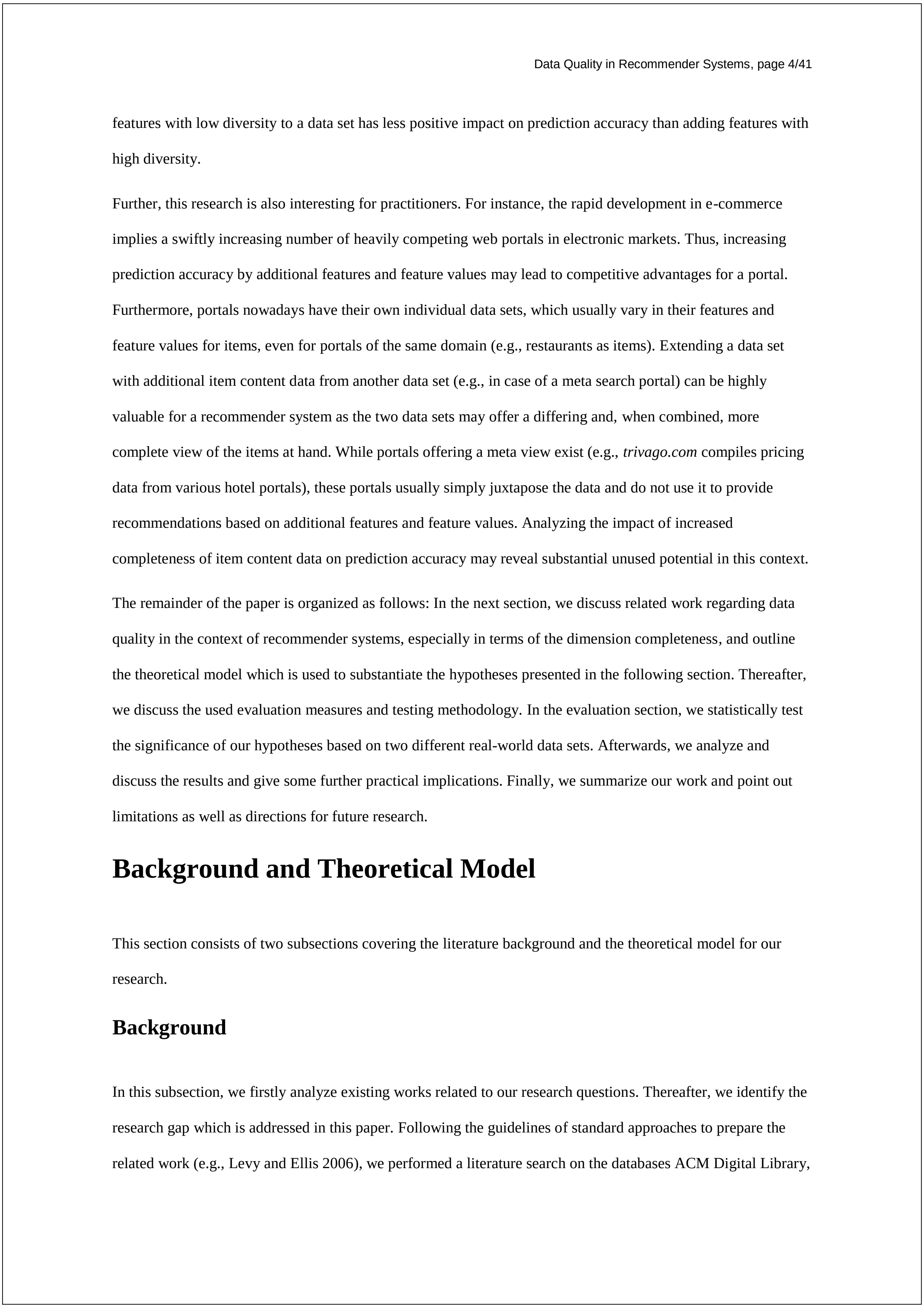
In general, data quality constitutes a multidimensional construct (Pipino et al. 2002; Wand and Wang 1996; Wang et al. 1995) comprising several dimensions such as correctness, completeness and currency of data (Batini and Scannapieco 2016; Heinrich et al. 2018b; Lee et al. 2002; Redman 1996). Some existing works investigate and assess the impact of data quality and its dimensions in decision making (Feldman et al. 2018; Heinrich and Hristova 2016). As recommender systems are an important category of decision support systems, especially in electronic markets, we aim to examine the impact of *item content* data and their quality on the determined recommendations. Here, capturing a more complete view of this item content data (i.e. more available features and feature values) is of particular relevance (Adomavicius and Tuzhilin 2005; Pazzani and Billsus 2007; Picault et al. 2011). After all, “some representations capture only certain aspects of the content, but there are many others that would influence a user’s experience” (Lops et al. 2011). Hence, in this paper, we focus on the data quality dimension *completeness*.

Batini et al. (2009) summarize that completeness can be understood as the amount to which an available data view includes data describing the corresponding set of considered real-world objects (cf., e.g., also Ballou and Pazer 1985; Redman 1996). Following this definition, we aim to investigate the impact of completeness on recommendation quality, with completeness being the amount of available features and their feature values describing the set of items. For instance, the movie feature *Genre* has multiple feasible feature values such as *Comedy*, *Drama*, *Thriller* and so forth, while the restaurant feature *Cuisine* has multiple feasible feature values such as *Italian, American* or *Mexican*. Providers covering such domains typically assign such feature values to items in order to describe and emphasize their (special) characteristics and thus, allow a more complete view on these items. Moreover, to assess the impact of completeness on recommendation quality, we examine the prediction accuracy, which is by far the most discussed quality measure in recommender systems literature (Shani and Gunawardana 2011). In this paper, prediction accuracy is assessed by the familiar evaluation measures Root Mean Squared Error (RMSE), Precision, Recall and F1-measure enabling a broad but also differentiated analysis of the results. To the best of our knowledge, no existing work analyzes the impact of the amount of available features or feature values (*completeness of item content data*) on prediction accuracy. Thus, we focus on the following two research questions:

RQa: *Does the amount of available item features influence the prediction accuracy of recommender systems?*

RQb: *Does the amount of filled up missing item feature values influence the prediction accuracy of recommender systems?*

We address these research questions by formulating ten hypotheses based on a theoretical model derived from the literature. Further, we test the statistical significance of these hypotheses by means of both a t-test and a moderated regression analysis concerning the impact of the amount of available item features and their feature values on prediction accuracy. The results show that completeness of the item content data generally has a significant positive impact on prediction accuracy. However, the results also reveal some findings which are contrary to statements in existing literature (Mitra et al. 2002; Tabakhi and Moradi 2015) stating that adding



features with low diversity to a data set has less positive impact on prediction accuracy than adding features with high diversity.

Further, this research is also interesting for practitioners. For instance, the rapid development in e-commerce implies a swiftly increasing number of heavily competing web portals in electronic markets. Thus, increasing prediction accuracy by additional features and feature values may lead to competitive advantages for a portal. Furthermore, portals nowadays have their own individual data sets, which usually vary in their features and feature values for items, even for portals of the same domain (e.g., restaurants as items). Extending a data set with additional item content data from another data set (e.g., in case of a meta search portal) can be highly valuable for a recommender system as the two data sets may offer a differing and, when combined, more complete view of the items at hand. While portals offering a meta view exist (e.g., *trivago.com* compiles pricing data from various hotel portals), these portals usually simply juxtapose the data and do not use it to provide recommendations based on additional features and feature values. Analyzing the impact of increased completeness of item content data on prediction accuracy may reveal substantial unused potential in this context.

The remainder of the paper is organized as follows: In the next section, we discuss related work regarding data quality in the context of recommender systems, especially in terms of the dimension completeness, and outline the theoretical model which is used to substantiate the hypotheses presented in the following section. Thereafter, we discuss the used evaluation measures and testing methodology. In the evaluation section, we statistically test the significance of our hypotheses based on two different real-world data sets. Afterwards, we analyze and discuss the results and give some further practical implications. Finally, we summarize our work and point out limitations as well as directions for future research.

Background and Theoretical Model

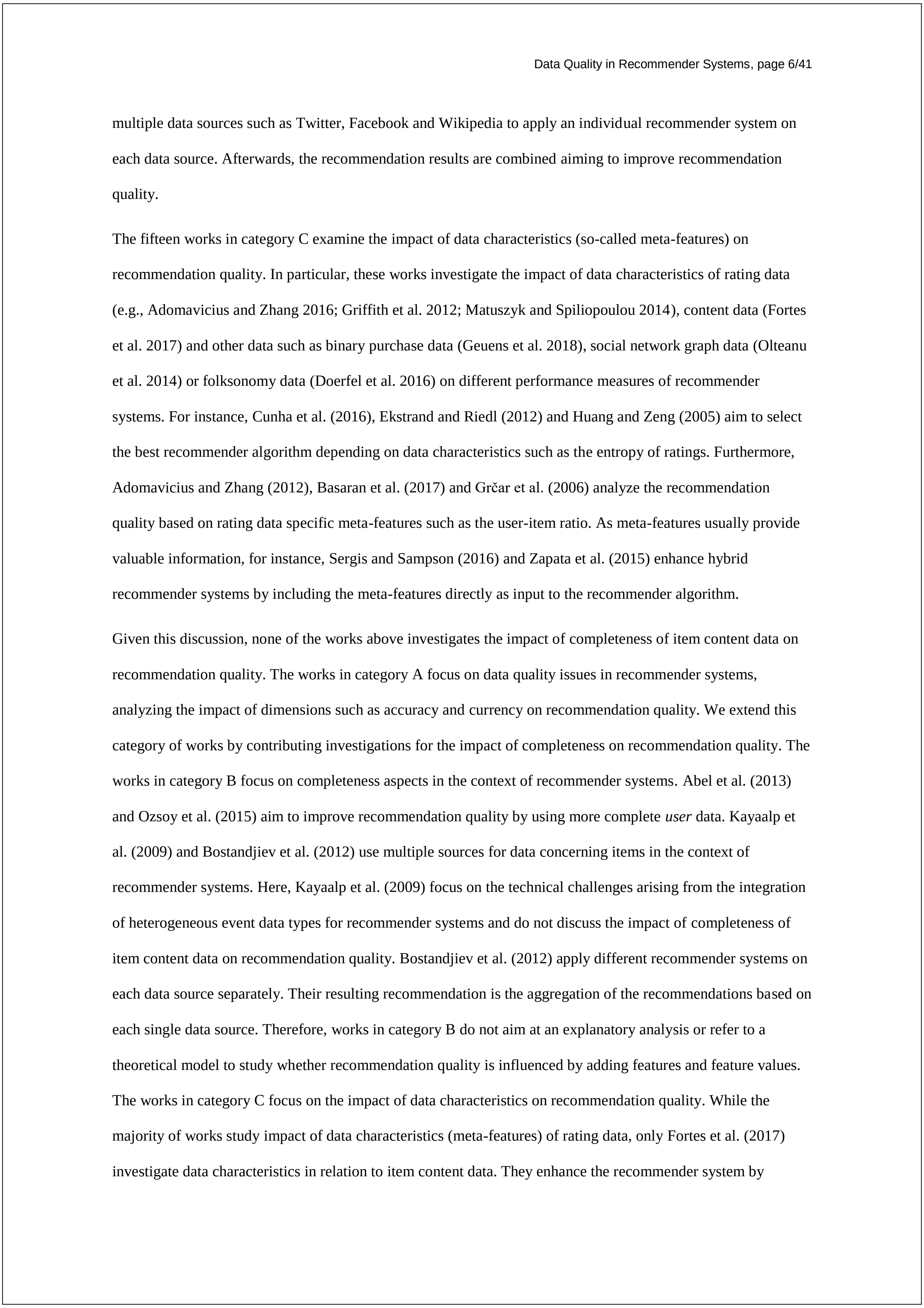
This section consists of two subsections covering the literature background and the theoretical model for our research.

Background

In this subsection, we firstly analyze existing works related to our research questions. Thereafter, we identify the research gap which is addressed in this paper. Following the guidelines of standard approaches to prepare the related work (e.g., Levy and Ellis 2006), we performed a literature search on the databases ACM Digital Library, AIS Electronic Library, IEEE Xplore, ScienceDirect and Springer as well as the proceedings of the European and International Conference on Information Systems, the ACM Conference on Recommender Systems and the International Conference on Information Quality. The resulting papers were examined based on title, abstract and keywords, leading to thirteen remaining papers. We performed an additional forward and backward search on these papers, leading to a total of twenty-seven relevant papers. These papers were analyzed in detail and could be organized within three categories A, B and C. Works of category A discuss data quality issues in the context of recommender systems, whereas works of category B present recommender systems which deal with a data set extended by using web data sources. Works of category C investigate the impact of data characteristics such as the entropy of the distribution of rating data on recommendation quality. In the following, we discuss the relevant papers of each category.

The eight works in category A explicitly recognize the importance of data quality for recommender systems from a *general* perspective (Amatriain et al. 2009; Berkovsky et al. 2012; Burke and Ramezani 2011; Konstan and Riedl 2012; Lathia et al. 2009; Levi et al. 2012; Pessemier et al. 2010; Sar Shalom et al. 2015), including several approaches that deal with data quality issues. For instance, as data sparsity and inaccuracy have been identified to influence recommendation quality, Lathia et al. (2009) suggest to choose data sources for the application of a recommender system user-dependently. Sar Shalom et al. (2015) tackle sparsity and redundancy issues by deleting or omitting certain users or items while Pessemier et al. (2010) analyze consumption data such as ratings in regard to currency. Further, Levi et al. (2012) use text mining on user reviews from various sources to alleviate the cold start problem of new users by assigning them to so-called context groups.

The four works in category B (implicitly) investigate completeness in recommender systems (Abel et al. 2013; Bostandjiev et al. 2012; Kayaalp et al. 2009; Ozsoy et al. 2015). More precisely, these works propose to use data from additional web sources to gain an extended data set and to increase recommendation quality in this way. Abel et al. (2013) study user profiles based on aggregated data sets from the social web and show that recommendation quality is improved by user profiles extended through several cross-system user-modelling strategies. Ozsoy et al. (2015) argue that recommendations can be improved by consolidating user data from multiple sources. In their experiments, they show that using multiple user features from several social networks produces an enhanced perspective of user behavior and preferences, leading to improved recommendations. Kayaalp et al. (2009) present an event recommender system for users of a social network. This system collects heterogeneous event data from various web pages to achieve an extended data set and proposes event recommendations on this basis. A further approach is proposed by Bostandjiev et al. (2012). They suggest to use



multiple data sources such as Twitter, Facebook and Wikipedia to apply an individual recommender system on each data source. Afterwards, the recommendation results are combined aiming to improve recommendation quality.

The fifteen works in category C examine the impact of data characteristics (so-called meta-features) on recommendation quality. In particular, these works investigate the impact of data characteristics of rating data (e.g., Adomavicius and Zhang 2016; Griffith et al. 2012; Matuszyk and Spiliopoulou 2014), content data (Fortes et al. 2017) and other data such as binary purchase data (Geuens et al. 2018), social network graph data (Olteanu et al. 2014) or folksonomy data (Doerfel et al. 2016) on different performance measures of recommender systems. For instance, Cunha et al. (2016), Ekstrand and Riedl (2012) and Huang and Zeng (2005) aim to select the best recommender algorithm depending on data characteristics such as the entropy of ratings. Furthermore, Adomavicius and Zhang (2012), Basaran et al. (2017) and Grčar et al. (2006) analyze the recommendation quality based on rating data specific meta-features such as the user-item ratio. As meta-features usually provide valuable information, for instance, Sergis and Sampson (2016) and Zapata et al. (2015) enhance hybrid recommender systems by including the meta-features directly as input to the recommender algorithm.

Given this discussion, none of the works above investigates the impact of completeness of item content data on recommendation quality. The works in category A focus on data quality issues in recommender systems, analyzing the impact of dimensions such as accuracy and currency on recommendation quality. We extend this category of works by contributing investigations for the impact of completeness on recommendation quality. The works in category B focus on completeness aspects in the context of recommender systems. Abel et al. (2013) and Ozsoy et al. (2015) aim to improve recommendation quality by using more complete *user* data. Kayaalp et al. (2009) and Bostandjiev et al. (2012) use multiple sources for data concerning items in the context of recommender systems. Here, Kayaalp et al. (2009) focus on the technical challenges arising from the integration of heterogeneous event data types for recommender systems and do not discuss the impact of completeness of item content data on recommendation quality. Bostandjiev et al. (2012) apply different recommender systems on each data source separately. Their resulting recommendation is the aggregation of the recommendations based on each single data source. Therefore, works in category B do not aim at an explanatory analysis or refer to a theoretical model to study whether recommendation quality is influenced by adding features and feature values. The works in category C focus on the impact of data characteristics on recommendation quality. While the majority of works study impact of data characteristics (meta-features) of rating data, only Fortes et al. (2017) investigate data characteristics in relation to item content data. They enhance the recommender system by including these data characteristics directly in the recommender algorithm as they aim for a predictive analysis. In contrast to the discussed works, which either focus on the consideration of *rating* data characteristics (e.g., entropy of rating distribution) or generate recommendations in a predictive analysis, we extend this category of works in two ways. Firstly, we explicitly investigate the impact of completeness of *item content* data on prediction accuracy. Secondly, we conduct an explanatory analysis based on causal hypotheses and a theoretical model, which strongly differs from predictive analytics (Shmueli and Koppius 2011). Both aspects have important implications in practice as the actual relevance of increasing the amount of available features and feature values for prediction accuracy is examined.

Theoretical Model

This subsection presents a theoretical model constituting a basis for the hypotheses discussed in the subsequent section. Research in the field of data quality shows an increasing tendency to study the impact of data quality of data views and data values (independent variable) on different evaluation criteria of decision support systems such as decision quality or data mining outcome (dependent variable) (e.g., Bharati and Chaudhury 2004; Blake and Mangiameli 2011; Feldman et al. 2018; Ge 2009; Woodall et al. 2015). More precisely, Blake and Mangiameli (2011) analyze the impact of the data quality dimensions accuracy, completeness, consistency and currency on data mining results in order to support decision-making. Woodall et al. (2015) investigate the impact of completeness on classification outcomes used for supporting users in their decision process. Bharati and Chaudhury (2004) examine the effects of accuracy, completeness and currency on the ability of an online analytical processing system to sustain decision-making. Ge (2009) focuses on accuracy, completeness and consistency and their impact on decision quality. Feldman et al. (2018) propose an analytical framework to investigate the impact of incomplete data sets on a binary classifier that serves for decision support.

The focus of these papers is to investigate in which way and to what extent the quality of data views and data values, especially the dimension completeness, influences evaluation criteria such as data mining outcome of particular decision support systems. Because recommender systems are a relevant category of decision support systems, especially in electronic markets, assisting users that face decision-making problems (Porcel and Herrera-Viedma 2010; Power et al. 2015), we derive the theoretical model from these works to examine the impact of completeness of item content data on prediction accuracy of recommender systems. Figure 1 presents this theoretical model.

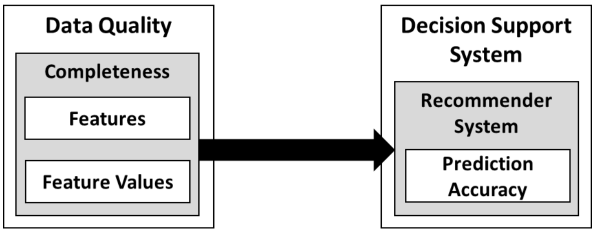


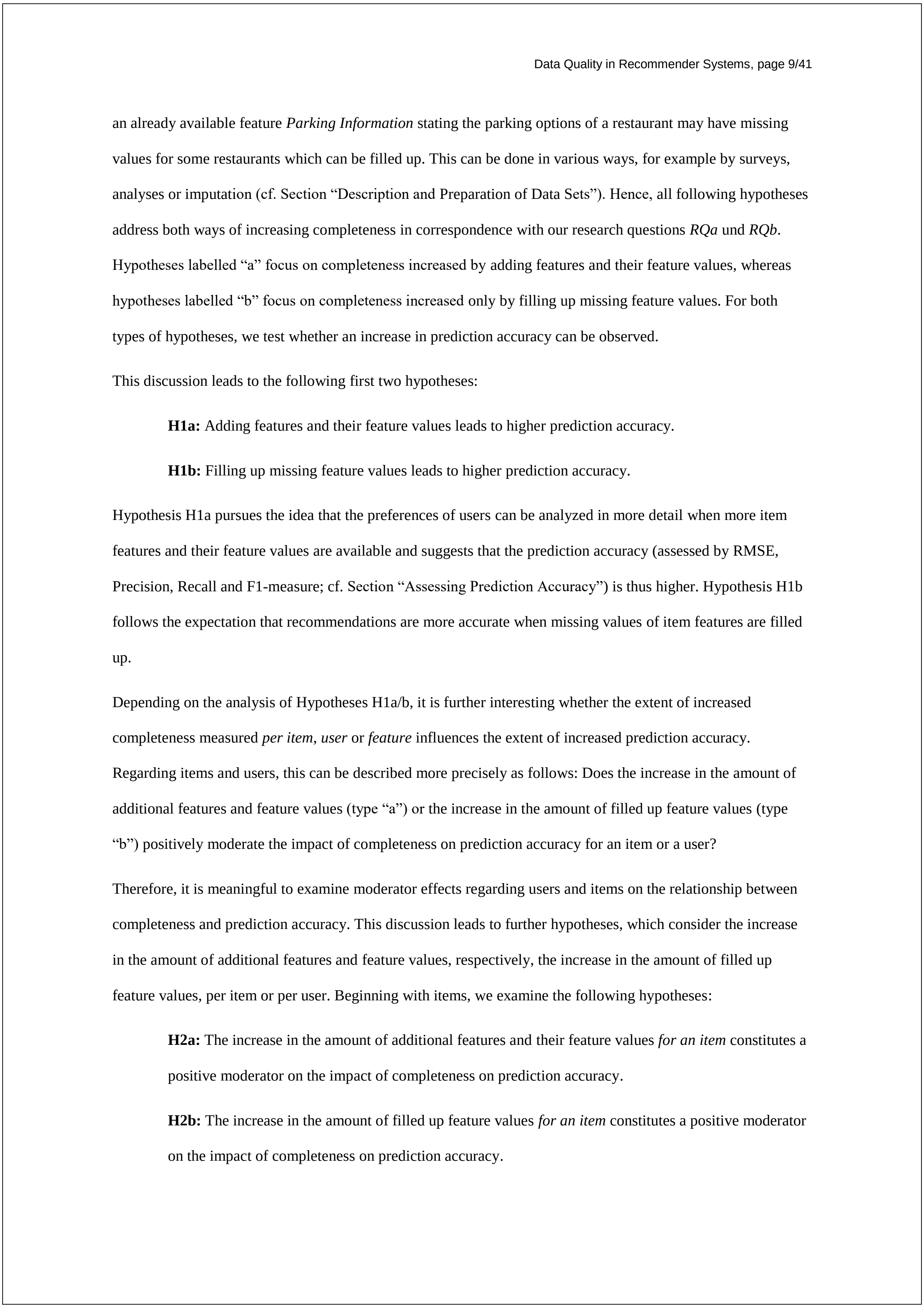
Figure 1. Theoretical Model

In the context of decision support systems, completeness is a frequently investigated dimension of data quality (Blake and Mangiameli 2011; Feldman et al. 2018; Ge 2009; Woodall et al. 2015). These works refer to completeness as the amount of available data views and data values. We take up this idea in the theoretical model and consider completeness by the amount of features and their feature values (cf. left side of Figure 1). As discussed above, features such as *Cuisine* can have multiple feasible feature values such as *Italian*, *American* or *Mexican*, which are assigned to items in order to describe and underline their characteristics enabling a more complete view on these items. Therefore, we focus on such features and their feature values when analyzing completeness. Similar to Bharati and Chaudhury (2004) and Ge (2009), the presented theoretical model in Figure 1 indicates a direct relation between data quality and evaluation criteria of decision support systems. In particular, the theoretical model suggests this relation between completeness of item content data and prediction accuracy of recommender systems (cf. right side of Figure 1). This model constitutes the foundation for the following hypotheses and is customized by different moderator variables to allow for a detailed analysis.

Hypotheses

Based on the theoretical model, we present ten hypotheses to address our research questions. Each hypothesis examines the impact of completeness of item content data on prediction accuracy from a different angle. Figure 2 at the end of this section shows an overview of all hypotheses.

Content-based and hybrid recommender systems, two major categories of recommender systems (Ning et al. 2015), operate on item content data to propose items to users that they are likely to be interested in (Lops et al. 2011). For this kind of data, increased completeness means that more features and/or more feature values are assigned to items (cf. Section “Theoretical Model”). Thus, increased completeness in this sense can be achieved in two ways: First, by adding features and their feature values to the feature set. For instance, a feature *Actors* can be added to the feature set for the movie domain. Second, by filling up missing feature values. For example,



an already available feature *Parking Information* stating the parking options of a restaurant may have missing values for some restaurants which can be filled up. This can be done in various ways, for example by surveys, analyses or imputation (cf. Section “Description and Preparation of Data Sets”). Hence, all following hypotheses address both ways of increasing completeness in correspondence with our research questions *RQa* und *RQb*. Hypotheses labelled “a” focus on completeness increased by adding features and their feature values, whereas hypotheses labelled “b” focus on completeness increased only by filling up missing feature values. For both types of hypotheses, we test whether an increase in prediction accuracy can be observed.

This discussion leads to the following first two hypotheses:

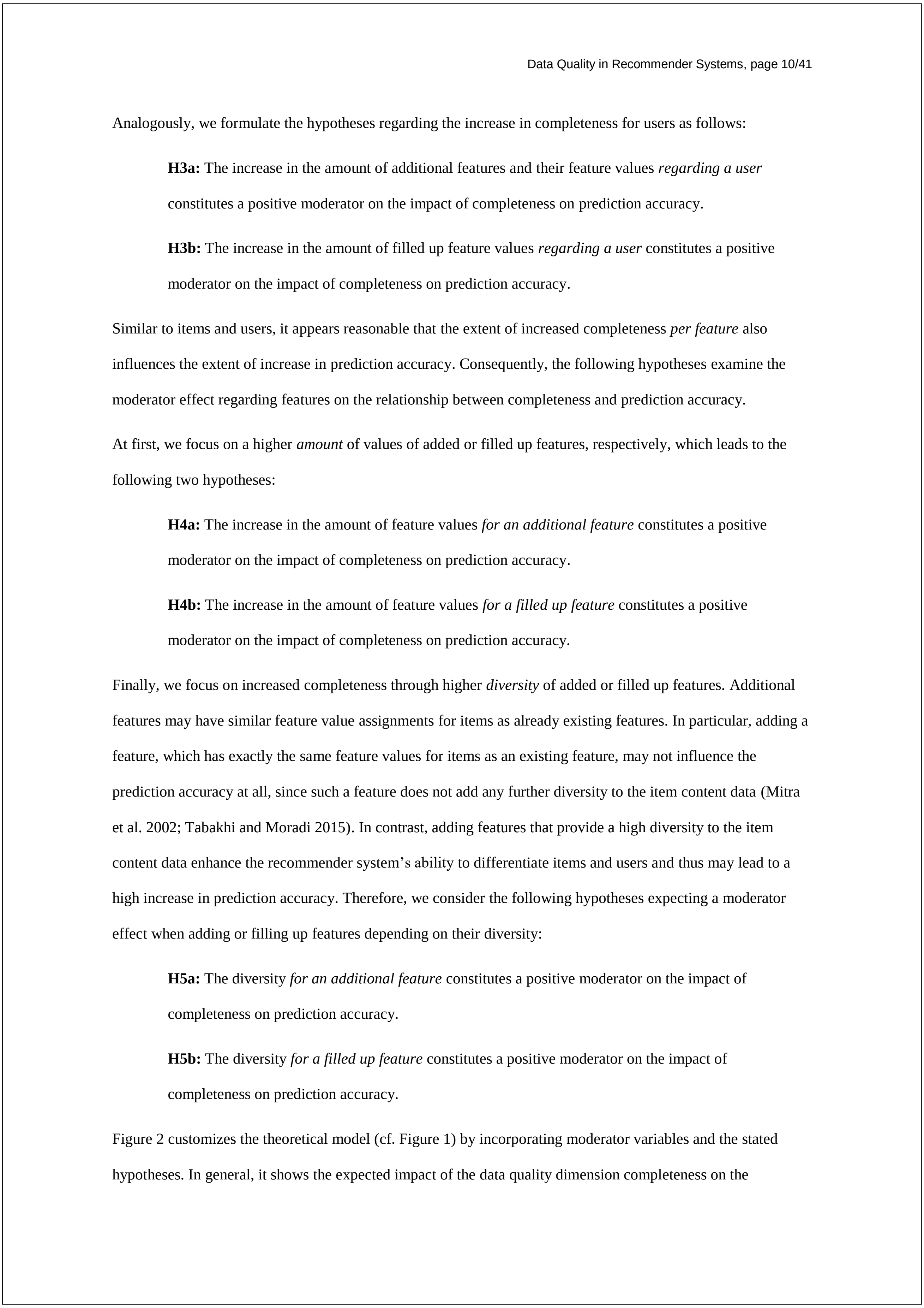
> **H1a:** Adding features and their feature values leads to higher prediction accuracy.
>
> **H1b:** Filling up missing feature values leads to higher prediction accuracy.

Hypothesis H1a pursues the idea that the preferences of users can be analyzed in more detail when more item features and their feature values are available and suggests that the prediction accuracy (assessed by RMSE, Precision, Recall and F1-measure; cf. Section “Assessing Prediction Accuracy”) is thus higher. Hypothesis H1b follows the expectation that recommendations are more accurate when missing values of item features are filled up.

Depending on the analysis of Hypotheses H1a/b, it is further interesting whether the extent of increased completeness measured *per item*, *user* or *feature* influences the extent of increased prediction accuracy. Regarding items and users, this can be described more precisely as follows: Does the increase in the amount of additional features and feature values (type “a”) or the increase in the amount of filled up feature values (type “b”) positively moderate the impact of completeness on prediction accuracy for an item or a user?

Therefore, it is meaningful to examine moderator effects regarding users and items on the relationship between completeness and prediction accuracy. This discussion leads to further hypotheses, which consider the increase in the amount of additional features and feature values, respectively, the increase in the amount of filled up feature values, per item or per user. Beginning with items, we examine the following hypotheses:

> **H2a:** The increase in the amount of additional features and their feature values *for an item* constitutes a positive moderator on the impact of completeness on prediction accuracy.
>
> **H2b:** The increase in the amount of filled up feature values *for an item* constitutes a positive moderator on the impact of completeness on prediction accuracy.



Analogously, we formulate the hypotheses regarding the increase in completeness for users as follows:

> **H3a:** The increase in the amount of additional features and their feature values *regarding a user* constitutes a positive moderator on the impact of completeness on prediction accuracy.
>
> **H3b:** The increase in the amount of filled up feature values *regarding a user* constitutes a positive moderator on the impact of completeness on prediction accuracy.

Similar to items and users, it appears reasonable that the extent of increased completeness *per feature* also influences the extent of increase in prediction accuracy. Consequently, the following hypotheses examine the moderator effect regarding features on the relationship between completeness and prediction accuracy.

At first, we focus on a higher *amount* of values of added or filled up features, respectively, which leads to the following two hypotheses:

> **H4a:** The increase in the amount of feature values *for an additional feature* constitutes a positive moderator on the impact of completeness on prediction accuracy.
>
> **H4b:** The increase in the amount of feature values *for a filled up feature* constitutes a positive moderator on the impact of completeness on prediction accuracy.

Finally, we focus on increased completeness through higher *diversity* of added or filled up features. Additional features may have similar feature value assignments for items as already existing features. In particular, adding a feature, which has exactly the same feature values for items as an existing feature, may not influence the prediction accuracy at all, since such a feature does not add any further diversity to the item content data (Mitra et al. 2002; Tabakhi and Moradi 2015). In contrast, adding features that provide a high diversity to the item content data enhance the recommender system’s ability to differentiate items and users and thus may lead to a high increase in prediction accuracy. Therefore, we consider the following hypotheses expecting a moderator effect when adding or filling up features depending on their diversity:

> **H5a:** The diversity *for an additional feature* constitutes a positive moderator on the impact of completeness on prediction accuracy.
>
> **H5b:** The diversity *for a filled up feature* constitutes a positive moderator on the impact of completeness on prediction accuracy.

Figure 2 customizes the theoretical model (cf. Figure 1) by incorporating moderator variables and the stated hypotheses. In general, it shows the expected impact of the data quality dimension completeness on the prediction accuracy as stated by Hypotheses H1a/b. Additionally, it illustrates the hypotheses examining a moderating effect for the increase in completeness per item (Hypotheses H2a/b), per user (Hypotheses H3a/b) and per feature (Hypotheses H4a/b, H5a/b).

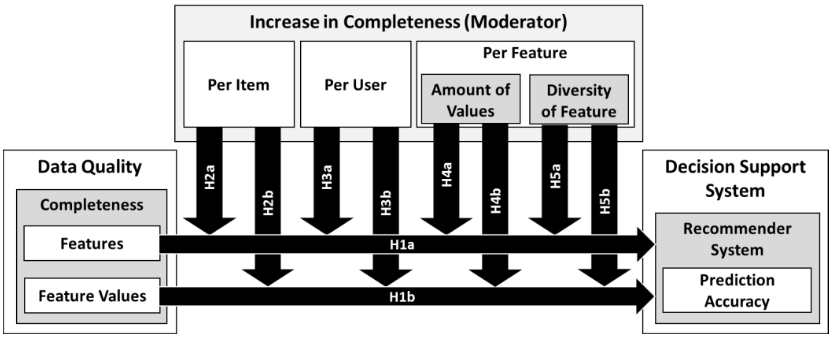


Figure 2. Overview for Hypotheses H1-H5

Methodology

In this section, we introduce the models used to test Hypotheses H1-H5. To do so and to assess prediction accuracy as the dependent variable, we first discuss selected measures which allow differentiated analyses and interpretations regarding the impact on prediction accuracy. Thereafter, we describe the testing methodology for Hypotheses H1a/b as well as the regression models for testing Hypotheses H2-H5.

