SOCIAL PRODUCT SEARCH – ENHANCING PRODUCT SEARCH WITH MINED (SPARSE) PRODUCT FEATURES

Research paper

Hirschmeier, Stefan, University of Cologne, Köln, Germany, hirschmeier@wim.uni-koeln.de

Egger, Marc, University of Cologne, Köln, Germany, egger@wim.uni-koeln.de

Abstract

Search functionality in web shops is limited today. Consumers can only search for a restricted set of standard product features for each product group. A major part of the relevant information, especially reviews, is only available as an extension to the product description, that is, after a product has been found and therefore quite late in the purchase process. With the growing numbers of reviews, reading review texts is a burden for consumers, and there is a need to rearrange and organize user-generated content. Integrating mined product features into the product search might therefore add value to the customer experience. Following design science principles, we propose an approach to mine frequent product feature sets from social media content and enhance product search with sets of product features to create a "social product search". We contribute a design science artefact in form of a situated implementation. For illustration, we present an example for the product group notebooks with 22480 reviews of 2745 products that we crawled from amazon.com. Further, we depict three application scenarios how mined frequent product feature sets can be integrated into the product search and enhance the consumer search experience.

Keywords: Product search, product feature extraction, frequent item set mining, product reviews.

1 Introduction

Search functionality in web shops is limited today. We depict an example: A user is in search of a notebook. He/she has some requirements – the size should be 13", with a 1TB hard disk, but most of all he/she wants a notebook which does not make any noise, as the fan of the previous notebook was annoying him/her. In the web shop, he/she can select the size of the notebook and the storage, but he/she cannot search for notebooks with a silent fan. For all notebooks that match his/her criteria of size and storage, he/she must read all the reviews and find out what users mentioned about the fan.

We notice a mismatch of what users write about products, and what they can search for. Current search functionalities allow users to search and filter for a limited set of facts about the product, such as storage size or screen size. All other aspects, among them decision-critical ones, are not accessible in the product search. The information that users write down in reviews depicts what matters to them, and there is strong indication that this information will also be decision-critical to other users (Cao, Duan, & Gan, 2011; Chevalier & Mayzlin, 2006), not only *after* they found a product they consider buying, but already when they search for it (i.e., earlier in the purchase process).

Several years ago, when there were only few product reviews (user reviews, customer reviews, consumer reviews, online reviews or simply reviews are synonyms used in literature), the information written down in user experiences could be consumed in reasonable time. Therefore, it was and is still common to provide the information *after* a user shows interest in a certain product (Huang, Etzioni, Zettlemoyer, Clark, & Lee, 2012). But the number of product reviews increased enormously in the last decade, and ecommerce customers now have a rich source of information at hand to gain insights into

the products they consider buying. With the growing number of reviews, the need increases to organize and rearrange reviews (Hu & Liu, 2004).

We notice the emergence of a problem and an opportunity at the same time: The emerging problem is that users are often overcharged with information as they can hardly read all the reviews (Chen, Shang, & Kao, 2009; Furner, Zinko, & Zhu, 2015; Park, Lee, & Han, 2006). The emerging opportunity is that the amount of reviews allows to extract a rich amount of product features (synonymously called product attributes, aspects, facets or product properties in literature) from the review texts. A set of mined product features can provide a condensed view on a product. Even more sophisticated, such inferred product features could be used in the product search process to provide a richer search functionality (e.g. search for notebooks with a "bright display").

Mining product features is basically understood, even though harmonization and aggregation remain a challenging task. However, when we set those mined product features into search and filter functionalities, we encounter two problems: a) The sparsity of product feature combinations and b) which product features to show to the user when space for display is limited. For example, when a user searches for "silent fan", he might obtain a handful of matching products. When he also searches for "bright display", he might end up with only two or three matching products (the only products in which some reviewers mentioned the silent fan and some other reviewers emphasized the bright screen). When adding a third keyword to the search, the result set could already end up being zero. Therefore, although user-generated content (UGC) is a rich source of information, nobody has control over which product features reviewers write about, and therefore the allocation between product and product features is incomplete.

The product feature sparsity problem is crucial for the user interaction. A search interface where users might quickly run into zero search results is a disappointing experience. Instead of letting the user run into scenarios with zero results, users might better be served with information which product features typically come together and how many products match. Therefore, we formulate the following research question:

RQ: How to design a product search that makes use of sparse product features (e.g. mined product features from user-generated content)?

To answer this question, we 1) elicit design requirements, 2) translate the design requirements to design principles, 3) build an instantiation according to the design principles and 4.) evaluate our instantiation against the design requirements.

Considering the empty result set problem, some mechanisms already exist in information filtering, such as soft filtering and ranking approaches. They however do not support the user in getting a quick overview of which features typically come together and understanding the product domain.

The idea of integrating product information from social media into the product search can relate to a "social product search". The term "social product search" has rarely been used in literature so far. In fact, only one source (Yi, Jiang, & Benbasat, 2017) recently used the term to describe two scenarios: a) the navigation through the product space by tags (= product features), and b) the access to high-quality content of socially endorsed individuals. Whereas Yi et al. assume that users create tags directly, we argue that this is not necessary, as tags (product features) can also be extracted automatically from reviews and assigned to products afterwards. In this perspective, it does not matter if the data source is a full text or a tag. Therefore, we subsume our approach under "social product search", but try to be more explicit by calling it product search with mined product features.

The logic of this paper's focus is as follows: Mined product features are a) sparse and b) can be used for a social product search as described above. Therefore, product search with *mined* product features is a special case of product search with *sparse* product features. The design of a product search with *mined* product features generalizes to product search with *sparse* product features, independent from the source of the product features (i.e., mined from reviews or created directly by users).

Following a design science approach, the remainder of this paper is structured as follows: in Section 2, we present related work for mining and aggregating product features. In Section 3, we depict our de-

sign science approach, followed by design requirements, design principles and the actual instantiation. Section 4 concludes with a discussion.

