Latent Customer Needs Elicitation by Use Case Analogical Reasoning From Sentiment Analysis of Online Product Reviews

Different from explicit customer needs that can be identified directly by analyzing raw data from the customers, latent customer needs are often implied in the semantics of use cases underlying customer needs information. Due to difficulties in understanding semantic implications associated with use cases, typical text mining-based methods can hardly identify latent customer needs, as opposite to keywords mining for explicit customer needs. This paper proposes a two-layer model for latent customer needs elicitation through use case reasoning. The first layer emphasizes sentiment analysis, aiming to identify explicit customer needs based on the product attributes and ordinary use cases extracted from online product reviews. Fuzzy support vector machines (SVMs) are developed to build sentiment prediction models based on a list of affective lexicons. The second layer is geared toward use case analogical reasoning, to identify implicit characteristics of latent customer needs by reasoning the semantic similarities and differences analogically between the ordinary and extraordinary use cases. Case-based reasoning (CBR) is utilized to perform case retrieval and case adaptation. A case study of Kindle Fire HD 7 in, tablet is developed to illustrate the potential and feasibility of the proposed method. [DOI: 10.1115/1.4030159]

1 Introduction

The success of a product or a service is largely dependent on to what extent the product or the service satisfies customer needs. In addition to explicit customer needs, latent customer needs are particularly critical to product innovation and success. Elicitation and understanding of latent customer needs have significant impacts on delighting customers or disgusting them. General approaches to customer needs elicitation involve such steps as (1) gathering raw data from customers; (2) interpreting collected data into customer needs; (3) organizing the customer needs into a hierarchy of primary, secondary, and if necessary tertiary needs; (4) prioritizing the customer needs with relative importance; and (5) reflecting on the results and the process [1]. There are various methods and procedures available for gathering raw data, such as interviews, focus groups, and observations [2]. Different methods often result in different amounts of time with varied effectiveness. While explicit customer needs tend to be self-explanatory as long as we can identify appropriate textual keywords in raw data, latent customer needs are often implied in the semantics of customer needs information. It is challenging to reveal latent customer needs underlying explicit customer needs and understand the semantic implications associated with use cases. The difficulties in latent customer needs elicitation can be observed as follows.

1.1 Technical Challenge

1.1.1 Multichannel Data Collection. First, different types of customers need to be identified for data collection, including lead users—customers who push a product to its limits, experience needs prior to the general population, and benefit significantly from having those needs fulfilled [3]. Nevertheless, it is hard to identify a large number of lead users [4]. Many latent needs might not even be expressed in language, but rather behaviors, displays, and physiological data [5]. Therefore, it may require sophisticated sensor networks to identify behavioral and physiological data for such latent customer needs, which is often costly and time-consuming.

1.1.2 Linguistic Analysis of Customer Needs. Customer needs are often expressed in linguistic terms, which tend to be abstract, fuzzy, and conceptual [6]. Although much effort has been devoted to leveraging tools and technologies (e.g., text mining) to enhance latent customer needs elicitation, still there is much space for improvement, especially the deep semantics underlying textual data, which can only be reasoned and inferred. Furthermore, it is difficult to create a high-quality information channel that runs directly among customers, marketing folks, and product designers. People in different domains express the needs in different set of contexts and differences in semantics and terminology always impair the ability to convey needs information effectively from customers to marketing folks and to designers [7].

1.1.3 Unawareness of Latent Customer Needs. Explicit customer needs can be easily articulated, leading to incremental changes in a product. However, latent needs may be nonobvious and very difficult to identify [8], and sometimes customers may not be consciously aware of them but are unexpectedly delighted if they are satisfied or disgusted if they are not satisfied [9]. For example, emotional needs are considered as latent customer needs. This is because emotional needs tend to be imprecise and ambiguous [10]. Traditionally, empathic design methods for examining the customer in the natural environment can identify latent needs, such as customer observation [11]. However, the

---

Roger Jianxin Jiao
The George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, 801 Ferst Drive, Atlanta, GA 30332
e-mail: rjiao@gatech.edu

Julie S. Linsey
The George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, 801 Ferst Drive, Atlanta, GA 30332
e-mail: julie.linsey@me.gatech.edu

Feng Zhou
The George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, 801 Ferst Drive, Atlanta, GA 30332
e-mail: fzhou35@gatech.edu

1Corresponding author.
Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received August 28, 2014; final manuscript received February 4, 2015; published online May 19, 2015. Assoc. Editor: Wei Chen.
observation data may be biased based on the interpretation of design engineers and analyzing the data is time-consuming. Another important method is to involve customers in the design process actively [12]. Despite many advantages and innovations of customer codesign, it may also be time-consuming and costly to identify, recruit, and understand the right customers and their roles.

1.2 Latent Customer Needs Elicitation Through Use Case Reasoning. Due to their implicit semantics in nature, latent customer needs can hardly be identified directly from explicit customer needs, but rather must be discovered by inferring the use cases articulated from analysis of explicit customer needs. We propose a two-layer approach to formulate latent customer needs elicitation by use case analogical reasoning from sentiment analysis of customer need information (see Fig. 1). Product attributes and use cases are first extracted from raw data about customer needs. Then, the first layer, i.e., sentiment analysis layer, aims to elicit explicit customer needs by mining customer preference information regarding ordinary use cases. The second layer, i.e., use case analogical reasoning layer, implies latent customer needs by adapting explicit customer needs based on the semantic similarities and differences between ordinary and extraordinary use cases.