Assessing Prediction Accuracy

To enable a detailed and careful analysis of the results of the Hypotheses H1-H5, we assessed prediction accuracy by means of different measures from literature, namely RMSE, Precision, Recall and F1-measure (Gunawardana and Shani 2015). RMSE as shown in Equation (1) is one of the most popular measures for assessing prediction accuracy (Gunawardana and Shani 2015) and is defined by the term

$$RMSE = \sqrt{\frac{1}{|T|} \cdot \sum_{(u,i)\in T} (\hat{r}_{ui} - r_{ui})^2}, \tag{1}$$

where T is a test set of user-item pairs (u, i) for which the ratings $\hat{r}_{ui}$ are predicted by the recommender system and the actual ratings r_{ui} are known. RMSE received special attention by the Netflix Prize Challenge in 2006 (Koren 2009). Its main characteristic is that higher errors (i.e., the difference between predicted and actual rating) are weighted stronger through its quadratic structure than lower errors. Further, usually the predicted ratings $\hat{r}_{ui}$ are continuous (real-valued) and the actual ratings r_{ui} are discrete (and ordered). Hence, minor RMSE value changes may not result in a different mapping (by rounding) of the continuous predicted rating $\hat{r}_{ui}$ to a discrete star rating $\widehat{dr}_{ui} \in \{1, \dots, 5\}$. This means that the mapping to a discrete star rating may not change, even with an improved RMSE value. Therefore, it is also necessary to assess whether the mapping of continuous predicted ratings $\hat{r}_{ui}$ to discrete star ratings $\widehat{dr}_{ui}$ changes or improves with the increase in completeness and the expected increase in prediction accuracy. To evaluate this, Precision, Recall, and F1-measure are the most important measures. These measures assess whether or not the predicted rating level $\widehat{dr}_{ui}$ exactly coincides with the actual true rating level r_{ui} for each user-item pair (u, i) (Aggarwal 2014). Precision and Recall are calculated as the average of the Precision and Recall values for each star rating level $k \in \{1,2,3,4,5\}$, which are given by the following terms.

$$Precision_k = \frac{TP_k}{TP_k + FP_k} \tag{2}$$

$$Recall_k = \frac{TP_k}{TP_k + FN_k} \tag{3}$$

Here, TP_k is the number of user-item pairs (u, i) with $r_{ui} = k$ and $\widehat{dr}_{ui} = k$ (“true positives”), FP_k as shown in Equation (2) is the number of user-item pairs (u, i) with $r_{ui} \neq k$ and $\widehat{dr}_{ui} = k$ (“false positives”), and FN_k as shown in Equation (3) is the number of user-item pairs (u, i) with $r_{ui} = k$ and $\widehat{dr}_{ui} \neq k$ (“false negatives”). F1-measure as shown in Equation (4) is then given by the harmonic mean of Precision and Recall

$$F1 = 2 \cdot \frac{Precision \ \cdot \ Recall}{Precision \ + \ Recall}. \tag{4}$$

The main difference in interpretation of these measures is that the Precision, Recall and F1-measure focus on correct or incorrect mappings of predicted and actual star ratings while ignoring the (real-valued) error size, which is in the focus of RMSE.

Model for Hypotheses H1a/b

Each of the Hypotheses H1a and H1b focuses on a comparison of the prediction accuracy of two item content data sets, one data set without increased completeness and the other data set with increased completeness (cf. Figure 3). In both cases, we initially do not consider any moderator variable. To test the significance of both hypotheses, we used the paired Student’s t-test, a broadly applied test in the evaluation of recommender systems to compare the results of two different settings, while in both settings the considered set of user ratings remains the same (Shani and Gunawardana 2011). More precisely, the t-test was used to compare each of the measures RMSE, Precision, Recall and the F1-measure (and thus the prediction accuracy) based on the data set with increased completeness (i.e., when adding features and their feature values or when filling up missing feature values) and based on the data set without increased completeness.

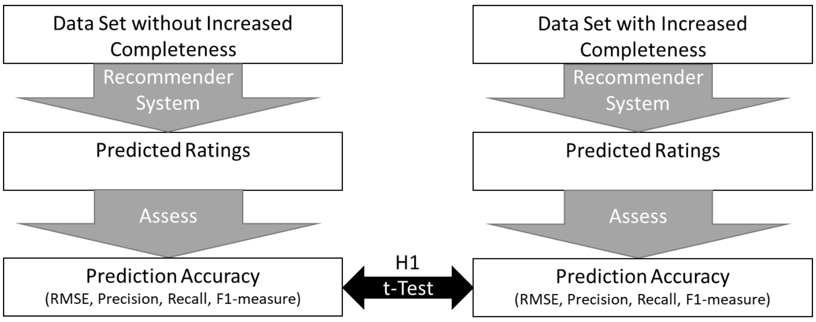

Data Set without Increased Completeness
Data Set with Increased Completeness
Recommender System
Recommender System
Predicted Ratings
Predicted Ratings
Assess
Assess
Prediction Accuracy (RMSE, Precision, Recall, F1-measure)
H1
t-Test
Prediction Accuracy (RMSE, Precision, Recall, F1-measure)

Figure 3. Testing Hypotheses H1a/b

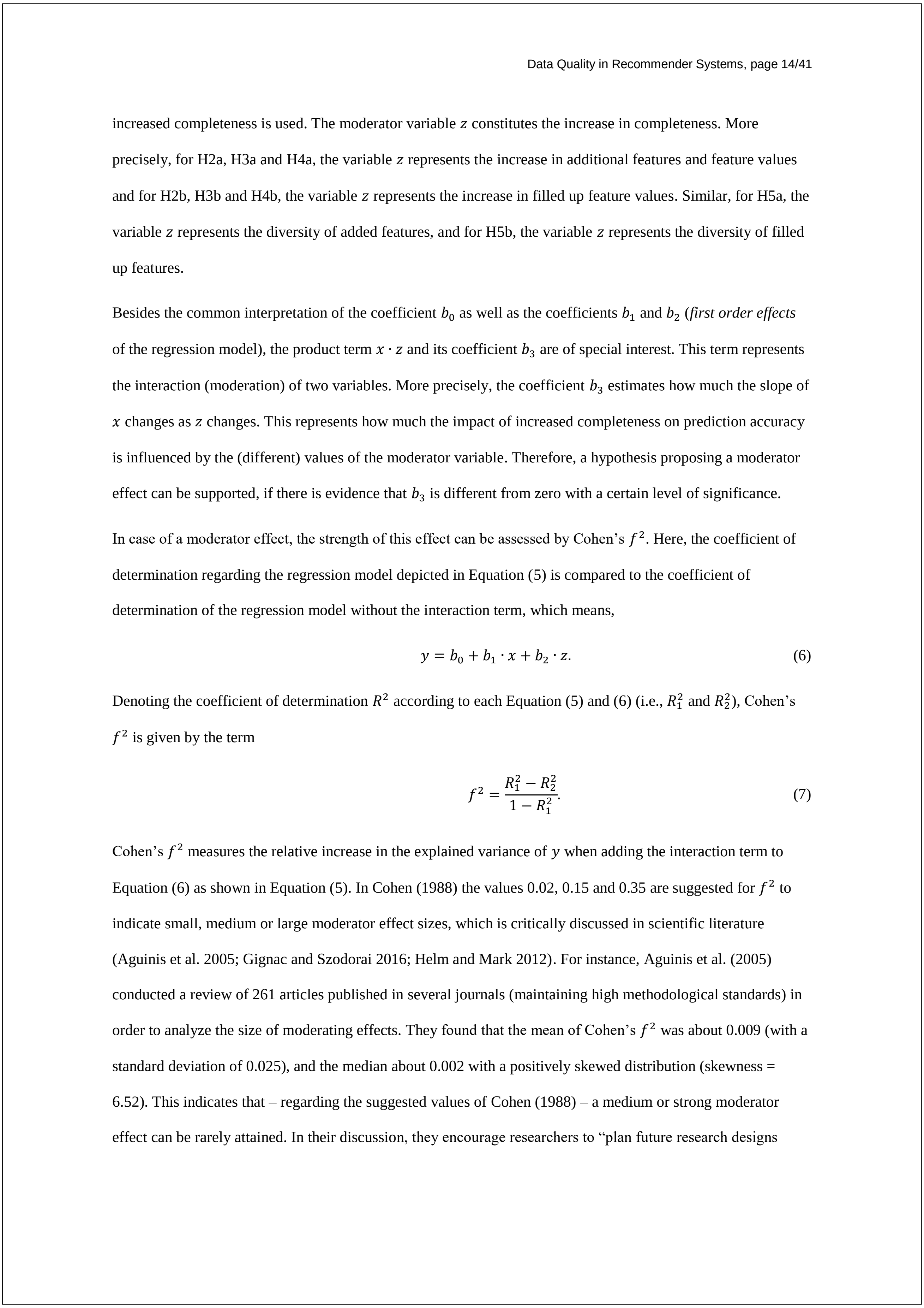
Model for Hypotheses H2-H5

The Hypotheses H2-H5 analyze whether the increase in completeness per item, user or feature moderates the impact of completeness on the increase in prediction accuracy caused by adding features and their feature values (hypotheses of type "a") or by filling up missing feature values (hypotheses of type "b"). This means that the tests of the Hypotheses H2-H5 are organized in a similar way. Therefore, we describe the general structure for all of these tests in the following.

To test moderator effects on the impact of completeness on increased prediction accuracy, we chose moderated regression analysis (cf., e.g., Cohen et al. 2003; Dawson 2014; Hayes 2013; Helm and Mark 2012) as it is a widespread statistical tool to test whether the relationship between two variables is dependent on a third variable (the moderator). The underlying regression model is represented by the equation

$$y = b_0 + b_1 \cdot x + b_2 \cdot z + b_3 \cdot x \cdot z. \tag{5}$$

Here, y is the dependent or endogenous variable (criterion), x is the independent or exogenous variable (predictor) and z is the moderator variable. Regarding Hypotheses H2-H5, the endogenous variable y constitutes the (expected) increase in prediction accuracy measured by RMSE, Precision, Recall and F1-measure while the exogenous variable x indicates whether the data set with increased completeness or the data set without



increased completeness is used. The moderator variable z constitutes the increase in completeness. More precisely, for H2a, H3a and H4a, the variable z represents the increase in additional features and feature values and for H2b, H3b and H4b, the variable z represents the increase in filled up feature values. Similar, for H5a, the variable z represents the diversity of added features, and for H5b, the variable z represents the diversity of filled up features.

Besides the common interpretation of the coefficient b_0 as well as the coefficients b_1 and b_2 (*first order effects* of the regression model), the product term $x \cdot z$ and its coefficient b_3 are of special interest. This term represents the interaction (moderation) of two variables. More precisely, the coefficient b_3 estimates how much the slope of x changes as z changes. This represents how much the impact of increased completeness on prediction accuracy is influenced by the (different) values of the moderator variable. Therefore, a hypothesis proposing a moderator effect can be supported, if there is evidence that b_3 is different from zero with a certain level of significance.

In case of a moderator effect, the strength of this effect can be assessed by Cohen's f^2. Here, the coefficient of determination regarding the regression model depicted in Equation (5) is compared to the coefficient of determination of the regression model without the interaction term, which means,

$$y = b_0 + b_1 \cdot x + b_2 \cdot z. \tag{6}$$

Denoting the coefficient of determination R^2 according to each Equation (5) and (6) (i.e., R_1^2 and R_2^2), Cohen's f^2 is given by the term

$$f^2 = \frac{R_1^2 - R_2^2}{1 - R_1^2}. \tag{7}$$

Cohen's f^2 measures the relative increase in the explained variance of y when adding the interaction term to Equation (6) as shown in Equation (5). In Cohen (1988) the values 0.02, 0.15 and 0.35 are suggested for f^2 to indicate small, medium or large moderator effect sizes, which is critically discussed in scientific literature (Aguinis et al. 2005; Gignac and Szodorai 2016; Helm and Mark 2012). For instance, Aguinis et al. (2005) conducted a review of 261 articles published in several journals (maintaining high methodological standards) in order to analyze the size of moderating effects. They found that the mean of Cohen's f^2 was about 0.009 (with a standard deviation of 0.025), and the median about 0.002 with a positively skewed distribution (skewness = 6.52). This indicates that – regarding the suggested values of Cohen (1988) – a medium or strong moderator effect can be rarely attained. In their discussion, they encourage researchers to "plan future research designs based on smaller (and more realistic) targeted effect sizes" (Aguinis et al. 2005) as long as the observed effect has a meaningful impact and interpretation for science and practice.

Evaluation

In this section, we outline the test procedure and results of our empirical evaluation. Initially, we describe both used real-world data sets. Afterwards, we introduce the recommender system which was applied to these data sets and outline in detail how we tested each hypothesis. We conclude the section by presenting the results of these tests.

Description and Preparation of Data Sets

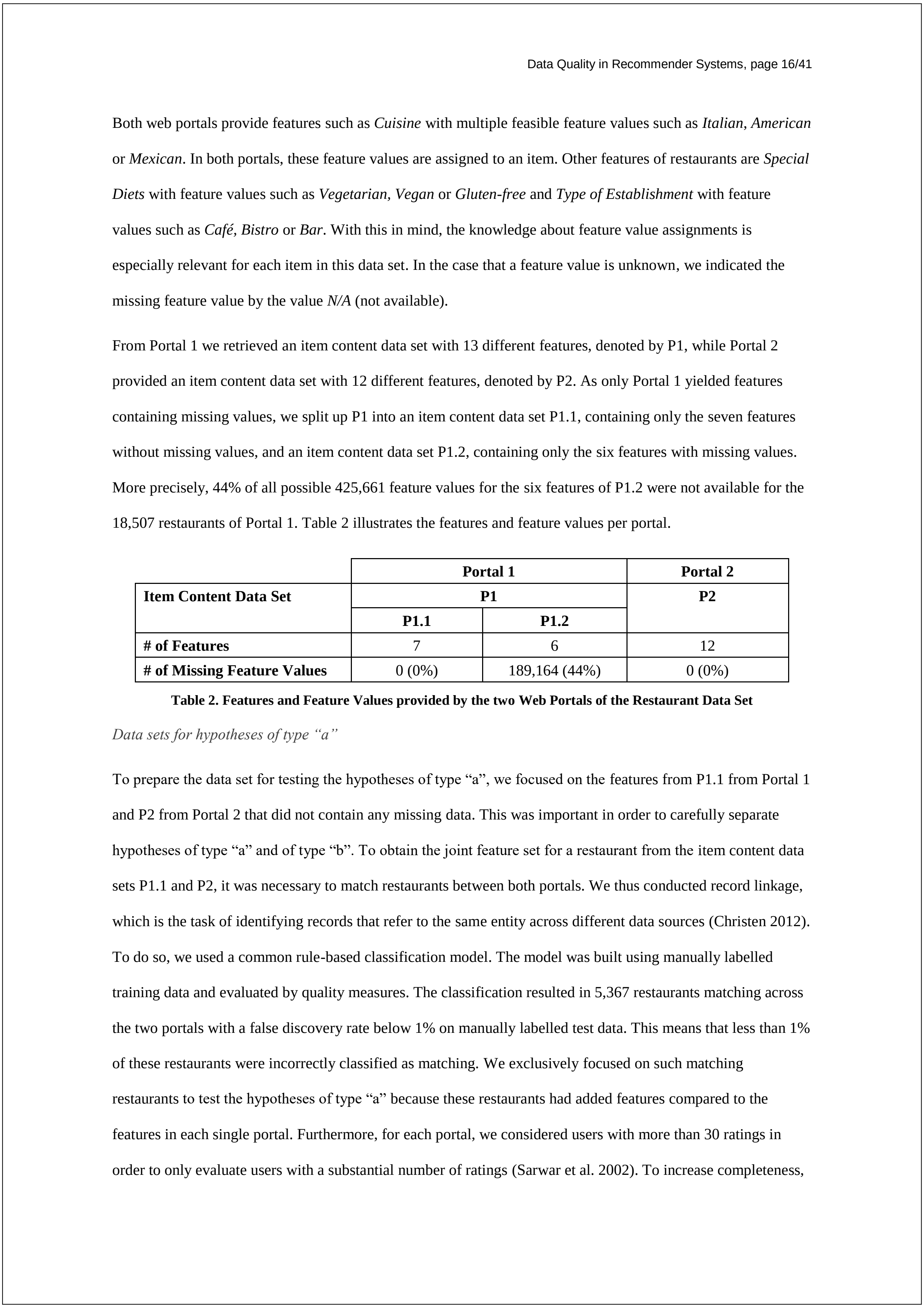
For testing our hypotheses, we prepared two real-world data sets. While the first data set contains a large number of user-generated ratings about restaurants and was retrieved from two leading advertising web portals, the second data set is based on the non-commercial movielens data set containing approximately one million ratings (Harper and Konstan 2015). In both data sets, the ratings are assessments of items by users and hence, each rating corresponds to exactly one user and one item. Further, the rating values are given on an ordinal, five-tier scale of stars, ranging from 1 star to 5 stars.

Restaurant Data Set

In the first data set, one portal (Portal 1) focuses on local businesses such as bars or restaurants and provided over 100 million ratings by 2018. The second portal (Portal 2) specializes on travel opportunities and businesses such as restaurants providing over 400 million ratings by 2018. Since each web portal provided a vast amount of data, we focused on an excerpt and chose rating data of restaurants from the area of New York City, USA, because the high number of restaurants in this area allows for testing each hypothesis on a sufficiently high number of items, users or features, respectively. This led to a data set with more than 2.2 million ratings provided by over 550,000 users on more than 18,500 restaurants from Portal 1 and more than 720,000 ratings from about 375,000 users for more than 8,600 restaurants from Portal 2. Table 1 describes the restaurant data set.

	Portal 1	**Portal 2**
# of Users	556,462	374,960
# of Restaurants	18,507	8,631
# of Ratings	2,252,224	721,416

Table 1. Description of the Restaurant Data Set



Both web portals provide features such as *Cuisine* with multiple feasible feature values such as *Italian*, *American* or *Mexican*. In both portals, these feature values are assigned to an item. Other features of restaurants are *Special Diets* with feature values such as *Vegetarian, Vegan* or *Gluten-free* and *Type of Establishment* with feature values such as *Café, Bistro* or *Bar*. With this in mind, the knowledge about feature value assignments is especially relevant for each item in this data set. In the case that a feature value is unknown, we indicated the missing feature value by the value *N/A* (not available).

From Portal 1 we retrieved an item content data set with 13 different features, denoted by P1, while Portal 2 provided an item content data set with 12 different features, denoted by P2. As only Portal 1 yielded features containing missing values, we split up P1 into an item content data set P1.1, containing only the seven features without missing values, and an item content data set P1.2, containing only the six features with missing values. More precisely, 44% of all possible 425,661 feature values for the six features of P1.2 were not available for the 18,507 restaurants of Portal 1. Table 2 illustrates the features and feature values per portal.

	Portal 1		**Portal 2**
Item Content Data Set	**P1**		**P2**
	P1.1	**P1.2**	
# of Features	7	6	12
# of Missing Feature Values	0 (0%)	189,164 (44%)	0 (0%)

Table 2. Features and Feature Values provided by the two Web Portals of the Restaurant Data Set

Data sets for hypotheses of type "a"

To prepare the data set for testing the hypotheses of type "a", we focused on the features from P1.1 from Portal 1 and P2 from Portal 2 that did not contain any missing data. This was important in order to carefully separate hypotheses of type "a" and of type "b". To obtain the joint feature set for a restaurant from the item content data sets P1.1 and P2, it was necessary to match restaurants between both portals. We thus conducted record linkage, which is the task of identifying records that refer to the same entity across different data sources (Christen 2012). To do so, we used a common rule-based classification model. The model was built using manually labelled training data and evaluated by quality measures. The classification resulted in 5,367 restaurants matching across the two portals with a false discovery rate below 1% on manually labelled test data. This means that less than 1% of these restaurants were incorrectly classified as matching. We exclusively focused on such matching restaurants to test the hypotheses of type "a" because these restaurants had added features compared to the features in each single portal. Furthermore, for each portal, we considered users with more than 30 ratings in order to only evaluate users with a substantial number of ratings (Sarwar et al. 2002). To increase completeness, features from Portal 2 were added to the feature set of Portal 1 and vice versa. This resulted in two cases used for testing the hypotheses of type “a”: The data for the first case originated from Portal 1, consisted of 5,367 items with 367,182 ratings of 8,138 users and was evaluated using the item content data sets P1.1 as baseline and P2 as set of additional features and their feature values. The data for the second case originated from Portal 2, comprised the same 5,367 items with 20,659 ratings of 505 users and was evaluated using the item content data sets P2 as baseline and P1.1 as set of additional features (cf. Table 3).

Data sets for hypotheses of type “b”

To prepare data for testing the hypotheses of type “b”, we focused on the first portal, as the second portal did not provide any features with missing values. In this case, to fill up missing feature values in the item content data set P1.2 containing six features, we used the common nearest neighbor imputation technique (Enders 2010). Similar to above, this imputation was evaluated by means of training and test data as well as quality measures. Missing values were imputed with a mean absolute error of only 0.299 for the test data. Again, we considered users with more than 30 ratings. This led to the data for testing the hypotheses of type “b” consisting of 18,507 restaurants with 731,395 ratings of 10,556 users, which was evaluated comparing the item content data sets P1.2 as baseline (consisting of 236,497 feature values) and P1.2’ as set of baseline features with filled up feature values (consisting of 425,661 feature values including the 189,164 filled up feature values) (cf. Table 3).

	Hypotheses of Type “a” originating from Portal 1		**Hypotheses of Type “a”** originating from Portal 2		**Hypotheses of Type “b”** originating from Portal 1	
Item Content Data Set	**P1.1** (Baseline)	**P1.1&P2** (Baseline & add. features)	**P2** (Baseline)	**P1.1&P2** (Baseline & add. features)	**P1.2** (Baseline)	**P1.2’** (Baseline & filled up feature values)
# of Features/ **# of Feature Values**	7	19	12	19	236,497	425,661
# of Items	5,367		5,367		18,507	
# of Ratings	367,182		20,659		731,395	
# of Users	8,138		505		10,556	

Table 3. Description of the Data Bases for Evaluating Hypotheses H1a/b-H5a/b on the Restaurant Data Set

Movie Data Set

The second data set focuses on movies and originates from the research lab grouplens, which provides data sets with up to 20 million ratings from the non-commercial web portal movielens by 2016. Since the movielens data sets have been updated since 1998, new features and feature values have been added in new versions. To enable an evaluation based on a larger amount of ratings, we consider the data set from 2003 with only one feature and its most recent version from 2016 with five additional features and their feature values. The old version (OldV) of the movielens data set from 2003 contains over one million ratings provided by over 6,000 users on approximately 3,900 movies, while the new version (NewV) consists of over 20 million ratings from about 140,000 users for more than 27,000 movies. Table 4 describes the movie data set.

	OldV	**NewV**
# of Users	6,040	138,493
# of Movies	3,883	27,278
# of Ratings	1,000,209	20,000,263

Table 4. Description of the Movie Data Set

Similar to the restaurant data set, both versions of the movielens data set provide the feature *Genre* with multiple feasible feature values such as *Comedy*, *Drama* or *Thriller*, while the new version provides additional features and their feature values such as *Actors* and *Country of Origin* each with according feature values. For example, the additional feature *Actors* in the version NewV indicates the top billed actors of the movie cast. Both versions do not yield features containing missing values, which means that only hypotheses of type "a" could be tested on the movie data set. Table 5 illustrates the features and feature values per version.

Item Content Data Set	**OldV**	**NewV**
# of Features	1	6
# of Missing Feature Values	0 (0%)	0 (0%)

Table 5. Features and Feature Values provided by the two Versions of the Movie Data Set

Data set for hypotheses of type "a"

Since the movie data set consists of an old and a new version, it is clear that the baseline item content data set is given by the old version and the item content data set with increased completeness is given by the union of both versions. Similar to the restaurant data set, the joint feature set for a movie was obtained by matching movies between both versions. As the movielens identifiers of the movies did not change between both versions (except from 24 movies, which were removed), record linkage was easy to conduct. Furthermore, the 6,040 users in both versions had at least 20 ratings, enabling a substantial number of ratings for the evaluation. Since 24 movies and their corresponding 2,175 ratings had been removed in the new version NewV, this resulted in content data sets consisting of 3,859 items with 998,034 ratings of 6,040 users and was evaluated using the item content data sets OldV as baseline and NewV as set of additional features and their feature values (cf. Table 6).

	Hypotheses of Type "a" originating from the old version of the movielens data set	
Item Content Data Set	**OldV** (Baseline)	**OldV&NewV** (Baseline & add. features)
# of Features	1	6
# of Items	3,859	
# of Ratings	998,034	
# of Users	6,040	

Table 6. Description of the Data Bases for Evaluating Hypotheses H1a-H5a on the Movie Data Set

Used Recommender System

For our evaluation, we used the hybrid recommender system approach *Content-Boosted Matrix Factorization* (CBMF) as presented by Forbes and Zhu (2011) and Nguyen and Zhu (2013). Matrix factorization approaches became very popular by the contest on the Netflix Grand Prize, which started 2006 and ended 2009 (Koren et al. 2009; Koren 2009). They are now state-of-the-art models in the research of recommender systems (Kim et al. 2016; Ning et al. 2017; Symeonidis 2016). CBMF is able to utilize both non-content data (ratings) and, in particular, content data (features and feature values of items). Like all matrix factorization models, CBMF models are learned by optimization and therefore, preliminary steps such as feature weighting or feature selection are not necessary for CBMF (Koren et al. 2009; Nguyen and Zhu 2013).

CBMF learns a d-dimensional vector of latent factors $p_u \in \mathbb{R}^d$ for each user u and a d-dimensional vector of latent factors $a_f \in \mathbb{R}^d$ for each feature f, such that the actual rating r_{ui} for a user-item pair (u, i) is approximated by the predicted star rating $\hat{r}_{ui} = p_u^T q_i$, with $q_i = \sum_{f \in F_i} a_f$ and F_i being the set of features that are assigned to item i. In our evaluation, we used the default configuration for CBMF as described, for instance, by Nguyen and Zhu (2013). Excepting this default configuration concerns the regularization penalty factor λ, which has to be adjusted depending on the data set (Koren et al. 2009). Thus, to determine this factor we conducted cross-validation tests as described by Koren et al. (2009). For instance, the value $\lambda = 10 \times 10^{-6}$ (cf. Figure 4) yielded the best results on test data from Portal 1 regarding the RMSE. All other parameter configurations were adopted from Nguyen and Zhu (2013).

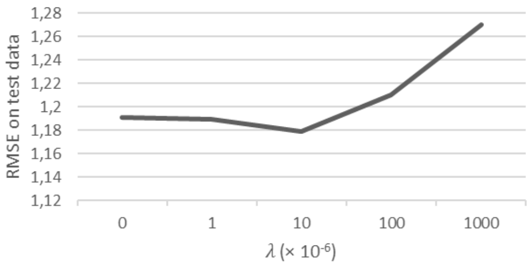


Figure 4. RMSE on the Test Data Depending on the Regularization Penalty Factor λ

Test Procedure and Results

For our evaluation, we split ratings into 50% training data for learning the CBMF model and 50% test data for assessing the prediction accuracy. On the one hand, dividing the data in half at random allowed to obtain a large *test* set (cf. also Nguyen and Zhu 2013), which is important for meaningful results when testing hypotheses. On the other hand, because of the large real-world data sets, 50% *training* data allowed us to learn the CBMF model.

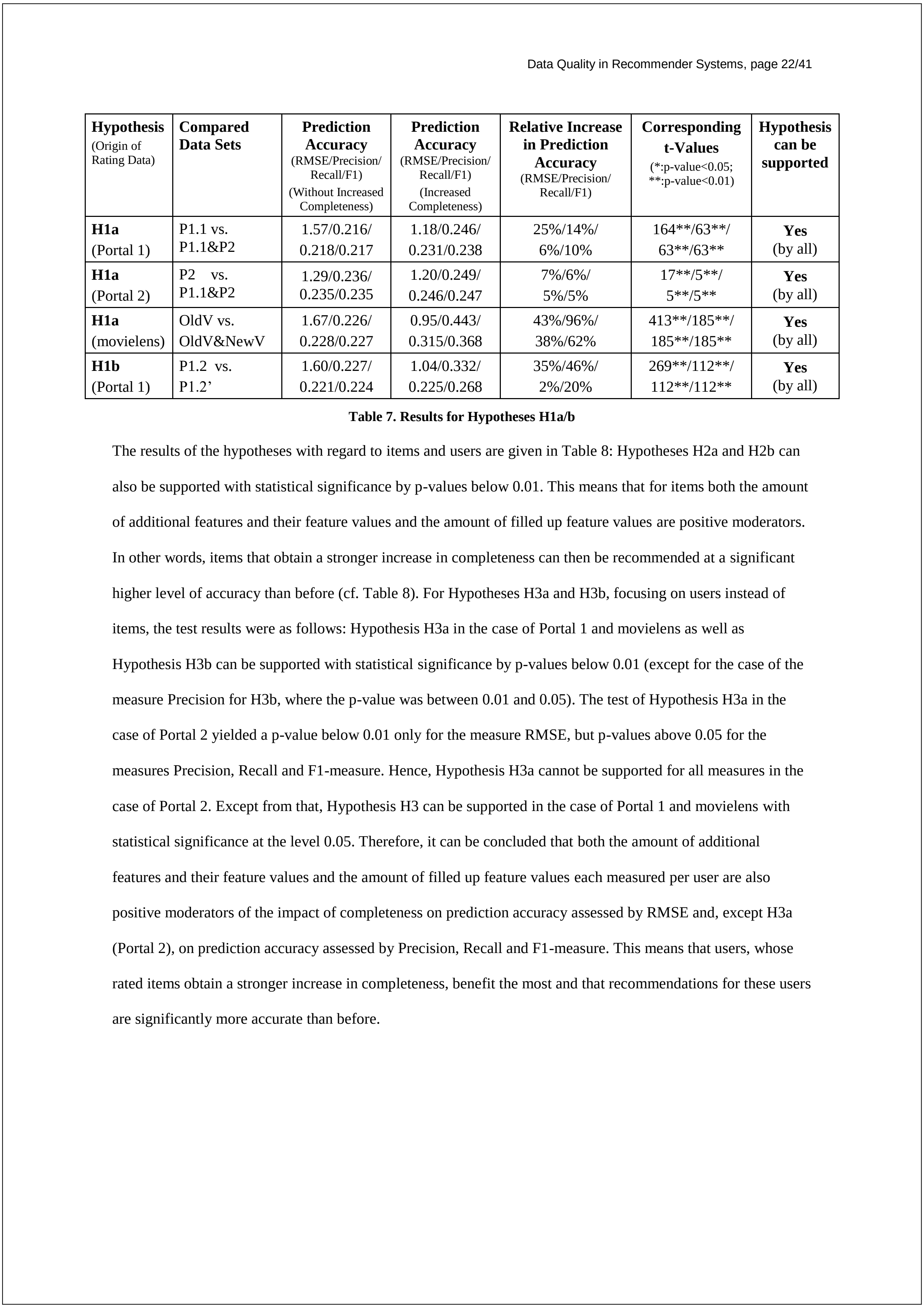
After that, we utilized the recommender system for each pair of item content data sets (with and without increased completeness) to predict ratings and assess the corresponding prediction accuracy. The increase in prediction accuracy assessed separately by Precision, Recall and F1-measure was determined by subtracting the prediction accuracy based on the baseline content from the prediction accuracy based on the content with increased completeness. As lower RMSE values indicate more accurate predictions, the negative difference was used in this case, accordingly.

A requirement for evaluating Hypotheses H1a/b using Student’s t-test is that sample groups should be normally distributed. Because of the large sample size in our evaluation, this requirement is obviously met (Boneau 1960). For evaluating moderator effects in Hypotheses H2-H5, we examined whether the selection of the linear regression model is appropriate or whether non-linear, for instance, quadratic regression models should be preferred (i.e., a curvilinear moderator effect is expected). Therefore, to test for potential non-linear moderator effects, we compared the fitness of the quadratic (non-linear) model and the linear model relying on the frequently discussed and used Bayesian Information Criterion (BIC) for model selection (cf. Schwarz 1978), for which smaller BIC values indicate the preferred model. These tests yielded almost the same BIC values for both models. For instance, for the first Hypothesis H2a the BIC value for the linear model was -5,464 and for the quadratic model -5,446 (e.g., regarding the measure Precision) and for the last Hypothesis H5b the BIC value for the linear model was -155 and for the quadratic model -148. Since the quadratic model did not or hardly improve the BIC values, the linear model was used because of its lower complexity, as suggested by literature (Cohen et al. 2003; MacCallum and Mar 1995).

The moderator variable for Hypotheses H2a/b was operationalized by the number of added or filled up feature value assignments *per item* (cf. Blake and Mangiameli 2011) relative to the number of feature value assignments *per item* in the baseline content data set. For Hypotheses H3a/b, the mean of the aforementioned operationalization across all rated items of *a user* was used as the moderator variable. In a similar way, the moderator variable for Hypothesis H4a was operationalized by the number of added feature value assignments for *a feature* relative to the number of feature value assignments in the baseline content data set. Hypothesis H4b was operationalized by the number of filled up feature value assignment for *a feature* relative to the number of feature value assignments for *this feature* in the baseline data set. The moderator for Hypotheses H5a/b was assessed by the mean cosine distance between the added/filled up features and the baseline features (Mitra et al. 2002; Tabakhi and Moradi 2015). Summing up the above, each operationalization of the moderator variables shares a similar concept as it was determined as the increase in completeness relative to the baseline content.

Furthermore, we used the two standard levels of significance 0.01 (indicated by ‘**’) and 0.05 (indicated by ‘*’) for the tests of all hypotheses (e.g., Shani and Gunawardana 2011).

In the following, we outline the evaluation results. In particular, we present the impact on prediction accuracy for all tests, which means, the values for each measure (RMSE, Precision, Recall and F1-measure), their relative increase in prediction accuracy and the significance of the t-values in case of H1a/b and the significance of the regression coefficients together with the effect sizes in case of H2-H5. Table 7 shows the results of our evaluation for the first two hypotheses: Hypotheses H1a and H1b can be supported with positive t-values and statistical significance by p-values below 0.01. This means that both adding features and their feature values as well as filling up missing feature values lead to significantly higher prediction accuracy as indicated by each of the evaluation measures in Table 7.