2 Related Work

Integrating product features into the product search requires techniques for mining and aggregating them. In the following, some approaches from the research field of product feature extraction and integrating it into search scenarios are presented.

First, we present two approaches that also follow the idea of using mined product features in a search context. Huang et al. present *RevMiner* (Huang et al., 2012), an extractive interface for navigating reviews. The authors extract attribute-value pairs from restaurant reviews and allow users to search for restaurants with specific properties (e.g., "huge portions" or "good service") and compare them. Feuerbach et al. (2017) extract attribute-value pairs and integrate them into an interactive recommendation process, depicted with a hotel search. Their hotel guide allows users to search for, e.g., "comfortable beds" or "quiet location".

A lot of research has been done on information extraction in general, usually using natural language processing (NLP) and domain knowledge in the form of ontologies. We focus on research with particular focus on product feature identification.

Thorough overviews of text-extraction, sentiment analysis and opinion mining are given by Liu (2012) and Jiang (2012). Lee et al. (2008) integrated traditional information-retrieval relevance rankings with database aggregation to model the knowledge within online product reviews and product descriptions in order to provide a needs-centric search wherein users can input free-text queries. Dave et al. (2003) proposed a classifier that draws on information-retrieval techniques for feature extraction and scoring in order to generate a list of product features (e.g., quality) and aggregate opinions about each of them. Aggregating subjective opinions is a challenge but leads to more intersubjective information, referred to as "objectivity by averaging" (Parameswaran & Whinston, 2007). Approaches for aggregating sentiments using ontologies have been proposed by Mukherjee and Joshi ((2013); also see (Mukherjee & Joshi, 2014)) and Grandi et al. (2014). Social business intelligence approaches (Francia, Golfarelli, & Rizzi, 2014) make up an interdisciplinary research area and combine data-mining technologies, natural language processing, and other promising techniques to identify product aspects. Gallinucci et al. (2013) proposed a way to aggregate topics for social business intelligence, and Hu and Liu (2004) proposed an approach to mine and summarize customer product reviews, focusing on product features only and generating feature-based summaries. Further, Popescu and Etzioni (2005) constructed an unsupervised information-extraction system, which mines reviews in order to build a model of important product features.

Yang et al. (2016) proposed a combined approach, which integrates local context information and global context information to extract and rank features based on feature score and frequency. Finding a new way of combining supervised and unsupervised learning techniques, Wang et al. (2014) propose two novel semi-supervised models for product aspect extraction. Quan and Ren (2014) propose a method of unsupervised product aspect extraction for feature-oriented opinion determination, where domain-specific features are extracted by measuring the similarity distance of domain vectors. Wei et al. (2009) propose a semantic-based product feature extraction technique that exploits a list of predefined positive and negative adjectives to recognize opinion words semantically and subsequently extract product features expressed in consumer reviews. Cruz et al. (2010) describe a domain-specific, resource-based opinion extraction system and suggest that domain-specific knowledge is a valuable resource in order to build precise opinion extraction systems.

Especially for ecommerce websites, it is vital to process and present product features in an appropriate way. Nevertheless, although diverse techniques exist to extract product features from full-text data and approaches have been proposed to integrate product features into early purchase-process phases, in practice, these approaches have rarely been applied.

3 Designing a Product Search with Sparse Product Features

3.1 Research Design

Our research follows design science principles (Hevner et al., 2004; March & Smith, 1995). Design science is a well-known research approach in IS and has been reemphasized in the last years (Gregor & Hevner, 2013; March & Smith, 1995). At the core of design science is the iterative design with continuous reflection and incremental refinement (Hevner et al., 2004; Takeda, Veerkamp, & Yoshikawa, 1990). Whereas many scholars have proposed guidance how to conduct design research (e.g., Hevner, 2007; Iivari, 2015; Jones & Gregor, 2007; Nunamaker, Chen, & Purdin, 1990; Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007), our research design reflects the ideas of Vaishnavi and Kuechler (2015). We also got inspired by the work of Meth et al. (2015). We conducted two design cycles, one for retail products on ecommerce platforms, and one for services in the tourism industry. In the end, we contribute a level 1 design science contribution in the terminology of Gregor and Hevner (2013).

Figure 1 presents the content of this section. We elicit three design requirements, that are addressed in two design principles. The two design principles then influence three of the process steps of our implementation.

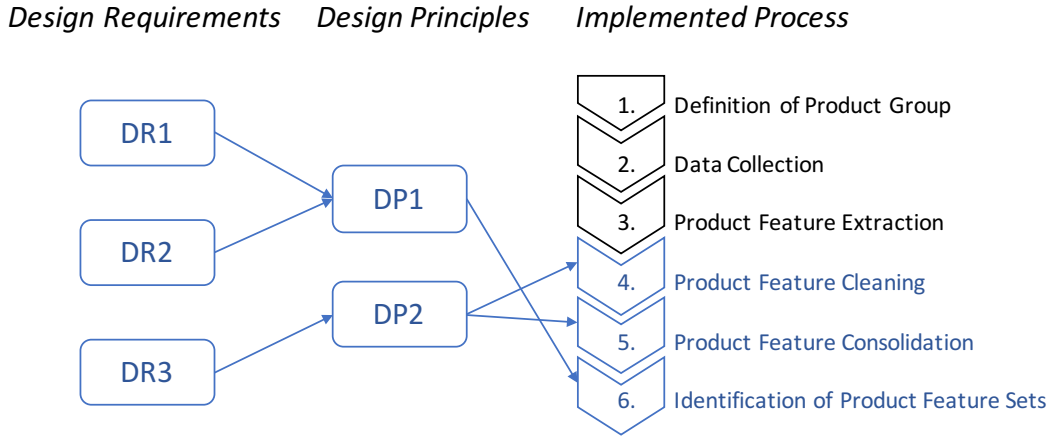

Figure 1. Elements of the design science approach

3.2 Design Requirements

To elicit design requirements, we first create valid scenarios for the integration of mined product features into a product search. Such a "social product search" is not commonly known in practice, and the integration into current product search concepts is not necessarily straightforward.