1.2.1 Sentiment Analysis. It is the computational study of opinions, sentiments, and emotions expressed in online texts [13], and three important steps are needed, including (1) what product attributes are evaluated, i.e., attribute extraction; (2) what is the evaluation polarity, i.e., sentiment prediction; and (3) aggregating sentiments over each product attribute. For example, I do not like the charger of my tablet, which expresses a negative opinion on the charger of the tablet. Of all the customer reviews, the number of positive opinions and the number of negative opinions indicate the trend of customer preferences about this attribute. For online product reviews, the majority of them are expressed with emotional tones, making it easy to identify customer likes and dislikes. Despite the fact that the quality of the review data varies, many reviews have detailed opinionated information about specific product attributes in terms of their quality, usability, aesthetics, etc. Therefore, sentiment analysis can be conducted with regard to specific product attributes rather than the general product, e.g., Refs. [14-16]. Such links between the opinionated information and product attributes give direct clues for eliciting explicit customer needs.

1.2.2 Use Case Analogical Reasoning. Another kind of important information that sentiment analysis can extract is different use cases from online product reviews. Compared with ordinary use cases, extraordinary ones tend to create a context to elicit latent customer needs. This is consistent with the key idea in Ref. [4], in which extraordinary use cases are created to break the mold of users’ thought process and usage pattern, thereby encouraging them to interact with the product in innovative ways and to articulate latent needs that lead to breakthrough products. In this sense, the users interacting with the product in extraordinary use cases become empathic extraordinary users [17]. Similarly, Chen et al. [18] proposed the usage of context-based design, in which product performance varies significantly under different use cases. This greatly affects customer preferences and choices. In this paper, we propose to apply CBR for use case analogical reasoning to elicit latent needs. CBR is the process of solving new problems based on the solutions of similar past problems and is one of the most powerful methods for analogy reasoning [19,20]. In this paper, explicit customer needs are elicited for ordinary use cases that are mined through sentiment analysis of online product reviews. Whenever an extraordinary case is identified, the most similar case in the database will be retrieved. The elicited customer needs for the retrieved case are reused to discover latent customer needs for extraordinary use cases by semantic case adaptation.

As a summary, the major contribution of this paper is to elicit latent customer needs, which can be elaborated at two levels. (a) At the fundamental level, we emphasize the importance of latent customer needs and identify three different kinds of latent customer needs, focusing on their similarities and differences. Such understanding offers guidance about how to elicit latent customer needs elicitation. (b) At the methodological level, we propose use case analogical reasoning from sentiment analysis for latent customer needs elicitation. The proposed method balances and trade-offs supervised and unsupervised data mining techniques. For one thing, sentiment analysis of online product reviews combines both text mining and designers’ scrutiny to facilitate and improve the processes of data collection and linguistic analysis of explicit customer needs. For another, use case analogical reasoning with CBR discovers latent needs using case adaptation by understanding the semantic similarities and differences between ordinary use cases and extraordinary use cases.

2 Related Work

2.1 Product Attribute Extraction. How customers use the product and what attributes they interact with indicate what customers like and dislike [21]. Recently, product attribute extraction has been one of the tasks of sentiment analysis of online product reviews. For example, Ghani et al. [22] proposed a text-mining
method to extract attribute-level pairs from textual product descriptions online. Putthividhya and Hu [23] combined supervised named entity recognition with bootstrapping to identify product attributes with a high precision of 90.33%. These two methods, however, are supervised and semisupervised, which need a laborious manual labeling process for training. Hu and Liu [16,24] applied association rule mining (ARM) to extract frequent items in the product reviews as candidates for product attributes. The advantage of this method is that it is unsupervised and thus no training process is needed. However, the number of features discovered for each product was over 100 with plenty of redundancy, despite the fact that feature pruning and compactness pruning were applied. For example, image quality and photo quality are often considered as two product attributes while semantically they are the same. Raju et al. [25] proposed an unsupervised approach to extract product attributes from† with 92% of precision but only with 62% of recall. The low recall is due to the fact that nouns and noun phrases are always considered as product attribute candidates.

From the perspective of design, more information about the extracted product attributes is needed. For example, Wassenaar et al. [26] mapped customer desires to design attributes related to engineering analyses, based on which a discrete choice demand model was proposed to assess product profits [27]. Tucker and Kim [28] proposed a statistical trend detection technique to classify them as standard, nonstandard, or obsolete based on customer preferences. Archak et al. [29] emphasized the weight customers placed on each product attribute and how these weights affect the product demand. Rai [30] proposed a text-mining method to partition online customer reviews into individual product attributes, and three important measures were presented to rank identified attributes. Stone and Choi [31,32] extracted consumer preferences from user-generated content and studied validation and uncertainty analysis, respectively. However, automatic attribute level extraction is still an open question, because many data mining algorithms cannot deal with unstructured textual data expressed by customers. In this paper, we propose an unsupervised method, i.e., ARM, to extract product attributes, based on which similarity measures based on WordNet [33] are used to refine extracted attributes by reducing attribute redundancy. Such an automated fashion greatly shortens the traditional customer needs elicitation process. Attribute-level pairs are identified automatically but are corrected by the designer’s scrutiny. Despite the human involvement in this step, the major difference from the previous research is that we concentrate on latent customer needs elicitation rather than product attributes extraction or explicit customer preferences, which is also one of the primary contributions of this paper.

2.2 Customer Preference and Needs Elicitation. Based on the extracted attributes, the next step is to summarize customer preferences and needs. Many traditional methods, such as interviews, focus groups, questionnaires, self-reports, and observations [2,12,34], have been proposed. Chen et al. [21] proposed analytic techniques, especially different discrete choice models to capture heterogeneous customer preferences so as to predict customer choices. Recently, sentiment analysis can be used to elicit customer preferences from online product reviews. It can classify product reviews into positive or negative opinions at the document level or at the sentence level [13]. Both supervised and unsupervised methods have been proposed. One of the unsupervised methods is to apply affective lexicons, which make use of semantic features (polarity tags and semantic orientations) in product reviews. For example, Hu and Liu [24] formed a list of lexicon seeds with known polarities, and then were expanded using WordNet [33] through synonym and antonym links to predict review orientation. Ding et al. [35] improved Hu and Liu’s method by including linguistic rules. Titov and McDonald [36] proposed a multi-aspect sentiment model, in which latent Dirichlet allocation was used to build topics representative of product attributes. These methods were domain independent and unsupervised, which made the whole system easy to implement and maintain. However, compared with supervised learning methods, the prediction accuracies from the above-mentioned methods can be limited, since there is no training process involved.