Hypothesis (Origin of Rating Data)	**Compared Data Sets**	**Prediction Accuracy** (RMSE/Precision/ Recall/F1) (Without Increased Completeness)	**Prediction Accuracy** (RMSE/Precision/ Recall/F1) (Increased Completeness)	**Relative Increase in Prediction Accuracy** (RMSE/Precision/ Recall/F1)	**Corresponding t-Values** (*:p-value<0.05; **:p-value<0.01)	**Hypothesis can be supported**
H1a (Portal 1)	P1.1 vs. P1.1&P2	1.57/0.216/ 0.218/0.217	1.18/0.246/ 0.231/0.238	25%/14%/ 6%/10%	164**/63**/ 63**/63**	**Yes** (by all)
H1a (Portal 2)	P2 vs. P1.1&P2	1.29/0.236/ 0.235/0.235	1.20/0.249/ 0.246/0.247	7%/6%/ 5%/5%	17**/5**/ 5**/5**	**Yes** (by all)
H1a (movielens)	OldV vs. OldV&NewV	1.67/0.226/ 0.228/0.227	0.95/0.443/ 0.315/0.368	43%/96%/ 38%/62%	413**/185**/ 185**/185**	**Yes** (by all)
H1b (Portal 1)	P1.2 vs. P1.2’	1.60/0.227/ 0.221/0.224	1.04/0.332/ 0.225/0.268	35%/46%/ 2%/20%	269**/112**/ 112**/112**	**Yes** (by all)

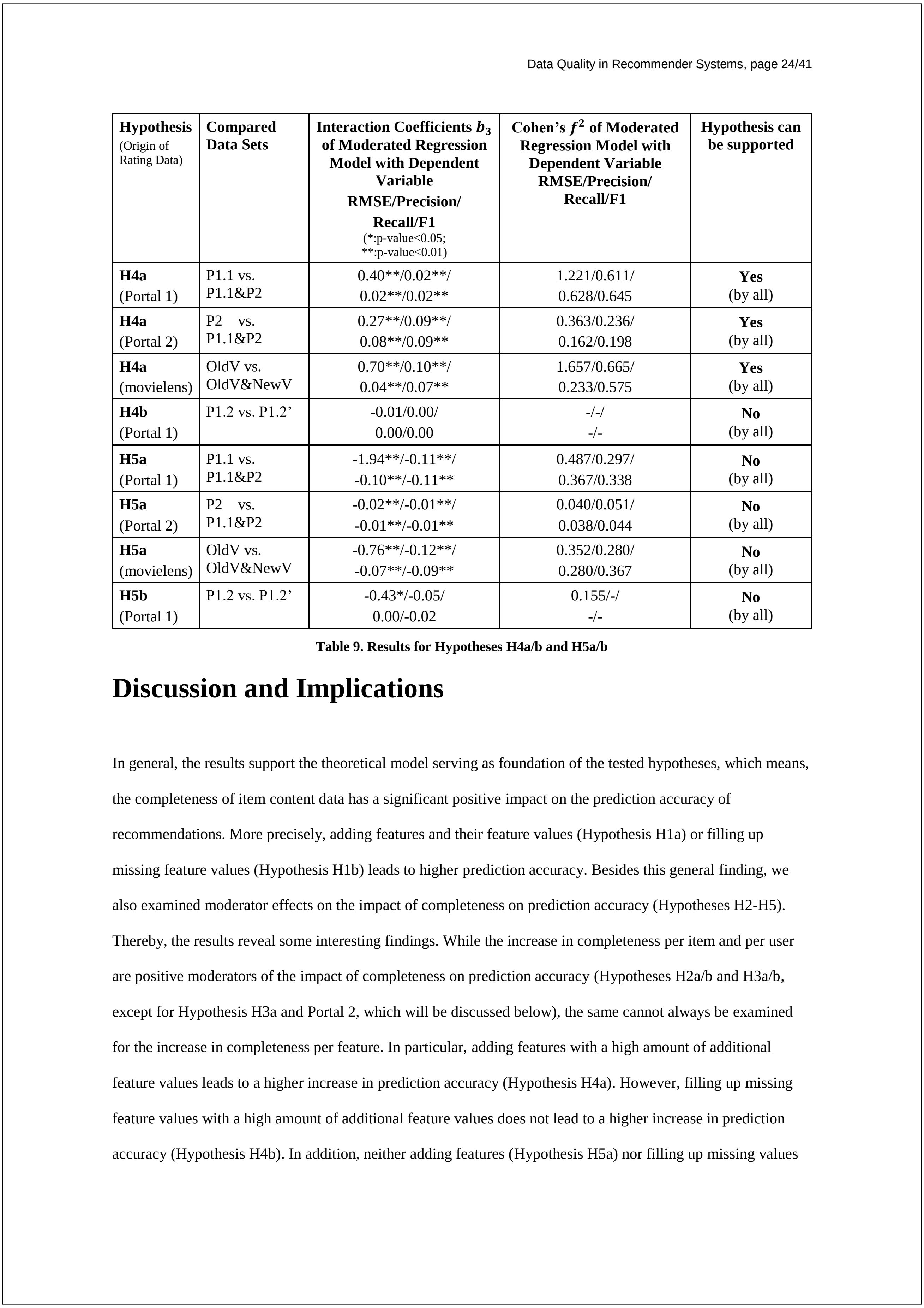
Table 7. Results for Hypotheses H1a/b

The results of the hypotheses with regard to items and users are given in Table 8: Hypotheses H2a and H2b can also be supported with statistical significance by p-values below 0.01. This means that for items both the amount of additional features and their feature values and the amount of filled up feature values are positive moderators. In other words, items that obtain a stronger increase in completeness can then be recommended at a significant higher level of accuracy than before (cf. Table 8). For Hypotheses H3a and H3b, focusing on users instead of items, the test results were as follows: Hypothesis H3a in the case of Portal 1 and movielens as well as Hypothesis H3b can be supported with statistical significance by p-values below 0.01 (except for the case of the measure Precision for H3b, where the p-value was between 0.01 and 0.05). The test of Hypothesis H3a in the case of Portal 2 yielded a p-value below 0.01 only for the measure RMSE, but p-values above 0.05 for the measures Precision, Recall and F1-measure. Hence, Hypothesis H3a cannot be supported for all measures in the case of Portal 2. Except from that, Hypothesis H3 can be supported in the case of Portal 1 and movielens with statistical significance at the level 0.05. Therefore, it can be concluded that both the amount of additional features and their feature values and the amount of filled up feature values each measured per user are also positive moderators of the impact of completeness on prediction accuracy assessed by RMSE and, except H3a (Portal 2), on prediction accuracy assessed by Precision, Recall and F1-measure. This means that users, whose rated items obtain a stronger increase in completeness, benefit the most and that recommendations for these users are significantly more accurate than before.

Hypothesis (Origin of Rating Data)	**Compared Data Sets**	**Interaction Coefficients b_3 of Moderated Regression Model with Dependent Variable RMSE/Precision/ Recall/F1** (*:p-value<0.05; **:p-value<0.01)	**Cohen's f^2 of Moderated Regression Model with Dependent Variable RMSE/Precision/ Recall/F1**	**Hypothesis can be supported**
H2a (Portal 1)	P1.1 vs. P1.1&P2	0.06**/0.02**/ 0.01**/0.01**	0.024/0.014/ 0.002/0.008	**Yes** (by all)
H2a (Portal 2)	P2 vs. P1.1&P2	0.12**/0.02**/ 0.02**/0.02**	0.042/0.001/ 0.001/0.001	**Yes** (by all)
H2a (movielens)	OldV vs. OldV& NewV	0.03**/0.01**/ 0.001**/0.004**	0.032/0.015/ 0.001/0.011	**Yes** (by all)
H2b (Portal 1)	P1.2 vs. P1.2'	0.24**/0.03**/ 0.02**/0.02**	0.436/0.019/ 0.009/0.015	**Yes** (by all)
H3a (Portal 1)	P1.1 vs. P1.1&P2	0.08**/0.02**/ 0.01**/0.01**	0.013/0.002/ 0.001/0.001	**Yes** (by all)
H3a (Portal 2)	P2 vs. P1.1&P2	0.18**/-0.01/ 0.00/-0.01	0.015/-/ -/-	**Only for RMSE measure**
H3a (movielens)	OldV vs. OldV& NewV	0.02**/0.004**/ 0.004**/0.004**	0.018/0.003/ 0.003/0.005	**Yes** (by all)
H3b (Portal 1)	P1.2 vs. P1.2'	0.39**/0.01*/ 0.02**/0.01**	0.094/0.0003/ 0.001/0.001	**Yes** (by all)

Table 8. Results for Hypotheses H2a/b and H3a/b

The results of Hypotheses H4a/b and H5a/b are given in Table 9. Hypothesis H4a can be supported with statistical significance by p-values below 0.01, whereas Hypothesis H4b cannot be supported indicated by negative coefficients b_3. In other words, only the amount of additional features and their feature values is a positive moderator of the impact on prediction accuracy (H4a), but not the amount of filled up feature values (H4b). Hypotheses H5a/b cannot be supported as indicated by negative coefficients or by p-values above 0.05. This suggests that the diversity for an additional feature or for a filled up feature is not a positive moderator.



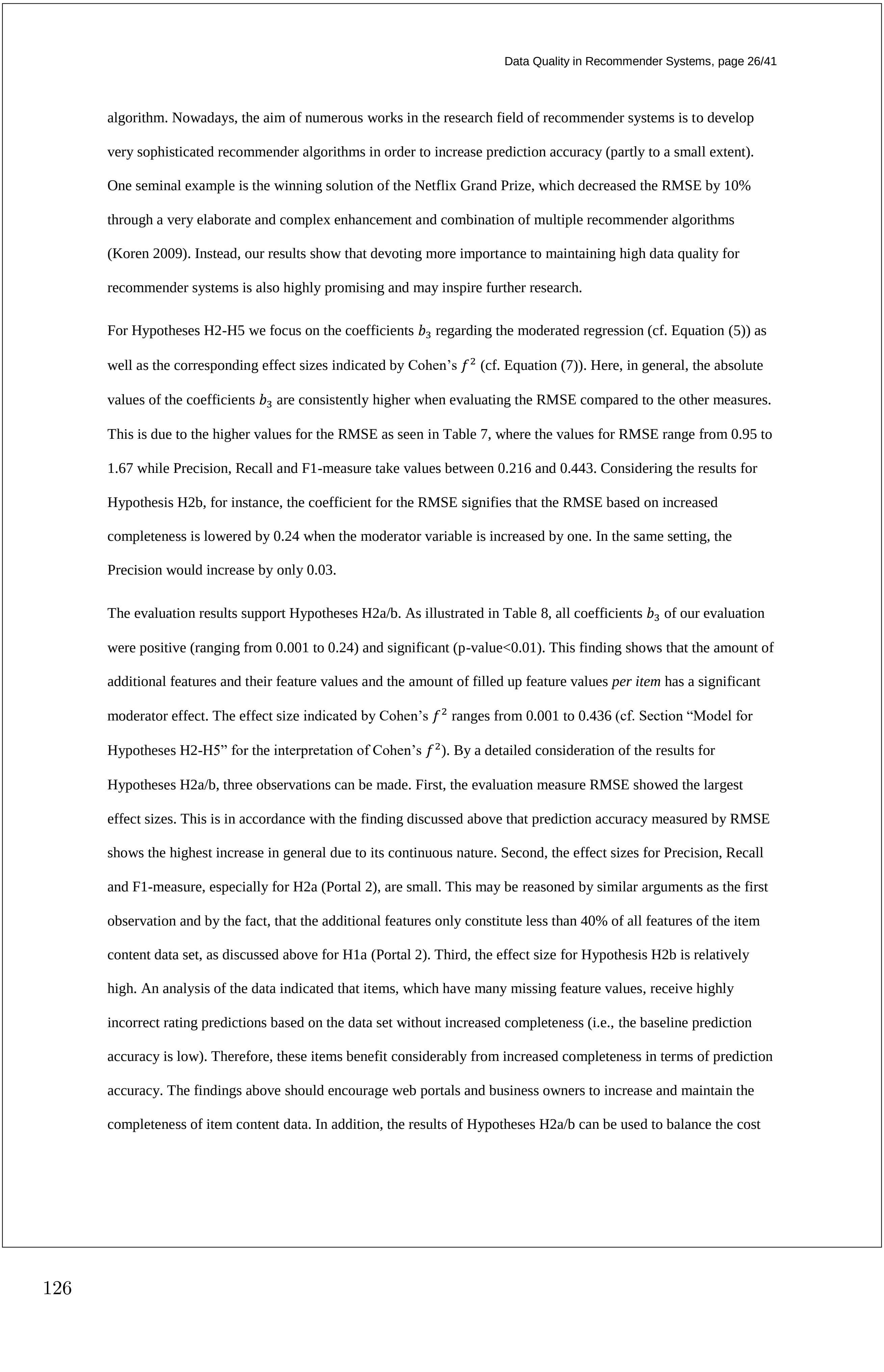
Hypothesis (Origin of Rating Data)	**Compared Data Sets**	**Interaction Coefficients b_3 of Moderated Regression Model with Dependent Variable RMSE/Precision/ Recall/F1** (*:p-value<0.05; **:p-value<0.01)	**Cohen's f^2 of Moderated Regression Model with Dependent Variable RMSE/Precision/ Recall/F1**	**Hypothesis can be supported**
H4a (Portal 1)	P1.1 vs. P1.1&P2	0.40**/0.02**/ 0.02**/0.02**	1.221/0.611/ 0.628/0.645	**Yes** (by all)
H4a (Portal 2)	P2 vs. P1.1&P2	0.27**/0.09**/ 0.08**/0.09**	0.363/0.236/ 0.162/0.198	**Yes** (by all)
H4a (movielens)	OldV vs. OldV&NewV	0.70**/0.10**/ 0.04**/0.07**	1.657/0.665/ 0.233/0.575	**Yes** (by all)
H4b (Portal 1)	P1.2 vs. P1.2'	-0.01/0.00/ 0.00/0.00	-/-/ -/-	**No** (by all)
H5a (Portal 1)	P1.1 vs. P1.1&P2	-1.94**/-0.11**/ -0.10**/-0.11**	0.487/0.297/ 0.367/0.338	**No** (by all)
H5a (Portal 2)	P2 vs. P1.1&P2	-0.02**/-0.01**/ -0.01**/-0.01**	0.040/0.051/ 0.038/0.044	**No** (by all)
H5a (movielens)	OldV vs. OldV&NewV	-0.76**/-0.12**/ -0.07**/-0.09**	0.352/0.280/ 0.280/0.367	**No** (by all)
H5b (Portal 1)	P1.2 vs. P1.2'	-0.43*/-0.05/ 0.00/-0.02	0.155/-/ -/-	**No** (by all)

Table 9. Results for Hypotheses H4a/b and H5a/b

Discussion and Implications

In general, the results support the theoretical model serving as foundation of the tested hypotheses, which means, the completeness of item content data has a significant positive impact on the prediction accuracy of recommendations. More precisely, adding features and their feature values (Hypothesis H1a) or filling up missing feature values (Hypothesis H1b) leads to higher prediction accuracy. Besides this general finding, we also examined moderator effects on the impact of completeness on prediction accuracy (Hypotheses H2-H5). Thereby, the results reveal some interesting findings. While the increase in completeness per item and per user are positive moderators of the impact of completeness on prediction accuracy (Hypotheses H2a/b and H3a/b, except for Hypothesis H3a and Portal 2, which will be discussed below), the same cannot always be examined for the increase in completeness per feature. In particular, adding features with a high amount of additional feature values leads to a higher increase in prediction accuracy (Hypothesis H4a). However, filling up missing feature values with a high amount of additional feature values does not lead to a higher increase in prediction accuracy (Hypothesis H4b). In addition, neither adding features (Hypothesis H5a) nor filling up missing values of features (Hypothesis 5b) with a high diversity leads to a higher increase in prediction accuracy, which constitutes a further interesting finding. In the following, we discuss each result in detail.

Both Hypotheses H1a and H1b are supported as indicated by t-values with positive sign and with p-values below 0.01. This means, as illustrated in Table 7, both adding features and their feature values as well as filling up missing feature values led to a considerable increase in prediction accuracy. After increasing completeness, the RMSE was between 7% and 43% lower than the RMSE before increasing completeness (corresponding to absolute decreases of RMSE between 0.09 and 0.72). Precision was between 6% and 96% higher, Recall was between 2% and 38% higher and F1-measure was between 5% and 62% higher. By a detailed consideration of the results for Hypotheses H1a/b, two interesting observations can be made. First, the relative increase in prediction accuracy is lower for H1a in the case of Portal 2 compared to all other cases of H1a. This may be due to the fact that the additional features only constitute less than 40% of all features of the item content data set with increased completeness in case of Portal 2 (7 of 19 features). In the other cases of H1a, the additional features constitute at minimum 60% of the features of the data set with increased completeness (12 of 19 features or 5 of 6 features). Second, the increase in prediction accuracy measured by RMSE and Precision is in almost all cases (considerably) higher than measured by Recall and F1-measure. In contrast to the discrete nature of the measures Precision, Recall and the F1-measure, the higher increase in prediction accuracy measured by RMSE may be reasoned by the fact that RMSE uses the predicted ratings as determined by the recommender system (i.e., as a continuous variable). Therefore, the errors between predicted and actual ratings are assessed by an interval-scaled difference. To analyze the high increases in Precision, we examined the results of H1a (movielens) in more detail, which shows the highest increase of Precision (+96%). Here, we found that, on the one hand, the *decreases* in the number of incorrect predictions (i.e., false positives) was the largest for the rating levels 1 star (-96%), 2 stars (-76%) and 5 stars (-60%). On the other hand, the largest *increases* in correct predictions (true positives) was achieved for the ratings levels 3 stars (+31%) and 4 stars (+207%). This means that by increasing completeness the used recommender system was less likely to incorrectly predict “extreme” ratings (i.e., very high or very low ratings) while mostly improving the correct prediction of “mainstream” ratings (the mean overall rating is 3.6). Hence, the Precision of most classes achieved a much higher increase than the Recall or F1-measure. In total, the results of Hypotheses H1a/b show that recommendations based on item content data sets with increased completeness are more accurate, which is valuable for achieving a high user satisfaction (Koren et al. 2009; Ricci et al. 2015). At this point, we want to emphasize that the increase in prediction accuracy is provided only by increasing data quality and not by enhancing the recommender



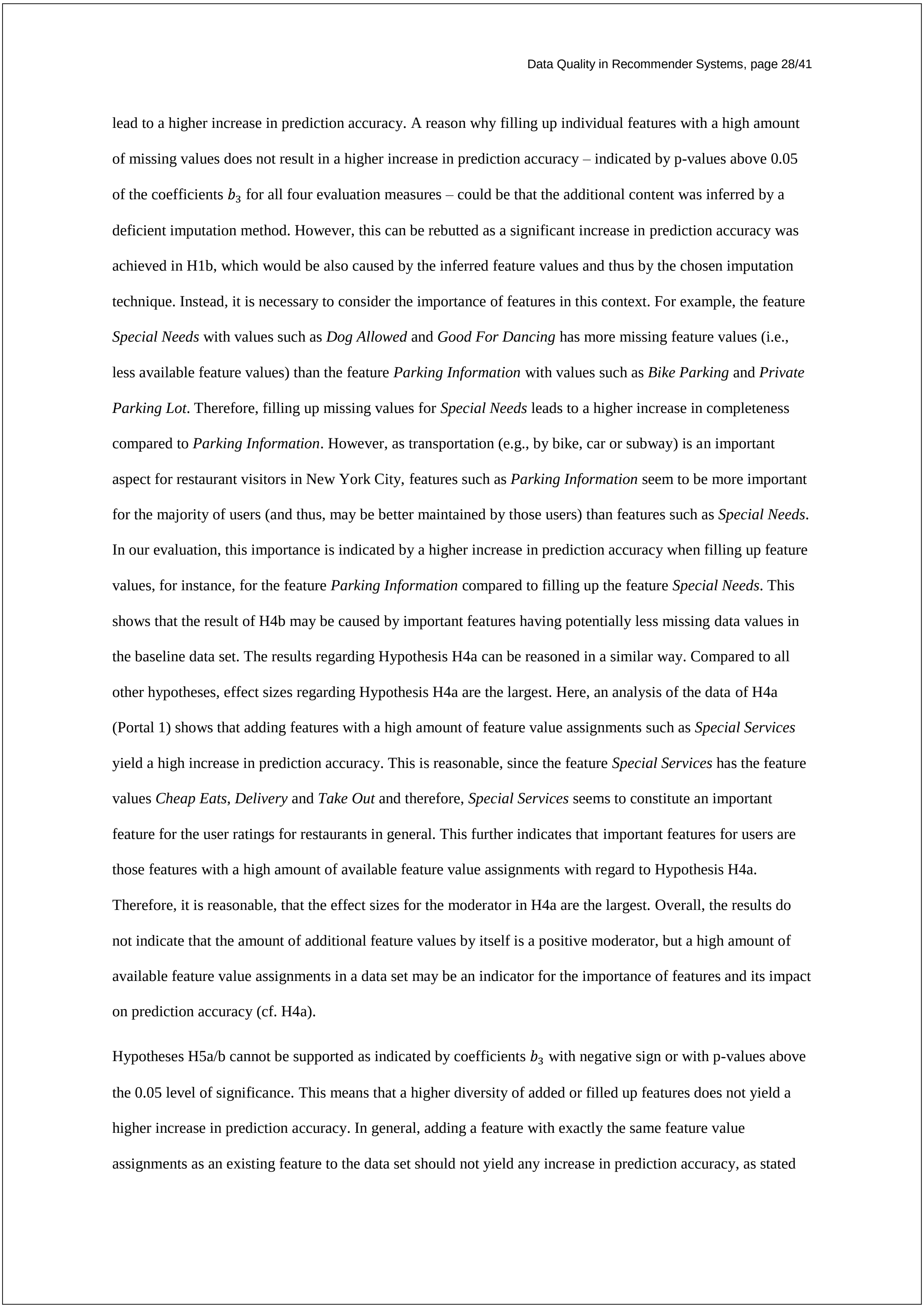
algorithm. Nowadays, the aim of numerous works in the research field of recommender systems is to develop very sophisticated recommender algorithms in order to increase prediction accuracy (partly to a small extent). One seminal example is the winning solution of the Netflix Grand Prize, which decreased the RMSE by 10% through a very elaborate and complex enhancement and combination of multiple recommender algorithms (Koren 2009). Instead, our results show that devoting more importance to maintaining high data quality for recommender systems is also highly promising and may inspire further research.

For Hypotheses H2-H5 we focus on the coefficients b_3 regarding the moderated regression (cf. Equation (5)) as well as the corresponding effect sizes indicated by Cohen's f^2 (cf. Equation (7)). Here, in general, the absolute values of the coefficients b_3 are consistently higher when evaluating the RMSE compared to the other measures. This is due to the higher values for the RMSE as seen in Table 7, where the values for RMSE range from 0.95 to 1.67 while Precision, Recall and F1-measure take values between 0.216 and 0.443. Considering the results for Hypothesis H2b, for instance, the coefficient for the RMSE signifies that the RMSE based on increased completeness is lowered by 0.24 when the moderator variable is increased by one. In the same setting, the Precision would increase by only 0.03.

The evaluation results support Hypotheses H2a/b. As illustrated in Table 8, all coefficients b_3 of our evaluation were positive (ranging from 0.001 to 0.24) and significant (p-value<0.01). This finding shows that the amount of additional features and their feature values and the amount of filled up feature values *per item* has a significant moderator effect. The effect size indicated by Cohen's f^2 ranges from 0.001 to 0.436 (cf. Section "Model for Hypotheses H2-H5" for the interpretation of Cohen's f^2). By a detailed consideration of the results for Hypotheses H2a/b, three observations can be made. First, the evaluation measure RMSE showed the largest effect sizes. This is in accordance with the finding discussed above that prediction accuracy measured by RMSE shows the highest increase in general due to its continuous nature. Second, the effect sizes for Precision, Recall and F1-measure, especially for H2a (Portal 2), are small. This may be reasoned by similar arguments as the first observation and by the fact, that the additional features only constitute less than 40% of all features of the item content data set, as discussed above for H1a (Portal 2). Third, the effect size for Hypothesis H2b is relatively high. An analysis of the data indicated that items, which have many missing feature values, receive highly incorrect rating predictions based on the data set without increased completeness (i.e., the baseline prediction accuracy is low). Therefore, these items benefit considerably from increased completeness in terms of prediction accuracy. The findings above should encourage web portals and business owners to increase and maintain the completeness of item content data. In addition, the results of Hypotheses H2a/b can be used to balance the cost and benefit of data quality improvement measures, a topic discussed in recent literature (Heinrich et al. 2018a). For instance, only items (e.g., products offered by a web portal) with a higher profit margin can be extended with additional content in a selective manner, avoiding a potentially expensive large-scale extension of the whole data set. This opens up an effective option to manage the item content data in an affordable manner, which can be a crucial factor for web portals.

Hypotheses H3a/b can be also supported except in the case of Portal 2 regarding the measures Precision, Recall and the F1-measure. In all other cases of H3a/b, our evaluation yields significant coefficients b_3 ranging from 0.004 to 0.39. This means that the amount of additional features and their feature values and the amount of filled up feature values *per user* show moderator effects. The effect size indicated by Cohen's f^2 ranges from 0.0003 to 0.094. By a detailed consideration of the results for Hypotheses H3a/b, two interesting observations can be made. First, similarly to the discussions above, RMSE shows the largest effect sizes. Second, in the case of H3a (Portal 2) the p-values of the coefficient b_3 were above the significance level of 0.05 for the measures Precision, Recall and F1-measure. This may be reasoned by the lower additional item content (7 of 19 features) as well as the lower number of users (505 users) in this particular evaluation. Thus, according to the results of H3 users with a stronger increase in the amount of additional features and their feature values or in the amount of filled up feature values are suggested to have a significantly higher increase in prediction accuracy. This means that web portals – similar to the discussion above – can manage and increase the prediction accuracy for specific users (e.g., users with low versus high sales volumes) by extending the content of items, which have been rated by these users or which may be interesting and recommended for them in the future. In addition, another promising option would be to give providers as well as users, which mainly rate items with a lower number of available features, an incentive to provide additional data for these items. In return, the user community would benefit in this way from more appropriate item recommendations.

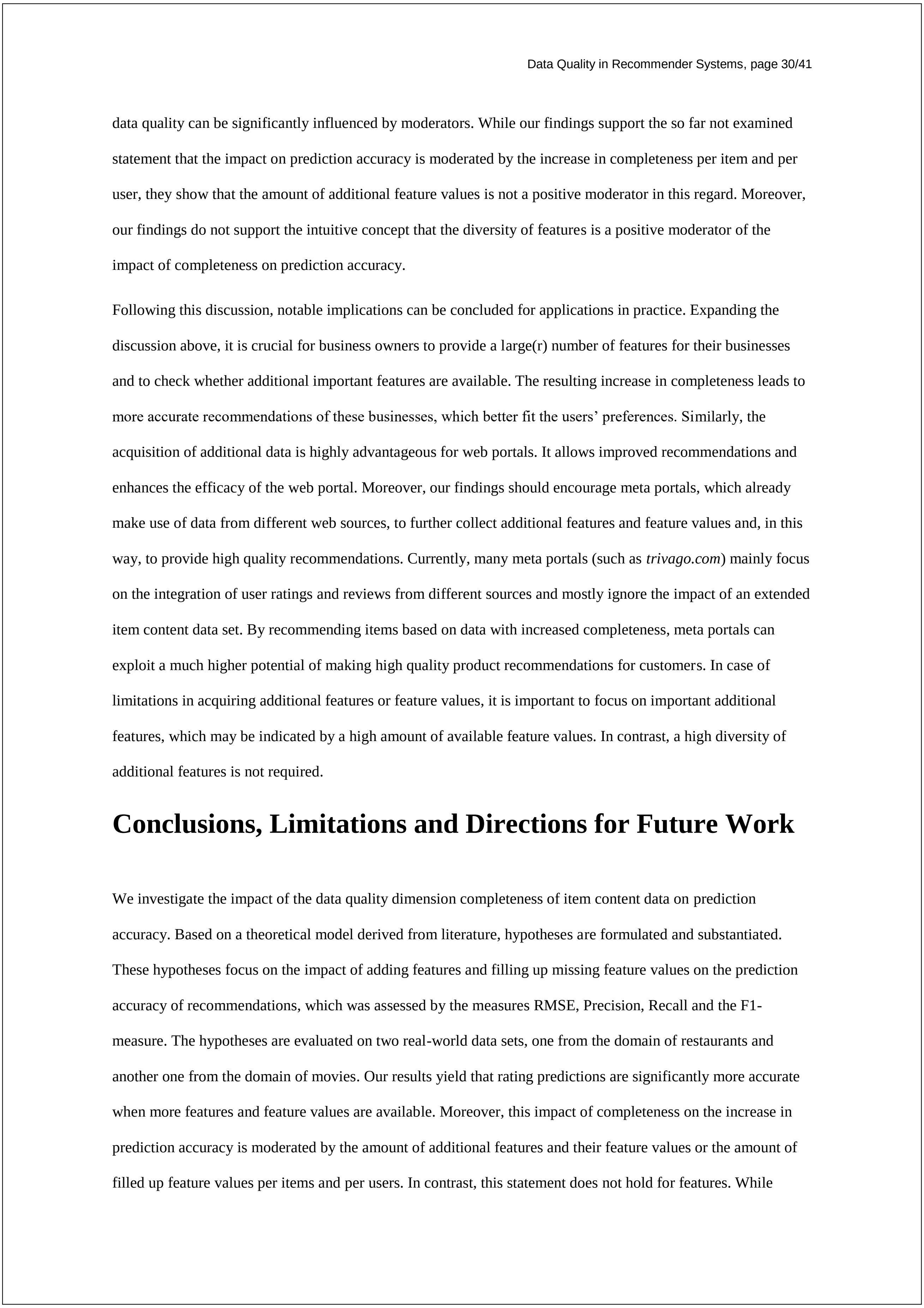
The results of Hypotheses H4a/b and H5a/b indicate that the *amount* and *diversity* of additional item content does *in general* not moderate the increase in prediction accuracy as intuition might suggest. Although Hypothesis H4a can be supported by our evaluation with relatively high moderator effects indicated by Cohen's f^2 ranging from 0.162 to 1.657 and with positive significant coefficients b_3 (ranging from 0.02 to 0.70), Hypothesis H4b cannot be supported. This means that portals aiming to extend item content data should primarily focus on (selected) additional features with a high amount of feature values, but filling up features with a high amount of additional feature values does not lead to a higher increase in prediction accuracy in general. At first sight, this result is counterintuitive, as one would have expected that more filled up feature values would



lead to a higher increase in prediction accuracy. A reason why filling up individual features with a high amount of missing values does not result in a higher increase in prediction accuracy – indicated by p-values above 0.05 of the coefficients b_3 for all four evaluation measures – could be that the additional content was inferred by a deficient imputation method. However, this can be rebutted as a significant increase in prediction accuracy was achieved in H1b, which would be also caused by the inferred feature values and thus by the chosen imputation technique. Instead, it is necessary to consider the importance of features in this context. For example, the feature *Special Needs* with values such as *Dog Allowed* and *Good For Dancing* has more missing feature values (i.e., less available feature values) than the feature *Parking Information* with values such as *Bike Parking* and *Private Parking Lot*. Therefore, filling up missing values for *Special Needs* leads to a higher increase in completeness compared to *Parking Information*. However, as transportation (e.g., by bike, car or subway) is an important aspect for restaurant visitors in New York City, features such as *Parking Information* seem to be more important for the majority of users (and thus, may be better maintained by those users) than features such as *Special Needs*. In our evaluation, this importance is indicated by a higher increase in prediction accuracy when filling up feature values, for instance, for the feature *Parking Information* compared to filling up the feature *Special Needs*. This shows that the result of H4b may be caused by important features having potentially less missing data values in the baseline data set. The results regarding Hypothesis H4a can be reasoned in a similar way. Compared to all other hypotheses, effect sizes regarding Hypothesis H4a are the largest. Here, an analysis of the data of H4a (Portal 1) shows that adding features with a high amount of feature value assignments such as *Special Services* yield a high increase in prediction accuracy. This is reasonable, since the feature *Special Services* has the feature values *Cheap Eats*, *Delivery* and *Take Out* and therefore, *Special Services* seems to constitute an important feature for the user ratings for restaurants in general. This further indicates that important features for users are those features with a high amount of available feature value assignments with regard to Hypothesis H4a. Therefore, it is reasonable, that the effect sizes for the moderator in H4a are the largest. Overall, the results do not indicate that the amount of additional feature values by itself is a positive moderator, but a high amount of available feature value assignments in a data set may be an indicator for the importance of features and its impact on prediction accuracy (cf. H4a).

Hypotheses H5a/b cannot be supported as indicated by coefficients b_3 with negative sign or with p-values above the 0.05 level of significance. This means that a higher diversity of added or filled up features does not yield a higher increase in prediction accuracy. In general, adding a feature with exactly the same feature value assignments as an existing feature to the data set should not yield any increase in prediction accuracy, as stated by the literature (Mitra et al. 2002; Tabakhi and Moradi 2015). Hence, the increase in prediction accuracy caused by adding features to a data set is expected to decrease with the similarity of these additional features to the existing features. Therefore, we would have anticipated that adding and filling up features with a high diversity would enable the recommender system to differentiate items in more details, thus leading to more accurate recommendations to users. However, an analysis shows that even features with a high diversity can be of low importance to users and thus, result in a low increase in prediction accuracy. For example, the additional feature *Production Company* with feature values such as *Paramount Pictures* or *Twentieth Century Fox* brings high diversity to the baseline feature *Genre*, as indicated by a mean cosine distance of 0.96 between the features *Production Company* and *Genre*. Nevertheless, adding only this feature has low impact on the increase in prediction accuracy (e.g., the RMSE decreased only by 0.002). This seems reasonable, as production companies produce diverse movies with different actors, directors and of different genres and therefore, the feature *Production Company* is usually of low importance for the majority of users. This underlines that features exist which have diverse feature value assignments, but their importance is low for users. Contrary to works such as (Mitra et al. 2002; Tabakhi and Moradi 2015), which propose to sort out features with high similarity (i.e., low diversity), this shows that the diversity or similarity of features may only be a subordinate factor for the impact of completeness on the prediction accuracy. In total, the increase in completeness by the amount of additional feature values (H4) as well as by the diversity of added/filled up features (H5) does not constitute a positive moderator of the impact of completeness on prediction accuracy.