We built a focus group of three researchers to concretize the properties of such a system. The method we used relates to design thinking, that is, considering information needs and habits of users, taking existing literature into account, integrating ideas, working out concepts in more detail, and continuously validating our results. In the end, we carved out three different scenarios, and validated them with a fourth researcher. The key insight was that the sparsity of product features may lead to three modes of presentation: a) a "static" presentation of product feature combinations, b) a "dynamic" presentation, where the user navigates iteratively through the space of possible combinations by using filters or c) a navigation by tags that are associated to the products.

The scenarios are depicted with the help of user interface wireframes.

Static product feature sets approach. The first approach displays the product feature combinations in predefined sets to the user. Users can learn from these combinations, as the combinations give information about product clusters. Some product clusters may be obvious, such as a cluster with product features like high-performance graphics and powerful CPU obviously determines gamer note-

books. Another example could be a cluster with product features like “lightweight product” and “short battery runtime”, which would be typical for smaller but handy notebooks. Apart from these obvious product clusters, product feature sets of other product clusters may be unexpected and surprising. Furthermore, since product features in reviews are not only mentioned in a positive way, user may learn about tradeoffs in the product group, e.g. a long-lasting battery often comes along with more weight, and consumers can better understand the compromises they probably have to make. This way, product feature sets not only help in fast navigation, but are also a means of understanding the product domain.

Displaying product feature sets supports the learning process of the customer and therefore can possibly increase both the quality and the speed of the decision process. Next to learning from product feature combinations, users could also directly filter for these predefined combinations.

The approach is depicted in Figure 2, which shows a wireframe of a web shop. Next to classic filters such as SSD size, a filter based on review information is presented that allows to select product feature sets based on UGC. In brackets, the remaining result set is displayed, as is the status quo in today’s web shops. Alternatively, product feature sets could be presented in clouds that the user may click on.

Figure 2. “Static” presentation of product feature sets as predefined filter combinations

Dynamic product feature sets approach. The second approach guides the user through an iterative filtering process, in order to avoid combinations that have no results. Instead of providing predefined product feature sets for filters as in the first scenario, single product features can be selected.

As reviews cover both positive and negative experiences with product features, filters can be designed accordingly. Whereas consumers select positive product features in filters, they may also want to exclude negative product features, i.e., it might make sense to provide positive filters to select positive product features (as familiar to users from their web shop experiences), but also to provide negative filters to exclude product features that were reviewed in a negative way.

Both positive and negative filters of mined product features are not necessarily symmetric. As an example, for 100 notebooks, we might find that the screen was found to be bright for 34 products, and in reviews for 15 products the screen was found to be quite dark. That means, for the remaining 51 notebooks we do not have any information about the screen, which means selecting a bright screen does not have the same effect on search results as excluding a dark screen.

Even worse, contradicting opinions could emerge: for the same product, some reviewers might find the screen to be bright and another reviewer might consider the screen to be too dark. So, when includ-

ing "bright display", we select 34 products out of 100; but when excluding "dark display", we end up with 85 products in the result set.

Figure 3 depicts the dynamic approach in three exemplary steps. First, from all available filters, "silent fan" is selected. The number of remaining search results for all other filters is adapted.

For negative filters, we did not indicate the numbers of the remaining result set, because it is not clear if consumers will expect that the number in brackets shows the number of products that remain or that will be excluded. This is however also not research objective for the paper at hand.

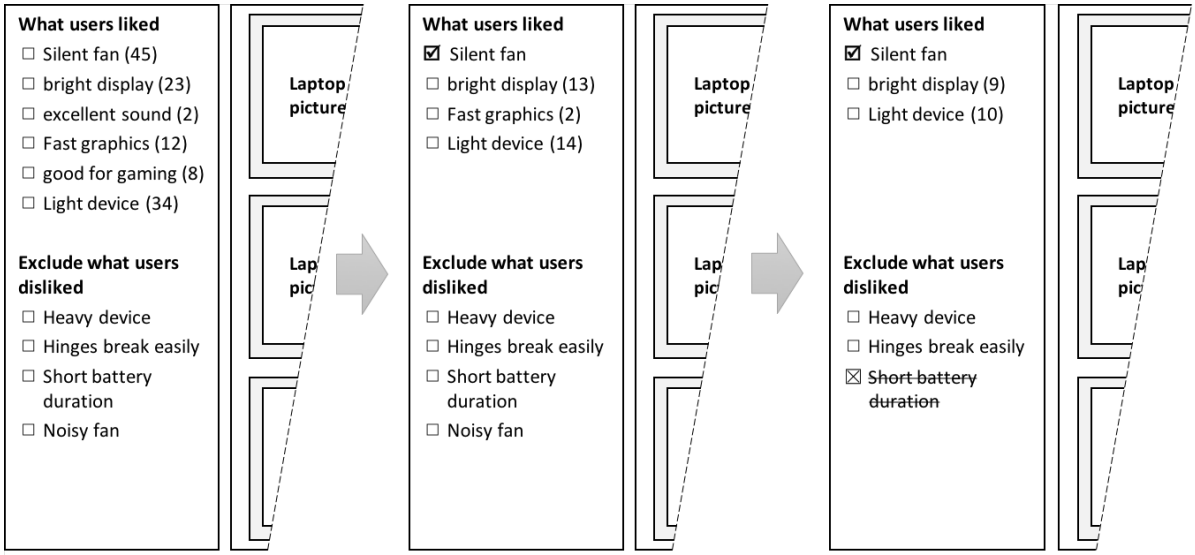

Figure 3. "Dynamic" presentation by iterative selection of positive and negative filters

Navigation by tags. In the third scenario, users can navigate through the product space by clicking tags which are associated to the products. The social commerce website stylehive.com is an example for such an interaction design. Therefore, we do not present a figure for this scenario.