For supervised methods, Jin et al. [14] integrated multiple linguistic features, including part-of-speech tags, tag patterns, and lexicons under a frame of hidden Markov models, which were used to extract product attributes and classify sentiments at the same time with good performance. Their computation-intensive training process was alleviated by a bootstrapping process. Chen et al. [15] proposed a model based on conditional random fields with similar linguistic features, and their model was able to outperform that of Jin et al. However, one main limitation of these supervised methods is that they are not product-independent and retraining models for other products is expensive [37]. In this paper, we combine both an unsupervised method, i.e., affective lexicons and a supervised method, i.e., fuzzy SVMs, for sentiment prediction. First, affective lexicons are domain-independent and can be applied to many online reviews for different types of products. In order to further improve the prediction accuracy with supervised methods, a perfect candidate of binary classifiers is SVMs, which excel at separating categories by a clear gap that is as wide as possible. SVMs have been widely applied and proved to be effective in different areas in pattern recognition [38].

2.3 User-Generated Online Product Reviews. Online customer reviews have an important role for retailers, customers, and designers. These user-generated product reviews describe product performance in terms of different product attributes in different use situations from various users’ perspectives [39]. Such use cases provide a specific channel for customer needs elicitation. However, the current literature mainly focuses on function on product choice decision making and predicting product demands. For example, strong positive correlations have been found between positive ratings and growth of product sales [40]. Similarly, Ghose and Ipeirotis [41] analyzed the review features, such as subjectivity, informativeness, and readability on product sales and their perceived usefulness. Miao et al. [42] applied sentiment analysis of online product reviews to generate a ranking mechanism with temporal opinion quality and relevance to facilitate product choice decision making for customers. Archak et al. [29] incorporated textual content of product reviews to learn customer preferences and predict consumers’ product choices and demands. As pointed out by Lee [43], however, prior analysis of online product reviews appears to have overlooked the role in customer needs elicitation. These online reviews can be a new approach for assessing rapidly changing customer needs.

3 Latent Customer Needs

Many researchers, e.g., Refs. [8] and [9], agree that, unlike explicit customer needs, latent ones are hard to articulate, because they are not obvious or customers may not even be aware of them. However, in the current literature, no clear understanding and common consensus can be achieved. We summarize three fundamental categories of latent customer needs.

3.1 Unexpected Delighter. From the degree of customer satisfaction point of view, latent customer needs yield paramount satisfaction. This is consistent with the attractive quality attributes in the Kano model [44]. When these needs are not addressed, it will result in dissatisfaction. However, when they are fulfilled, customers are delighted unexpectedly as customers are often unaware of them. Nevertheless, these latent needs will become basic needs over time. One good example is probably the vanity mirror mounted on the sun visor in the car. Such a mirror often delighted customers.
women passengers when it was first introduced, and then as time went by, it became a basic product attribute in the car.

In order to obtain such latent needs, empathic design is often used, that is, to observe and capture customers' interactions with products in their own environment, especially subtle ones, such as body languages and facial expressions [45]. Deszca et al. [46] argued that users in empathic design are almost as involved in product design as designers and engineers. Therefore, another possible method to elicit latent customer needs is customer code-sign [12], in which customers are actively involved in the product design process with design engineers.

3.2 Lead Users' Needs. Lead users are capable of eliciting latent needs. They experience needs that will be general in a marketplace, but face them months or years before the bulk of that marketplace encounters them; and benefit significantly from having those needs fulfilled [3]. Hence, from the perspective of lead users, these needs are latent with regard to the nonlead users. This is possible according to Rogers [47], who showed that new products typically diffuse through a society over a period of time, and those of the very first individuals who adopted a product were called innovators. In this sense, lead users are consistent with the innovators in the innovation diffusion process within a social network.

Lead users' needs foretell the general demand of a new product, and lead users adopt a product far in advance of the general market. They tend to redesign the existing products to satisfy their needs. The greater the benefit a given user can obtain from the redesign process, the greater his effort to obtain a solution will be [3]. Hence, latent needs elicitation can be enabled by interviewing lead users. However, it is difficult to obtain such lead users.

3.3 Extraordinary Users' Needs. Another kind of latent needs is experienced under certain extraordinary circumstances. Users under such cases are called extraordinary users. They are designed to break the mold of the ordinary users’ thought process and usage pattern, thus interacting with the product in radically new ways to articulate latent needs [4]. For example, by putting users in situations so that they are hard to see or hear, the use case makes users interact with products in extraordinary ways [17]. In this sense, these situationally disabled users become extraordinary users.

Lin and Seepersad [4] proposed a similar idea to transform ordinary users into empathic lead users by creating extraordinary use cases (e.g., mitts on users’ hands in the dark to simulate dunk on a cold day) for tent assembly. Experimental results from empathic lead user interviews significantly increased the effectiveness in latent customer needs discovery. Chen et al. [18] proposed usage context-based design, and by changing the use context, latent customer needs were uncovered to show their decision making process. In this paper, we aim to elicit such latent customer needs. First, the proposed method identifies explicit customer needs by sentiment analysis, based on which CBR is applied for use case analogical reasoning to discover latent customer needs by reasoning the semantic similarities and differences between ordinary use cases and extraordinary use cases.