Based on these findings and the above discussion, the contribution of our work to the existing body of knowledge can be outlined. Blake and Mangiameli (2011), Feldman et al. (2018) and Woodall et al. (2015) proposed and substantiated that completeness – in the sense of the amount of available feature values – has a significant impact on evaluation criteria such as decision quality of specific considered decision support systems. Complementary to these works, our results show that not only a higher amount of available feature values, but also adding new features to the feature set can have a significant impact on evaluation criteria of decision support systems and in particular recommender systems. Furthermore, so far, the impact of data quality was validated for different evaluation criteria. The works of Bharati and Chaudhury (2004) and Ge (2009) supported the impact on the evaluation criteria decision-making satisfaction and decision quality. Blake and Mangiameli (2011), Feldman et al. (2018) and Woodall et al. (2015) demonstrated the impact on data mining outcome. Supplementing these findings, our work is the first to analyze the impact of data quality – in particular completeness – on the evaluation criterion prediction accuracy. Moreover, our results show that the impact of



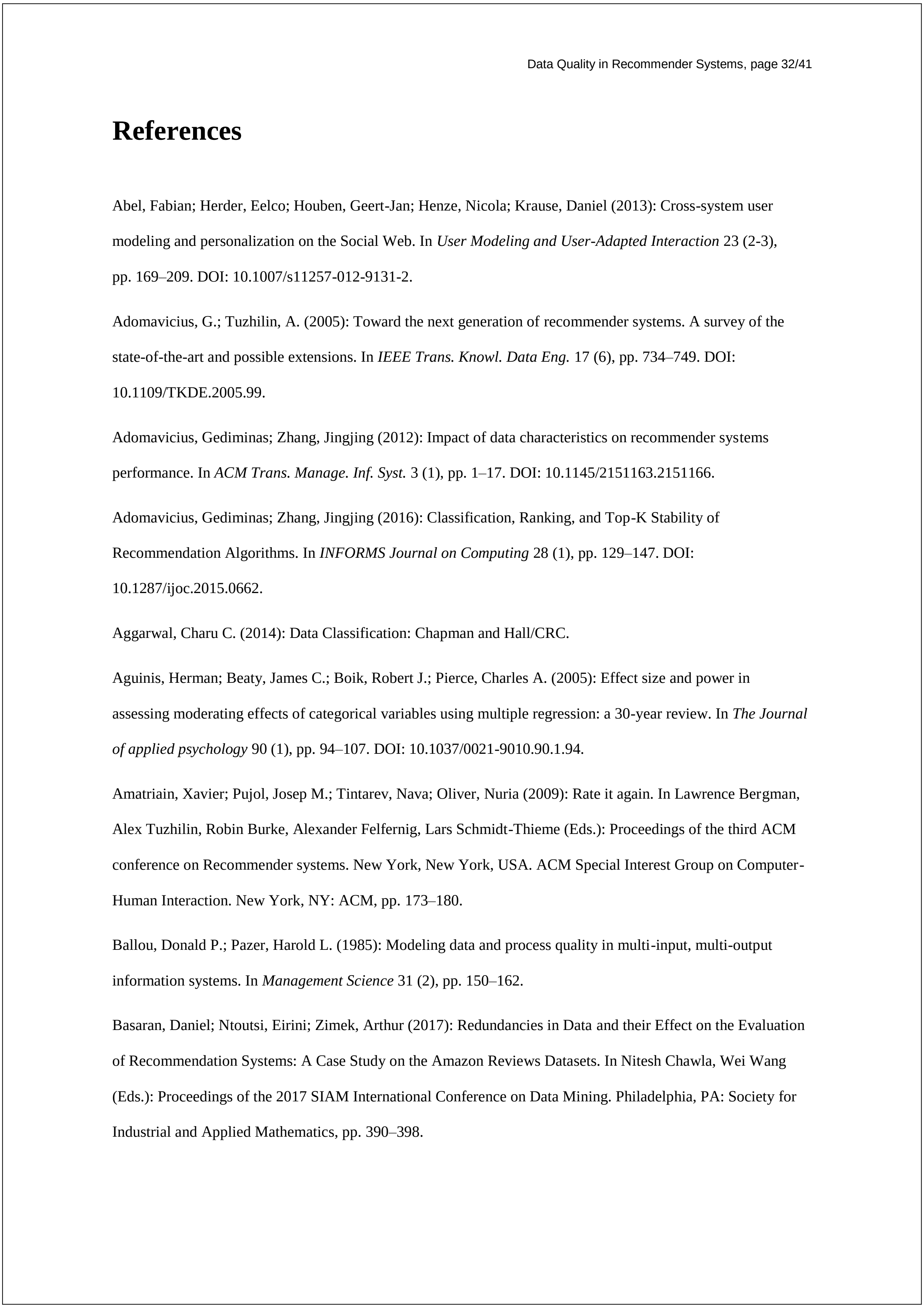
data quality can be significantly influenced by moderators. While our findings support the so far not examined statement that the impact on prediction accuracy is moderated by the increase in completeness per item and per user, they show that the amount of additional feature values is not a positive moderator in this regard. Moreover, our findings do not support the intuitive concept that the diversity of features is a positive moderator of the impact of completeness on prediction accuracy.

Following this discussion, notable implications can be concluded for applications in practice. Expanding the discussion above, it is crucial for business owners to provide a large(r) number of features for their businesses and to check whether additional important features are available. The resulting increase in completeness leads to more accurate recommendations of these businesses, which better fit the users' preferences. Similarly, the acquisition of additional data is highly advantageous for web portals. It allows improved recommendations and enhances the efficacy of the web portal. Moreover, our findings should encourage meta portals, which already make use of data from different web sources, to further collect additional features and feature values and, in this way, to provide high quality recommendations. Currently, many meta portals (such as *trivago.com*) mainly focus on the integration of user ratings and reviews from different sources and mostly ignore the impact of an extended item content data set. By recommending items based on data with increased completeness, meta portals can exploit a much higher potential of making high quality product recommendations for customers. In case of limitations in acquiring additional features or feature values, it is important to focus on important additional features, which may be indicated by a high amount of available feature values. In contrast, a high diversity of additional features is not required.

Conclusions, Limitations and Directions for Future Work

We investigate the impact of the data quality dimension completeness of item content data on prediction accuracy. Based on a theoretical model derived from literature, hypotheses are formulated and substantiated. These hypotheses focus on the impact of adding features and filling up missing feature values on the prediction accuracy of recommendations, which was assessed by the measures RMSE, Precision, Recall and the F1-measure. The hypotheses are evaluated on two real-world data sets, one from the domain of restaurants and another one from the domain of movies. Our results yield that rating predictions are significantly more accurate when more features and feature values are available. Moreover, this impact of completeness on the increase in prediction accuracy is moderated by the amount of additional features and their feature values or the amount of filled up feature values per items and per users. In contrast, this statement does not hold for features. While adding features with a high amount of feature values leads to a higher increase in prediction accuracy, filling up a high amount of feature values or adding features to the existing content with a high diversity does not lead to a higher increase in prediction accuracy. Here, our results suggest that the importance of features to users is an essential factor for the increase in prediction accuracy. Our findings are not only valuable from a scientific perspective but also in practice for business owners as well as for web portals and meta portals.

Our work also has some limitations, which could be starting points for future research. In this paper, we increased completeness by adding features from other web portals as well as by imputing missing feature values. Nevertheless, other approaches to increase completeness are possible. For example, a feature set could be extended with features based on user-generated item tags as proposed by Zhang et al. (2010). Similarly, feature values could be filled up by analyzing additional textual data using text mining to extract non-available feature values (Ghani et al. 2006). Another limitation are the costs of data preparation and computation caused by adding features and their feature values or by filling up missing feature values. In our evaluation settings, the necessary additional time and costs are reasonable: For example, the computation time of CBMF for training and evaluating the model for Hypothesis H1a (Portal 1) was raised from 285 seconds to 488 seconds for all users/items, for Hypothesis H1a (Portal 2) from 10 seconds to 24 seconds. However, these costs might indeed be relevant for applications with a vast amount of additional item content data. Furthermore, it would be highly interesting to test the impact of other data quality dimensions such as currency on recommendation quality. Additionally, in this paper we focus on different metrics for prediction accuracy as the most important quality measures for recommender systems (Shani and Gunawardana 2011). However, as the goals of a recommender system can be very diverse (e.g., introducing customers to the full product spectrum) further metrics can be of particular interest for other application scenarios (Jannach et al. 2016). Thus, further research on the impact of data quality assessed by other quality measures such as coverage, serendipity or scalability (Herlocker et al. 2004; Shani and Gunawardana 2011) would also be relevant. Finally, in the future, tests similar to ours could also be conducted using data sets from further domains, such as recommendations for music songs.



References

Abel, Fabian; Herder, Eelco; Houben, Geert-Jan; Henze, Nicola; Krause, Daniel (2013): Cross-system user modeling and personalization on the Social Web. In *User Modeling and User-Adapted Interaction* 23 (2-3), pp. 169–209. DOI: 10.1007/s11257-012-9131-2.

Adomavicius, G.; Tuzhilin, A. (2005): Toward the next generation of recommender systems. A survey of the state-of-the-art and possible extensions. In *IEEE Trans. Knowl. Data Eng.* 17 (6), pp. 734–749. DOI: 10.1109/TKDE.2005.99.

Adomavicius, Gediminas; Zhang, Jingjing (2012): Impact of data characteristics on recommender systems performance. In *ACM Trans. Manage. Inf. Syst.* 3 (1), pp. 1–17. DOI: 10.1145/2151163.2151166.

Adomavicius, Gediminas; Zhang, Jingjing (2016): Classification, Ranking, and Top-K Stability of Recommendation Algorithms. In *INFORMS Journal on Computing* 28 (1), pp. 129–147. DOI: 10.1287/ijoc.2015.0662.

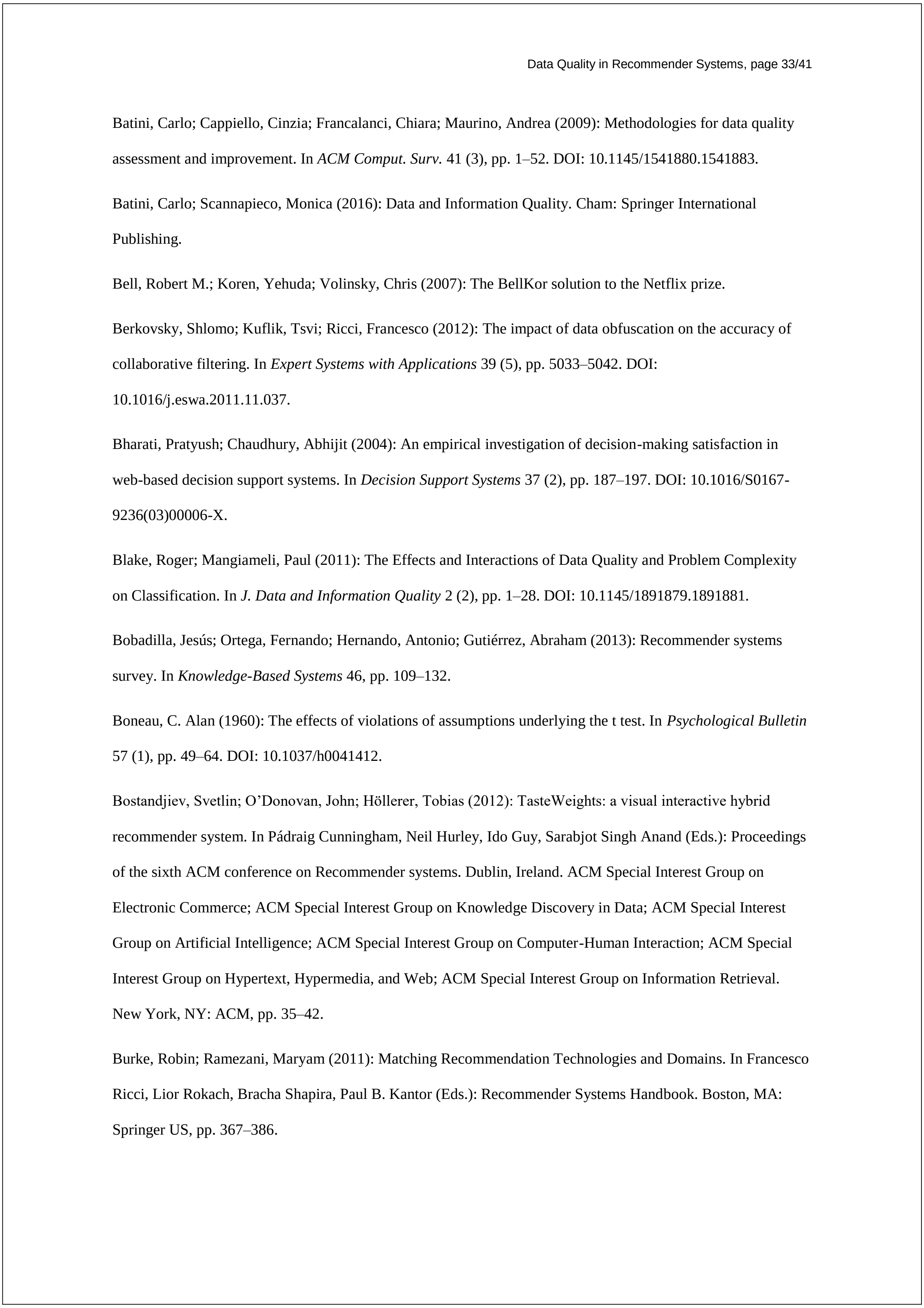
Aggarwal, Charu C. (2014): Data Classification: Chapman and Hall/CRC.

Aguinis, Herman; Beaty, James C.; Boik, Robert J.; Pierce, Charles A. (2005): Effect size and power in assessing moderating effects of categorical variables using multiple regression: a 30-year review. In *The Journal of applied psychology* 90 (1), pp. 94–107. DOI: 10.1037/0021-9010.90.1.94.

Amatriain, Xavier; Pujol, Josep M.; Tintarev, Nava; Oliver, Nuria (2009): Rate it again. In Lawrence Bergman, Alex Tuzhilin, Robin Burke, Alexander Felfernig, Lars Schmidt-Thieme (Eds.): Proceedings of the third ACM conference on Recommender systems. New York, New York, USA. ACM Special Interest Group on Computer-Human Interaction. New York, NY: ACM, pp. 173–180.

Ballou, Donald P.; Pazer, Harold L. (1985): Modeling data and process quality in multi-input, multi-output information systems. In *Management Science* 31 (2), pp. 150–162.

Basaran, Daniel; Ntoutsi, Eirini; Zimek, Arthur (2017): Redundancies in Data and their Effect on the Evaluation of Recommendation Systems: A Case Study on the Amazon Reviews Datasets. In Nitesh Chawla, Wei Wang (Eds.): Proceedings of the 2017 SIAM International Conference on Data Mining. Philadelphia, PA: Society for Industrial and Applied Mathematics, pp. 390–398.



Batini, Carlo; Cappiello, Cinzia; Francalanci, Chiara; Maurino, Andrea (2009): Methodologies for data quality assessment and improvement. In *ACM Comput. Surv.* 41 (3), pp. 1–52. DOI: 10.1145/1541880.1541883.

Batini, Carlo; Scannapieco, Monica (2016): Data and Information Quality. Cham: Springer International Publishing.

Bell, Robert M.; Koren, Yehuda; Volinsky, Chris (2007): The BellKor solution to the Netflix prize.

Berkovsky, Shlomo; Kuflik, Tsvi; Ricci, Francesco (2012): The impact of data obfuscation on the accuracy of collaborative filtering. In *Expert Systems with Applications* 39 (5), pp. 5033–5042. DOI: 10.1016/j.eswa.2011.11.037.

Bharati, Pratyush; Chaudhury, Abhijit (2004): An empirical investigation of decision-making satisfaction in web-based decision support systems. In *Decision Support Systems* 37 (2), pp. 187–197. DOI: 10.1016/S0167-9236(03)00006-X.

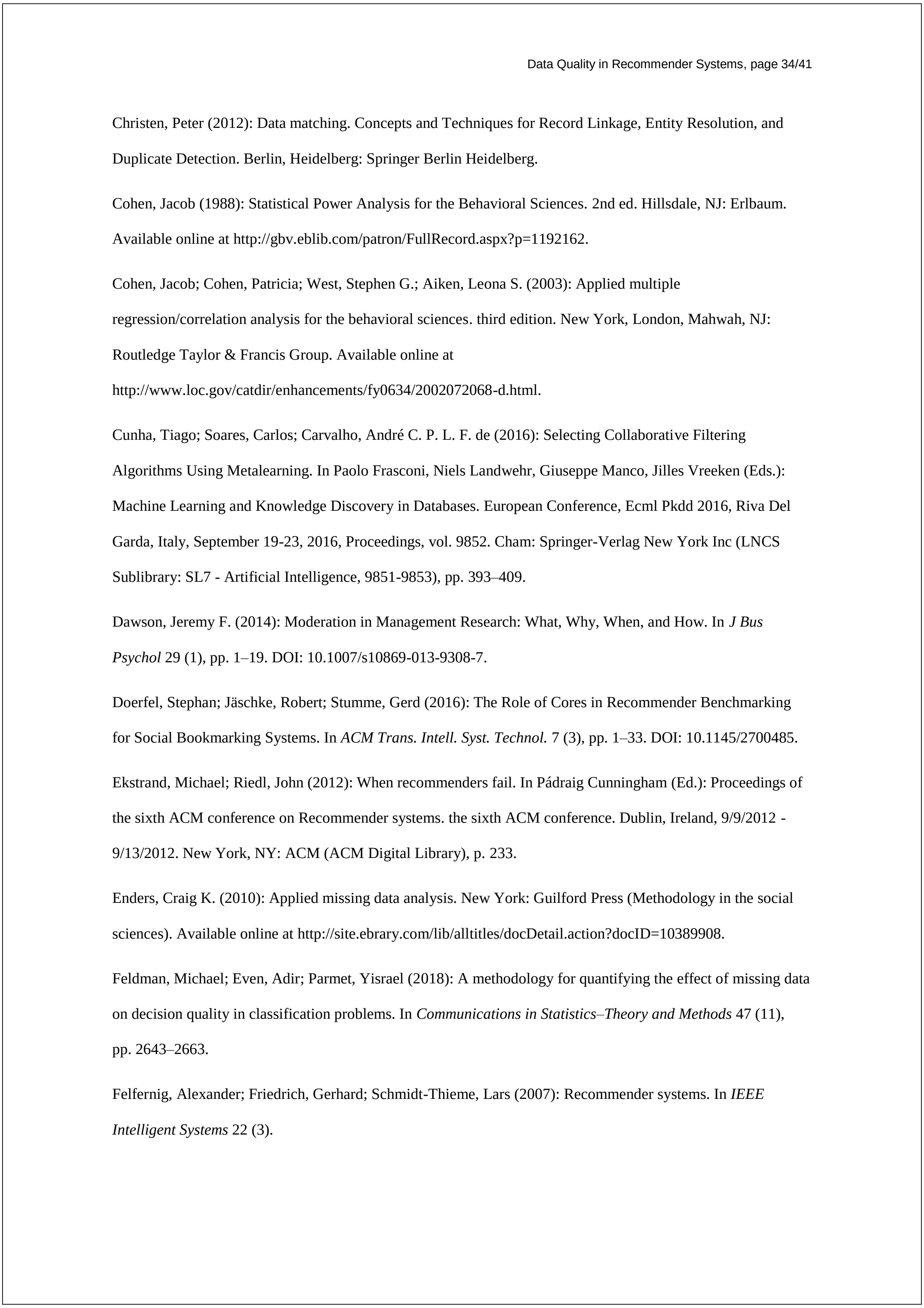
Blake, Roger; Mangiameli, Paul (2011): The Effects and Interactions of Data Quality and Problem Complexity on Classification. In *J. Data and Information Quality* 2 (2), pp. 1–28. DOI: 10.1145/1891879.1891881.

Bobadilla, Jesús; Ortega, Fernando; Hernando, Antonio; Gutiérrez, Abraham (2013): Recommender systems survey. In *Knowledge-Based Systems* 46, pp. 109–132.

Boneau, C. Alan (1960): The effects of violations of assumptions underlying the t test. In *Psychological Bulletin* 57 (1), pp. 49–64. DOI: 10.1037/h0041412.

Bostandjiev, Svetlin; O’Donovan, John; Höllerer, Tobias (2012): TasteWeights: a visual interactive hybrid recommender system. In Pádraig Cunningham, Neil Hurley, Ido Guy, Sarabjot Singh Anand (Eds.): Proceedings of the sixth ACM conference on Recommender systems. Dublin, Ireland. ACM Special Interest Group on Electronic Commerce; ACM Special Interest Group on Knowledge Discovery in Data; ACM Special Interest Group on Artificial Intelligence; ACM Special Interest Group on Computer-Human Interaction; ACM Special Interest Group on Hypertext, Hypermedia, and Web; ACM Special Interest Group on Information Retrieval. New York, NY: ACM, pp. 35–42.

Burke, Robin; Ramezani, Maryam (2011): Matching Recommendation Technologies and Domains. In Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor (Eds.): Recommender Systems Handbook. Boston, MA: Springer US, pp. 367–386.



Christen, Peter (2012): Data matching. Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Berlin, Heidelberg: Springer Berlin Heidelberg.

Cohen, Jacob (1988): Statistical Power Analysis for the Behavioral Sciences. 2nd ed. Hillsdale, NJ: Erlbaum. Available online at http://gbv.eblib.com/patron/FullRecord.aspx?p=1192162.

Cohen, Jacob; Cohen, Patricia; West, Stephen G.; Aiken, Leona S. (2003): Applied multiple regression/correlation analysis for the behavioral sciences. third edition. New York, London, Mahwah, NJ: Routledge Taylor & Francis Group. Available online at http://www.loc.gov/catdir/enhancements/fy0634/2002072068-d.html.

Cunha, Tiago; Soares, Carlos; Carvalho, André C. P. L. F. de (2016): Selecting Collaborative Filtering Algorithms Using Metalearning. In Paolo Frasconi, Niels Landwehr, Giuseppe Manco, Jilles Vreeken (Eds.): Machine Learning and Knowledge Discovery in Databases. European Conference, Ecml Pkdd 2016, Riva Del Garda, Italy, September 19-23, 2016, Proceedings, vol. 9852. Cham: Springer-Verlag New York Inc (LNCS Sublibrary: SL7 - Artificial Intelligence, 9851-9853), pp. 393–409.

Dawson, Jeremy F. (2014): Moderation in Management Research: What, Why, When, and How. In *J Bus Psychol* 29 (1), pp. 1–19. DOI: 10.1007/s10869-013-9308-7.

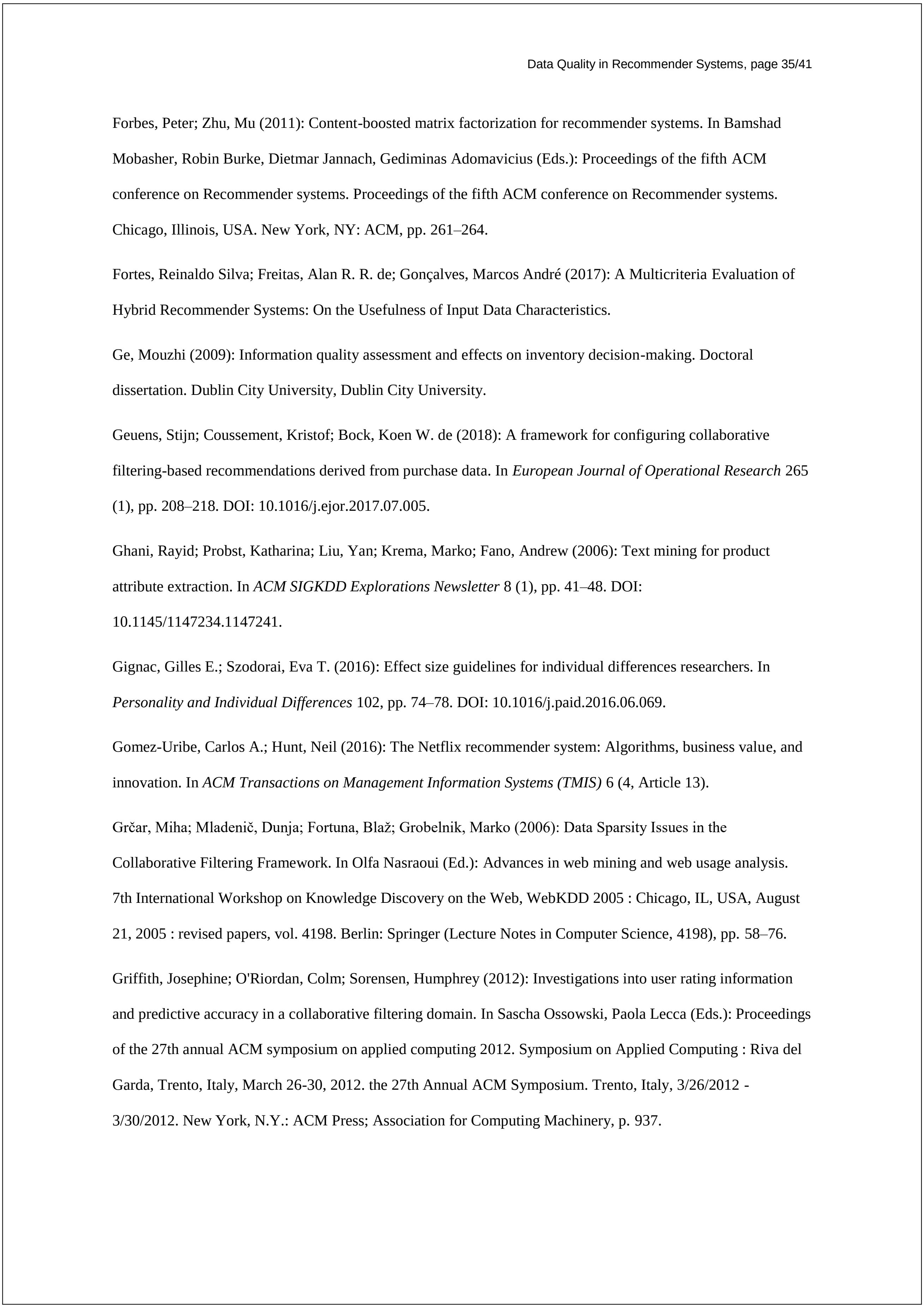
Doerfel, Stephan; Jäschke, Robert; Stumme, Gerd (2016): The Role of Cores in Recommender Benchmarking for Social Bookmarking Systems. In *ACM Trans. Intell. Syst. Technol.* 7 (3), pp. 1–33. DOI: 10.1145/2700485.

Ekstrand, Michael; Riedl, John (2012): When recommenders fail. In Pádraig Cunningham (Ed.): Proceedings of the sixth ACM conference on Recommender systems. the sixth ACM conference. Dublin, Ireland, 9/9/2012 - 9/13/2012. New York, NY: ACM (ACM Digital Library), p. 233.

Enders, Craig K. (2010): Applied missing data analysis. New York: Guilford Press (Methodology in the social sciences). Available online at http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10389908.

Feldman, Michael; Even, Adir; Parmet, Yisrael (2018): A methodology for quantifying the effect of missing data on decision quality in classification problems. In *Communications in Statistics–Theory and Methods* 47 (11), pp. 2643–2663.

Felfernig, Alexander; Friedrich, Gerhard; Schmidt-Thieme, Lars (2007): Recommender systems. In *IEEE Intelligent Systems* 22 (3).



Forbes, Peter; Zhu, Mu (2011): Content-boosted matrix factorization for recommender systems. In Bamshad Mobasher, Robin Burke, Dietmar Jannach, Gediminas Adomavicius (Eds.): Proceedings of the fifth ACM conference on Recommender systems. Proceedings of the fifth ACM conference on Recommender systems. Chicago, Illinois, USA. New York, NY: ACM, pp. 261–264.

Fortes, Reinaldo Silva; Freitas, Alan R. R. de; Gonçalves, Marcos André (2017): A Multicriteria Evaluation of Hybrid Recommender Systems: On the Usefulness of Input Data Characteristics.

Ge, Mouzhi (2009): Information quality assessment and effects on inventory decision-making. Doctoral dissertation. Dublin City University, Dublin City University.

Geuens, Stijn; Coussement, Kristof; Bock, Koen W. de (2018): A framework for configuring collaborative filtering-based recommendations derived from purchase data. In *European Journal of Operational Research* 265 (1), pp. 208–218. DOI: 10.1016/j.ejor.2017.07.005.

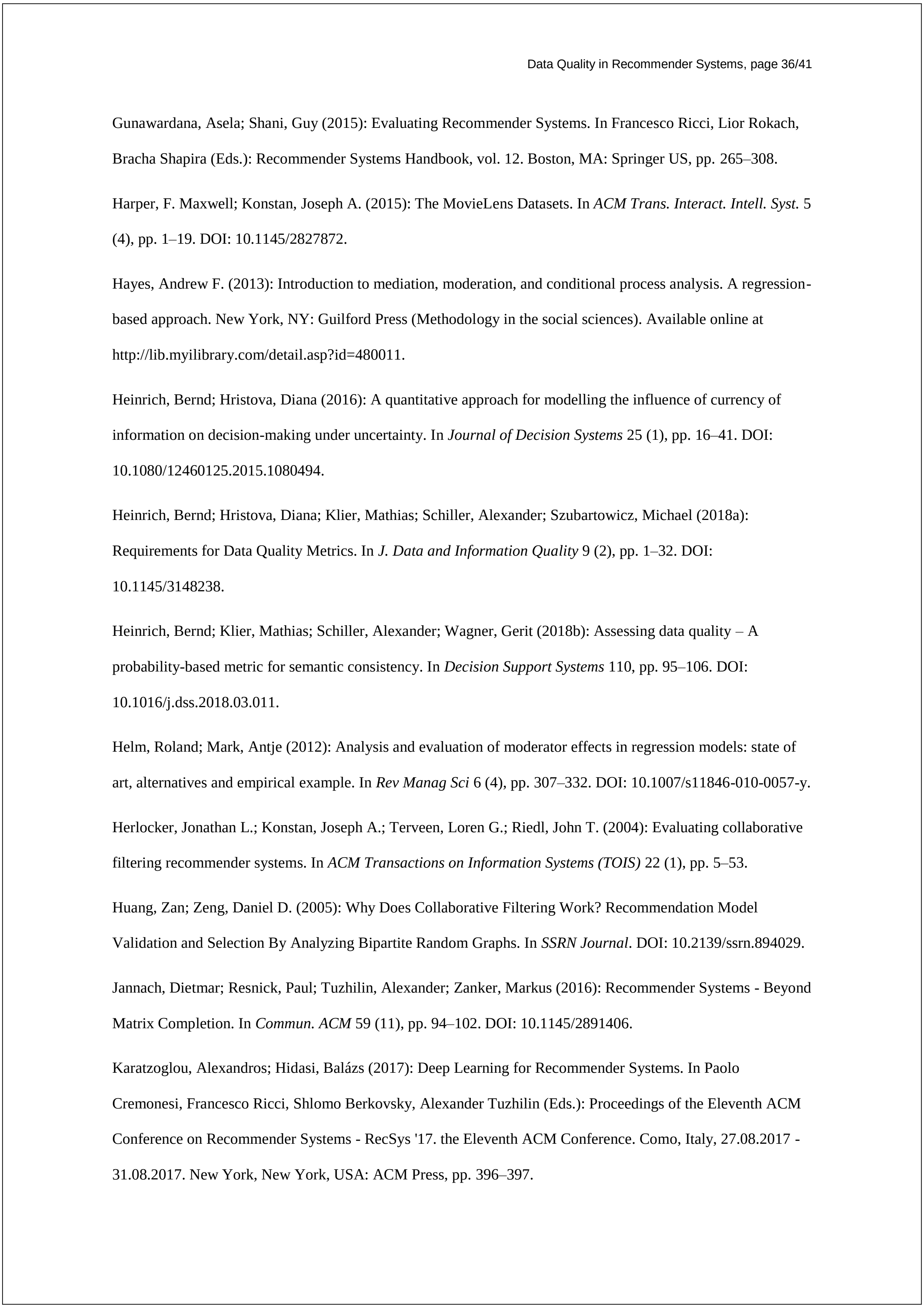
Ghani, Rayid; Probst, Katharina; Liu, Yan; Krema, Marko; Fano, Andrew (2006): Text mining for product attribute extraction. In *ACM SIGKDD Explorations Newsletter* 8 (1), pp. 41–48. DOI: 10.1145/1147234.1147241.

Gignac, Gilles E.; Szodorai, Eva T. (2016): Effect size guidelines for individual differences researchers. In *Personality and Individual Differences* 102, pp. 74–78. DOI: 10.1016/j.paid.2016.06.069.

Gomez-Uribe, Carlos A.; Hunt, Neil (2016): The Netflix recommender system: Algorithms, business value, and innovation. In *ACM Transactions on Management Information Systems (TMIS)* 6 (4, Article 13).

Grčar, Miha; Mladenič, Dunja; Fortuna, Blaž; Grobelnik, Marko (2006): Data Sparsity Issues in the Collaborative Filtering Framework. In Olfa Nasraoui (Ed.): Advances in web mining and web usage analysis. 7th International Workshop on Knowledge Discovery on the Web, WebKDD 2005 : Chicago, IL, USA, August 21, 2005 : revised papers, vol. 4198. Berlin: Springer (Lecture Notes in Computer Science, 4198), pp. 58–76.

Griffith, Josephine; O'Riordan, Colm; Sorensen, Humphrey (2012): Investigations into user rating information and predictive accuracy in a collaborative filtering domain. In Sascha Ossowski, Paola Lecca (Eds.): Proceedings of the 27th annual ACM symposium on applied computing 2012. Symposium on Applied Computing : Riva del Garda, Trento, Italy, March 26-30, 2012. the 27th Annual ACM Symposium. Trento, Italy, 3/26/2012 - 3/30/2012. New York, N.Y.: ACM Press; Association for Computing Machinery, p. 937.



Gunawardana, Asela; Shani, Guy (2015): Evaluating Recommender Systems. In Francesco Ricci, Lior Rokach, Bracha Shapira (Eds.): Recommender Systems Handbook, vol. 12. Boston, MA: Springer US, pp. 265–308.

Harper, F. Maxwell; Konstan, Joseph A. (2015): The MovieLens Datasets. In *ACM Trans. Interact. Intell. Syst.* 5 (4), pp. 1–19. DOI: 10.1145/2827872.

Hayes, Andrew F. (2013): Introduction to mediation, moderation, and conditional process analysis. A regression-based approach. New York, NY: Guilford Press (Methodology in the social sciences). Available online at http://lib.myilibrary.com/detail.asp?id=480011.

Heinrich, Bernd; Hristova, Diana (2016): A quantitative approach for modelling the influence of currency of information on decision-making under uncertainty. In *Journal of Decision Systems* 25 (1), pp. 16–41. DOI: 10.1080/12460125.2015.1080494.

