Information Systems Frontiers
https://doi.org/10.1007/s10796-020-10071-y

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

Bernd Heinrich[1] · Marcus Hopf[1] · Daniel Lohninger[1] · Alexander Schiller[1] · Michael Szubartowicz[1]

Accepted: 25 September 2020

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

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

✉ Bernd Heinrich
Bernd.Heinrich@ur.de

[1] Department of Management Information Systems, University of Regensburg, Universitätsstraße 31, Regensburg 93053, Germany

Published online: 22 October 2020

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.

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

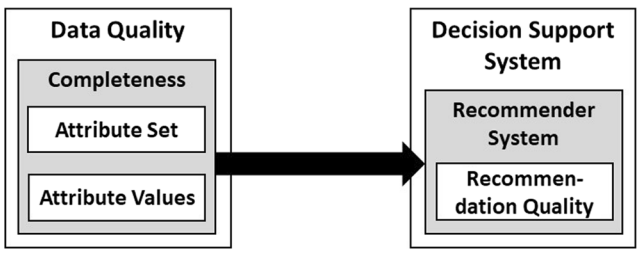

Fig. 1 Theoretical model (according to Heinrich et al. 2019)

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 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.

[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.

(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 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.

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).

In **Task 1.1**, the data for duplicate detection is standardized and prepared. This is necessary because different portals often

[3] If more than two data sets are available, the procedure can be applied iteratively.

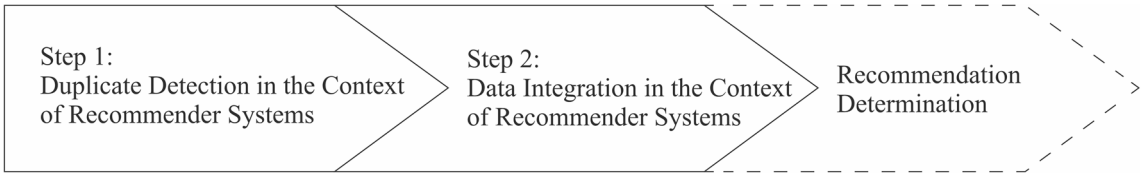

Fig. 2 Procedure to extend an item content data set in the context of recommender systems

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

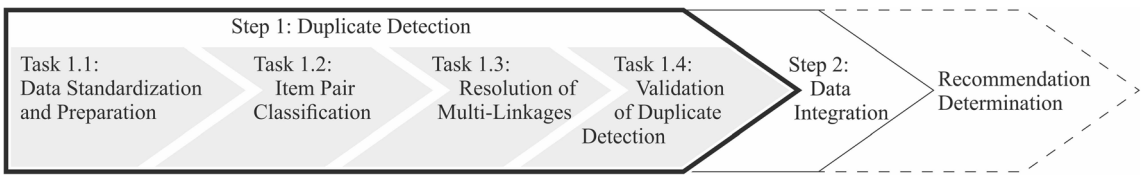

Fig. 3 The step duplicate detection in detail

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".

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.

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).

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.

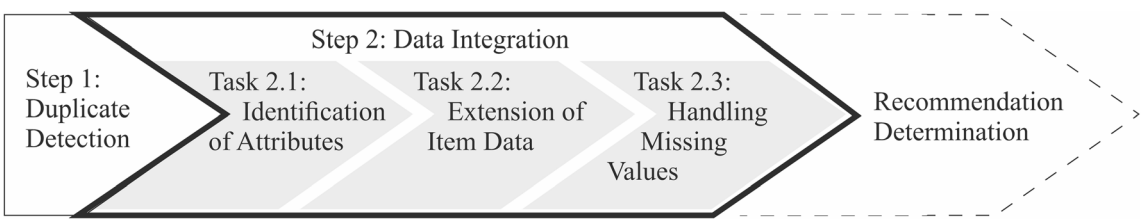

Fig. 4 The step data integration in detail

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 $\widehat{S}_i = \widehat{\beta}_0 + \widehat{\beta}_1(P_i)$ 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 $\widehat{S}_i$.
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 $\widehat{r}_{ui} = p_u^T q_i$, 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.

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.

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.

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%)

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 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%)

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

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})

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.

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

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

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.

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

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%

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

[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.

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.

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.

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 as 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.

Funding Open Access funding enabled and organized by Projekt DEAL.

Compliance with Ethical Standards

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: 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: 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: 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: 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: Practical Innovations, Open Solutions, 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: Springer.

de Pessemier, T., 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: ACM. https://doi.org/10.1145/1864708.1864767 .

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: ACM. https://doi.org/10.1145/2043932.2043979 .

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

GitHub. (2020). Procedure completeness: Extending item content data. https://github.com/ProcedureCompleteness/ExtendingItemContentDataSets. Accessed 14 Sept 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: 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: 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,: 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: 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: 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: 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., Harper, M., Terveen, F., & 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: 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: 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 .

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: 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: Wiley.

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: IEEE Service Center.

Ricci, F., Rokach, L., & Shapira, B. (Eds.). (2015a). *Recommender Systems Handbook*. Boston: Springer US.

Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender systems: Introduction and challenges. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender Systems Handbook* (pp. 1–34). Boston: 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: 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: 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: 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: 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: Curran.

Xu, H., Liu, B., Shu, L., & Yu, P. (2019). 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* (pp. 2324–2335). Minneapolis: 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