4 Use Case Analogical Reasoning From Sentiment Analysis

Given the online product reviews, we decompose the latent customer needs elicitation into a two-layer problem, including how to elicit explicit customer needs and discover latent customer needs. However, both explicit and latent customer needs are based on identification of product attributes and use cases. Thus, we formulate the problem of latent customer needs elicitation from online product reviews as follows:

Given a set of review web pages, How to identify product attributes, attribute levels, and use cases,

How to elicit explicit customer needs, How to discover latent customer needs,

Subject to constraints of stakeholders and company goals.

In order to solve this problem, we propose use case analogical reasoning from sentiment analysis of online product reviews. Figure 2 shows the steps of the proposed method, including data collection, attribute/case identification, sentiment prediction, and use case analogical reasoning.

1) The first step is data collection, in which the input is a set of review web pages from1, and the output is a file of product reviews with customer ratings (see Fig. 3). Specifically, PYTHON 2.72 is used to crawl online reviews and removes meaningless symbols (e.g., HTML symbols). The cleaned reviews are segmented into sentences and these sentences are labeled with either positive or negative with the help of user-provided ratings. This process is automated by PYTHON with a natural language toolkit3, including data collection, tokenization, part of speech tagging, and sentence segmentation.

2) In the second step, ARM is first used to extract a set of product attributes that customers reviewed [16], and then a similarity-matching algorithm is used to generate a set of refined product attributes. The same process is also conducted to identify refined use cases (see Sec. 8.3). This step is also automated, except the predefined product attributes and interaction elements in the use cases.

3) In the third step, a model based on fuzzy SVMs is used to predict sentiments of product reviews with regard to individual product attributes. First, we construct a list of affective lexicons based on ANEW (affective norms for English words) [48], and then WordNet [33] is used to expand the sentiment lexicon seeds based on a standard label propagation algorithm [37]. Second, the fuzzy SVM model with different kernel functions is built to predict sentiments of product reviews. Third, customer preference information about their attribute levels can be summarized based on sentiment orientations of product reviews. This step is also automated because the training labels in the SVM algorithm are provided by the user ratings of the reviews.

4) In the last step, CBR is used for analogical reasoning between ordinary and extraordinary use cases in order to elicit latent customer needs. First, customer opinions are summarized to translate customer opinions into explicit customer needs for ordinary use cases. Then, ordinary use cases are reused and customized to elicit latent customer needs with CBR for analogical reasoning in an automated way. However, design engineers are involved to scrutinize elicited explicit and latent customer needs for the purpose of consistency.

5 Case Example

In order to illustrate the proposed method, a Kindle Fire HD 7 in. tablet (released on September 14, 2012) is used as a case example. The unstructured review data are collected from1 from October 2, 2012 to November 20, 2013 with user-provided ratings (from 1 star to 5 stars). Figure 3 shows a typical review about this product and we can see its rating (5 stars), the product he/she commented, and the review content. These reviews are then segmented into sentences. After attribute extraction, the sentences that do not contain product attributes are discarded. If a product attribute is described in multiple sentences, only the first sentence is kept for the purpose of normalization. For those sentences that contain product attributes, we extract the data features (see Sec. 6.2.1), based on which SVM models are used to predict polarity of the sentence that describes the product attribute(s). The ground truth is obtained by the customer ratings provided, i.e., those with ratings of 1 star or 2 stars are considered as negative, and those

1www.python.org 2www.nltk.org
with ratings of 4 stars or 5 stars are considered as positive. Those with ratings of 3 stars account for 20.6% of all the reviews and are discarded. This is because these reviews tend to have mixed opinions for the product under evaluation, which complicates the labeling process. However, if a majority of reviews have a star of 3, a manual process may be needed to label for them the labeling process. A total number of 4475 opinionated sentences that contain product attributes were generated from randomly selected reviews that are longer than 5 sentences.

6 Sentiment Prediction

6.1 Lexicon Construction and Propagation

A list of sentiment lexicons is built on ANEW and WordNet. The words in ANEW are rated against three dimensions between 1 and 9, including valence, arousal, and dominance. All the 1033 words are regarded as seed lexicons and are then expanded through synonym and antonym links in WordNet [33]. In order to overcome ambiguities raised by polysemy, the expanded list in WordNet is accompanied with simple part-of-speech tags, including noun (n), verb (v), adjective (a), and adverb (r). Hence, not only a list of synonyms and antonyms links in WordNet [33]. In order to overcome ambiguities raised by polysemy, the expanded list in WordNet is accompanied with simple part-of-speech tags, including noun (n), verb (v), adjective (a), and adverb (r). Hence, not only a list of

According to Blair-Goldensohn et al. [37], we set $m = 5$ and $\lambda = 0.2$, since larger values of $m$ do not improve performance and larger values of $\lambda$ lead to a too skewed distribution of the scores. Finally, $s_{m}^{n}$ is also normalized as follows:

$$s_{m}^{n} = s_{m}^{n} / \text{abs}(\text{max}(s_{m}^{n})) * \text{abs}(s_{m})$$

where abs() is a function to obtain its absolute value. For example, the 750th word is “radiant” in ANEW, $s_{750}$ = {“beaming.a,” “beamy.a,” “bright.a,” “dull.a”}, and beaming.a and beamy.a are synonymous with each other. Then, the final score is $s_{750}^{50, 2} = [0.496, 0.299, 0.299, 0.224, -0.224]^{T}$ for valence. Of all the scores we obtained, most of them are consistent with affective semantics if not all.