All three approaches were presented to outline possible instantiations of a social product search with mined product features. Hybrid approaches and other approaches are imaginable.

Based on the three scenarios, we were able to formulate user requirements as an intermediate step, and from there to derive design requirements for the system that hold true for all three scenarios.

User requirements	System Design Requirements (DR)
Search results should not end up being zero, i.e. no product combinations with zero matching products should be displayed. (would be a disappointing experience and unexpected from today's product searches)	**DR1:** Establish a data structure for possible product feature combinations, that can be efficiently used.
Search results must cover available products. (which is a standard expectation derived from common product searches)	**DR2:** As the availability of products changes constantly (new products coming in, old products dropping out), the computation of result sets has to be congruent with the available products.
Product feature sets should be consolidated and meaningful to the user. Feature sets may also contain negative product features. (Filters or tags in today's product searches are expected to be comprehensible and consolidated)	**DR3:** Preprocessing should include a manual step for validation and consolidation of mined product features.

Table 1. User requirements and system design requirements.

3.3 Constructing Design Principles

3.3.1 Design Principle 1

In order to construct design principles, we take a closer look at the relationship between products and mined product features in a network analysis.

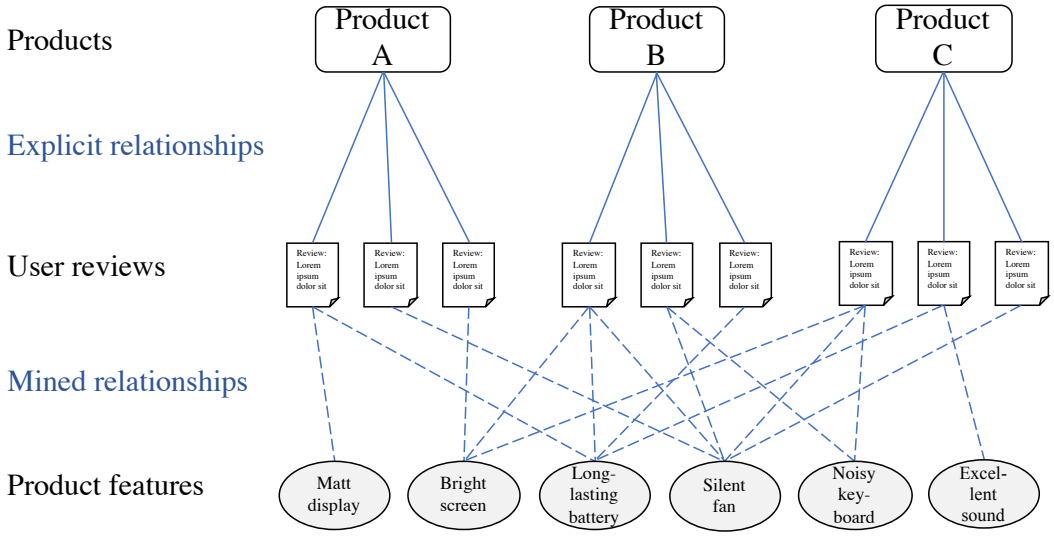

Figure 4. Three-mode graph with products, reviews, and product features

Figure 4 shows a three-mode graph with products, reviews, and product features. Products and reviews are interconnected by explicit relationships, reviews and product features by mined relationships.

For the purpose of product search, we do not need the review layer, as during search, we can omit the information from which review the product features were mined. We therefore delete the review layer and obtain a two-mode graph. The information, which review accounts for the product features, may however be stored in the product feature vertices. Figure 5 shows the same graph with a deleted review layer.

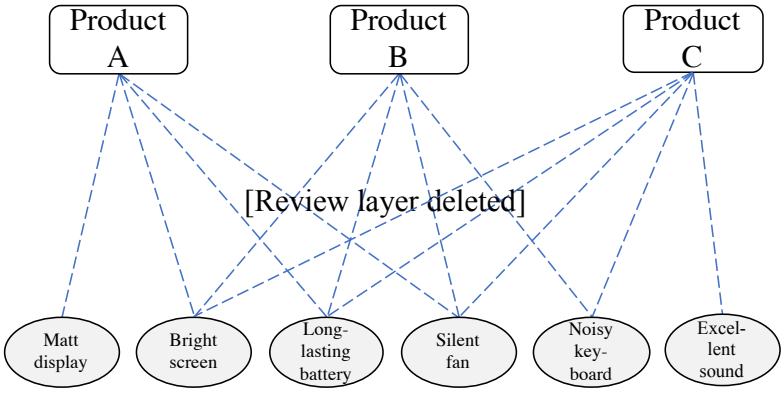

Figure 5. Two-mode graph with products and product features only

To obtain a one-mode graph with product features as vertices only, we eliminate the product nodes as well. Between each two product features, we count how many ways there are to get from one product feature to the other, i.e. how many products are connected to both features. We use this information as a weight for the edges. In order not to lose the information, which products represent the edges, we store this information separately. Figure 6 shows the resulting one-mode graph.

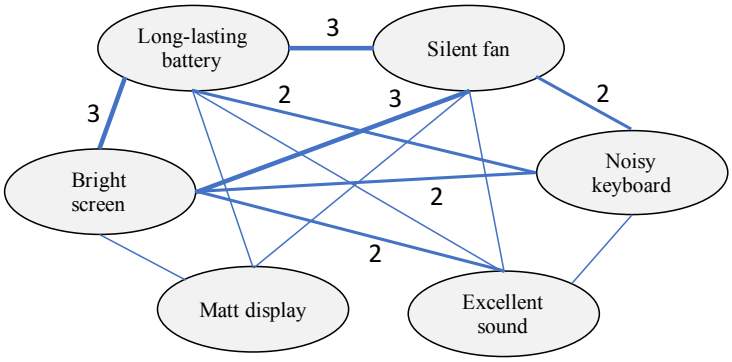

Figure 6. One-mode graph with product features only

The problem of finding frequent product feature sets is the same problem as to find cliques, that is, complete subgraphs in the one-mode graph, ideally with the biggest edges possible (i.e. for the same products). The problem of finding cliques in a graph however is a problem that is supposed to be NP-complete (Karp, 1972).