Heinrich, Bernd; Hristova, Diana; Klier, Mathias; Schiller, Alexander; Szubartowicz, Michael (2018a): Requirements for Data Quality Metrics. In *J. Data and Information Quality* 9 (2), pp. 1–32. DOI: 10.1145/3148238.

Heinrich, Bernd; Klier, Mathias; Schiller, Alexander; Wagner, Gerit (2018b): Assessing data quality – A probability-based metric for semantic consistency. In *Decision Support Systems* 110, pp. 95–106. DOI: 10.1016/j.dss.2018.03.011.

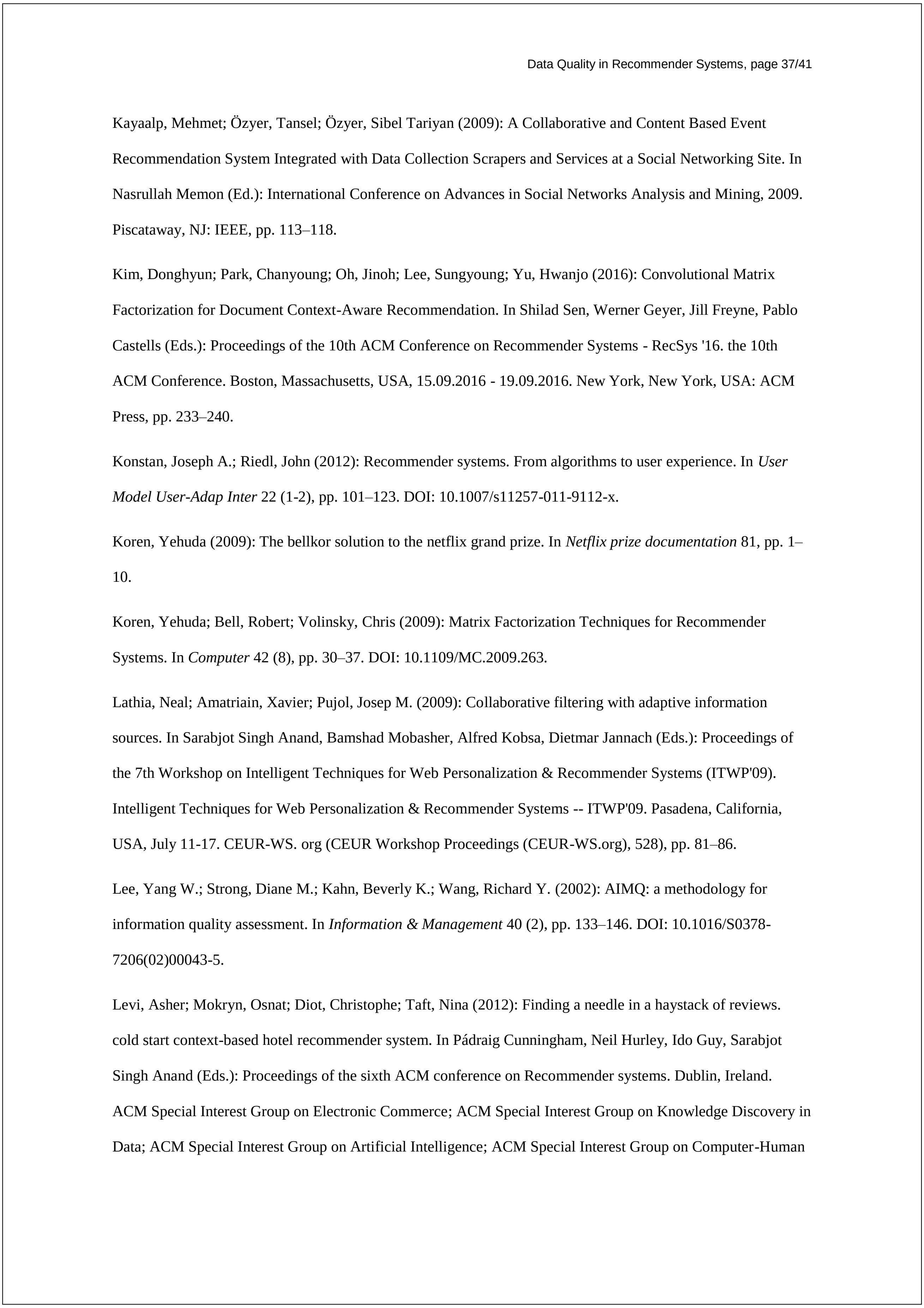
Helm, Roland; Mark, Antje (2012): Analysis and evaluation of moderator effects in regression models: state of art, alternatives and empirical example. In *Rev Manag Sci* 6 (4), pp. 307–332. DOI: 10.1007/s11846-010-0057-y.

Herlocker, Jonathan L.; Konstan, Joseph A.; Terveen, Loren G.; Riedl, John T. (2004): Evaluating collaborative filtering recommender systems. In *ACM Transactions on Information Systems (TOIS)* 22 (1), pp. 5–53.

Huang, Zan; Zeng, Daniel D. (2005): Why Does Collaborative Filtering Work? Recommendation Model Validation and Selection By Analyzing Bipartite Random Graphs. In *SSRN Journal*. DOI: 10.2139/ssrn.894029.

Jannach, Dietmar; Resnick, Paul; Tuzhilin, Alexander; Zanker, Markus (2016): Recommender Systems - Beyond Matrix Completion. In *Commun. ACM* 59 (11), pp. 94–102. DOI: 10.1145/2891406.

Karatzoglou, Alexandros; Hidasi, Balázs (2017): Deep Learning for Recommender Systems. In Paolo Cremonesi, Francesco Ricci, Shlomo Berkovsky, Alexander Tuzhilin (Eds.): Proceedings of the Eleventh ACM Conference on Recommender Systems - RecSys '17. the Eleventh ACM Conference. Como, Italy, 27.08.2017 - 31.08.2017. New York, New York, USA: ACM Press, pp. 396–397.



Kayaalp, Mehmet; Özyer, Tansel; Özyer, Sibel Tariyan (2009): A Collaborative and Content Based Event Recommendation System Integrated with Data Collection Scrapers and Services at a Social Networking Site. In Nasrullah Memon (Ed.): International Conference on Advances in Social Networks Analysis and Mining, 2009. Piscataway, NJ: IEEE, pp. 113–118.

Kim, Donghyun; Park, Chanyoung; Oh, Jinoh; Lee, Sungyoung; Yu, Hwanjo (2016): Convolutional Matrix Factorization for Document Context-Aware Recommendation. In Shilad Sen, Werner Geyer, Jill Freyne, Pablo Castells (Eds.): Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16. the 10th ACM Conference. Boston, Massachusetts, USA, 15.09.2016 - 19.09.2016. New York, New York, USA: ACM Press, pp. 233–240.

Konstan, Joseph A.; Riedl, John (2012): Recommender systems. From algorithms to user experience. In *User Model User-Adap Inter* 22 (1-2), pp. 101–123. DOI: 10.1007/s11257-011-9112-x.

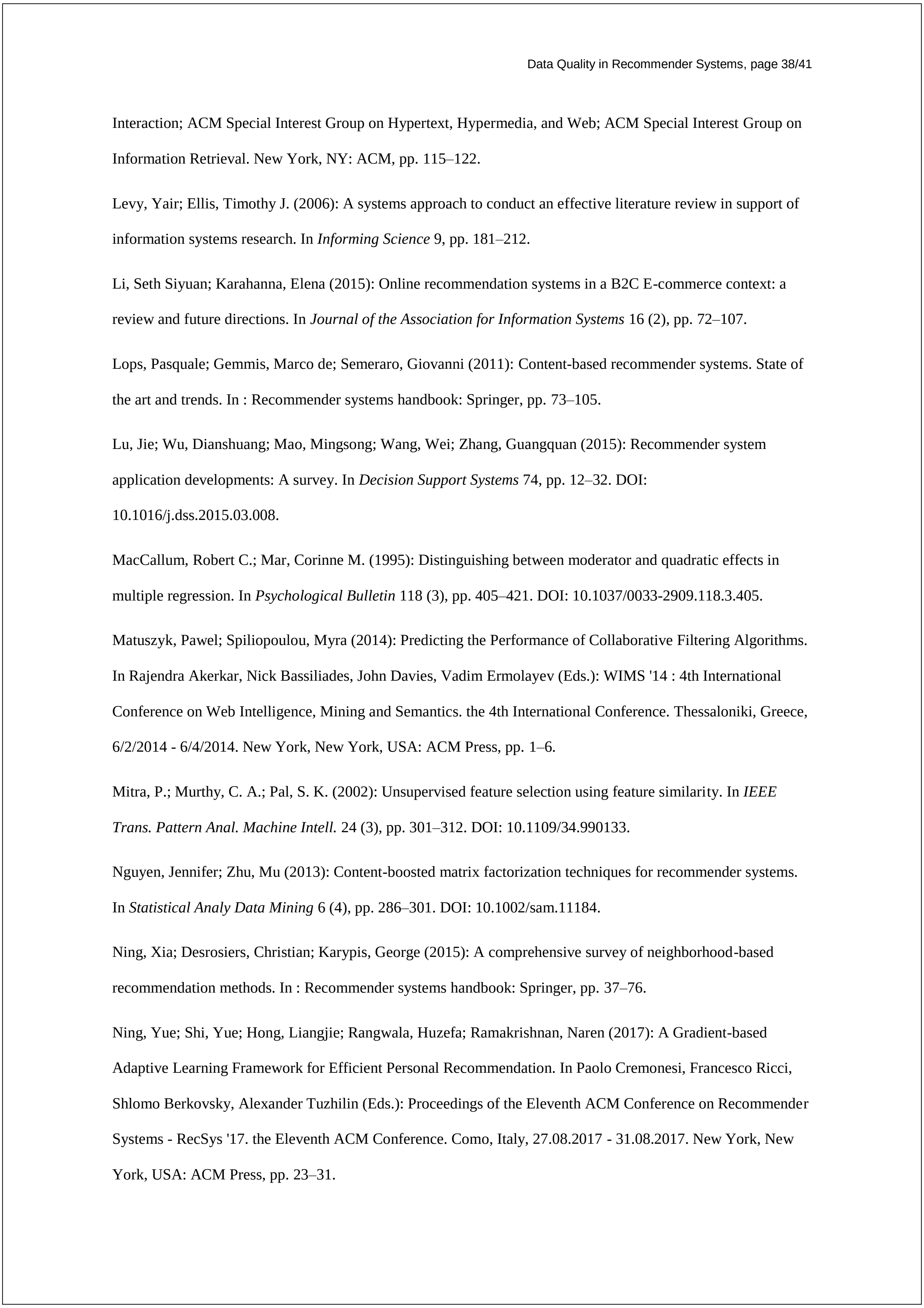
Koren, Yehuda (2009): The bellkor solution to the netflix grand prize. In *Netflix prize documentation* 81, pp. 1–10.

Koren, Yehuda; Bell, Robert; Volinsky, Chris (2009): Matrix Factorization Techniques for Recommender Systems. In *Computer* 42 (8), pp. 30–37. DOI: 10.1109/MC.2009.263.

Lathia, Neal; Amatriain, Xavier; Pujol, Josep M. (2009): Collaborative filtering with adaptive information sources. In Sarabjot Singh Anand, Bamshad Mobasher, Alfred Kobsa, Dietmar Jannach (Eds.): Proceedings of the 7th Workshop on Intelligent Techniques for Web Personalization & Recommender Systems (ITWP'09). Intelligent Techniques for Web Personalization & Recommender Systems -- ITWP'09. Pasadena, California, USA, July 11-17. CEUR-WS. org (CEUR Workshop Proceedings (CEUR-WS.org), 528), pp. 81–86.

Lee, Yang W.; Strong, Diane M.; Kahn, Beverly K.; Wang, Richard Y. (2002): AIMQ: a methodology for information quality assessment. In *Information & Management* 40 (2), pp. 133–146. DOI: 10.1016/S0378-7206(02)00043-5.

Levi, Asher; Mokryn, Osnat; Diot, Christophe; Taft, Nina (2012): Finding a needle in a haystack of reviews. cold start context-based hotel recommender system. In Pádraig Cunningham, Neil Hurley, Ido Guy, Sarabjot Singh Anand (Eds.): Proceedings of the sixth ACM conference on Recommender systems. Dublin, Ireland. ACM Special Interest Group on Electronic Commerce; ACM Special Interest Group on Knowledge Discovery in Data; ACM Special Interest Group on Artificial Intelligence; ACM Special Interest Group on Computer-Human



Interaction; ACM Special Interest Group on Hypertext, Hypermedia, and Web; ACM Special Interest Group on Information Retrieval. New York, NY: ACM, pp. 115–122.

Levy, Yair; Ellis, Timothy J. (2006): A systems approach to conduct an effective literature review in support of information systems research. In *Informing Science* 9, pp. 181–212.

Li, Seth Siyuan; Karahanna, Elena (2015): Online recommendation systems in a B2C E-commerce context: a review and future directions. In *Journal of the Association for Information Systems* 16 (2), pp. 72–107.

Lops, Pasquale; Gemmis, Marco de; Semeraro, Giovanni (2011): Content-based recommender systems. State of the art and trends. In : Recommender systems handbook: Springer, pp. 73–105.

Lu, Jie; Wu, Dianshuang; Mao, Mingsong; Wang, Wei; Zhang, Guangquan (2015): Recommender system application developments: A survey. In *Decision Support Systems* 74, pp. 12–32. DOI: 10.1016/j.dss.2015.03.008.

MacCallum, Robert C.; Mar, Corinne M. (1995): Distinguishing between moderator and quadratic effects in multiple regression. In *Psychological Bulletin* 118 (3), pp. 405–421. DOI: 10.1037/0033-2909.118.3.405.

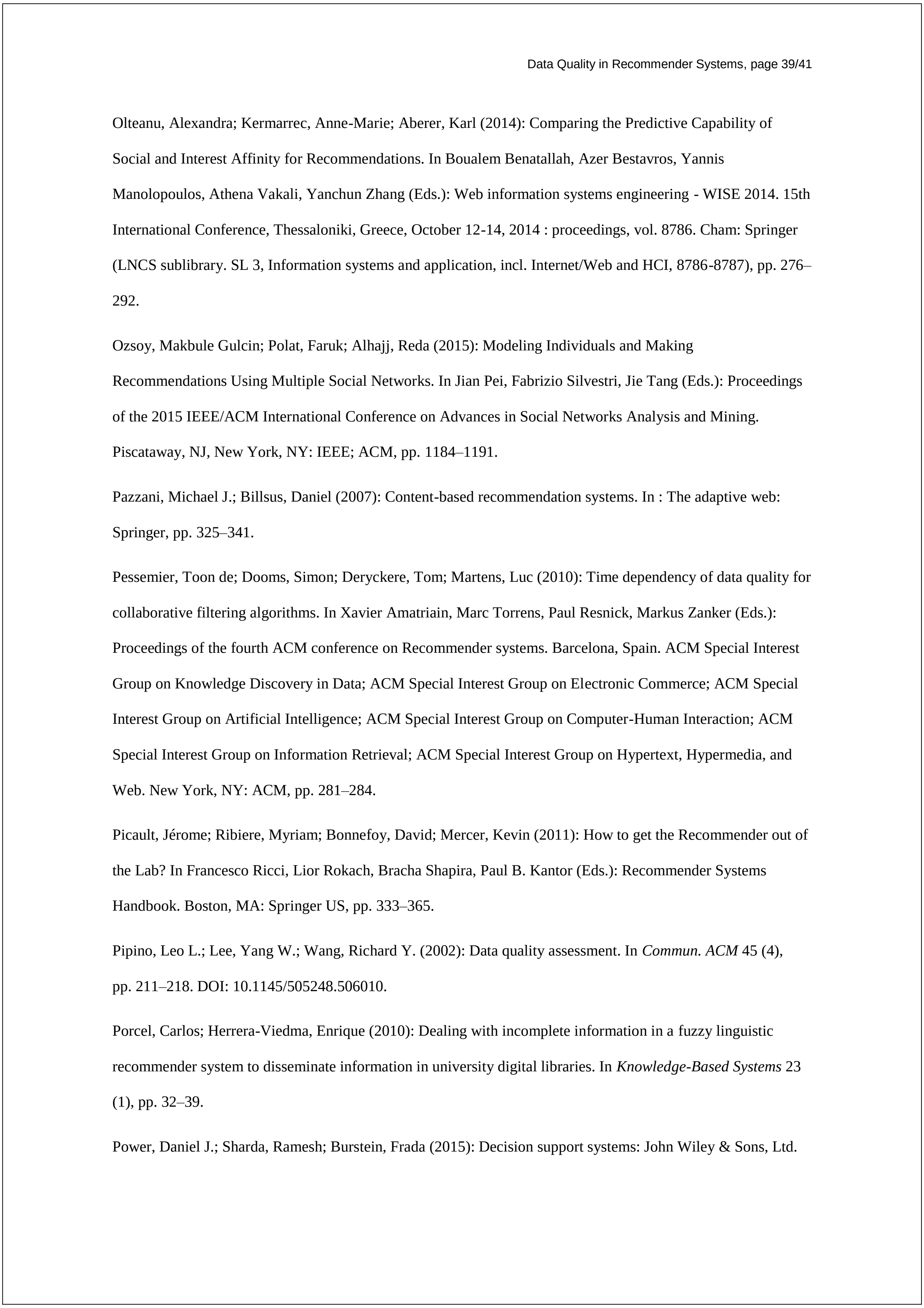
Matuszyk, Pawel; Spiliopoulou, Myra (2014): Predicting the Performance of Collaborative Filtering Algorithms. In Rajendra Akerkar, Nick Bassiliades, John Davies, Vadim Ermolayev (Eds.): WIMS '14 : 4th International Conference on Web Intelligence, Mining and Semantics. the 4th International Conference. Thessaloniki, Greece, 6/2/2014 - 6/4/2014. New York, New York, USA: ACM Press, pp. 1–6.

Mitra, P.; Murthy, C. A.; Pal, S. K. (2002): Unsupervised feature selection using feature similarity. In *IEEE Trans. Pattern Anal. Machine Intell.* 24 (3), pp. 301–312. DOI: 10.1109/34.990133.

Nguyen, Jennifer; Zhu, Mu (2013): Content-boosted matrix factorization techniques for recommender systems. In *Statistical Analy Data Mining* 6 (4), pp. 286–301. DOI: 10.1002/sam.11184.

Ning, Xia; Desrosiers, Christian; Karypis, George (2015): A comprehensive survey of neighborhood-based recommendation methods. In : Recommender systems handbook: Springer, pp. 37–76.

Ning, Yue; Shi, Yue; Hong, Liangjie; Rangwala, Huzefa; Ramakrishnan, Naren (2017): A Gradient-based Adaptive Learning Framework for Efficient Personal Recommendation. In Paolo Cremonesi, Francesco Ricci, Shlomo Berkovsky, Alexander Tuzhilin (Eds.): Proceedings of the Eleventh ACM Conference on Recommender Systems - RecSys '17. the Eleventh ACM Conference. Como, Italy, 27.08.2017 - 31.08.2017. New York, New York, USA: ACM Press, pp. 23–31.



Olteanu, Alexandra; Kermarrec, Anne-Marie; Aberer, Karl (2014): Comparing the Predictive Capability of Social and Interest Affinity for Recommendations. In Boualem Benatallah, Azer Bestavros, Yannis Manolopoulos, Athena Vakali, Yanchun Zhang (Eds.): Web information systems engineering - WISE 2014. 15th International Conference, Thessaloniki, Greece, October 12-14, 2014 : proceedings, vol. 8786. Cham: Springer (LNCS sublibrary. SL 3, Information systems and application, incl. Internet/Web and HCI, 8786-8787), pp. 276–292.

Ozsoy, Makbule Gulcin; Polat, Faruk; Alhajj, Reda (2015): Modeling Individuals and Making Recommendations Using Multiple Social Networks. In Jian Pei, Fabrizio Silvestri, Jie Tang (Eds.): Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. Piscataway, NJ, New York, NY: IEEE; ACM, pp. 1184–1191.

Pazzani, Michael J.; Billsus, Daniel (2007): Content-based recommendation systems. In : The adaptive web: Springer, pp. 325–341.

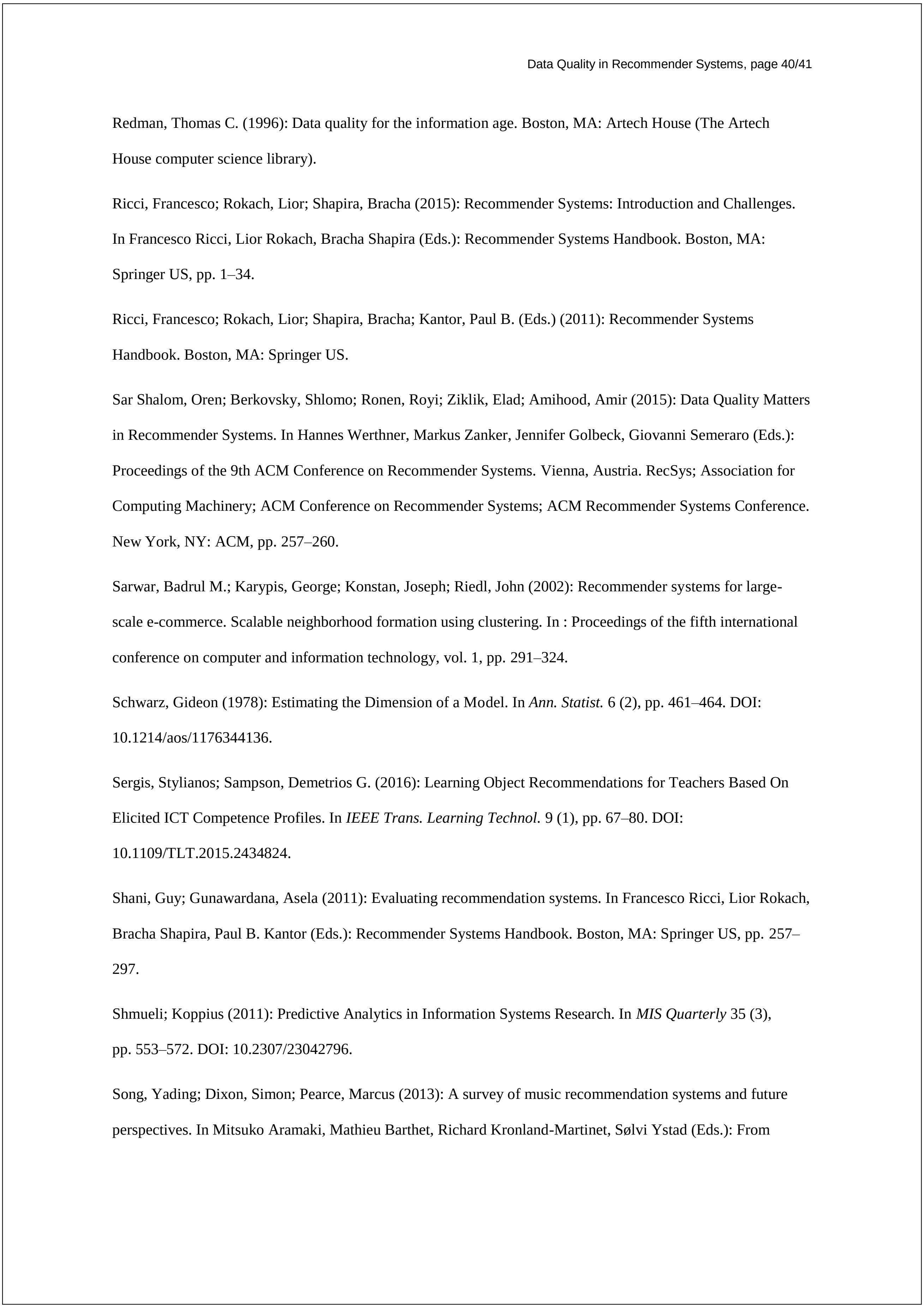
Pessemier, Toon de; Dooms, Simon; Deryckere, Tom; Martens, Luc (2010): Time dependency of data quality for collaborative filtering algorithms. In Xavier Amatriain, Marc Torrens, Paul Resnick, Markus Zanker (Eds.): Proceedings of the fourth ACM conference on Recommender systems. Barcelona, Spain. ACM Special Interest Group on Knowledge Discovery in Data; ACM Special Interest Group on Electronic Commerce; ACM Special Interest Group on Artificial Intelligence; ACM Special Interest Group on Computer-Human Interaction; ACM Special Interest Group on Information Retrieval; ACM Special Interest Group on Hypertext, Hypermedia, and Web. New York, NY: ACM, pp. 281–284.

Picault, Jérome; Ribiere, Myriam; Bonnefoy, David; Mercer, Kevin (2011): How to get the Recommender out of the Lab? In Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor (Eds.): Recommender Systems Handbook. Boston, MA: Springer US, pp. 333–365.

Pipino, Leo L.; Lee, Yang W.; Wang, Richard Y. (2002): Data quality assessment. In *Commun. ACM* 45 (4), pp. 211–218. DOI: 10.1145/505248.506010.

Porcel, Carlos; Herrera-Viedma, Enrique (2010): Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries. In *Knowledge-Based Systems* 23 (1), pp. 32–39.

Power, Daniel J.; Sharda, Ramesh; Burstein, Frada (2015): Decision support systems: John Wiley & Sons, Ltd.



Redman, Thomas C. (1996): Data quality for the information age. Boston, MA: Artech House (The Artech House computer science library).

Ricci, Francesco; Rokach, Lior; Shapira, Bracha (2015): Recommender Systems: Introduction and Challenges. In Francesco Ricci, Lior Rokach, Bracha Shapira (Eds.): Recommender Systems Handbook. Boston, MA: Springer US, pp. 1–34.

Ricci, Francesco; Rokach, Lior; Shapira, Bracha; Kantor, Paul B. (Eds.) (2011): Recommender Systems Handbook. Boston, MA: Springer US.

Sar Shalom, Oren; Berkovsky, Shlomo; Ronen, Royi; Ziklik, Elad; Amihood, Amir (2015): Data Quality Matters in Recommender Systems. In Hannes Werthner, Markus Zanker, Jennifer Golbeck, Giovanni Semeraro (Eds.): Proceedings of the 9th ACM Conference on Recommender Systems. Vienna, Austria. RecSys; Association for Computing Machinery; ACM Conference on Recommender Systems; ACM Recommender Systems Conference. New York, NY: ACM, pp. 257–260.

Sarwar, Badrul M.; Karypis, George; Konstan, Joseph; Riedl, John (2002): Recommender systems for large-scale e-commerce. Scalable neighborhood formation using clustering. In : Proceedings of the fifth international conference on computer and information technology, vol. 1, pp. 291–324.

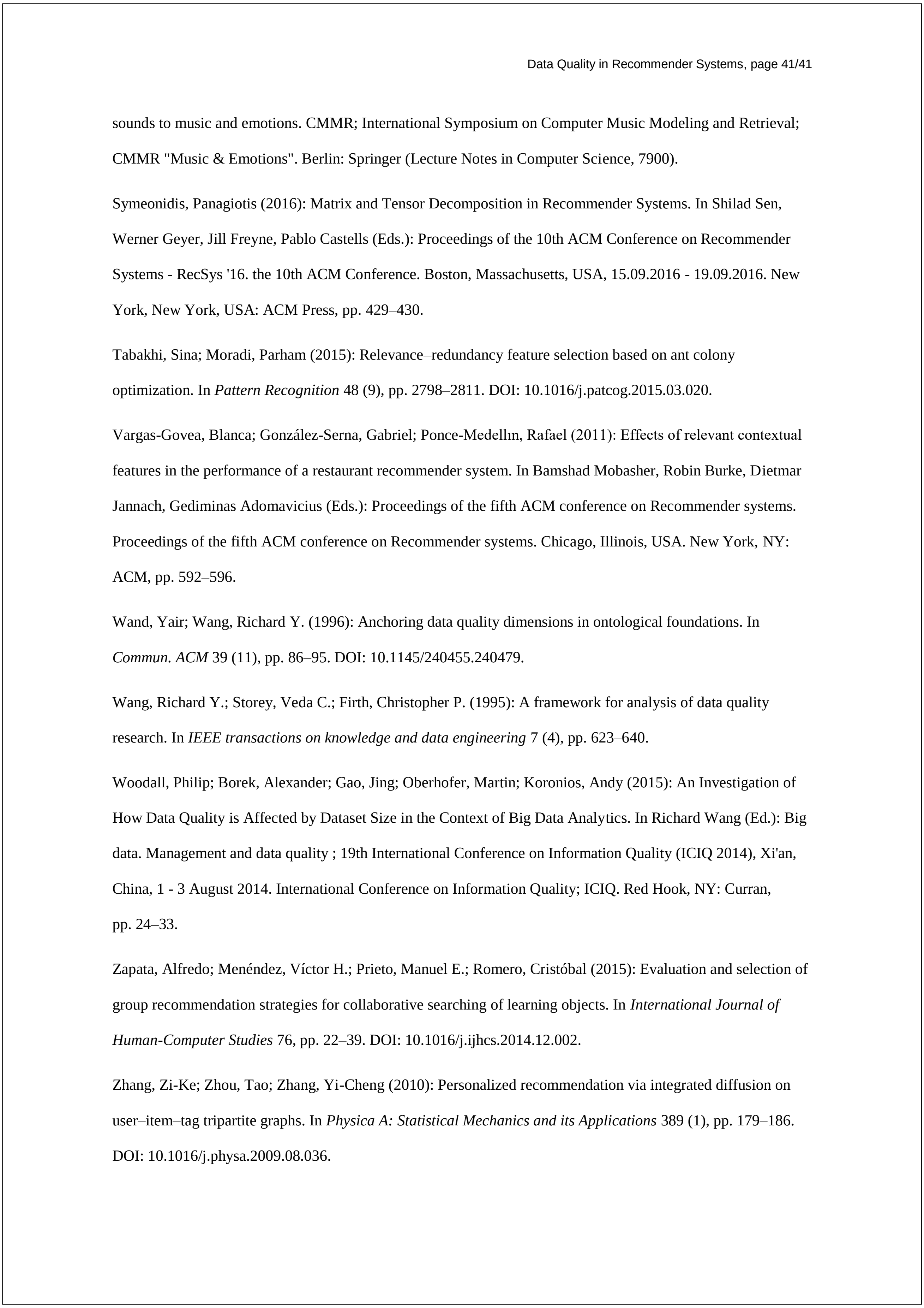
Schwarz, Gideon (1978): Estimating the Dimension of a Model. In *Ann. Statist.* 6 (2), pp. 461–464. DOI: 10.1214/aos/1176344136.

Sergis, Stylianos; Sampson, Demetrios G. (2016): Learning Object Recommendations for Teachers Based On Elicited ICT Competence Profiles. In *IEEE Trans. Learning Technol.* 9 (1), pp. 67–80. DOI: 10.1109/TLT.2015.2434824.

Shani, Guy; Gunawardana, Asela (2011): Evaluating recommendation systems. In Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor (Eds.): Recommender Systems Handbook. Boston, MA: Springer US, pp. 257–297.

Shmueli; Koppius (2011): Predictive Analytics in Information Systems Research. In *MIS Quarterly* 35 (3), pp. 553–572. DOI: 10.2307/23042796.

Song, Yading; Dixon, Simon; Pearce, Marcus (2013): A survey of music recommendation systems and future perspectives. In Mitsuko Aramaki, Mathieu Barthet, Richard Kronland-Martinet, Sølvi Ystad (Eds.): From



sounds to music and emotions. CMMR; International Symposium on Computer Music Modeling and Retrieval; CMMR "Music & Emotions". Berlin: Springer (Lecture Notes in Computer Science, 7900).

Symeonidis, Panagiotis (2016): Matrix and Tensor Decomposition in Recommender Systems. In Shilad Sen, Werner Geyer, Jill Freyne, Pablo Castells (Eds.): Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16. the 10th ACM Conference. Boston, Massachusetts, USA, 15.09.2016 - 19.09.2016. New York, New York, USA: ACM Press, pp. 429–430.

Tabakhi, Sina; Moradi, Parham (2015): Relevance–redundancy feature selection based on ant colony optimization. In *Pattern Recognition* 48 (9), pp. 2798–2811. DOI: 10.1016/j.patcog.2015.03.020.

Vargas-Govea, Blanca; González-Serna, Gabriel; Ponce-Medellın, Rafael (2011): Effects of relevant contextual features in the performance of a restaurant recommender system. In Bamshad Mobasher, Robin Burke, Dietmar Jannach, Gediminas Adomavicius (Eds.): Proceedings of the fifth ACM conference on Recommender systems. Proceedings of the fifth ACM conference on Recommender systems. Chicago, Illinois, USA. New York, NY: ACM, pp. 592–596.

Wand, Yair; Wang, Richard Y. (1996): Anchoring data quality dimensions in ontological foundations. In *Commun. ACM* 39 (11), pp. 86–95. DOI: 10.1145/240455.240479.

Wang, Richard Y.; Storey, Veda C.; Firth, Christopher P. (1995): A framework for analysis of data quality research. In *IEEE transactions on knowledge and data engineering* 7 (4), pp. 623–640.

Woodall, Philip; Borek, Alexander; Gao, Jing; Oberhofer, Martin; Koronios, Andy (2015): An Investigation of How Data Quality is Affected by Dataset Size in the Context of Big Data Analytics. In Richard Wang (Ed.): Big data. Management and data quality ; 19th International Conference on Information Quality (ICIQ 2014), Xi'an, China, 1 - 3 August 2014. International Conference on Information Quality; ICIQ. Red Hook, NY: Curran, pp. 24–33.

Zapata, Alfredo; Menéndez, Víctor H.; Prieto, Manuel E.; Romero, Cristóbal (2015): Evaluation and selection of group recommendation strategies for collaborative searching of learning objects. In *International Journal of Human-Computer Studies* 76, pp. 22–39. DOI: 10.1016/j.ijhcs.2014.12.002.

Zhang, Zi-Ke; Zhou, Tao; Zhang, Yi-Cheng (2010): Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs. In *Physica A: Statistical Mechanics and its Applications* 389 (1), pp. 179–186. DOI: 10.1016/j.physa.2009.08.036.