6.2 Prediction Based on Fuzzy SVMs

6.2.1 Data Feature Extraction. Since the lexicon list cannot cover all the English words, a backup opinion lexicon list is provided with 2006 positive words and 4783 negative words [24]. Simple valence values are assigned to them (i.e., 0.5 to all the positive words). Besides, it includes useful properties of online texts, such as spellings, morphological variants, slang, and social media markups. We choose ten features from a sentence by a sequential feature selection method among 14 proposed features, described as follows:

- (1) number of words with positive valence, i.e., $N_{pos}$
- (2) number of words with negative valence, i.e., $N_{neg}$
- (3) the average of valence, arousal, and dominance, i.e., $V_{ave}, A_{ave}, D_{ave}$
- (4) the maximum valence and corresponding arousal and dominance, i.e., $V_{max}, A, D$, if $N_{pos} > N_{neg}$; the minimum valence and corresponding arousal and dominance, i.e., $V_{min}, A, D$, if $N_{pos} \leq N_{neg}$
- (5) number of negation words, i.e., $N_{neg}$
- (6) number of words that denote adversative relations, i.e., $N_{a}$

The features in (1) and (2) are extracted by string comparison between the target sentence and the affective lexicon lists with PYTHON, while the features in (3) and (4) are obtained from ANEW and the lexicon propagation algorithm described in Sec. 6.1. The features in (5) and (6) are also obtained by string comparison between the target sentence and a predefined list of negation words (e.g., no, never, none) and a predefined list of adversative words (e.g., but, nevertheless), respectively. The last feature is useful when the sentence has a mixed sentiment. For example,
“the screen of the tablet is wonderful, but the battery charger sucks.” In such a case, we will further divide the sentence into two subsentences automatically whenever we detect words that indicate an adversative relation (i.e., but), i.e., “the screen of the tablet is wonderful” and “the battery charger sucks.” This helps accurately predict customer preferences for individual product attributes.

### 6.2.2 SVM Model Construction and Prediction Results

In this paper, we apply fuzzy SVMs [49] with different kernel functions to predict sentiment, including linear kernels, the radial basis function (RBF) kernel, polynomial kernels, as well as orthogonal polynomial kernels [38]. We utilize fuzzy SVMs coded in MATLAB to predict sentiments for each product attribute in individual sentences. Specifically, a tenfold cross-validation strategy is used to tune the parameters, in which a minimization procedure is used with the optimization tool in MATLAB. In order to avoid local minima, ten different initial values are randomly generated for the minimization process to obtain the possible global minima.

A tenfold cross-validation method is also adopted for sentiment prediction, in which we make the numbers of sentences in both the negative sentiment and the positive sentiment equal to obtain unbiased results. Precision, recall, and F-scores, see Ref. [50], are reported in Table 1, respectively. The bold numbers represent the best result. First, the average values of precision, recall, and F-scores are 72.6%, 73.6%, and 73.2%, respectively, indicating the proposed method gains a medium accuracy. Among all SVMs, the one with the four-item Hermite kernel function outperforms other SVMs in terms F-score, which is the harmonic mean of precision and recall. Note that the SVM model with the RBF kernel function also achieves comparable results with those of orthogonal polynomial kernels.

<table>
<thead>
<tr>
<th>Kernel function</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
<th>F-score (%)</th>
<th>#SV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear kernel</td>
<td>71.6</td>
<td>71.7</td>
<td>71.7</td>
<td>1137</td>
</tr>
<tr>
<td>RBF kernel</td>
<td>73.6</td>
<td>73.7</td>
<td>73.7</td>
<td>1134</td>
</tr>
<tr>
<td>Polynomial kernel ($d = 4$)</td>
<td>69.4</td>
<td>69.4</td>
<td>69.4</td>
<td>513</td>
</tr>
<tr>
<td>Four-item Chebyshev</td>
<td>73.0</td>
<td>74.7</td>
<td>73.8</td>
<td>637</td>
</tr>
<tr>
<td>Four-item Legendre</td>
<td>73.9</td>
<td>75.4</td>
<td>74.7</td>
<td>648</td>
</tr>
<tr>
<td>Four-item Laguerre</td>
<td>71.6</td>
<td>75.8</td>
<td>73.7</td>
<td>664</td>
</tr>
<tr>
<td>Four-item Hermite</td>
<td><strong>75.1</strong></td>
<td><strong>75.2</strong></td>
<td><strong>75.1</strong></td>
<td>663</td>
</tr>
<tr>
<td>Average</td>
<td>72.6</td>
<td>73.7</td>
<td>73.2</td>
<td>—</td>
</tr>
</tbody>
</table>

#### 7.2 Attribute Refinement by Similarity Matching
Five measures of semantic relatedness between two words, $v_i$ and $v_j$, in Python that can make use of WordNet⁵ are selected for similarity matching, including Hirst–St-Onge similarity, Leacock–Chodorow similarity, Resnik similarity, Jiang–Conrath similarity, and Lin similarity [52]. Taking two words, “picture” and “photo,” as an example, we can calculate the similarity between them using the above-mentioned five measures, and the results are 0.333, 2.539, 5.562, 0.246, and 0.733, respectively. Since they are not in the same scale, each is normalized between 0 and 1 by dividing its maximum similarity scores. For attributes with more than one word, we use $p_a = [u_1, \ldots, u_m]$ to denote a term attribute in the user predefined set, and $f_a = [v_1, \ldots, v_m]$ to denote a term attribute in frequent attributes mined through ARM. The similarity between them is calculated as the word average similarity between them [51], i.e.,

$$\text{ave}(p_a, f_a) = \frac{1}{m} \sum_{i=1}^{m} \max_i (ws(u_i, v_i)) + \sum_{i=1}^{m} \max_i (ws(u_i, v_i))/2,$$

where $\max_i (ws(u_i, v_i))$ is the maximum word similarity measure between $u_i$ and the words in $f_a$, and the five word similarity measures can be applied here.