Luckily, the problem of identifying frequently mentioned product feature sets can be modelled analogously to a well-known data mining problem – the Frequent Item Set Mining Problem in the research area of Association Rule Mining. Its most popular application is the market basket analysis or, more abstract, affinity analysis.

Within frequent item set identification, the Apriori-Algorithm (Agrawal & Srikant, 1994) has become popular, and a lot of variants, improvements and alternatives have been proposed in literature, such as AprioriClose, AprioriTID, FPGrowth (Han, Pei, & Yin, 2000), FPClose (Grahne & Zhu, 2005), and FPMax (Ziani & Ouinten, 2009).

The algorithms find, in a set of transactions, similar item sets and also provide their support, i.e. in how many transactions the identified frequent item set occurs. Different frequent item sets can be mined (Fournier-Viger, 2018; Pasquier, Bastide, Taouil, & Lakhal, 1999):

(1) Frequent Item Sets: A frequent item set is an item set appearing in at least min_support transactions from the transaction database, where min_support is a parameter given by the user.

(2) Frequent closed item set: A frequent closed item set is a frequent item set that is not included in a proper superset having exactly the same support. The set of frequent closed item sets is thus a subset of the set of frequent item sets.

(3) Frequent maximal item set: A frequent maximal item set is a frequent item set that is not included in a proper superset that is a frequent item set. The set of frequent maximal item sets is thus a subset of the set of frequent closed item sets.

Regarding runtimes, the FPGrowth /FPClose /FPMax family of frequent item set mining has shown the best performance within the landscape of all algorithms. The speed of the frequent item set mining is important, as product availability is a dynamic context and the number of available products for a product feature set may need to be recalculated ad hoc, or at least regularly.

Therefore, adressing DR1 and DR2, we can formulate a design principle as follows:

DP1: Product searches with mined product features should use best-performing frequent item set mining algorithms. Best-performing implementations to date are FPGrowth, FPClose and FPMax.

3.3.2 Design Principle 2

Reviewers might express the same product feature in various ways: “good battery”, “long-lasting battery”, “good battery life” and “excellent battery runtime” all denote the same product feature. The example makes clear that not only the adjectives might differ, but also various combinations of entities

and adjectives might refer to the same product feature. As in our context, product features should be aggregated as much as possible in order to obtain a limited and easy-to-consume set of product features with strong support, there is a need to consolidate the extracted product features as good as possible. A consolidation procedure requires human interaction, at least as a one-time effort.

Once a consolidation has been done manually, domain-specific knowledge is built up and can be stored in taxonomies and dictionaries with synonyms for further automation. Of course, this domain knowledge refers to product groups and needs to be elicited for every product group separately. Also, it needs to get updated regularly. Two raters should to independently aggregate product features, to approach inter-rater consistency.

DP2: Data preparation for product searches with mined product features should include a manual intermediate step for cleaning and consolidating product features.

3.4 Process Instantiation According to Design Principles

In the following, we depict our approach to extract product features from reviews and determine frequent product feature sets. As an example for illustration, we use the product category notebooks. The approach follows a six-step process, which reflects the design principles identified in Section 3.3 and roughly refers to the approach of Gensler et al. [23]. The process steps are (1) definition of the data set, (2) data collection, (3) extraction of product features, (4) product feature cleaning, (5) product feature consolidation, and (6) identification of product feature sets that were frequently mentioned in reviews.

3.4.1 Definition of Product Group

In the first step, we choose a product group for which we want to extract product features. Logically, a product group is the suitable scope of consideration for product feature extraction, as products within the same product group share product features and products from different product groups have different product features. As an example, for this paper we chose the product group notebooks with reviews from amazon.com.

3.4.2 Data Collection

To collect notebook reviews from amazon.com, we set up a web crawler with a screen scraper that collected all reviews connected to notebooks and saved the scraped texts into a database. We collected a total amount of 22480 reviews in German language for 2745 notebooks. In a fast-changing product group like notebooks, a three-year timeframe seemed sufficient for practical purposes. To obtain a richer database for this example, we however set a ten year timeframe from September 2005 to September 2015.

3.4.3 Product Feature Extraction

To extract product features, it is first of all important to understand how reviewers explicate their opinions and what we denote with product feature. Prominent work on product feature extraction has been done by Liu (2012). Liu defines an opinion as a *quintuple* (e_i, a_{ij}, s_{ijkl}, h_k, t_l), where e_i is the name of an entity (e.g. "fan"), a_{ij} is an aspect of e_i (e.g. "silent"), s_{ijkl} is the sentiment on aspect a_{ij} of entity e_i, h_k is the opinion holder, and t_l is the time when the opinion is expressed by h_k. Of course, when talking about product features, the entity alone does not provide much informational value – e.g. it is not surprising that the notebook has a fan. The informational value is given by the combination of e_i and a_{ij}. So, in this study, when talking about product features, we mean the tuple (e_i, a_{ij}), i.e. an entity with a qualifying aspect. The representation of e_i and a_{ij} as adjective and noun is not only a frequent pattern in texts, in which reviewers explicate their opinions, but also the most condensed form of presenting the extracted product features.