7 Paper 5: Something's Missing? A Procedure for Extending Item Content Data Sets in the Context of Recommender Systems

Something's Missing? A Procedure for Extending Item Content Data Sets in the Context of Recommender Systems

Bernd Heinrich*

Department of Management Information Systems, University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

Marcus Hopf

Department of Management Information Systems, University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

Daniel Lohninger

Department of Management Information Systems, University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

Alexander Schiller

Department of Management Information Systems, University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

Michael Szubartowicz

Department of Management Information Systems, University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

Abstract

The rapid development of e-commerce has led to a swiftly increasing number of competing providers in electronic markets, which maintain their own, individual data describing the offered items. Recommender systems are popular and powerful tools relying on this data to guide users to their individually best item choice. Literature suggests that data quality of item content data has substantial influence on recommendation quality. Thereby, the dimension completeness is expected to be particularly important. Herein resides a considerable chance to improve recommendation quality by increasing completeness via extending an item content data set with an additional data set of the same domain. This paper therefore proposes a procedure for such a systematic data extension and analyzes effects of the procedure regarding items, content and users based on real-world data sets from four leading web portals. The evaluation results suggest that the proposed procedure is indeed effective in enabling improved recommendation quality.

Keywords

Completeness, Data Extension, Data Quality, Recommender System

** Corresponding author:*
Prof. Dr. Bernd Heinrich
E-mail address: Bernd.Heinrich@ur.de
Telephone number: +49 941 943-6101
Fax number: +49 941 943-6120

1 Introduction

In line with the emergence and proliferation of the internet, e-commerce has developed into a major disruptor for retail business. Indeed, in 2020, retail e-commerce sales worldwide are estimated to hit $4.2 trillion, with its share of global retail reaching 16.1% and rising further to 22% in 2023 (Statista 2019). This rapid development of e-commerce has implied a swiftly increasing number of competing providers in electronic markets (e.g., *Amazon* and *Walmart* in general retail, *Booking.com* and *HRS* in hotel bookings, *Yelp* and *TripAdvisor* in restaurant bookings). Providers – even of the same domain – maintain their own, individual data sets containing information regarding the offered items (e.g., products or services), which usually vary in their attributes (content) to describe even the same items. For instance, *Booking.com* provides detailed data on location score and furniture of hotels, which is not offered by *HRS*. This data as well as the recommender systems commonly present on such e-commerce platforms aim at guiding users to their individually best item choice, improving user stickiness and increasing platform revenue (Zhou 2020). Such supporting systems are mandatory as customers regularly need to make a choice between a plethora of items (e.g., songs, movies, restaurants, hotels) on e-commerce platforms (Kamis et al. 2010; Levi et al. 2012; Richthammer and Pernul 2018; Tang et al. 2017; Vargas-Govea et al. 2011). It is thus hardly surprising that recommender systems in particular have been established as one of the most powerful and popular tools in the field of e-commerce in recent years (Ricci et al. 2015a; Scholz et al. 2017; Smith and Linden 2017).

As recommender systems are data-driven tools, the quality of the data which a recommender system is based on is assessed to be one of the issues recommender systems research is strongly involved with (Bunnell et al. 2019) and may have substantial influence on the resulting recommendations (Picault et al. 2011; Sar Shalom et al. 2015). Here, data quality is a multidimensional construct comprising several dimensions such as accuracy, completeness and currency of data (Batini and Scannapieco 2016; Pipino et al. 2002; Wand and Wang 1996), with each dimension providing a distinct view on data quality (e.g., Heinrich et al. 2018). For recommender systems examining the item content data (attributes and attribute values of items), achieving a more complete view on these items seems to be especially important (Adomavicius and Tuzhilin 2005; Picault et al. 2011), as "some representations capture only certain aspects of the content, but there are many others that would influence a user's experience" (Picault et al. 2011). This means that the data quality dimension completeness is of particular relevance for recommender systems.

Herein resides a considerable chance to improve recommendation quality by increasing completeness via extending an item content data set (e.g., from an e-commerce platform such as *TripAdvisor*) with additional attributes and attribute values from another data set in the same domain (e.g., from an e-commerce platform such as *Yelp*). This opportunity is particularly promising for search portals offering a meta view by compiling information from various platforms (e.g., *trivago.com*), which currently simply juxtapose the data and do not use an extended data set for the application of a recommender system. Yet, how to systematically achieve more complete item content data sets and realize the expected advantages for recommender systems is left unanswered in existing research. Thus, the paper at hand investigates the following research question:

How can an item content data set be systematically extended with respect to the data quality dimension completeness, aiming to improve recommendation quality?

As recommender systems are an important category of decision support systems (Power et al. 2015), this research is in line with recent works which have revealed a significant impact of data quality dimensions, especially completeness, on data-driven decision support systems (e.g., Feldman et al. 2018; Heinrich et al. 2019; Woodall et al. 2015).

The remainder of the paper is organized as follows. In the next section, the general and theoretical background as well as the related work are discussed. Thereafter, a procedure for the systematic extension of an item content data set with attributes and attribute values from another item content data set is presented, providing the basis for determining recommendations. In the fourth section, the proposed procedure is evaluated in two e-commerce real-world scenarios and resulting effects on recommendation quality are analyzed. The final section summarizes the work and discusses limitations as well as directions for future research.

2 Foundation

This section first discusses the positioning of recommender systems in the field of decision support systems in e-commerce as general background of our research. The second part of this section presents a theoretical model regarding the relationship between data quality and decision support systems – especially recommender systems – based on a discussion of existing literature. The third part of the section discusses related work and identifies the research gap addressed by this paper.

2.1 General Background

Recommender systems have become a highly relevant category of decision support systems (Power et al. 2015). In particular, in e-commerce, recommender systems are often necessary as users regularly need to make decisions for purchase, consumption or utilization of items (e.g., songs, movies, restaurants or hotels) from a plethora of possible alternatives available in information systems (IS) on e-commerce platforms (Kamis et al. 2010; Levi et al. 2012; Richthammer and Pernul 2018; Tang et al. 2017; Vargas-Govea et al. 2011).

More precisely, the high number of items together with the high number of users on e-commerce platforms lead to the problem of information overload, which is widely discussed by many researchers in the past decades and thus, constitutes a major subject of IS research in fields such as e-commerce (Lu et al. 2015) or management of business organizations (Edmunds and Morris 2000). In particular, information overload denotes the phenomenon regarding an individual's ability to appropriately cope with solving problems (e.g., making a choice) when more information is available than the individual can assimilate (Edmunds and Morris 2000). This is, users often do not have the skills and experience to adequately evaluate the large number of available alternatives for making their choice (Ricci et al. 2015b; Scholz et al. 2017). The resulting problem leaves users of e-commerce IS unable to make effective decisions due to this large volume of information (e.g., items) to which users are exposed to (Hasan et al. 2018; Lu et al. 2015; Richthammer and Pernul 2018; Scholz et al. 2017). In order to address the problem of information overload, the literature suggests for IS providers in e-commerce to incorporate decision support systems, in particular recommender systems, to assist users in their decision-making (Bunnell et al. 2019; Karimova 2016; Lu et al. 2015). Therefore, recommender systems aim at individually preselecting smaller sets of relevant items for each single user (i.e., information filtering; cf. Lu et al. 2015) to allow for good decision-making in a personalized and comfortable way avoiding to overwhelm the user (Manca et al. 2018).

Here, recommender systems are especially suitable to tackle the information overload problem, since they constitute data-driven systems, which enables them to individually support each user's decision-making in an automated manner (Bunnell et al. 2019; Karumur et al. 2018; Lu et al. 2015). A variety of IS research aims to tackle the information overload problem in the field of e-commerce by developing different approaches for recommender systems (e.g., Content-Based Filtering; cf. Aggarwal 2016; Jannach et al. 2012; Ricci et al. 2015a). In particular, recommender systems process different types of data (e.g., user rating data or item content data) in order to derive the individual users' preferences, which are stored in a user profile, based on data such as users' historical evaluations of other items (cf. Peska and Vojtas 2015; Ricci et al. 2015a). To enable recommendations of high precision, the matching of the user profile against item profiles (i.e., the content data of an item) or against other user profiles is highly relevant (Ricci et al. 2015a). This further emphasizes the key role of data (e.g., item content data) for recommender systems to enable individualized decision support for a large number of users in e-commerce settings (e.g., during shopping experiences on e-commerce websites; cf. Heinrich et al. 2019; Kamis et al. 2010).

In e-commerce, recommender systems not only assist users and make their experience on e-commerce platforms more comfortable, but they also create business value for the IS providers (Bunnell et al. 2019). By integrating recommender systems into a wide variety of e-commerce activities such as browsing, purchasing, rating or reviewing items, the resulting diversity of generated data (e.g., item content data, user rating data or click-stream data) can be used for modeling of user profiles and thus support certain marketing activities such as cross-selling, advertising or product promotion (Karimova 2016; Lu et al. 2015). It is thus hardly surprising that in recent years, recommender systems as data-driven tools have emerged to be among the most frequently applied decision support systems in the field of IS in e-commerce (Ricci et al. 2015a; Scholz et al. 2017; Smith and Linden 2017).

As recommender systems support user choices mainly on the basis of data, it seems promising to investigate how the data quality (e.g., completeness of item content data) influences the quality of recommender systems in the field of e-commerce.

2.2 Theoretical Background

The systematic procedure presented in this paper aims to contribute to further research investigating the relationship between data quality and (data-driven) decision support systems. At first glance, it might seem natural and obvious to suggest that more data always has a positive influence on decision support (especially when provided by a system). However, research in different areas shows that more data does not always lead to better results of decision support systems in general (e.g., when selecting features based on which a decision is obtained; cf. Mladenić and Grobelnik 2003; Vanaja and Mukherjee 2019), as different data sets (e.g., with more or fewer attributes) may lead to varying results of decision support. Thus, the impact of the data quality of data values on different evaluation criteria of decision support systems such as decision quality or data mining outcome has been studied in existing literature (e.g., Bharati and Chaudhury 2004; Blake and Mangiameli 2011; Feldman et al. 2018; Ge 2009; Heinrich et al. 2019; Woodall et al. 2015). Yet, this research neither focuses on how to systematically achieve more complete item content data sets nor on how to define a well-founded procedure, but instead tries to explain the relationship between data quality and evaluation criteria of decision support systems. In this regard, such explanatory models are the theoretical background in data quality research which we aim to support by our work. Thus, this background is briefly discussed in the following.

Bharati and Chaudhury (2004) assess the effects of the data quality dimensions accuracy, completeness and currency on the ability of an online analytical processing system to sustain decision-making. Ge (2009) discusses accuracy, completeness and consistency and their impact on decision quality. Blake and Mangiameli (2011) assess the impact of accuracy, completeness, consistency and currency on data mining results in order to support decision-making in companies. Woodall et al. (2015) analyze the impact of completeness on classification outcomes used for supporting users in their decision process. Feldman et al. (2018) propose an analytical framework to investigate the effects of incomplete data sets on a binary classifier that serves for decision support. Heinrich et al. (2019) examine the impact of the amount of available attributes and attribute values on the prediction accuracy of recommender systems.

Summing up, the focus of these papers is to investigate in which way and to what extent improving the quality of data values, especially the dimension completeness, leads to an improvement in evaluation criteria of particular decision support systems. A relevant and widely used category of decision support systems which assists users facing decision-making problems are recommender systems (Porcel and Herrera-Viedma 2010; Power et al. 2015). Based on this and in line with Heinrich et al. (2019), we refer to the theoretical model for describing the relationship between data quality and decision support systems, presented in Fig. 1.

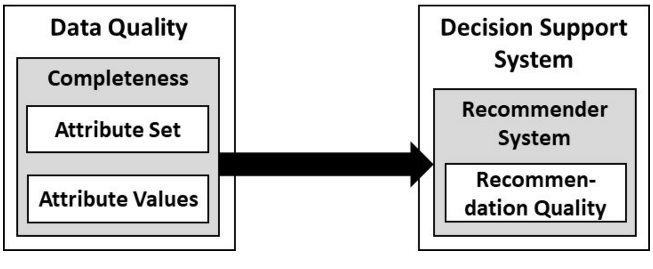


Fig. 1 Theoretical Model (according to Heinrich et al. 2019)

The theoretical model in Fig. 1 indicates a direct relationship between data quality and decision support systems. In particular, the theoretical model suggests this relationship between completeness of item content data (attributes and attribute values) and recommendation quality of recommender systems. With this model as theoretical background, the procedure presented in this paper proposes how to systematically extend items in an item content data set with attributes and attribute values of the same items from a second item content data set in order to gain a more complete view on the considered real-world entities (e.g., movies or restaurants). Thus, this systematic procedure forms the basis for an even more precise and well-founded investigation of the impact of completeness on the recommendation quality of data-driven decision support systems (especially recommender systems) in the future.[1] In particular, it enables theoretical relationships (i.e., similar to Fig. 1) for different data sets to be analyzed in a transparent and comprehensible manner. Furthermore, this procedure can serve as an already evaluated template for future procedures in order to support the investigation of further data quality dimensions (e.g., consistency) in other data-driven decision support systems.

2.3 Related Work and Research Gap

In this section, we present approaches dealing with data extension in the context of recommender systems and analyze relevant works discussing data quality aspects related to recommender systems.[2] Thereafter, we summarize existing contributions and identify the research gap addressed by this paper.

To prepare the related work, we followed the guidelines of standard approaches (e.g., Levy and Ellis 2006). In particular, we performed a literature search on the databases *ACM Digital Library*, *AIS Electronic Library*, *IEEE Xplore*, *ScienceDirect* and *Springer* as well as the proceedings of the *European and International Conference on Information Systems*, the *International Conference on Information Quality* and the *ACM Conference on Recommender Systems*. Subsequently, we examined whether these works represent relevant approaches for our research by reading title, keywords, abstract and summary and also conducted a forward and backward search in order to find further relevant works. After analyzing the resulting papers in detail, eighteen articles were deemed relevant. These papers could be organized within two separate categories, with each category containing nine works.

(1): The first category of works copes with some kind of data extension in the context of recommender systems. For these works, the effect on decision quality and in particular recommendation quality is vital ("fitness for use"). This is a crucial difference to general approaches for data extension (e.g., in the context of data warehouses), where the effect on decision quality is often unclear or difficult to assess. Although all papers of the first category consider data extension and its effect on recommendation quality, none of the approaches describes the systematic extension of an item content data set with additional data from the same domain in the form of a procedure in the context of recommender systems, which is the contribution of our research. Moreover, the approaches differ in the kind of extended data (1A), the entities extended with data (1B) and in the usage of different methods for data extension (1C).

(1A): Several recent articles focus on the extension of data with data from a distinct area, for example, data from different domains such as music and film (cross-domain data sets; Abel et al. 2013; Ntoutsi and Stefanidis 2016; Ozsoy et al. 2016), context information such as time and location (multi-dimensional data sets; Abel et al. 2013; Kayaalp et al. 2009) or data from different social and semantic web sources such as *Wikipedia*, *Facebook* and *Twitter* (heterogeneous data sets; Abel et al. 2013; Bostandjiev et al. 2012; Chang et al. 2018; Kayaalp et al. 2009; Ozsoy et al. 2016). These approaches examine whether the diversity of data types leads to improved recommendation quality but do not systematically extend item content data with additional data from the same domain.

(1B): Other works in literature analyze user profiles from different social networks (Abel et al. 2013; Li et al. 2018; Ozsoy et al. 2016; Raad et al. 2010). The matching user profiles are merged across different

[1] In this regard, an implementation of the procedure is available on GitHub (GitHub 2020).

[2] Some approaches for data extension with regard to completeness (e.g., cf. Naumann et al. 2004; Bleiholder and Naumann 2008; Scannapieco and Batini 2004) mainly deal with technical issues (e.g., wrapper architecture, database architecture) or model-oriented aspects (e.g., schema mapping, operators, join approaches), which are not within the scope of this work.

networks to produce a positive effect on recommendation quality. However, these works do not focus on item content data at all.

(1C): Finally, some recent works focus on the extension of item or user data from multiple data sources in the context of recommender systems (Abel et al. 2013; Bostandjiev et al. 2012; Bouadjenek et al. 2018; Ozsoy et al. 2016). These approaches rely on tools such as *BlogCatalog*, *Google Social Graph API*, *Google Search API* or *OpenID*, which provide information for the matching of users or items. However, these works do not focus on describing the systematic extension of an item content data set and instead use external, non-transparent methods for data extension, which severely limits their applicability in other scenarios.

(2): The second category of works explicitly recognizes the importance of data quality for recommender systems (Amatriain et al. 2009; Basaran et al. 2017; Berkovsky et al. 2012; Burke and Ramezani 2011; Heinrich et al. 2019). In particular, Heinrich et al. (2019) examine the impact of the number of available attributes and attribute values on prediction accuracy of recommender systems by testing hypotheses but do not provide a procedure for extending an item content data set with additional attributes and attribute values. Further approaches give rise to concepts that deal with data quality issues in the context of recommender systems. For instance, data sources used by a recommender system can be chosen user-dependently as data sparsity and inaccuracy have been identified to impact recommendation quality (Lathia et al. 2009). Sar Shalom et al. (2015) tackle sparsity and redundancy issues by deleting or omitting certain users or items while Pessemier et al. (2010) analyze consumption data such as ratings in regard to currency. Further, Levi et al. (2012) use text mining on user reviews from various sources to alleviate the cold start problem of new users by assigning them to so called context groups.

In summary, none of these works provides a systematic procedure for the extension of a data set with item content data of another data set from the same domain. The works in category (1A) focus on the extension with data from a different area, but they do not target on data representing the *same* items, which is a decisive characteristic of our research. The works in category (1B) do not focus on item content data but instead analyze user profiles from various social networks. In contrast to this, we provide a procedure for the matching and extension of *item* content data. The works in category (1C) use existing tools for data extension, especially for user data. Such an extension is non-transparent, highly dependent on these tools as well as the application scenario and does not allow to support the analysis of theoretical relationships (cf. Fig. 1) between different data sets in a verifiable and comprehensible manner. Additionally, no explicit procedure for extending an item content data set with additional attributes and attribute values in detail is given. The works of the second category analyze the impact of data quality on recommender systems. However, only Heinrich et al. (2019) analyze effects of a more complete view on items by data set extension. Yet, this work does not aim to provide a procedure for the extension of item content data in the context of recommender systems. In contrast, the authors present an explanatory analysis based on hypotheses testing. To conclude, none of these approaches presents a systematic procedure for the extension of a data set with item content data of another data set from the same domain.

3 A Procedure for Extending an Item Content Data Set

In this section, we propose a procedure for the systematic extension of a data set in the context of recommender systems, aiming to improve the quality of the resulting recommendations. We discuss and substantiate in detail how to extend a data set DS1 containing items and item attributes from a certain domain (e.g., movies, restaurants or hotels) by using a data set DS2 containing items and item attributes from the same domain.[3] In particular, items in DS1 are extended with attributes and attribute values of the same items from DS2. This means that in a first step *duplicates have to be detected* before in a second step, the *data sets can be actually integrated into one data set*.

The exact elaboration of these two steps in the context of recommender systems addresses our research question and thus represents the contribution of this paper. In a subsequent step, the resulting data set extension can be evaluated by *determining recommendations* based on the extended data set and assessing

[3] If more than two data sets are available, the procedure can be applied iteratively.

the resulting recommendation quality. Since different existing content-based or hybrid recommender systems can be used for this step, it is not a core element of the procedure. The procedure is illustrated in Fig. 2 and described in the following.

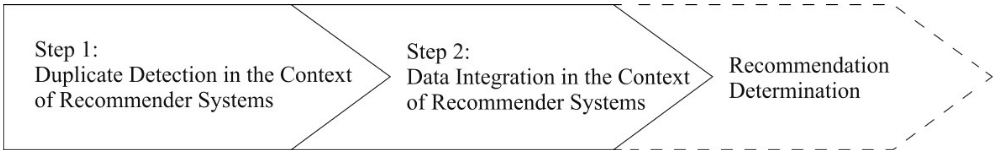

Step 1:
Duplicate Detection in the Context of Recommender Systems
Step 2:
Data Integration in the Context of Recommender Systems
Recommendation Determination

Fig. 2 Procedure to Extend an Item Content Data Set in the Context of Recommender Systems

3.1 Duplicate Detection in the Context of Recommender Systems

An item in a data set DS1 usually has different attributes and attribute values compared to its corresponding duplicate item in a data set DS2 (e.g., because the portals have heterogeneous data policies), making duplicate detection in the context of recommender systems a non-trivial task. Here, duplicate detection is a binary classification of item pairs (one item from DS1 and one item from DS2) with the two classes *duplicate* and *non-duplicate*. Due to a potentially large number of items per data set, duplicate detection should be carried out in a widely automated manner. To assist this task, literature proposes *similarity measure functions* (SMFs; e.g., the Jaro-Winkler function; Winkler 1990) to determine the similarity of *key attributes* (e.g., "Name" and "Geolocation" of a restaurant) between items from DS1 and DS2, with high similarity values indicating possible duplicates. We propose the following four Tasks 1.1-1.4 to configure and perform duplicate detection, acknowledging peculiarities in the context of recommender systems (cf. Fig. 3).

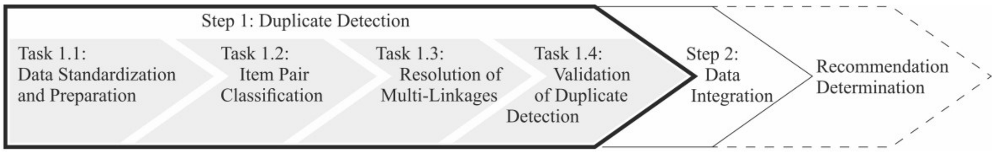

Step 1: Duplicate Detection
Task 1.1:
Data Standardization and Preparation
Task 1.2:
Item Pair Classification
Task 1.3:
Resolution of Multi-Linkages
Task 1.4:
Validation of Duplicate Detection
Step 2:
Data Integration
Recommendation Determination

Fig. 3 The Step Duplicate Detection in Detail

In **Task 1.1**, the data for duplicate detection is standardized and prepared. This is necessary because different portals often specify varying values for (key) attributes (e.g., due to heterogeneous data policies). Furthermore, as the data is usually decentrally generated by many different users, these users often enter attribute values on their very own interpretation, leading to data quality problems in e-commerce platforms. These issues make duplicate detection for recommender systems data sets highly complex. For example, one and the same US phone number could be entered as "+1-212-283-1100" in one data set and as "(212) 283-1100" in the other data set. Here, it is clear that a standardization of both phone numbers to "area code: 212, phone number: 2831100" helps determining that these numbers refer to the same phone connection in an automated manner. The standardization of the key attributes can be conducted by utilizing specific parsing tools which standardize the values of the key attributes (e.g., the python package "phonenumbers" for the key attribute "Phone"). After standardization, the values for all key attributes of both data sets DS1 and DS2 are stored in a common standard format. Nevertheless, even after standardization, duplicate items in DS1 and DS2 may differ in key value attributes caused by varying entered values (e.g., "283-100" instead of "283-1100"). Subsequent to standardization, item pairs are prepared for binary classification in the next task. Here, each item from DS1 in combination with each item from DS2 is considered as an item pair. It is clear that most of these pairs are non-duplicates. Therefore, it is beneficial to discard the item pairs which are obvious non-duplicates (e.g., restaurants with a GPS distance larger than 1,000 meters), which is referred to as blocking in literature (Steorts et al. 2014).

Task 1.2 comprises the binary classification of item pairs as duplicates or non-duplicates. In many contexts, this classification can be performed rather easily in a supervised manner. However, in the context of recommender systems, generally, no substantial amount of labeled training data (i.e., item pairs labelled as (non-)duplicates) is available for a supervised classification. Therefore, it is crucial to perform item pair classification in an unsupervised manner, not requiring any labeled training data (cf., e.g., Jurek et al. 2017). In the following, we describe the basic ideas of such an algorithm and emphasize the crucial peculiarities of the algorithm in the context of recommender systems. The algorithm starts with an initialization, followed by the proper classification and ends with all item pairs being classified as duplicate or non-duplicate.

The initialization consists of the selection of SMFs that are used for the classification. For each key attribute available in both data sets DS1 and DS2, adequate SMFs have to be specified. The choice of SMFs primarily depends on the data type of the respective key attribute. In particular, for key attributes containing string values and key attributes containing numerical values, different SMFs have to be used (e.g., the haversine SMF for GPS data values and the Jaro-Winkler SMF for string data values; cf. Table 1). Here, it is important to not only select one SMF per key attribute, but to select multiple SMFs with different characteristics, since the compared values of the key attributes may also exhibit varying deviations and specifications. For string attribute values with different suffixes (e.g., a restaurant is represented by “Fluffy’s New York” in DS1 and by “Fluffy’s Café & Pizzeria” in DS2), a SMF that focuses on the initial characters of a string such as the Jaro-Winkler SMF is appropriate. Further, for string attribute values with typographical errors (e.g., a restaurant is represented by “Fulffy’s” in DS1 and by “Fluffys” in DS2), a SMF addressing this special deviation such as the Levenshtein SMF is suitable. Therefore, it is important to utilize multiple SMFs for item pair classification to cope with the challenges of highly diverse data values in the context of recommender systems. To further elaborate on the specification of SMFs for item pair classification, we give a broader discussion of selected SMFs with different characteristics in Table 1 based on Christen (2012) and state their properties and examples in the context of recommender systems.

The proper classification is then conducted via an unsupervised ensemble self-learning algorithm, which improves results compared to just using the values of SMFs for classification (Jurek et al. 2017). This self-learning algorithm starts with training a certain binary classifier. The training is conducted on a small set of training data, which consists of the item pairs with the highest similarity values (implicitly labeled as duplicates) and item pairs with the lowest similarity values (implicitly labeled as non-duplicates) and thus does not need to be labeled manually. This binary classifier is then used to predict the classes of all other item pairs. The item pairs classified with a high certainty are then added to the training data. Subsequently, the binary classifier is trained again and the steps are gradually repeated until all item pairs are classified as either duplicates or non-duplicates by this certain binary classifier. To further increase the robustness of the classification result, multiple such binary classifiers are used with the described self-learning method and the obtained results are then aggregated to obtain the final stable result of the item pair classification.

Table 1. Selected Similarity Measure Functions and their Application in the Context of Recommender Systems

Similarity measure functions	*Properties*	*Examples in the context of recommender systems*
Levenshtein The Levenshtein SMF is based on the minimum number of edit operations of single characters necessary to transform a string s_1 into a string s_2.	• Appropriate for misspellings/ typographical errors • Inappropriate for truncated/ shortened strings and divergent pre-/suffixes • Complexity: $O(\|s_1\| * \|s_2\|)$	Typographical error in the attribute "Restaurant Name": "Fulffy's" vs. "Fluffys".
Jaro The Jaro SMF is based on the number of agreeing characters c contained in the strings s_1 and s_2 within half the length of the longer string, and the number of transpositions t in the set of common substrings.	• Appropriate for misspellings/ typographical errors • Inappropriate for long divergent pre-/suffixes • Complexity: $O(\|s_1\| + \|s_2\|)$	Misspelling in the attribute "Restaurant Name": "Fluffy's Café" vs. "Flufy's Café".
Jaro-Winkler The Jaro-Winkler SMF extends the Jaro SMF, putting more emphasis on the beginning of the strings.	• Appropriate for misspellings/typographical errors and divergent suffixes • Inappropriate for long divergent prefixes • Complexity: $O(\|s_1\| + \|s_2\|)$	Divergent suffixes of the attribute "Restaurant Name": "Fluffy's New York" vs. "Fluffy's Café & Pizzeria".
Haversine This SMF is based on the haversine formula, which measures the distance between two locations on earth.	• Appropriate for geographical coordinates given in latitude/longitude	"40.711, -73.966" vs. "40.710, -73.965".

In **Task 1.3,** it is necessary to resolve multi-linkages of duplicates resulting from Task 1.2. This problem may arise as an item from DS1 can be contained in more than one item pair classified as a duplicate. Thus, this item from DS1 is linked to more than one item from DS2. Similarly, an item from DS2 can be linked to more than one item from DS1. As the matched items will be proposed to users in the recommendation step, it is important to resolve these multi-linkages of items to avoid redundant and multiple recommendations of individual items. To resolve the multi-linkages, the prediction scores of the ensemble classifier from Task 1.2 are used. Considering an item from DS1 linked to multiple items from DS2, only the linkage with the highest prediction score is retained and all other linkages are discarded. Analogously, only one linkage is kept when an item from DS2 is linked to multiple items from DS1. In this way, the n-to-n reference of items from DS1 and DS2 is firstly reduced to 1-to-n references and then to 1-to-1 references.

Step 1 concludes with the validation of the results of the duplicate detection in **Task 1.4**, which is necessary to assess the quality of the duplicate detection. This quality plays an important role in the context of recommender systems, as false duplicates would result in erroneous data integrations in the next step of the procedure, and thereby, to negative effects on item recommendations. On the other hand, false negatives would result in feasible data integrations not being carried out, thus reducing the benefit of the procedure. Therefore, a small excerpt of item pairs, serving as test data, needs to be labeled as duplicates or non-duplicates for validation purposes. Here, a random selection of item pairs to be labeled would result in the vast majority of these item pairs being labeled as non-duplicates, since most item pairs are indeed non-duplicates. Therefore, it is important to take the calculated values of the SMFs into account and to also label item pairs which are more likely to be a real duplicate. Building on this labeled test data, the number of correct classifications (i.e., "true positives" and "true negatives") and the number of errors (i.e., "false positives" and "false negatives") can be determined. Based on these numbers, evaluation metrics such as precision, recall and F1-measure can be assessed. If these evaluation metrics report unsatisfactory results, the classification errors may be analyzed and tackled. The evaluation metrics thus enable to ensure a high quality of the conducted duplicate detection and to provide data suitable for the next step of the procedure, which concludes Task 1.4 and thus Step 1.

3.2 Data Integration in the Context of Recommender Systems

In Step 2 of the procedure, attributes and attribute values of DS1 and DS2 are integrated to obtain the envisioned more complete view on items. In particular, *matching* attributes (i.e., attributes of DS2 also existing in DS1) and *additional* attributes (i.e., attributes only existing in DS2) have to be identified and the items' attribute values have to be extended. To perform this integration in the context of recommender systems, we propose the following three Tasks 2.1-2.3 (cf. Fig. 4).

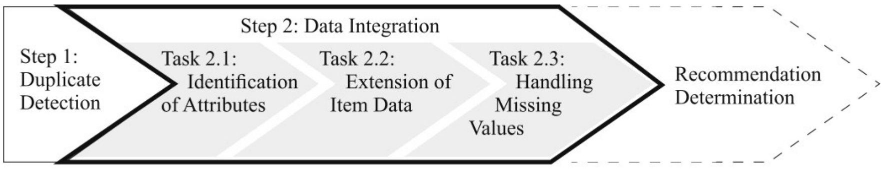

Step 1: Duplicate Detection
Step 2: Data Integration
Task 2.1: Identification of Attributes
Task 2.2: Extension of Item Data
Task 2.3: Handling Missing Values
Recommendation Determination

Fig. 4 The Step Data Integration in Detail

The goal of **Task 2.1** is to identify matching attributes. To do so, the attributes of DS2 have to be compared to the attributes of DS1. The automated identification of matching attributes can prove to be non-trivial in the context of recommender systems because different portals often use varying names for the same attribute (e.g., "Artist" and "Performer") due to heterogeneous data policies. An incorrect matching of attributes can lead to items being assigned wrong data and thus have a direct detrimental impact on recommendation quality. As this task is of relatively low complexity for humans, the identification may be performed in a manual manner (e.g., the manual matching of 143 attributes in DS1 to 251 attributes in DS2 in the application scenario regarding restaurants of our evaluation took approximately one hour and exhibited almost perfect inter-coder reliability). In contrast, an automated identification (e.g., using WordNet) may be error-prone, as it is difficult for an algorithm to directly identify attributes such as "Artist" and "Performer" as matching attributes. Furthermore, an automated identification requires a subsequent manual verification by humans, which is also time-consuming. Overall, an automated identification should only be performed when the number of attributes is extremely high, rendering a manual identification ineffective. In any case, all attributes of DS2 not matched to an attribute of DS1 are identified as additional attributes.

In **Task 2.2**, the item content data is extended for each item in DS1. More precisely, the item content data subsequently consists of the attributes of DS1 and the additional attributes of DS2. Additional attributes allow a more complete view on the considered item and may improve recommendation quality. In particular, additional attribute values can have enormous leverage for users with many item reviews in the context of recommender systems, since a large number of affected rated items can be described in more detail with the additional content. Depending on the recommender system used or under trade-off considerations, it may be helpful to limit the number of the additional attributes considered for data extension. To identify a subset of additional attributes for which a strong improvement of recommendation quality is expected (e.g., attributes with very many missing values may hardly impact recommendation quality), several options are possible (e.g., the use of an attribute selection algorithm; cf. Chandrashekar and Sahin 2014; Molina et al. 2002). These options are discussed in more detail in Section 4.3. After selecting the additional attributes, for each item in DS1 for which a duplicate in DS2 was identified and for each additional attribute chosen, the respective attribute values of the duplicate are inserted into the item content data.

After Task 2.2, some attribute values of items in the extended data set may still be missing because they are not provided by either data set (e.g., the values of the attribute "Genres" are not given for all items in the movie domain). These missing values have to be addressed in **Task 2.3**, since many recommender systems cannot operate on missing attribute values. Moreover, missing attribute values may be detrimental to recommendation quality. Therefore, a further extension of item content data is enabled by imputation methods. More precisely, missing attribute values can be inferred via imputation based on non-missing attribute values in the extended data set. Here, the presented procedure provides an advantage compared to imputing values based on just DS1 as the attribute values from both data sets DS1 and DS2 are available and can be used as basis for the imputation. Table 2 discusses selected imputation methods and their relevance in the context of recommender systems based on Enders (2010). In addition to these imputation methods, it is also feasible to impute values in a user-specific way which is more flexible than assigning fixed values for the missing values in the extended data set. In this case, the missing values of all items rated by a user can be handled by an imputation method from Table 2 (e.g., Arithmetic Mean Imputation) to capture the user's preferences more accurately when generating her/his user profile.

Table 2. Selected Methods for Handling Missing Values and their Application in the Context of Recommender Systems

Imputation methods	*Properties*	*Examples in the context of recommender systems*
Arithmetic Mean Imputation (AMI) Missing attribute values are replaced with the mean attribute value of all items, where the values for this attribute are not missing.	• AMI is convenient to implement • AMI attenuates standard deviation and variance	Each missing value of the attribute "Runtime" is replaced with the mean value of "Runtime" (as an indicator) over all movies that do have a value for "Runtime".
Regression Imputation (RI) Missing values are replaced with predicted scores from regression equations. The regression equations are estimated by analyzing the extended data set.	• RI is complicated to implement • RI attenuates standard deviation and variance (but less than AMI)	For two hotel attributes "Price" (P_i) and "Service" (S_i), there are only missing values for "Service". A regression equation $\hat{S}_i = \hat{\beta}_0 + \hat{\beta}_1(P_i)$ for the attribute "Service", depending on the attribute "Price", is estimated by analyzing the hotels with given values for "Service". The missing values S_i of "Service" are replaced by $\hat{S}_i$.
Hot Deck Imputation (HDI) Missing attribute values of an item are replaced with the corresponding values of the most similar item.	• HDI is convenient to implement • HDI attenuates standard deviation and variance (but less than AMI)	The movie "The Dark Knight" is the most similar movie to "The Dark Knight Rises", as both movies belong to the batman trilogy of the director "Christopher Nolan". The value of "The Dark Knight" for the attribute "Genres" is "Action" and thus, the missing value of "The Dark Knight Rises" for "Genres" is inferred with the value "Action".