First, in order to evaluate five similarity measures, a redundancy reduction measure is adopted as follows [51]:

$$\text{RedundancyReduction} = \frac{N_{mfa} - N_{nfa}}{N_{fa}}$$

where $N_{mfa}$ is the number of matched attributes in the frequent attributes discovered by ARM, $N_{fa}$ is the number of frequent attributes, and $N_{nfa}$ is the number of matched attributes in the user predefined set. Note that the attributes that have exactly the same words and/or terms do not count in the calculation of redundancy reduction. Through ARM, we identify 116 product attributes and attribute levels for the Kindle Fire HD 7-in. model. Table 3 shows the results of redundancy reduction with different empirical thresholds. Here, the threshold is defined as a function of the standard deviation of similarity scores. Note that this measure penalizes itself by increasing the value of $N_{nfa}$ when a low

---

⁵www.mathworks.com/products/optimization

⁶www.nltk.org/howto/wordnet.html

---

**Table 1** Sentiment prediction results of the 7 in. model Kindle Fire HD

<table>
<thead>
<tr>
<th>Kernel function</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
<th>F-score (%)</th>
<th>#SV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear kernel</td>
<td>71.6</td>
<td>71.7</td>
<td>71.7</td>
<td>1137</td>
</tr>
<tr>
<td>RBF kernel</td>
<td>73.6</td>
<td>73.7</td>
<td>73.7</td>
<td>1134</td>
</tr>
<tr>
<td>Polynomial kernel ($d = 4$)</td>
<td>69.4</td>
<td>69.4</td>
<td>69.4</td>
<td>513</td>
</tr>
<tr>
<td>Four-item Chebyshev</td>
<td>73.0</td>
<td>74.7</td>
<td>73.8</td>
<td>637</td>
</tr>
<tr>
<td>Four-item Legendre</td>
<td>73.9</td>
<td>75.4</td>
<td>74.7</td>
<td>648</td>
</tr>
<tr>
<td>Four-item Laguerre</td>
<td>71.6</td>
<td>75.8</td>
<td>73.7</td>
<td>664</td>
</tr>
<tr>
<td>Four-item Hermite</td>
<td><strong>75.1</strong></td>
<td><strong>75.2</strong></td>
<td><strong>75.1</strong></td>
<td>663</td>
</tr>
<tr>
<td>Average</td>
<td>72.6</td>
<td>73.7</td>
<td>73.2</td>
<td>—</td>
</tr>
</tbody>
</table>

**Table 2** Confusion matrix of best performance by four-item Hermite SVM

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
<th>Total</th>
<th>Recall (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td>990</td>
<td>274</td>
<td>1264</td>
<td>78.3</td>
</tr>
<tr>
<td>Predicted</td>
<td>356</td>
<td>908</td>
<td>1264</td>
<td>71.8</td>
</tr>
<tr>
<td>Precision</td>
<td>73.6%</td>
<td>76.8%</td>
<td>75.1%</td>
<td>F-score = 75.1</td>
</tr>
</tbody>
</table>

**Table 3** Reduction of product attribute redundancy with different thresholds and classification accuracy

<table>
<thead>
<tr>
<th>Similarity</th>
<th>0.2 std</th>
<th>0.4 std</th>
<th>0.6 std</th>
<th>0.8 std</th>
<th>1 std</th>
<th>F-score (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hirst–St-Onge</td>
<td>0.218</td>
<td>0.245</td>
<td>0.303</td>
<td>0.337</td>
<td>0.342</td>
<td>68.5</td>
</tr>
<tr>
<td>Leacock–Chodorow</td>
<td>0.226</td>
<td>0.270</td>
<td>0.292</td>
<td>0.321</td>
<td>0.337</td>
<td>65.4</td>
</tr>
<tr>
<td>Resnik</td>
<td>0.224</td>
<td>0.279</td>
<td>0.352</td>
<td>0.359</td>
<td>0.363</td>
<td>71.2</td>
</tr>
<tr>
<td>Jiang–Conrath</td>
<td>0.279</td>
<td>0.287</td>
<td>0.344</td>
<td>0.371</td>
<td>0.372</td>
<td>70.8</td>
</tr>
<tr>
<td>Lin</td>
<td>0.273</td>
<td>0.315</td>
<td>0.362</td>
<td>0.372</td>
<td>0.410</td>
<td>75.8</td>
</tr>
</tbody>
</table>
threshold is used. Of all the five similarity measures, the similarity measures of Jiang–Conrath and Lin persistently perform better than other measures, and it seems that more redundancy reduction is obtained with higher thresholds.

Second, in order to evaluate the classification accuracy, we first manually pick the typical product attributes from the extracted attributes to form the seed categories. We selected three words or phrases in each category to describe the product attributes and attribute levels that span the semantic space as large as possible. Then, the five similarity measures with their respective optimal thresholds are used to classify product attributes based on the word’s semantic similarity. The ground truth is built on manual sorting with three graduate students in engineering design with a strategy of majority voting. This process also corrects any invalid attribute levels automatically identified. It generates 13 categories (see Fig. 4) with one general category, i.e., Kindle Fire HD 7 in. (KFHD 7 in.). Then, the average prediction accuracy in terms of F-score is also shown in Table 3.

8 Latent Customer Needs Reasoning

8.1 Compiling Customer Opinions on Product Attributes. Based on the sentiment analysis of individual product attributes, we are able to associate customer opinions with these product attributes in terms of their positive reviews versus negative reviews. First, based on the attribute extraction and refinement process, 13 major product attributes are identified in Fig. 4, including the product itself as an attribute, i.e., KFHD 7 in. First, more positive reviews were received than negative reviews on KFHD 7 in. Second, screen, video, audio, reader, connectivity, dimensions, and prices had more positive opinions than negative opinions, showing that generally customers were satisfied with these attributes, especially audio and price. Third, customers complained disappointedly of camera and battery, whereas their opinions about storage, interface, online service, and customer service were mixed.