The product feature extraction technique we used resembles very much the approach used by Egger (Egger & Schoder, 2017). We quickly summarize the approach, which is a combination of tokeniza-

tion, part-of-speech (PoS) tagging and sequence analysis. The reviews are disassembled into smaller pieces with tokenization (Feldman & Sanger, 2006), to obtain several sentences for every review and a set of words for every sentence. For every single word, a PoS tag is given according to the approach of Schmid (1994). Therefore, we obtain a secondary representation of the text solely made up of PoS tags. These PoS tags allow to use pattern recognition for extracting opinions about product features.

3.4.4 Product Feature Cleaning

In the set of all product features for the product group, we find several that are not specific enough, such as "nice notebook" or "daily use". Also, some reviewers report known metadata such as "128GB RAM" or information about the price. We delete such product features (in line with design principle DP2), as they have no additional value for the product search process.

3.4.5 Product Feature Consolidation

According to DP2, product features should be consolidated. It took us less than two hours for one person to manually screen, clean up and consolidate all product features. For the product group notebooks, we obtain a list of 41 product features for the product group notebooks (see Table 2).

bright screen	excellent sound	high-quality chassis	small charger
built-in SSD	fast graphics	high-resolution screen	small keyboard
built-in battery	fast hard disk	lightweight device	small touchpad
cheap chassis	fast SSD	long-lasting battery	slow hard disk
classy design	good WLAN reception	low-resolution screen	solid hinges
comfortable keyboard	good contrast	matt screen	thin device
comfortable touchpad	good pressure point	noisy fan	upgradeable RAM
dark display	good service	non-reflective display	weak battery
dedicated graphics	good value for money	reflective screen	weird keyboard
disappointing sound	heavy device	silent fan	weird touchpad
excellent screen			

Table 2. Consolidated product features for product group notebooks

3.4.6 Identification of Product Feature Sets

The identification of product feature sets is designed by using SPMF, a Java open-source pattern mining library (Fournier-Viger et al., 2014). It is a collection of frequent item set mining algorithms that suffice DP1. For illustration, Table 3 shows an example of source data for frequent item set mining, and Table 4 shows three different frequent item sets for the data in Table 3.

Product	**Mined product features**
Product A	{long-lasting battery, silent fan, high-resolution screen}
Product B	{long-lasting battery, silent fan, high-resolution screen, excellent sound}
Product C	{silent fan, high-resolution screen, excellent sound}
Product D	{long-lasting battery, bright display}
Product E	{long-lasting battery, silent fan, high-resolution screen, excellent sound}

Table 3. Mined product features, building the source for Table 4.

Frequent item sets *(Apriori / FPGrowth)*	Support	Frequent closed item sets *(Apriori Close / FPClose)*	Support	Frequent maximal item sets *(FPMax)*	Support
{1}	4	{1}	4	{1, 2, 4}	3
{2}	4	{2, 4}	4	{2, 4, 5}	3
{4}	4	{1, 2, 4}	3		
{5}	3	{2, 4, 5}	3		
{1, 2}	3				
{1, 4}	3			1: long-lasting battery 2: silent fan 3: bright display 4: high-resolution screen 5: excellent sound	
{2, 4}	4				
{2, 5}	3				
{4, 5}	3				
{1, 2, 4}	3				
{2, 4, 5}	3				

Table 4. Frequent, closed and maximal item sets for Table 3 with min_support = 0.5

3.5 Evaluation

To evaluate the approach against the design requirements, we used a sample of 22480 reviews of 2745 notebooks. With 41 consolidated product features (see Table 4) we get the following results (Table 5):

	FPGrowth	FPClose	FPMax
	with min_support = 0.01		
item sets	400182	104276	8757
runtime	606 ms	22 s	811 ms
	with min_support = 0.1		
item sets	390	390	108
runtime	51 ms	60 ms	59 ms
	with min_support = 0.2		
item sets	42	42	17
runtime	20 ms	21 ms	38 ms

Table 5. Number of frequent product feature sets for the product group notebooks and their runtimes with different algorithms

To describe the results, we refer to the largest product feature set we found, because this is also typically one of the sets with the lowest support.

The largest feature set we found for min_support = 0.01 included 12 product features *({high-quality chassis, excellent screen, comfortable keyboard, long-lasting battery, high-resolution screen, silent fan, lightweight device, bright screen, reflective screen, classy design, excellent sound, low-resolution screen}*) with a support of 28 products.

For min_support = 0.1, the largest product feature set included 6 product features *({high-quality chassis, excellent screen, comfortable keyboard, long-lasting battery, high-resolution screen, noisy fan}*) with a support of 245 products.

Using a min_support of 0.2 resulted in a largest product feature set of 4 product features (*{high-quality chassis, excellent screen, comfortable keyboard, long-lasting battery}*) with a support of 556 products.

Results for the product group notebooks allow a positive evaluation for DR1. The computed product feature sets seem appropriate for user consumption and allow effective handling. Also DR2 can be evaluated positively: Runtimes allow for fast computation that suffices both regular pre-calculation as well as ad-hoc-calculation if necessary. DR3 as a manual step was not evaluated.

Apart from design principles, one might want to ask the question what are the best frequent item sets and the optimal parameters for the approach at hand. The answer depends on the application scenario. For displaying predefined "static" frequent product feature sets during the product search to the consumers (scenario 1), the frequent maximal patterns provided by the FPMax algorithm seem suited best, because the product sets are most diverse from each other. For the purpose of dynamic filtering (scenario 2) and navigation by tags (scenario 3), results from FPGrowth seem most suitable, as they are most detailed.

Without focusing on exact tuning of parameters, the results show that reasonable results can be obtained in reasonable time.

4 Discussion, Limitations and Further Research

In this paper, we motivated to enhance product search with information from reviews in the form of product feature sets that are frequently mentioned in reviews. The main contribution of the paper is a level 1 design science contribution in the terminology of Gregor and Hevner (2013). We illustrated the instantiation of the identification of product feature sets with the product category notebooks. Furthermore, we gave an idea in three scenarios of how applications of enhanced product searches may look like. While other approaches may exist, we add one further approach to the landscape of approaches how to use mined product features in search systems and contribute to the pluralism of techniques.