3.3 Subsequent Step: Recommendation Determination

Subsequent to duplicate detection and data integration, recommendations for users on e-commerce platforms can be inferred by applying a recommender system based on the extended data set and evaluating the resulting recommendations. This step is also necessary to analyze the effects of data set extension on recommendation quality. As our approach is tailored to data sets containing item content data in addition to rating data, it is feasible to apply both content-based as well as hybrid recommender systems that leverage both data types

(Ricci et al. 2015b). Handling item content data is very important in e-commerce settings, because the recommender system can map the potentially extensive needs of customers more accurately due to the more precise description of the items (e.g., proposal of tailored products based on product preferences). Therefore, for this *subsequent* step of our procedure, we suggest to apply the state-of-the-art hybrid recommender system approach Content-Boosted Matrix Factorization (CBMF; cf. Forbes and Zhu 2011), which utilizes both rating data and, in particular, item content data and is thus more comprehensive than collaborative filtering recommender systems. Matrix factorization approaches have become very popular through the Netflix contest, which started in 2006 and ended in 2009 (Koren 2009; Koren et al. 2009), and now constitute state-of-the-art recommender systems (Kim et al. 2016; Ning et al. 2017). As a matrix factorization approach, CBMF learns a model by optimizing a loss function based on training data and therefore, preliminary steps such as attribute weighting or attribute selection are not necessary for CBMF (Koren 2009; Nguyen and Zhu 2013).

The basic idea of matrix factorization recommender systems is to decompose the rating matrix R (users as rows; items as columns) into two low-rank matrices P (representing users) and Q (representing items), with $PQ \approx R$. Then, the idea of CBMF is to further decompose the matrix Q into a low-rank matrix A and the matrix F, with $AF^T = Q$ and F containing the attribute vectors of items (items as rows; attributes as columns). Hence, the overall idea is that the rating matrix R can be approximated by $R \approx PAF^T$. In particular, CBMF learns a n-dimensional vector of latent factors $p_u \in \mathbb{R}^n$ for each user u and a n-dimensional vector of latent factors $a_f \in \mathbb{R}^n$ for each attribute f, such that the actual rating r_{ui} for a user-item pair (u, i) is approximated by the predicted star rating $\hat{r}_{ui} = p_u^T q_i$, with $q_i = \sum_{f \in F_i} a_f$ and F_i being the set of attributes that are assigned to the item i. Finally, to evaluate the effects of the data set extension on recommendation quality, the rating data is split into training data for learning the parameters of the CBMF model (p_u and a_f) and test data to assess the recommendation quality via quality measures such as Root-Mean-Square-Error (RMSE; cf. Shani and Gunawardana 2011).

4 Evaluating the Procedure in Real-world Scenarios

In this section, we evaluate the proposed procedure in two real-world e-commerce scenarios. First, the reasons for selecting these scenarios are discussed and the used data sets are described. Thereafter, the evaluation of the procedure with respect to these data sets is outlined. Finally, important effects of the data set extension regarding items, content and users on recommendation quality are presented.

4.1 Selection and Description of the Real-world Scenarios

We evaluated the procedure in two real-world e-commerce scenarios regarding the domains of restaurants and movies. While these domains are frequent subjects of IS research in e-commerce (Chang and Jung 2017; Nguyen et al. 2018; Wei et al. 2013; Yan et al. 2015), both domains exhibit versatile facets and different challenges for a procedure for data set extension. Thereby, analyzing these two domains allows for a broader evaluation of the proposed procedure in e-commerce application scenarios.

First, we selected the domain of restaurants because this domain is very challenging regarding duplicate detection (i.e., Step 1 of the procedure, e.g., the resolution of multi-linkages of duplicates (Task 1.3)) in the context of recommender systems. In comparison to other domains (e.g., the domain of movies as second scenario) there are items with the same name being found in the immediate vicinity (i.e., in the case of restaurant chains such as McDonald's or Subway), which makes this domain especially challenging. For the real-world scenario in the domain of restaurants, we prepared data sets of two leading advertising web portals which provide crowd-sourced ratings about businesses (e.g., restaurants). The first portal (Portal R1) focuses on travel opportunities and businesses such as restaurants and provided over 650 million ratings whereas the second portal (Portal R2) specializes on local businesses such as bars or restaurants and provided over 150 million ratings by 2020. These portals were chosen because an initial check revealed that, while both portals contain data about an overlapping set of real-world entities, they offer an interestingly different view (i.e., different attributes) on these entities. In particular, we selected the area of New York City (USA) as both portals provided a large number of items, users and ratings for this area. In this way, the evaluation of the procedure and the analysis regarding its effects on recommendation quality could be performed on a sufficiently large data basis. Here, the data from Portal R1 consists of more than 8,900 items representing restaurants in the area of New York City, rated by over 380,000 users with approximately 850,000 ratings. The data from Portal R2 consists of over 18,500 items representing restaurants in the same area, rated by more than 580,000 users with around 2.4 million ratings. Each item of Portal R1 is described by the key attributes “Name”, “Postal Code”, “Geolocation”, “Address”, “Phone” and “District”, category attributes such as “Italian Cuisine” or “Pizza”, and business information attributes such as “Parking Available” or “Waiter Service”. In Portal R2, items are described by key attributes equaling those in Portal R1 as well as (partly different) category attributes and business information attributes. The data from Portal R1 contains around 3,000 missing values for one attribute whereas the data from Portal R2 contains more than 190,000 missing values for 26 attributes. In our evaluation, we extended the data from Portal R1 with the data from Portal R2 (i.e., the data from Portal R1 served as DS_{R1} and the data from Portal R2 served as DS_{R2}). Table 3 describes the restaurant data sets.

Table 3. Description of the Restaurant Data Sets

	Portal R1 (DS_{R1})	**Portal R2 (DS_{R2})**
# of items (restaurants)	8,909	18,507
# of users	386,958	583,815
# of ratings	855,357	2,396,643
# of key attributes	6	6
# of further attributes (category attributes and business information attributes)	143	251
# of possible attribute values	1,247,260	4,589,736
# of missing values	3,253 (0.26%)	190,789 (4.16%)

In addition, we selected the domain of movies because this domain exhibits further but different challenges regarding item content data extension in the context of recommender systems. In comparison to the restaurant domain, the detection of duplicates and in particular the resolution of multi-linkages of duplicates is less challenging in the movie domain, since different movies have usually different titles (as key attribute) due to copyright standards. Nevertheless, Step 1 of the procedure is still favorable for movies in order to detect non-trivial movie duplicates in case the movie titles do not exactly match, as key attributes can (slightly) vary between different portals in some cases (e.g., the movie titles “Mission: Impossible – Ghost Protocol” and “Mission: Impossible – Ghost Protocol (2011)” represent the same item). Moreover, an initial check revealed that the amount of missing values in the data sets of both movie web portals (Portal M1 and Portal M2) is very high compared to other domains (e.g., restaurants). This means that Step 2 of the procedure including the task of handling missing values is even more important for the real-world scenario in the movie domain. Hence, we prepared data sets of two leading web portals which provide crowd-sourced ratings about movies. Here, the data from Portal M1 consists of approximately 29,000 movie items, rated by over 425,000 users with nearly 530,000 ratings. The data from Portal M2 consists of over 12,500 movie items, rated by approximately 230,000 users with nearly 410,000 ratings. Each item of Portal M1 is described by the key attribute “Title” and further attributes such as “Brand”. In Portal M2, items are described by the same key attribute as in Portal M1 as well as by further attributes such as “Cast” and “Language”. The data from Portal M1 contains over 245,000 missing values for all attributes whereas the data from Portal M2 contains more than 1 million missing values for all attributes. In our evaluation, we extended the data from Portal M1 with the data from Portal M2 (i.e., the data from Portal M1 served as DS_{M1} and the data from Portal M2 served as DS_{M2}). Table 4 describes the movie data sets.

Table 4. Description of the Movie Data Sets

	Portal M1 (DS_{M1})	Portal M2 (DS_{M2})
# of items (movies)	28,973	12,842
# of users	428,519	230,151
# of ratings	528,777	409,935
# of key attributes	1	1
# of further attributes	13	103
# of possible attribute values	376,649	1,322,726
# of missing values	247,341 (65.67%)	1,082,387 (81.83%)

4.2 Evaluation of the Procedure

In this section, we discuss the evaluation of the procedure for extending data sets with item content data in the restaurant and movie domain and present the evaluation results for each step for both domains.

4.2.1 Evaluation of Step 1 – Duplicate Detection

In the following, we outline the evaluation of the duplicate detection step. More precisely, the goal of this section is to assess the evaluation metrics precision, recall and F1-measure of duplicate detection. Therefore, we first discuss how we conducted and validated the tasks of this step and then present the evaluation results. Since this step is more challenging for restaurants, we especially focus on this domain.

To begin with, in Task 1.1, the key attribute values (cf. Table 5) of DS_{R1} and DS_{R2} were standardized due to inconsistent values caused by heterogeneous data policies among restaurant portals. For example, the postal code in DS_{R1} was given in the format "ZIP+4" (containing the standard five-digit postal code with four additional digits for postal delivery, e.g., "10019-2132") and in DS_{R2} in the format "ZIP" (containing the standard five-digit postal code, e.g., "10019"). Hence, "Postal Code" was restricted to only the standard five-digit postal code "ZIP" (e.g., "10019") to achieve standardized key attribute values. In the data preparation subtask, pairs of restaurants which were more than 1,000 meters apart from each other based on the key attribute "Geolocation" were removed, due to these restaurant pairs being obvious non-duplicates. This led to a total of 11,492 item pairs, constituting the data for the next task "Item Pair Classification". Task 1.2 was initialized by selecting adequate SMFs for all key attributes, following the argumentations given in Section 3. For example, the SMFs "Jaro-Winkler" and "Levenshtein" were proved as useful for the key attributes "Name" and "Address" and the SMF "Haversine" was beneficial for "Geolocation" (Kamath et al. 2013). These key attributes were selected as no natural unique IDs for the restaurants were available across DS_{R1} and DS_{R2}. The duplicate detection then yielded at first 6,226 pairs classified as duplicates and 5,266 item pairs classified as non-duplicates. In Task 1.3, multi-linkages of items were resolved. For example, the restaurant "Sushi You" in DS_{R1} was contained in two item pairs classified as duplicates (with the restaurant "Sushi You" from DS_{R2} in the first pair and with the restaurant "Sushi Ko" from DS_{R2} in the second pair). Here, the prediction score of the first pair was higher than the score of the second one and therefore, only the first pair was retained. After resolving such multi-linkages, the number of duplicate item pairs decreased to 5,919. With regard to Task 1.4, 500 item pairs (250 items presumed to be duplicates and 250 items presumed to be non-duplicates) were selected to validate our duplicate detection step. Thereby, the item pairs were examined by a web-based search which involved 1) visiting the homepages of the restaurants, 2) searching the restaurants via *Google Maps* and 3) using *Google Street View* to check the identity of restaurants. This method was necessary to reliably determine actual duplicates and non-duplicates as some non-duplicate item pairs were hard to identify. For example, the restaurants "Murray's Cheese Shop" in DS_{R1} located at "254 Bleecker St" in "West Village" and "Murray's Cheese Bar" in DS_{R2} at "264 Bleecker St" in "West Village", which seem to be very similar at first sight, turned out to be non-duplicates after the examination. The validation of the duplicate detection yielded a precision of 95.9% (i.e., 235 of 245 classified duplicates were real duplicates; 240 of 255 classified non-duplicates were real non-duplicates), a recall of 94.0% (i.e., 235 of 250 real duplicates were classified as duplicates; 240 of 250 real non-duplicates were classified as non-duplicates) and a F1-measure of 94.9%, demonstrating a very high quality. Summing up, the first step of the procedure yielded 5,919 duplicate restaurant item pairs of high quality constituting the basis for Step 2 of the procedure.

Table 5. Key Attributes of both Restaurant Portals

Key attributes	*Data type*	*Example key attribute values from both portals for a duplicate*
Name	String	“9 Ten Restaurant” (in DS_{R1}), “9 10 Restaurant” (in DS_{R2})
Postal Code	Number	“10019-2132” (in DS_{R1}), “10019” (in DS_{R2})
Geolocation	Geographic coordinates (latitude and longitude)	“N 40.76591° / W -73.97979°” (in DS_{R1}), “N 40.7659964050293° / W -73.9797178100586°” (in DS_{R2})
Address	String	“910 Seventh Avenue” (in DS_{R1})’ “910 7th Av” (in DS_{R2})
Phone	Number	“+1 917-639-3366” (in DS_{R1}), “(917) 639 3666” (in DS_{R2})
District	String	“Midtown” (in DS_{R1}), “Midtown West” (in DS_{R2})

Next, we briefly outline the first step of the procedure for the movie domain. As described before, the duplicate detection step for the movie domain is in general less challenging than for the restaurant domain due to copyright standards. However, titles of movie duplicates do not always exactly match, since different movie portals have heterogeneous data policies (e.g., the movie titles “Mission: Impossible – Ghost Protocol” and “Mission: Impossible – Ghost Protocol (2011)” represent the same item). Hence, standardization of the key attribute “Title” in both data sets DS_{M1} and DS_{M2} is necessary (e.g., removing the year of the movie’s release). Thereafter, many duplicates can be detected directly by matching the standardized “Title” of movies in a large part of the cases (cf. Section 4.1). Similar as for restaurants, pairs of movies which were obvious non-duplicates (based on similarities of the key attribute “Title”) were removed during blocking leading to 10,160 item pairs as result of Task 1.1. Since DS_{M1} also contained items going beyond regular cinematographic movies (e.g., other film material such as “The Theory of Evolution: A History of Controversy”), item pairs could only be identified for the mentioned 10,160 items in DS_{M1}. In Task 1.2, SMFs such as “Jaro-Winkler” and “Levenshtein” were used for the key attribute “Title” for conducting item pair classification similarly as for restaurants. With no multi-linkages present in the result of Task 1.2 (i.e., Task 1.3 could be skipped), 9,438 movie item pairs were detected as duplicates. Similarly, as for restaurants, 500 item pairs were prepared to validate duplicate detection by a manual web-based search. The validation of the duplicate detection for movies in Task 1.4 yielded a precision of 95.1%, a recall of 96.7% and a F1-measure of 95.9%, demonstrating a very high quality for detecting duplicates. Summing up, the first step of the procedure yielded 9,438 duplicate movie item pairs of high quality constituting the basis for Step 2 of the procedure.

4.2.2 Evaluation of Step 2 – Data Integration

In this section, we outline the evaluation of the data integration step. The goal of this section is to assess how the completeness of the item content data could be increased through data integration. Therefore, we first establish how we conducted and validated the tasks of Step 2 of the procedure and then present the results of the evaluation. Since the number of further attributes in DS_{M2} (compared to DS_{M1}) and the numbers of missing attribute values in DS_{M1} and DS_{M2} are very high (cf. Table 4), Step 2 is of particular relevance for the real-world scenario regarding the movie domain. Nevertheless, Step 2 is also crucial for the real-world scenario regarding restaurants, as in this step the actual data set extension is performed.

Following Task 2.1, as heterogeneous data policies among portals in the restaurant domain had led to different names of the same attribute and different levels of granularity used across DS_{R1} and DS_{R2}, all attributes of DS_{R2} were compared to the attributes of DS_{R1} to identify matching and additional attributes. Thereby, 57 attributes of DS_{R2} such as "Japanese", "Pizza" or "Vegan" were identified as matching attributes and 194 attributes of DS_{R2} such as "Attire", "Karaoke" or "Take Out" were identified as additional attributes in a manual check requiring approximately one hour of work, exhibiting almost perfect inter-coder reliability. According to Task 2.2, these additional attributes are to be analyzed regarding an extension of DS_{R1}. Here, for a first evaluation regarding the effects on recommendation quality, we used all additional attributes for the extension of DS_{R1}. Thus, the extended data set contained all attributes of DS_{R1} and all additional attributes of DS_{R2}. Thereafter, the item content data of DS_{R1} was extended and attribute values of duplicates were inserted. Further, we validated Task 2.3, which means, the remaining missing attribute values were imputed in a first step. To this end, we evaluated the use of the Hot Deck Imputation method (cf. Table 2), allowing the replacement of all missing values and yielding an item content data set without missing values. In total, the evaluation shows that the completeness of the item content data of DS_{R1} can be increased by integrating 194 additional attributes from DS_{R2} and by imputation of 3,253 values in DS_{R1} and 190,789 values in DS_{R2}. This emphasizes the superior data quality of the resulting extended data set compared to the basis data set DS_{R1} regarding the dimension completeness.

In the case of the movie data sets, all 103 attributes of DS_{M2} such as "Genres", "Cast" or "Language" were identified as additional attributes in Task 2.1. In Task 2.2, for a first evaluation regarding the effects on recommendation quality, we used all additional attributes of DS_{M2} for the extension of DS_{M1} similar to the case of restaurants. Thus, the attributes and values were inserted for the identified duplicates and thus, the extended data set contained all attributes of DS_{M1} and all attributes of DS_{M2}. In Task 2.3, the remaining missing attribute values were imputed by means of the Hot Deck Imputation method (cf. Table 2) yielding an item content data set without missing values. In total, the evaluation shows that the completeness of the item content data of DS_{M1} can be increased by integrating 103 additional attributes from DS_{M2} and by imputation of 247,341 values in DS_{M1} and 1,082,387 values in DS_{M2}. Therefore, the resulting extended data set shows strongly increased data quality compared to the basis data set DS_{M1} regarding the dimension completeness.

4.2.3 Evaluation of Subsequent Step – Recommendation Determination

Finally, we discuss the evaluation of the recommendation determination based on the extended data sets with increased completeness regarding both domains. After the data set extension in the first two steps of the procedure, the recommendations based on the extended data sets could be computed. As indicated in Section 3, we validated whether the hybrid recommender system approach CBMF (Forbes and Zhu 2011; Nguyen and Zhu 2013) can be utilized. We followed Nguyen and Zhu (2013) in regard to the default configuration for CBMF, with the only exception being the regularization penalty factor λ, which has to be adjusted depending on the data set at hand (Koren et al. 2009). To this end, we compared the results of cross-validation tests of different values for λ as described by Koren et al. (2009). In these tests, the value $\lambda = 10^{-5}$ yielded the best results in terms of RMSE. After the execution of CBMF, the recommendations were evaluated by the following standard technique (cf., e.g., Shani and Gunawardana 2011). The ratings of DS_{R1} and DS_{M1} were randomly split into a training set (67% of ratings) to learn the parameters of the CBMF model (p_u and a_f, cf. Section 3) and a test set (33% of ratings) for assessing recommendation quality. We quantified recommendation quality by the RMSE between the real ratings and the predicted ratings of the CBMF in the test set. To assess the recommendation quality based on the extended data sets compared to just data sets DS_{R1} or DS_{M1}, respectively, the training of the CBMF parameters and the assessment of recommendation quality were validated on either the item content data of the extended data set or just on the item content data of DS_{R1} or DS_{M1}. Here, in both cases (extended data set compared to the basis data set) the train-test-split remained the same such that a meaningful comparison of both cases was possible for both domains. The recommendation determination could be applied in each case without restrictions and yielded recommendations for each user. In particular, our procedure was able to successfully navigate numerous challenges in this context (cf. Table 6), which are common when trying to extend an item content data set with respect to the data quality dimension completeness. This successful validation of the determined recommendations concludes the evaluation of the proposed procedure in both real-world scenarios.

Table 6. Challenges in the Context of Recommender Systems

Topics	*Challenges in the context of recommender systems*	*References to procedure step / task*
Data / Content	• Decentral data capturing by many different users results in data quality problems requiring standardization • Heterogeneous data policies among portals lead to different characteristics of the data across data sets, also requiring standardization • Item content data is a central decisive factor for e-commerce business models and respective recommender systems	**1.1 Data Standardization and Preparation**
Key Attributes and Item Pair Classification	• Labeled training data is missing in the context of recommender systems for a supervised item pair classification • No natural unique IDs are available for items (e.g. restaurants) • Values of key attributes are entered in a decentral way and depend on the users' own interpretation leading to highly diverse data values • Items with the same name referring to the same organization (e.g., "McDonald's") and items with similar names referring to different organizations (e.g., "Sushi You" vs. "Sushi Ko") in the restaurant domain are potentially in close proximity in urban areas; however, they have to be distinguished as separate items	**1.2 Item Pair Classification**
Matching Attributes	• Heterogeneous data policies among portals lead to different names of the same attribute (e.g., "Bar" vs. "Pub") • Portals potentially use different levels of granularity when describing the attributes (e.g., "Asian Cuisine" vs. "Japanese Cuisine")	**2.1 Identification of Attributes**
Additional Attributes	• Attributes and their values (e.g., eight times more attributes after data set extension in the movie domain) directly affect the quality of the recommender system and the resulting recommendations	**2.2 Extension of Item Data**
Missing Values	• Many recommender system techniques cannot handle missing values (e.g., 75% missing attribute values had to be imputed in the movie domain)	**2.3 Handling Missing Values**

4.3 Effects on Recommendation Quality

In addition to the evaluation of the procedure itself in Section 4.2, we observed and examined effects of our procedure on recommendation quality in both e-commerce real-world scenarios. These effects can serve as a starting point for further investigations of the impact of completeness on the recommendation quality based on our procedure (cf. Section 2.2). In particular, besides evaluating the general impact of increased completeness on recommendation quality when applying the proposed procedure (Effect 1), we also investigated effects in detail on the results of the procedure from the three major dimensions related to (content-based and hybrid) recommendations in e-commerce (Heinrich et al. 2019): Items (Effect 2), content in form of attributes (Effect 3) and attribute values (Effect 4), and users (Effect 5). An overview of the results regarding these effects for both the restaurant and the movie domain is given in Table 7.

Effect 1. Extending the basis data set (DS_{R1} and DS_{M1}, respectively) by applying the proposed procedure improved recommendation quality considerably.

Scenario regarding restaurants: Indeed, the more complete view on restaurants provided by the extended data set led to an improvement in recommendation quality of 13.2% (the RMSE achieved for the extended data set is 0.89, while the RMSE for just DS_{R1} is 1.02). The more complete view and its effect can be illustrated by an example considering the user "Michelle", who had submitted 43 ratings overall. This user had, in reality, rated the restaurant "ShunLee" with a score of 4 stars. The rating of this restaurant as estimated by CBMF based on just DS_{R1} was 1 star, which means that there was a huge discrepancy between the real and the estimated rating. In the extended data set, the item vector of "ShunLee" was extended by all additional attributes and attribute values of its duplicate in DS_{R2} as described above. This extension led to a large improvement, as CBMF based on the extended data set determined a rating of 3 stars, which is much closer to the real rating of the user. Overall, the recommendations for "Michelle" based on the extended data set and based on just DS_{R1} resulted in RMSEs of 0.56 and 3.78, respectively. This example further illustrates the (considerable) improvement of recommendation quality.

Scenario regarding movies: Compared to the restaurant domain, the overall effect of the procedure in the movie domain is even stronger, as the extension of DS_{M1} led to an improvement in recommendation quality of 24.6%. However, the baseline RMSE of 3.15 based on just DS_{M1} is inferior for the movie domain compared to the restaurant domain with a baseline RMSE of 1.02, which means, improving a higher baseline RMSE is comparatively easier. This puts the high improvement in recommendation quality in perspective. Besides this, individual analyses of users regarding improvements in recommendation quality can be performed analogously to the description above for restaurants.

Effect 2. A sophisticated duplicate detection as proposed by our procedure yielded a high improvement in recommendation quality.

Scenario regarding restaurants: In order to investigate the importance of duplicate detection (cf. Section 3.1) on the resulting recommendation quality, we further instantiated and evaluated the procedure with an alternative rule-based duplicate detection algorithm (cf. Christen 2012). To evaluate this alternative algorithm, we performed Task 1.1, Task 1.3 and Task 1.4 in the same way, but for Task 1.2, we chose the following decision-rule aiming for a simple but transparent classification of item pairs (A, B):

If $jaro\_winkler\_similarity_{name}(A, B) > T_1$ **and** $haversine\_similarity_{geolocation}\ (A, B) > T_2$ **then** item B is classified as a duplicate of item A **else** item B is not classified as a duplicate of item A.

We evaluated different threshold configurations for T_1 and T_2 resulting in the best validation results for the thresholds $T_1 = 0.9$ and $T_2 = 0.909$ (corresponding to a distance of 100 meters), which were used for the rule-based item pair classification. As the rule-based duplicate detection was rather restrictive with judging pairs of items to be a duplicate, the fewer pairs of items identified as duplicates by the rule-based duplicate detection were almost all correctly classified, resulting in a high precision of 96.8% (compared to 95.9% precision of the sophisticated duplicate detection). However, the rule-based duplicate detection mainly just identified the rather obvious duplicates, leading to this high precision but a quite low recall. More precisely, it was only able to identify 72.8% of duplicates as indicated by the recall (compared to 94.0% recall of the sophisticated duplicate detection). Thus, the rule-based duplicate detection also exhibited an overall lower F1-measure of 83.1% compared to 94.9% for the sophisticated duplicate detection, demonstrating the higher quality of the sophisticated duplicate detection. The assessed improvement in recommendation quality when conducting the remainder of the procedure using this duplicate detection with lower quality was only 9.8% (compared to 13.2% improvement for the sophisticated duplicate detection with higher quality assessed on the same test set of ratings as in Effect 1). These results show that the sophisticated duplicate detection algorithm proposed by our procedure led to a significantly higher improvement in recommendation quality.

Scenario regarding movies: Similarly, as for restaurants, we instantiated and evaluated a rule-based duplicate detection algorithm in the movie domain yielding 85.3% for F1-measure (compared to 95.9% for the sophisticated duplicate detection). Nevertheless, even the procedure with the rule-based duplicate detection yields an improvement in recommendation quality by 23.9%, which is smaller than the improvement based on the sophisticated duplicate detection, which is 24.6%.

Effect 3. The extension of the basis data set (DS_{R1} and DS_{M1}, respectively) with further attributes (of DS_{R2} and DS_{M2}, respectively) generally supported the increase in recommendation quality, with the extent of improvement depending on the attribute set used for the extension.

Scenario regarding restaurants: To analyze and separate the effect of additional attributes for extension in Task 2.2, we split all additional attributes from DS_{R2} into two equally sized groups based on the absolute number of available values per attribute. First, we extended DS_{R1} with the set of additional attributes from DS_{R2} with a low number of available attribute values (Set 1), leading to an improvement in recommendation quality of just 0.1%. Second, the extension of DS_{R1} with the set of additional attributes with a high number of available attribute values (Set 2) achieved an improvement of 12.6%. In comparison, the extension of DS_{R1} with all additional attributes of DS_{R2} (Set 3) led to an improvement of 12.7%.[4] These results show that while the extension with additional attributes generally contributed to an improvement of recommendation quality, the extent of improvement depended on the number of available attribute values of the additional attributes. Thus, these results indicate that the increase in recommendation quality could mainly be traced back to attributes with a high number of available attribute values. Moreover, we investigated the extension of DS_{R1} with *all attributes* of DS_{R2} (Set 4; i.e., additional attributes *and* matching attributes from DS_{R2}) in order to further analyze this effect. This means, we omitted the identification of matching attributes (cf. Task 2.1) and extended DS_{R1} with all attributes of DS_{R2} (i.e., additional and matching attributes). Although another 57 (matching) attributes were added compared to the extension with only additional attributes, the improvement of recommendation quality decreased slightly by 0.1% to 12.6%. This finding based on our chosen real-world scenario supports that more data (i.e., more attributes and attribute values) does not always lead to better results of decision support systems and, in particular, recommender systems (cf. Section 2.2). Therefore, the additional and more complete data provided by the matching attributes did not yield any added value, which is in line with works such as Bleiholder and Naumann (2008). In our application context, the matching of attributes led to just a slight improvement of the recommendation quality (0.1%), however, there may be application areas in which the matching of attributes contributes even more to an improvement of the recommendation quality and therefore Task 2.1 of the procedure is essential.

Since both adding attributes and identifying matching attributes may cause effort, it would be interesting to further investigate how to choose an adequate balance between these efforts and the resulting benefits of improved recommendation quality. For instance, when the efforts for adding attributes are low, all additional attributes can be selected for extension. Otherwise, a limitation to a smaller set of (additional) attributes (e.g., attributes with a high number of available attribute values) may be reasonable to reduce high efforts while simultaneously accomplishing a similarly high improvement of recommendation quality.

Scenario regarding movies: As for restaurants, we analyzed four sets of additional attributes (Set 1-4) from DS_{M2} regarding an improvement in recommendation quality. Since the scenario regarding movies did not yield matching attributes, all attributes of DS_{M2} constituted additional attributes and thus, the attribute sets Set 3 and Set 4 were identical. Here, the results regarding this effect for movies further underline the findings identified for restaurants as the improvement of 1.7% in recommendation quality for Set 1 was small compared to high improvements of 17.4% for the Sets 2-4. That is, the increase in recommendation quality could mainly be traced back to attributes with a high number of available attribute values.

Effect 4. More attribute values (i.e., less missing values) resulted in increased recommendation quality.

[4] The difference between the improvement of 12.7% in Effect 3 and the improvement of 13.2% in Effect 1 can be attributed to the fact that imputation of missing values is omitted in Effect 3.

Scenario regarding restaurants: In addition to the analysis of the set of attributes, we also investigated effects of item content data with respect to (missing) attribute values. We fixed the set of attributes in the extended data set and focused on the imputation of missing attribute values (cf. Task 2.3) in order to separate Effect 4. We examined three settings with a varying number of (missing) attribute values. In the first setting, we imputed all missing values according to Task 2.3, resulting in no missing values in the item content data set used. The second setting used the extended data set without imputing missing values. In our real-world scenario regarding restaurants, however, only four percent of attribute values were missing, which could limit the extent of potential effects of missing attribute values. Therefore, we considered a third setting, in which we randomly removed an additional ten percent of attribute values from the extended item content data set to examine the effect of missing attribute values more generally in the restaurant domain. This led to a total of fourteen percent of missing attribute values in this third setting. We evaluated all three settings regarding resulting improvements in recommendation quality (i.e., RMSE based on the extended data set vs. RMSE based on just DS_{R1}). The results showed an improvement in recommendation quality of 13.2% for the first setting, 12.7% for the second setting and 6.5% for the third setting.

Scenario regarding movies: In contrast to the scenario regarding restaurants, the movie data sets showed high numbers of missing attribute values (cf. Table 4) making this scenario especially promising for analyzing the effect of imputing missing values (in Step 2 of the procedure) on recommendation quality in a real-world e-commerce application scenario. Similarly, as for restaurants, we examined the three settings with a varying number of missing attribute values. The results showed an improvement in recommendation quality of 24.6% for the first setting (i.e., the extended data set with imputed missing values), 17.4% for the second setting (i.e., the extended data set without imputed missing values) and 13.7% for the third setting (i.e., the extended data set without imputed missing values and 10% further removed attribute values).

These results emphasize that recommendation quality benefits significantly from having more attribute values and, in particular, from imputing missing values, which constitutes a main task in the proposed procedure (cf. Task 2.3).

Effect 5. Users with a high number of submitted ratings benefitted more from the data set extension than users with a low number of submitted ratings.

Scenario regarding restaurants: For the analysis of this effect, we examined the relation between the number of ratings submitted by users and the increase in recommendation quality. To do so, we grouped all users into three equally sized groups based on their number of submitted ratings in the training set and examined the three groups individually regarding their improvement in recommendation quality. The first group containing users with the highest number of ratings (averaging about 29 ratings submitted per user) achieved a RMSE improvement of 17.1%. The second group, whose users had on average submitted about 15 ratings, recorded a RMSE improvement of 16.3%. Finally, the third group of users, with an average of about 10 ratings submitted per user, achieved the lowest improvement of recommendation quality, accomplishing a RMSE improvement of 9.9%.

Scenario regarding movies: Analogous as for restaurants, we grouped the users in the movie scenario into three equally sized groups. The first group, whose users had on average submitted about 4 ratings, achieved the highest RMSE improvement of 45.4%. The second group, whose users had submitted about 2 ratings on average, still recorded a high RMSE improvement of 42.7%. Finally, the third group of users, with an average of about 1 rating submitted per user, achieved the lowest improvement of recommendation quality, accomplishing a RMSE improvement of only 6.0%. Although the improvement for the third user group is small, it is still noteworthy as these users with just 1 submitted rating have only rating data in either the training set or the test set. In particular, this means that even users without ratings at all (i.e., without ratings in the training set) benefit from extending the item content data set, which is of high relevance for e-commerce applications, as the case of new users occurs very frequently.

Overall, these results indicate that the improvement of recommendation quality depended on the number of ratings submitted by users, and that users with a higher number of submitted ratings benefitted more. In a detailed analysis, we concluded that this effect can be attributed to the fact that users with a higher number of submitted ratings mainly rated items for whom more item content was added. Thus, the extended data set enabled the recommender system to infer these users' ratings even more accurately.