In order to have a clear understanding of customer preferences, with the examination of the designers for correcting the results obtained by the algorithm, we further identified attribute-level pairs, as shown in Fig. 5. For example, storage has two levels, i.e., 16 GB and 32 GB. We summarize their percentages of positive reviews versus negative reviews within individual product attributes and their levels. Such information gives more insight into what customers like and dislike. For example, for storage, most customers complained of the 16 GB version rather than the 32 GB version, indicating that 16 GB is not sufficient for most of the customers. For another example, despite the fact that dimension received a large portion of positive reviews, there were still about 1/3 of the customers not satisfied with the size. Furthermore, based on the comment information, we can rank them in terms of their frequency, as shown in Fig. 6. By removing the general
The importance of these product attributes to a large extent [30]. The frequency of attribute levels is shown in Fig. 7, which shows the relative importance of levels within that product attribute.

8.2 Translating Customer Opinions Into Customer Needs for Ordinary Use Cases. We divide the use cases into three interaction elements, including user types, interaction environments, and activity contexts. Note case extraction and refinement are achieved similarly to that of product attributes extraction with ARM and similarity matching. In other words, interaction elements can be considered as attributes of use cases. We also propose predefined interaction elements involved in the use cases. For example, we predefined trip as one element in the contextual events, and those similar to the word trip, including travel, vacation, tour, outing, expedition, cruise, and voyage, and so on, are extracted by ARM and refined with the similarity-matching algorithm in Sec. 7.2. Finally, we extract use cases as illustrated in Fig. 8 based on the frequency commented on the review. Empirically, we set those above 30% of frequency as interaction elements of ordinary use cases, including typical adults, indoor with day light, and seated form the ordinary interaction use case. Other interaction elements with frequency below 30% are considered as those in extraordinary cases, as shown in red rectangular in Fig. 8.

The figure demonstrates that extraordinary use cases consist of relatively rarer interaction elements compared with ordinary use cases. For example, kids/students are a combined set of children, nephews, nieces, grandsons, granddaughters, sons, daughters, and kids mined from the reviews and they account for 26.5% of all the user types. We translate customer opinions into needs for the ordinary use cases based on which CBR is used to elicit latent customer needs for the extraordinary cases.

Sentiment analysis groups all the customer reviews about one product attribute or attribute level together in terms of positive and negative sentiments. This information facilitates transformation from customer opinions into customer needs. One advantage is that customer needs can be expressed as an attribute of the product, which ensures consistency and, in turn, facilitates subsequent translation into product specifications [1]. As an example, we show some customer reviews, and how they are translated into customer needs in Table 4 for the ordinary use case: typical adult, indoor with day light, and seated. From the positive customer reviews, the interpreted needs are expressed using positive phrasing with different levels of details that are as close to the raw data as possible. For example, the second review, the customer commented, “The kindle is easy to read and easy to use and see.” The algorithm first predicts that it is a positive review about the kindle (specifically, to read, to use, and see), which is further interpreted as the interface by the designer. Finally, the need is expressed as “The user interface is easy to use and see.” For the negative customer reviews, the opposite meaning will be the interpreted needs. As a rule of thumb, they are also expressed using positive phrasing with different levels of details. For example, the interpreted need for the last review, “I am disappointed that it has no back facing cam[era],” is “The tablet has a back facing camera.”

8.3 Use Case Analogical Reasoning. We propose CBR for use case analogical reasoning. It employs a hybrid reasoning method by combining case-based and rule-based reasoning for case understanding to elicit latent customer needs [19]. High-level customer needs are elicited first based on CBR and then a customized knowledge model compatible with rule-based reasoning is utilized to adapt an ordinary use case for an extraordinary use case. It has three major modules, including case database organization, case retrieval, and case adaptation.

8.3.1 Case Database Organization. The case database is denoted as \( C^C = (C^C_{1}, \ldots, C^C_{k}, \text{Ind}, R_{C}) \), where \( M_{CR} \) is the total number of refined cases for the time being, \( \text{Ind} = (\text{Activity}, \text{User}, \text{Ent}, \text{Evt}) \) is a case indexing model, and \( R_{C} \) is a domain customized knowledge model in terms of rules for case adaptation. Cases are organized based on the case indexing model, including use class Activity, user type User, interaction environment Ent, and contextual event Evt. Use cases shown in Fig. 8 can be constructed in terms of use case diagram using unified modeling language, resulting in corresponding cases in the case database. As an example, we construct an ordinary use case and an extraordinary use case in Fig. 9 as a form of case representation. In the extraordinary case, it describes that a boy (User) is surfing the Internet (Activity) on a trip (Evt), on the beach with strong sunlight (Ent). Whenever such a case is identified, we need to retrieve the most similar case to reuse the customer needs and adapt them with \( R_{C} \) to elicit latent customer needs.

8.3.2 Case Retrieval. Case retrieval is the process of finding prior solved cases that are closest to the current case. Here, solved cases mean that the customer needs are elicited for those cases. Given the extraordinary case \( C^E \), the retrieval process proceeds with the following pseudo algorithm in Fig. 10. It searches from the use class activity to check whether there is any case in the database that matches that of \( C^C \); if so, it will go further to check whether the case user type matches that of \( C^C \). It continues until it checks the interaction environment and the contextual event. If one case matches all the four variables, then the algorithm
calculates the similarity between $C_e$ and the retrieved case $C_{ijkl}$. If it only matches the first three variables, then the algorithm calculates the similarity between $C_e$ and $C_{ijk} / C_3$, where $C_{ijk} / C_3$ indicates a case with any contextual event. The same pattern applies to $C_{ij} / C_3 / C_3$ and $C_i / C_3 / C_3 / C_3$. The algorithm in Fig. 10 searches Activity first, which may miss the use cases with the same User, Ent, and Evt. Hence, we also include another three similar algorithms, which search User, Ent, and Evt first, respectively. When the similar cases are extracted, we need to determine which one is the most similar case. Hence, a similarity measure is designed. We assume that the four variables (i.e., Activity, User, Ent, Evt) of two cases $C_i$ and $C_j$ are associated with $m$ characteristics (e.g., age, gender for user type), i.e., $C_i = (c_{i1}, c_{i2}, \ldots, c_{im})$ and $C_j = (c_{j1}, c_{j2}, \ldots, c_{jm})$. If the $i$th characteristic is nominal, then $C_c(i, j) = 1$ when $c_{ii} = c_{jj}$ and 0 otherwise. If it assumes a real value or integer number, then