Our approach has some limitations. First, we did not develop a higher-valued level 2 design theory (according to Gregor and Hevner (2013)). We feel that we need more design cycles to do that. The second limitation is, we did not perform a user study so far, addressing questions like the impact on quality and speed of the purchase process. Third limitation is, the three scenarios we elaborated on for the intended product search do not have external validation with professionals from the industry yet. As a fourth limitation, we need to address that our approach is not free of typical problem areas of natural language processing. Challenges occur when users do not accurately or only implicitly describe product features, use non-standard terms or sloppy language.

The design of a "social product search" bears a lot of further questions how to design such a search, reaching from data quality to usability issues. We point out a few for further research:

Positive filters with UGC data bring the problem along, that a lot of products are filtered out, because all products that have no information in their reviews are automatically removed from the result set. Therefore, in data sources with a lot of missing data, positive filters affect the result set more than possibly useful. Negative filters (excluding filters) in contrast do not filter out products that have no mentions about product features – they just cut a small fraction of products out of the result set. Negative filters however include the problem that filtered out product will never be seen again during the search, i.e. there is no possibility to later read the full text reviews of those products and to prove if the properties were correct. The use of negative filters might however be of interest because of two reasons: First, some consumers read negative reviews first because they expect to find the decision-critical issues in there, and second, the result set will be affected less than with positive filters.

As the use of negative filters is unfamiliar however, further issues may arise. Future research could focus on the question, if negative information about product features is useful during the product search. Further questions arise: Does a product search with mined product features have any effect on consumer learning, on faster decisions, on conversion? How do consideration sets change, and are there systematic biases towards certain products, whereas other products systematically suffer?

Still, beneath open questions, this paper may hopefully serve as a basis for further research on enhanced product search possibilities with review information, in order to further leverage the enormous

potential of UGC. Future web shops will have to find a way to cope with information overload and to structure and present product information from reviews in a more structured and consumable way.

References

Agrawal, R., and R. Srikant. (1994). "Fast Algorithms for Mining Association Rules in Large Databases". In *Proceedings of the 20th International Conference on Very Large Data Bases* (pp. 487–499). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Cao, Q., W. Duan, and Q. Gan. (2011). "Exploring Determinants of Voting for the "Helpfulness" of Online User Reviews: A Text Mining Approach". *Decis. Support Syst.*, *50*(2), 511–521.

Chen, Y.-C., R.-A. Shang, and C.-Y. Kao. (2009). "The Effects of Information Overload on Consumers' Subjective State towards Buying Decision in The Internet Shopping Environment". *Electronic Commerce Research and Applications*, *8*, 48–58.

Chevalier, J. A., and D. Mayzlin. (2006). "The Effect of Word of Mouth on Sales: Online Book Reviews". *Journal of Marketing Research*, *43*(3), 345–354.

Cruz, F. L., J. A. Troyano, F. Enríquez, F. J. Ortega, and C. G. Vallejo. (2010). "A Knowledge-rich Approach to Feature-based Opinion Extraction from Product Reviews". In *Proceedings of the 2Nd International Workshop on Search and Mining User-generated Contents* (pp. 13–20). New York, NY, USA: ACM.

Dave, K., S. Lawrence, and D. M. Pennock. (2003). "Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews". In *Proceedings of the 12th International Conference on World Wide Web* (pp. 519–528). New York, NY, USA: ACM.

Egger, M., and D. Schoder. (2017). "Consumer-Oriented Tech Mining: Integrating the Consumer Perspective into Organizational Technology Intelligence - The Case of Autonomous Driving". *Hawaii International Conference on System Sciences 2017 (HICSS-50)*.

Feldman, R., and J. Sanger. (2006). "*Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*". New York, NY, USA: Cambridge University Press.

Feuerbach, J., B. Loepp, C.-M. Barbu, and J. Ziegler. (2017). "Enhancing an Interactive Recommendation System with Review-based Information Filtering". In P. Brusilovsky, M. de Gemmis, A. Felfernig, P. Lops, J. O'Donovan, N. Tintarev, & C. M. Willemsen (Eds.), *IntRS 2017: Interfaces and Human Decision Making for Recommender Systems : Proceedings of the 4th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems co-located with ACM Conference on Recommender Systems (RecSys 2017)* (Vol. 1884, pp. 2–9).

Fournier-Viger, P. (2018). "An Open-Source Data Mining Library Documentation".

Fournier-Viger, P., A. Gomariz, T. Gueniche, A. Soltani, C.-W. Wu, and V. S. Tseng. (2014). "SPMF: A Java Open-source Pattern Mining Library". *J. Mach. Learn. Res.*, *15*(1), 3389–3393.

Francia, M., M. Golfarelli, and S. Rizzi. (2014). "A methodology for social BI". *International Database Engineering & Applications Symposium*, 207–216.

Furner, C., R. Zinko, and Z. Zhu. (2015). "The Influence of Information Overload on the Development of Trust and Purchase Intention Based on Online Product Reviews in a Mobile vs. Web Environment: A Research Proposal". *WHICEB 2015 Proceedings*.

Gallinucci, E., M. Golfarelli, and S. Rizzi. (2013). "Meta-stars: multidimensional modeling for social business intelligence". In *Proceedings of the sixteenth international workshop on Data warehousing and OLAP* (pp. 11–18).

Grahne, G., and J. Zhu. (2005). "Fast Algorithms for Frequent Itemset Mining Using FP-Trees". *IEEE Trans. on Knowl. and Data Eng.*, *17*(10), 1347–1362.

Grandi, U., A. Loreggia, F. Rossi, and V. A. Saraswat. (2014). "From Sentiment Analysis to Preference Aggregation". In *Proceedings of the 2014 International Symposium on Artificial Intelligence and Mathematics*.