Table 7. Overview of Improvements in Recommendation Quality for each Effect

Effects	*Evaluation configurations*		*Relative improvements in recommendation quality (RMSE) by procedure application*	
			Restaurants	*Movies*
1	Standard procedure configuration (as outlined in section 4.2)		13.2%	24.6%
2	Procedure with simplified rule-based duplicate detection		9.8%	23.9%
3	Procedure without imputation and …	additional attributes with low number of available attribute values (Set 1)	0.1%	1.7%
		additional attributes with high number of available attribute values (Set 2)	12.6%	17.4%
		all additional attributes (Set 3)	12.7%	17.4%
		all attributes of DS2 (Set 4)	12.6%	17.4%
4	Standard procedure configuration (as outlined in section 4.2) (Setting 1)		13.2%	24.6%
	Procedure without imputation (Setting 2)		12.7%	17.4%
	Procedure without imputation and further removed attribute values (Setting 3)		6.5%	13.7%
5	Procedure for users with high rating numbers (Group 1)		17.1%	45.4%
	Procedure for users with moderate rating numbers (Group 2)		16.3%	42.7%
	Procedure for users with low rating numbers (Group 3)		9.9%	6.0%

5 Conclusion, Limitations and Directions for Future Work

Researchers have highlighted the relationship between data quality and decision support systems, and in particular recommender systems, in the field of IS. Based on a theoretical model, we present a procedure for the systematic extension of a data set DS1 with additional item content (attributes and attribute values) from another data set DS2 in the same domain. Thereby, the procedure aims to address data quality, especially by increasing the completeness of data sets and, in consequence, to improve recommendation quality of recommender systems. In a first step, an approach to detect duplicate items across data sets DS1 and DS2 is proposed. In a second step, we outline how item content data in DS1 can be extended by integrating the item content data of a data set DS2 as well as by imputing missing values. Based on these two steps, the resulting extended data set can be used by an arbitrary content-based or hybrid recommender system to determine recommendations in a subsequent step. We evaluate the procedure by using two real-world data sets regarding restaurants and movies, which constitute commonly analyzed domains in IS research on e-commerce, and discuss effects on recommendation quality. Here, the results show that the presented procedure is indeed capable of improving recommendations considerably by means of item content data extension, which is in line with existing research (cf. Heinrich et al. 2019). Furthermore, we investigate different effects on the results of the procedure from the three dimensions items, content and users, revealing that the procedure was valuable in each investigated case and indicating under which circumstances a substantial improvement in recommendation quality was achieved. Complementary to existing research proposing general relationships between data quality and decision support systems, this work provides and evaluates a tangible procedure which enables to increase data completeness with the aim of improving recommendation quality. Moreover, this procedure serves an evaluated template for future procedures to support the investigation of further data quality dimensions (e.g., consistency) for decision support systems in various e-commerce applications.

The rapid proliferation of e-commerce has cemented the tremendous relevance of recommender systems. These systems are powerful data-driven decision support systems incorporated in many e-commerce platforms guiding users to their individually best item choice among a plethora of alternatives. Thereby, recommender systems address the problem of information overload, which constitutes a major subject of IS research in the field of e-commerce. While the steady increasing volume of information (e.g., about attributes of items) would further aggravate the problem of information overload for users, recommender systems actually can somehow invert this effect. In contrast to the limited cognitive capabilities of users, for recommender systems as automated data-driven systems, more information (e.g., item content data; i.e., attributes and attribute values) is highly useful to individually support the user's decision-making and thus to further reduce the problem of information overload. To do so, increasing the completeness of the data (i.e., item content data) a recommender system is based on seems to constitute a promising way, which is studied in this paper by proposing a procedure for data set extension. Especially in established e-commerce domains (e.g., restaurants and movies), a higher completeness can significantly improve the recommendation quality for users (e.g., the selection of restaurants and movies), which in the long run strengthens the relationship between providers and users.

Here, our evaluation encourages IS providers in e-commerce (e.g., online portals) to improve data quality by providing a straightforward way to increase completeness without the need of manual tasks such as visiting items' websites or social media pages. Our procedure shows that achieving high data quality is indeed beneficial for companies, as the resulting improved recommendations support the various goals and purposes of recommender systems such as promoting cross- and up-selling or increasing customer loyalty (Jannach and Adomavicius 2016). Moreover, our results open up a way for portals with limited resources to balance the efforts and benefits associated to the procedure. For instance, as recommending items based on massively extended item content data can prove to be time-consuming, portals may prefer to focus on a subset of users or additional attributes based on the evidence found in Section 4.

However, our work also has some limitations, which could be starting points for future research. First, while we focused on completeness as a highly relevant data quality dimension, extensions of data sets in the context of recommender systems could also take into account other data quality dimensions such as accuracy or currency. Second, we considered the extension of item content data based on additional structured data in this paper. Here, it would be promising to leverage modern information extraction approaches, such as aspect extraction with language models (e.g., BERT; cf. Xu et al. 2019). Thereby, data sets already used by IS providers could be extended by extracted features from unstructured textual data sources (e.g., online customer reviews). Moreover, another interesting perspective might be to incorporate the extension of user data into the procedure, which could in some cases be realized by, for instance, user linkage based on online social network accounts. Finally, the approach could also be applied to further data sets, possibly from other domains outside the field of e-commerce, in order to validate and substantiate the resulting effects on recommendation quality.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

Abel, F., Herder, E., Houben, G.-J., Henze, N., & Krause, D. (2013). Cross-system user modeling and personalization on the Social Web. *User Modeling and User-Adapted Interaction, 23*, 169–209. https://doi.org/10.1007/s11257-012-9131-2.

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*, 734–749. https://doi.org/10.1109/TKDE.2005.99.

Aggarwal, C. C. (2016). *Recommender Systems*. Cham: Springer International Publishing.

Amatriain, X., Pujol, J. M., Tintarev, N., & Oliver, N. (2009). Rate it again. In L. Bergman, A. Tuzhilin, R. Burke, A. Felfernig, & L. Schmidt-Thieme (Eds.), *The third ACM conference on Recommender systems, New York, New York, USA* (pp. 173–180). New York, NY: ACM. https://doi.org/10.1145/1639714.1639744.

Basaran, D., Ntoutsi, E., & Zimek, A. (2017). Redundancies in Data and their Effect on the Evaluation of Recommendation Systems: A Case Study on the Amazon Reviews Datasets. In N. Chawla & W. Wang (Eds.), *The 2017 SIAM International Conference on Data Mining, Houston, Texas, USA* (pp. 390–398). Philadelphia, PA: Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611974973.44.

Batini, C., & Scannapieco, M. (2016). *Data and Information Quality*. Cham: Springer International Publishing.

Berkovsky, S., Kuflik, T., & Ricci, F. (2012). The impact of data obfuscation on the accuracy of collaborative filtering. *Expert Systems with Applications, 39*, 5033–5042. https://doi.org/10.1016/j.eswa.2011.11.037.

Bharati, P., & Chaudhury, A. (2004). An empirical investigation of decision-making satisfaction in web-based decision support systems. *Decision Support Systems, 37*, 187–197. https://doi.org/10.1016/S0167-9236(03)00006-X.

Blake, R., & Mangiameli, P. (2011). The Effects and Interactions of Data Quality and Problem Complexity on Classification. *Journal of Data and Information Quality, 2*, 1–28. https://doi.org/10.1145/1891879.1891881.

Bleiholder, J., & Naumann, F. (2008). Data fusion. *ACM Computing Surveys, 41*, 1–41. https://doi.org/10.1145/1456650.1456651.

Bostandjiev, S., O'Donovan, J., & Höllerer, T. (2012). TasteWeights: a visual interactive hybrid recommender system. In P. Cunningham, N. Hurley, I. Guy, & S. S. Anand (Eds.), *The sixth ACM conference on Recommender systems, Dublin, Ireland* (pp. 35–42). New York, NY: ACM. https://doi.org/10.1145/2365952.2365964.

Bouadjenek, M. R., Pacitti, E., Servajean, M., Masseglia, F., & Abbadi, A. E. (2018). A Distributed Collaborative Filtering Algorithm Using Multiple Data Sources. *arXiv preprint arXiv:1807.05853*.

Bunnell, L., Osei-Bryson, K.-M., & Yoon, V. Y. (2019). RecSys Issues Ontology: A Knowledge Classification of Issues for Recommender Systems Researchers. *Information Systems Frontiers, 97*, 667. https://doi.org/10.1007/s10796-019-09935-9.

Burke, R., & Ramezani, M. (2011). Matching Recommendation Technologies and Domains. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 367–386). Boston, MA: Springer US.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering, 40*(1), 16–28.

Chang, J.-H., Tsai, C.-E., & Chiang, J.-H. (2018). Using Heterogeneous Social Media as Auxiliary Information to Improve Hotel Recommendation Performance. *IEEE Access, 6*, 42647–42660. https://doi.org/10.1109/ACCESS.2018.2855690.

Chang, W.-L., & Jung, C.-F. (2017). A hybrid approach for personalized service staff recommendation. *Information Systems Frontiers, 19*, 149–163. https://doi.org/10.1007/s10796-015-9597-7.

Christen, P. (2012). *Data matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Edmunds, A., & Morris, A. (2000). The problem of information overload in business organisations: a review of the literature. *International Journal of Information Management, 20*, 17–28. https://doi.org/10.1016/S0268-4012(99)00051-1.

Enders, C. K. (2010). *Applied missing data analysis* (Methodology in the social sciences). New York: Guilford Press.

Feldman, M., Even, A., & Parmet, Y. (2018). A methodology for quantifying the effect of missing data on decision quality in classification problems. *Communications in Statistics–Theory and Methods, 47*(11), 2643–2663.

Forbes, P., & Zhu, M. (2011). Content-boosted matrix factorization for recommender systems. In B. Mobasher, R. Burke, D. Jannach, & G. Adomavicius (Eds.), *The fifth ACM conference on Recommender systems, Chicago, Illinois, USA* (pp. 261–264). New York, NY: ACM. https://doi.org/10.1145/2043932.2043979.

Ge, M. (2009). *Information quality assessment and effects on inventory decision-making*. Doctoral dissertation. Dublin City University, Dublin.

GitHub. (2020). Procedure Completeness: Extending Item Content Data. https://github.com/ProcedureCompleteness/ExtendingItemContentDataSets. Accessed 14 September 2020.

Hasan, M. R., Jha, A. K., & Liu, Y. (2018). Excessive use of online video streaming services: Impact of recommender system use, psychological factors, and motives. *Computers in Human Behavior, 80*, 220–228. https://doi.org/10.1016/j.chb.2017.11.020.

Heinrich, B., Hopf, M., Lohninger, D., Schiller, A., & Szubartowicz, M. (2019). Data quality in recommender systems: the impact of completeness of item content data on prediction accuracy of recommender systems. *Electronic Markets, 23*, 169. https://doi.org/10.1007/s12525-019-00366-7.

Heinrich, B., Klier, M., Schiller, A., & Wagner, G. (2018). Assessing data quality – A probability-based metric for semantic consistency. *Decision Support Systems, 110*, 95–106. https://doi.org/10.1016/j.dss.2018.03.011.

Jannach, D., & Adomavicius, G. (2016). Recommendations with a Purpose. In S. Sen & W. Geyer (Eds.), *The 10th ACM Conference on Recommender Systems, Boston, Massachusetts, USA* (pp. 7–10). New York, NY, USA: Association for Computing Machinery.

Jannach, D., Zanker, M., Ge, M., & Gröning, M. (2012). Recommender Systems in Computer Science and Information Systems – A Landscape of Research. *E-Commerce and Web Technologies, 123*, 76–87. https://doi.org/10.1007/978-3-642-32273-0_7.

Jurek, A., Hong, J., Chi, Y., & Liu, W. (2017). A novel ensemble learning approach to unsupervised record linkage. *Information Systems, 71*, 40–54. https://doi.org/10.1016/j.is.2017.06.006.

Kamath, K. Y., Caverlee, J., Lee, K., & Cheng, Z. (2013). Spatio-temporal dynamics of online memes: a study of geo-tagged tweets. In D. Schwabe (Ed.), *The 22nd International Conference on the World Wide Web, Rio de Janeiro, Brazil* (pp. 667–678). New York, NY: ACM. https://doi.org/10.1145/2488388.2488447.

Kamis, A., Stern, T., & Ladik, D. M. (2010). A flow-based model of web site intentions when users customize products in business-to-consumer electronic commerce. *Information Systems Frontiers, 12*, 157–168. https://doi.org/10.1007/s10796-008-9135-y.

Karimova, F. (2016). A Survey of e-Commerce Recommender Systems. *European Scientific Journal, ESJ, 12*, 75. https://doi.org/10.19044/esj.2016.v12n34p75.

Karumur, R. P., Nguyen, T. T., & Konstan, J. A. (2018). Personality, User Preferences and Behavior in Recommender systems. *Information Systems Frontiers, 20*, 1241–1265. https://doi.org/10.1007/s10796-017-9800-0.

Kayaalp, M., Özyer, T., & Özyer, S. T. (2009). A Collaborative and Content Based Event Recommendation System Integrated with Data Collection Scrapers and Services at a Social Networking Site. In N. Memon (Ed.), *International Conference on Advances in Social Networks Analysis and Mining, 2009, Athens, Greece* (pp. 113–118). Piscataway, NJ: IEEE. https://doi.org/10.1109/ASONAM.2009.41.

Kim, D., Park, C., Oh, J., Lee, S., & Yu, H. (2016). Convolutional Matrix Factorization for Document Context-Aware Recommendation. In S. Sen, W. Geyer, J. Freyne, & P. Castells (Eds.), *The 10th ACM Conference on Recommender Systems, Boston, Massachusetts, USA* (pp. 233–240). New York, New York, USA: ACM Press. https://doi.org/10.1145/2959100.2959165.

Koren, Y. (2009). The bellkor solution to the netflix grand prize. *Netflix prize documentation, 81*, 1–10.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer, 42*, 30–37. https://doi.org/10.1109/MC.2009.263.

Lathia, N., Amatriain, X., & Pujol, J. M. (2009). Collaborative filtering with adaptive information sources. In S. S. Anand, B. Mobasher, A. Kobsa, & D. Jannach (Eds.), *7th Workshop on Intelligent Techniques for Web Personalization & Recommender Systems, Pasadena, California, USA* (pp. 81–86, CEUR Workshop Proceedings (CEUR-WS.org), Vol. 528).

Levi, A., Mokryn, O., Diot, C., & Taft, N. (2012). Finding a needle in a haystack of reviews: cold start context-based hotel recommender system. In P. Cunningham, N. Hurley, I. Guy, & S. S. Anand (Eds.), *The sixth ACM conference on Recommender systems, Dublin, Ireland* (pp. 115–122). New York, NY: ACM. https://doi.org/10.1145/2365952.2365977.

Levy, Y., & Ellis, T. J. (2006). A systems approach to conduct an effective literature review in support of information systems research. *Informing Science, 9*, 181–212.

Li, Y., Zhang, Z., Peng, Y., Yin, H., & Xu, Q. (2018). Matching user accounts based on user generated content across social networks. *Future Generation Computer Systems, 83*, 104–115. https://doi.org/10.1016/j.future.2018.01.041.

Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: A survey. *Decision Support Systems, 74*, 12–32. https://doi.org/10.1016/j.dss.2015.03.008.

Manca, M., Boratto, L., & Carta, S. (2018). Behavioral data mining to produce novel and serendipitous friend recommendations in a social bookmarking system. *Information Systems Frontiers, 20*, 825–839. https://doi.org/10.1007/s10796-015-9600-3.

Mladenić, D., & Grobelnik, M. (2003). Feature selection on hierarchy of web documents. *Decision Support Systems, 35*, 45–87. https://doi.org/10.1016/S0167-9236(02)00097-0.

Molina, L. C., Belanche, L., & Nebot, À. (2002). Feature selection algorithms: a survey and experimental evaluation. In V. Kumar (Ed.), *IEEE International Conference on Data Mining, Maebashi City, Japan* (pp. 306–313). Los Alamitos, CA: IEEE Computer Society.

Naumann, F., Freytag, J.-C., & Leser, U. (2004). Completeness of integrated information sources. *Information Systems, 29*, 583–615. https://doi.org/10.1016/j.is.2003.12.005.

Nguyen, J., & Zhu, M. (2013). Content-boosted matrix factorization techniques for recommender systems. *Statistical Analysis and Data Mining, 6*, 286–301. https://doi.org/10.1002/sam.11184.

Nguyen, T. T., Maxwell Harper, F., Terveen, L., & Konstan, J. A. (2018). User Personality and User Satisfaction with Recommender Systems. *Information Systems Frontiers, 20*, 1173–1189. https://doi.org/10.1007/s10796-017-9782-y.

Ning, Y., Shi, Y., Hong, L., Rangwala, H., & Ramakrishnan, N. (2017). A Gradient-based Adaptive Learning Framework for Efficient Personal Recommendation. In P. Cremonesi, F. Ricci, S. Berkovsky, & A. Tuzhilin (Eds.), *The Eleventh ACM Conference on Recommender Systems, Como, Italy* (pp. 23–31). New York, New York, USA: ACM Press. https://doi.org/10.1145/3109859.3109909.

Ntoutsi, E., & Stefanidis, K. (2016). Recommendations beyond the ratings matrix. In Association for Computing Machinery (Ed.), *The Workshop on Data-Driven Innovation on the Web, Hannover, Germany* (pp. 1–5). New York, New York, USA: ACM Press. https://doi.org/10.1145/2911187.2914580.

Ozsoy, M. G., Polat, F., & Alhajj, R. (2016). Making recommendations by integrating information from multiple social networks. *Applied Intelligence, 45*, 1047–1065. https://doi.org/10.1007/s10489-016-0803-1.

Peska, L., & Vojtas, P. (2015). Using Implicit Preference Relations to Improve Content Based Recommending. *E-Commerce and Web Technologies, 239*, 3–16. https://doi.org/10.1007/978-3-319-27729-5_1.

Pessemier, T. de, Dooms, S., Deryckere, T., & Martens, L. (2010). Time dependency of data quality for collaborative filtering algorithms. In X. Amatriain, M. Torrens, P. Resnick, & M. Zanker (Eds.), *The fourth ACM conference on Recommender systems, Barcelona, Spain* (pp. 281–284). New York, NY: ACM. https://doi.org/10.1145/1864708.1864767.

Picault, J., Ribiere, M., Bonnefoy, D., & Mercer, K. (2011). How to get the Recommender out of the Lab? In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 333–365). Boston, MA: Springer US.

Pipino, L. L., Lee, Y. W., & Wang, R. Y. (2002). Data quality assessment. *Communications of the ACM, 45*, 211–218. https://doi.org/10.1145/505248.506010.

Porcel, C., & Herrera-Viedma, E. (2010). Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries. *Knowledge-Based Systems, 23*(1), 32–39.

Power, D. J., Sharda, R., & Burstein, F. (2015). *Decision support systems*. Hoboken, New Jersey, USA: John Wiley & Sons, Ltd.

Raad, E., Chbeir, R., & Dipanda, A. (2010). User Profile Matching in Social Networks. In T. Enokido (Ed.), *13th International Conference on Network-Based Information Systems (NBIS), 2010* (pp. 297–304). Piscataway, NJ: IEEE Service Center.

Ricci, F., Rokach, L., & Shapira, B. (Eds.). (2015a). *Recommender Systems Handbook*. Boston, MA: Springer US.

Ricci, F., Rokach, L., & Shapira, B. (2015b). Recommender Systems: Introduction and Challenges. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender Systems Handbook* (pp. 1–34). Boston, MA: Springer US.

Richthammer, C., & Pernul, G. (2018). Situation awareness for recommender systems. *Electronic Commerce Research, 37*, 85. https://doi.org/10.1007/s10660-018-9321-z.

Sar Shalom, O., Berkovsky, S., Ronen, R., Ziklik, E., & Amihood, A. (2015). Data Quality Matters in Recommender Systems. In H. Werthner, M. Zanker, J. Golbeck, & G. Semeraro (Eds.), *9th ACM Conference on Recommender Systems, Vienna, Austria* (pp. 257–260). New York, NY: ACM. https://doi.org/10.1145/2792838.2799670.

Scannapieco, M., & Batini, C. (2004). Completeness in the Relational Model: a Comprehensive Framework. In *International Conference on Information Quality, Cambridge, Massachusetts, USA* (pp. 333–345).

Scholz, M., Dorner, V., Schryen, G., & Benlian, A. (2017). A configuration-based recommender system for supporting e-commerce decisions. *European Journal of Operational Research, 259*(1), 205–215.

Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 257–297). Boston, MA: Springer US.

Smith, B., & Linden, G. (2017). Two decades of recommender systems at Amazon. com. *Ieee internet computing, 21*(3), 12–18.

Statista. (2019). Statistics and Market Data about E-commerce. https://www.statista.com/markets/413/e-commerce/. Accessed 3 June 2020.

Steorts, R. C., Ventura, S. L., Sadinle, M., & Fienberg, S. E. (2014). A Comparison of Blocking Methods for Record Linkage. In J. Domingo-Ferrer (Ed.), *Privacy in Statistical Databases* (Vol. 8744, pp. 253–268, Lecture Notes in Computer Science). Cham: Springer International Publishing.

Tang, H., Lee, C. B. P., & Choong, K. K. (2017). Consumer decision support systems for novice buyers – a design science approach. *Information Systems Frontiers, 19*, 881–897. https://doi.org/10.1007/s10796-016-9639-9.

Vanaja, R., & Mukherjee, S. (2019). Novel Wrapper-Based Feature Selection for Efficient Clinical Decision Support System. In L. Akoglu, E. Ferrara, M. Deivamani, R. Baeza-Yates, & P. Yogesh (Eds.), *Third International Conference on Intelligent Information Technologies, Chennai, India* (Vol. 941, pp. 113–129, Communications in Computer and Information Science, Vol. 941). Singapore: Springer Singapore. https://doi.org/10.1007/978-981-13-3582-2_9.

Vargas-Govea, B., González-Serna, G., & Ponce-Medellın, R. (2011). Effects of relevant contextual features in the performance of a restaurant recommender system. In B. Mobasher, R. Burke, D. Jannach, & G. Adomavicius (Eds.), *The fifth ACM conference on Recommender systems, Chicago, Illinois, USA* (pp. 592–596). New York, NY: ACM.

Wand, Y., & Wang, R. Y. (1996). Anchoring data quality dimensions in ontological foundations. *Communications of the ACM, 39*, 86–95. https://doi.org/10.1145/240455.240479.

Wei, C., Khoury, R., & Fong, S. (2013). Web 2.0 Recommendation service by multi-collaborative filtering trust network algorithm. *Information Systems Frontiers, 15*, 533–551. https://doi.org/10.1007/s10796-012-9377-6.

Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In *Proceedings of the Section on Survey Research Methods, Alexandria, Virginia*. Alexandria, Virginia, USA: American Statistical Association.

Woodall, P., Borek, A., Gao, J., Oberhofer, M., & Koronios, A. (2015). An Investigation of How Data Quality is Affected by Dataset Size in the Context of Big Data Analytics. In R. Wang (Ed.), *19th*

International Conference on Information Quality, Xi'an, China (pp. 24–33, Management and data quality). Red Hook, NY: Curran.

Xu, H., Liu, B., Shu, L., & Yu, P. (2019). BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. In (pp. 2324–2335). Minneapolis, Minnesota: Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1242.

Yan, X., Wang, J., & Chau, M. (2015). Customer revisit intention to restaurants: Evidence from online reviews. *Information Systems Frontiers, 17*, 645–657. https://doi.org/10.1007/s10796-013-9446-5.

Zhou, L. (2020). Product advertising recommendation in e-commerce based on deep learning and distributed expression. *Electronic Commerce Research, 20*, 321–342. https://doi.org/10.1007/s10660-020-09411-6.

8 Conclusion

While each paper is self-contained and closes with its findings and outlook, major findings of the dissertation as a whole can also be derived. Given a need for processing or analyzing big data it is worthwhile to explore the use or modification of existing approaches from the field of AI to meet demands. Developing completely new methods is not always necessary. This first major finding can be demonstrated in multiple paper:

1. To facilitate the adaptation of process models, the presented approach relies heavily on existing ways to define a planning domain and generate a plan in the field of automated planning. Each adaptation case identifies the possible consequences of a need for change and acts them out in accordance to those principles. As the evaluation shows, this guarantees the generation of correct and complete process models while significantly outperforming planning process models from scratch.

2. When evaluating the presented unified model to explain star ratings, BERT, an existing feature extraction method (Devlin et al., 2019), is used to extract features of the feature perspectives *item aspects*, *user characteristics* and *user contexts*. Similarly, the explanatory model (GOPM, cf. Binder et al., 2019) was taken from literature. To achieve considerably high explanatory power, effort was put into carefully selecting an input (i.e., the feature perspectives) for existing models. No methodological contribution was needed.

3. Factorization machines constitute the state-of-the-art in context-aware recommender systems (Lahlou et al., 2017). The aim of GroupFM is to build upon the flexibility of this approach and apply it in a group recommendation scenario. By doing so, baseline approaches can be outperformed in terms of recommendation accuracy.

Each research strand defines and develops new concepts and methods. By carefully selecting and combining these concepts and methods, new insights can be gained. This constitutes the second major finding of this dissertation. It is substantiated by the following examples:

1. The Multiple Pathway Anchoring and Adjustment Model (Cohen & Reed, 2006) from the field of marketing is used to derive the different feature perspectives contained in the unified model. Similarly, the Five Factor Model used in psychology to describe and distinguish human personalty (McCrae & John, 1992) serves as a basis for the operationalization of user characteristics.

2. In order to apply the presented procedure to systematically extend item content data, at each step appropriate approaches have to be selected and evaluated. This includes data standardization approaches, similarity measure functions as well as imputation methods. Taken together, this procedure can lead to a considerable improvement in recommendation quality.

With this in mind, future research endeavors might consider creating an overview of existing approaches within the considered research area as well as neighboring areas. In that way, promising concepts and methods might be found that were not previously explored in certain application scenarios. Furthermore, the possibility to combine different approaches or transfer the general idea of one concept into another might support researchers to gain deeper insights or solve existing problems.

Big data is not going away. Technical developments lead to new ways of generating and gathering data. Therefore, new data-driven application scenarios will arise regularly. It is in the hands of researchers and practitioners to identify such new applications of big data and develop suitable approaches, products and services.

References

Abbasi, A., Sarker, S., & Chiang, R. (2016). Big data research in information systems: Toward an inclusive research agenda. *Journal of the Association for Information Systems*, *17*(2), I–XXXII. https://doi.org/10.17705/1jais.00423

Akter, S., & Wamba, S. F. (2016). Big data analytics in e-commerce: A systematic review and agenda for future research. *Electronic Markets*, *26*(2), 173–194. https://doi.org/10.1007/s12525-016-0219-0

Badakhshan, P., Conboy, K., Grisold, T., & vom Brocke, J. (2019). Agile business process management. *Business Process Management Journal*, *26*(6), 1505–1523. https://doi.org/10.1108/BPMJ-12-2018-0347

Badue, C., Guidolini, R., Carneiro, R. V., Azevedo, P., Cardoso, V. B., Forechi, A., Jesus, L., Berriel, R., Paixão, T. M., Mutz, F., de Paula Veronese, L., Oliveira-Santos, T., & de Souza, A. F. (2021). Self-driving cars: A survey. *Expert Systems with Applications*, *165*, 113816. https://doi.org/10.1016/j.eswa.2020.113816

Batini, C., Rula, A., Scannapieco, M., & Viscusi, G. (2015). From data quality to big data quality. *Journal of Database Management*, *26*(1), 60–82. https://doi.org/10.4018/JDM.2015010103

Bazzaz Abkenar, S., Haghi Kashani, M., Mahdipour, E., & Jameii, S. M. (2021). Big data analytics meets social media: A systematic review of techniques, open issues, and future directions. *Telematics and informatics*, *57*, 101517. https://doi.org/10.1016/j.tele.2020.101517

Binder, M., Heinrich, B., Klier, M., Obermeier, A., & Schiller, A. P. R. (2019). Explaining the stars: Aspect-based sentiment analysis of online customer reviews. *Proceedings of the 27th European Conference on Information Systems (ECIS)*.

Chardonnens, T., Cudre-Mauroux, P., Grund, M., & Perroud, B. (2013). Big data analytics on high velocity streams: A case study. *2013 IEEE International Conference on Big Data*, 784–787. https://doi.org/10.1109/BigData.2013.6691653

Chau & Xu. (2012). Business intelligence in blogs: Understanding consumer interactions and communities. *MIS Quarterly*, *36*(4), 1189. https://doi.org/10.2307/41703504

Chowdhary, K. R. (2020). *Fundamentals of artificial intelligence*. Springer India. https://doi.org/10.1007/978-81-322-3972-7

Cohen, J. B., & Reed, A. (2006). A multiple pathway anchoring and adjustment (mpaa) model of attitude generation and recruitment. *Journal of Consumer Research*, *33*(1), 1–15.

de Mauro, A., Greco, M., & Grimaldi, M. (2015). What is big data? a consensual definition and a review of key research topics. *Proceedings of the 4th International Conference on Integrated Information*, 97–104. https://doi.org/10.1063/1.4907823

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 conference of the north american chapter of the human language technologies (naacl)* (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Diebold, F. X. (2021). What's the big idea? "big data" and its origins. *Significance*, *18*(1), 36–37. https://doi.org/10.1111/1740-9713.01490

Gambini, M., La Rosa, M., Migliorini, S., & ter Hofstede, A. H. M. (2011). Automated error correction of business process models. In S. Rinderle-Ma, F. Toumani, & K. Wolf (Eds.), *Business process management* (pp. 148–165). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-23059-2_14

Ghasemaghaei, M., & Calic, G. (2019). Can big data improve firm decision quality? the role of data quality and data diagnosticity. *Decision Support Systems*, *120*, 38–49. https://doi.org/10.1016/j.dss.2019.03.008

Heinrich, B., Klier, M., & Zimmermann, S. (2012). Automated planning of process models. In S. Smolnik, F. Teuteberg, & O. Thomas (Eds.), *Semantic technologies for business and information systems engineering* (pp. 169–194). IGI Global. https://doi.org/10.4018/978-1-60960-126-3.ch009

Heinrich, B., Krause, F., & Schiller, A. (2019). Automated planning of process models: The construction of parallel splits and synchronizations. *Decision Support Systems*, *125*, 113096. https://doi.org/10.1016/j.dss.2019.113096

Heinrich, B., & Schön, D. (2015). Automated planning of context-aware process models. https://doi.org/10.18151/7217352

Khayyat, Z., Ilyas, I. F., Jindal, A., Madden, S., Ouzzani, M., Papotti, P., Quiané-Ruiz, J.-A., Tang, N., & Yin, S. (2015). Bigdansing. In T. Sellis, S. B. Davidson, & Z. Ives (Eds.), *Proceedings of the 2015 acm sigmod international conference on management of data* (pp. 1215–1230). ACM. https://doi.org/10.1145/2723372.2747646

Lahlou, F. Z., Benbrahim, H., & Kassou, I. (2017). Context aware recommender system algorithms: State of the art and focus on factorization based methods. *Electronic Journal of Information Technology*, *0*(0).

Laney, D. (2001). 3d data management: Controlling data volume, velocity and variety. *META group research note*, *6*(70), 1.

Liu, J., Li, J., Li, W., & Wu, J. (2016). Rethinking big data: A review on the data quality and usage issues. *ISPRS Journal of Photogrammetry and Remote Sensing*, *115*, 134–142. https://doi.org/10.1016/j.isprsjprs.2015.11.006

Marrella, A. (2019). Automated planning for business process management. *Journal on Data Semantics*, *8*(2), 79–98. https://doi.org/10.1007/s13740-018-0096-0

Marrella, A., & Chakraborti, T. (2021). Applications of automated planning for business process management. In A. Polyvyanyy, M. T. Wynn, A. van Looy, & M. Reichert (Eds.), *Business process management* (pp. 30–36). Springer International Publishing. https://doi.org/10.1007/978-3-030-85469-0_4

Mashey, J. R. (1998). Big data ... and the next wave of infrastress. Retrieved March 4, 2022, from https://static.usenix.org/event/usenix99/invited_talks/mashey.pdf

McCrae, R. R., & John, O. P. (1992). An introduction to the five-factor model and its applications. *Journal of personality*, *60*(2), 175–215. https://doi.org/10.1111/j.1467-6494.1992.tb00970.x

Naseem, U., Razzak, I., Khan, S. K., & Prasad, M. (2021). A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *20*(5), 1–35. https://doi.org/10.1145/3434237

Picault, J., Ribière, M., Bonnefoy, D., & Mercer, K. (2011). How to get the recommender out of the lab? In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 333–365). Springer US. https://doi.org/10.1007/978-0-387-85820-3_10

Pipino, L. L., Lee, Y. W., & Wang, R. Y. (2002). Data quality assessment. *Communications of the ACM*, *45*(4), 211–218. https://doi.org/10.1145/505248.506010

Reisert, C., Zelt, S., & Wacker, J. (2018). How to move from paper to impact in business process management: The journey of sap. In J. vom Brocke & J. Mendling (Eds.), *Business process management cases* (pp. 21–36). Springer International Publishing. https://doi.org/10.1007/978-3-319-58307-5_2

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

Ridzuan, F., & Wan Zainon, W. M. N. (2019). A review on data cleansing methods for big data. *Procedia Computer Science*, *161*, 731–738. https://doi.org/10.1016/j.procs.2019.11.177

Roser, M., & Ritchie, H. (2013). Technological progress. *Our World in Data*.

Roy, S., Sajeev, A. S. M., Bihary, S., & Ranjan, A. (2014). An empirical study of error patterns in industrial business process models. *IEEE Transactions on Services Computing*, *7*(2), 140–153. https://doi.org/10.1109/TSC.2013.10

Russell, S. J., Norvig, P., Davis, E., & Edwards, D. (2016). *Artificial intelligence: A modern approach* (Third edition, Global edition). Pearson.

Saha, B., & Srivastava, D. (2014). Data quality: The other face of big data. *2014 IEEE 30th International Conference on Data Engineering*, 1294–1297. https://doi.org/10.1109/ICDE.2014.6816764

Sar Shalom, O., Berkovsky, S., Ronen, R., Ziklik, E., & Amihood, A. (2015). Data quality matters in recommender systems. In H. Werthner, M. Zanker, J. Golbeck, & G. Semeraro (Eds.), *Proceedings of the 9th acm conference on recommender systems* (pp. 257–260). ACM. https://doi.org/10.1145/2792838.2799670

Siering, M., & Janze, C. (2019). Information processing on online review platforms. *Journal of Management Information Systems*, *36*(4), 1347–1377. https://doi.org/10.1080/07421222.2019.1661094

Taleb, I., Serhani, M. A., & Dssouli, R. (2018). Big data quality: A survey. *2018 IEEE International Congress on Big Data (BigData Congress)*, 166–173. https://doi.org/10.1109/BigDataCongress.2018.00029

Vallurupalli, V., & Bose, I. (2020). Exploring thematic composition of online reviews: A topic modeling approach. *Electronic Markets*, *30*(4), 791–804. https://doi.org/10.1007/s12525-020-00397-5

Wang, J., Xu, C., Zhang, J., & Zhong, R. (2021). Big data analytics for intelligent manufacturing systems: A review. *Journal of Manufacturing Systems*. https://doi.org/10.1016/j.jmsy.2021.03.005

Zheng, K., Zhang, Z., & Song, B. (2020). E-commerce logistics distribution mode in big-data context: A case analysis of jd.com. *Industrial Marketing Management*, *86*, 154–162. https://doi.org/10.1016/j.indmarman.2019.10.009