Gregor, S., and A. R. Hevner. (2013). "Positioning and presenting design science research for maximum impact", *37*(2), 337-A336.

Han, J., J. Pei, and Y. Yin. (2000). "Mining Frequent Patterns Without Candidate Generation". In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (pp. 1–12). New York, NY, USA: ACM.

Hevner, A. (2007). "A Three Cycle View of Design Science Research". *Scandinavian Journal of Information Systems*, *19*.

Hevner, A. R., S. T. March, J. Park, and S. Ram. (2004). "Design Science in Information Systems Research". *MIS Quarterly*, *28*(1), 75–105.

Hu, M., and B. Liu. (2004). "Mining and Summarizing Customer Reviews". In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 168–177). New York, NY, USA: ACM.

Huang, J., O. Etzioni, L. S. Zettlemoyer, K. Clark, and C. Lee. (2012). "RevMiner: an extractive interface for navigating reviews on a smartphone.". In R. Miller, H. Benko, & C. Latulipe (Eds.), *UIST* (pp. 3–12). ACM.

Iivari, J. (2015). "Distinguishing and contrasting two strategies for design science research". *European Journal of Information Systems*, *24*(1), 107–115.

Jiang, J. (2012). "Information Extraction from Text". In C. C. Aggarwal & C. Zhai (Eds.), *Mining Text Data* (pp. 11–41). Springer US.

Jones, D., and S. Gregor. (2007). "The Anatomy of a Design Theory". *Journal of the Association for Information Systems*, *8*(5), 1.

Karp, R. M. (1972). "Reducibility among Combinatorial Problems". In R. E. Miller, J. W. Thatcher, & J. D. Bohlinger (Eds.), *Complexity of Computer Computations* (pp. 85–103). Springer US.

Lee, T. Y., S. Li, and R. Wei. (2008). "Needs-Centric Searching and Ranking Based on Customer Reviews". In *IEEE Conference on E-Commerce Technology* (Vol. 2008, pp. 128–135).

Liu, B. (2012). "*Sentiment Analysis and Opinion Mining*". Morgan & Claypool.

March, S. T., and G. F. Smith. (1995). "Design and natural science research on information technology". *Decision Support Systems*, *15*(4), 251–266.

Meth, H., B. Mueller, and A. Maedche. (2015). "Designing a Requirement Mining System". *Journal of the Association for Information Systems*, *16*(9).

Mukherjee, S., and S. Joshi. (2013). "Sentiment Aggregation using ConceptNet Ontology". In *International Joint Conference on Natural Language Processing* (pp. 570–578). Nagoya, Japan.

Mukherjee, S., and S. Joshi. (2014). "Author-Specific Sentiment Aggregation for Polarity Prediction of Reviews". In *Proceedings of the Ninth International Conference on Language Resources and Evaluation* (pp. 3092–3099). Reykjavik, Iceland: ELRA.

Nunamaker, J. F., M. Chen, and T. D. M. Purdin. (1990). "Systems Development in Information Systems Research". *Journal of Management Information Systems*, *7*(3), 89–106.

Parameswaran, M., and A. B. Whinston. (2007). "Research Issues in Social computing". *Journal of the Association for Information Systems*, *8*(6), Article 22, pp. 336-350.

Park, D.-H., J. Lee, and I. Han. (2006). "Information Overload and its Consequences in the Context of Online Consumer Reviews". *PACIS 2006 Proceedings*.

Pasquier, N., Y. Bastide, R. Taouil, and L. Lakhal. (1999). "Discovering Frequent Closed Itemsets for Association Rules". In *Database Theory — ICDT'99* (pp. 398–416). Springer, Berlin, Heidelberg.

Peffers, K., T. Tuunanen, M. Rothenberger, and S. Chatterjee. (2007). "A Design Science Research Methodology for Information Systems Research". *J. Manage. Inf. Syst.*, *24*(3), 45–77.

Popescu, A.-M., and O. Etzioni. (2005). "Extracting Product Features and Opinions from Reviews". In *Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (Vol. 2005, pp. 339–346).

Quan, C., and F. Ren. (2014). "Unsupervised Product Feature Extraction for Feature-oriented Opinion Determination". *Inf. Sci.*, *272*(C), 16–28.

Schmid, H. (1994). "Probabilistic Part-of-Speech Tagging Using Decision Trees". In *International Conference on New Methods in Language Processing* (pp. 44–49). Manchester, UK.

Takeda, H., P. Veerkamp, and H. Yoshikawa. (1990). "Modeling Design Process". *AI Magazine*, *11*(4), 37.

Vaishnavi, V. K., and W. Kuechler. (2015). "*Design Science Research Methods and Patterns: Innovating Information and Communication Technology*".

Wang, T., Y. Cai, H. Leung, R. Y. K. Lau, Q. Li, and H. Min. (2014). "Product aspect extraction supervised with online domain knowledge". *Knowledge-Based Systems*, *71*, 86–100.

Wei, C.-P., Y.-M. Chen, C.-S. Yang, and C. C. Yang. (2009). "Understanding what concerns consumers: a semantic approach to product feature extraction from consumer reviews". *Information Systems and e-Business Management*, *8*(2), 149–167.

Yang, L., B. Liu, H. Lin, and Y. Lin. (2016). "Combining Local and Global Information for Product Feature Extraction in Opinion Documents". *Inf. Process. Lett.*, *116*(10), 623–627.

Yi, C., Z. (Jack) Jiang, and I. Benbasat. (2017). "Designing for Diagnosticity and Serendipity: An Investigation of Social Product-Search Mechanisms". *Information Systems Research*, *28*(2), 413–429.

Ziani, B., and Y. Ouinten. (2009). "Mining maximal frequent itemsets: A java implementation of FPMAX algorithm". In *2009 International Conference on Innovations in Information Technology (IIT)* (pp. 330–334).