Fig. 8 Extracted use cases from online user-generated product reviews (those percentages show the frequency of interaction elements)

Table 4 Examples of how customer opinions are translated into customer needs for the ordinary use case

<table>
<thead>
<tr>
<th>Sentiment</th>
<th>Customer review</th>
<th>Interpreted needs</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>(1) It is so convenient that I can carry the kindle wherever I want in the house</td>
<td>The tablet is portable; the tablet has good connectivity</td>
<td>Dimension/connectivity</td>
</tr>
<tr>
<td></td>
<td>(2) The kindle is easy to read and easy to use and see</td>
<td>The user interface is easy to use and see</td>
<td>User interface</td>
</tr>
<tr>
<td></td>
<td>(3) I purchased the kindle for reading cheaper e-books. I love it</td>
<td>The kindle supports a variety of e-books with lower costs</td>
<td>Reader</td>
</tr>
<tr>
<td>Negative</td>
<td>(4) It is fully charged and after I use it for a while it is still with a good amount of battery left. Yet when I put it in sleep mode, it is dead in a few hours</td>
<td>The battery life is long enough</td>
<td>Battery</td>
</tr>
<tr>
<td></td>
<td>(5) The kindle is kind a heavy when I hold it for a long time, reading, watching movies, or playing games</td>
<td>The tablet has a support stand or is light enough to hold for reading, watching movies, or playing games for a long time</td>
<td>Dimension</td>
</tr>
<tr>
<td></td>
<td>(6) I am disappointed that it has no back facing cam[era]</td>
<td>The tablet has a back facing camera</td>
<td>Camera</td>
</tr>
</tbody>
</table>

Fig. 9 Case representation of (1) an ordinary use case and (2) an extraordinary use case

Fig. 10 Case retrieval pseudo algorithm
Then, the key differences between the ordinary use case and the extraordinary use case are highlighted. Therefore, the inferred latent customer needs for the given extraordinary case and then save it in the case database for the future use.

### 8.3.3 Case Adaptation

The domain customized knowledge model $R_d$ adapts new cases to elicit latent customer needs. This is implemented by integrating substitution and a rule-based method [19], including the following steps:

1. **Substitution**: It replaces invalid parts of the old use case with new content, according to key differences of a new case from the old one.
2. **Rule-based adaptation**: The system further refines the solution according to the customized knowledge model $R_d$.
3. **Evaluation**: The design engineer performs evaluation and feedback for improvement.
4. **Storage**: If the adaptation is successful, the new use case, along with adaptation knowledge, is stored for the future use. The customized knowledge model $R_d$ is also updated if necessary.

As an example, we will use the extraordinary use case in Fig. 9 as the new case, and the ordinary use case in Fig. 9 as the most similar case retrieved. The customer needs in Table 4 are translated into latent customer needs by case adaptation as shown in Table 5. First, the interpreted needs are usually assumed for the ordinary use case so that they are conditioned with braces (\{\}) (see the first column in Table 5) to facilitate case adaptation. Then, the key differences between the ordinary use case and the extraordinary use case are highlighted. Therefore, the inferred latent customer needs are obtained by substitution in braces (\{\}) (see the second column of Table 5). The customized knowledge model $R_d$ is simple and intuitive. Typical IF-THEN rules include:

- **IF** the user type is kids/students, **THEN** the need is conditioned by parental control or is children appropriate.
- **IF** the interaction environment is outdoor with sunlight, **THEN** the need is conditioned by outdoor with sunlight, and
- **IF** the contextual event is a trip, **THEN** the need is conditioned by trip characteristics (e.g., no WIFI connection, no power available).

According to these IF-THEN rules, the customized latent needs are shown in parentheses in the second column of Table 5. Finally, design engineers will examine the adaptation for refinement and improvement. If it is successful, the new case will be stored in the case database for the future use. In such a way, the case database will grow progressively. Finally, the customer needs will be summarized with regard to each product attribute.

### 9 Concluding Remarks

#### 9.1 Online User-Generated Product Reviews

Evidence has shown that these reviews have become an important information venue for customer needs elicitation. The case study in this paper extracts reviews for one product\(^1\), which gives us opportunities to elicit customer needs for this product. Liu et al. [53] pointed out that a long review covers user preferences, mentions many different product attributes, and includes likes and dislikes of the product. Hence, these long reviews tend to be more diagnostic than short ones in that they not only help designers in understanding and evaluating the quality and performance of products sold online but also lead to a profound understanding of product use in different cases that contribute to latent customer needs elicitation.

In our study, about 38.7% of the reviews comment the product itself without pointing out specific product attributes. Therefore, we exclude these reviews for customer needs elicitation though the preference information about the product shows the general popularity of the product (see Fig. 4). We assume that the product reviews cover all the possible product attributes. However, ARM only recovers those with frequent comments. Therefore, it is possible that product attributes with a very small number of comments are omitted in the study. However, on the one hand, these product attributes may be obsolete compared with popular ones [28]. On the other hand, they may be considered as must be attributes in Kano model that are taken for granted when fulfilled, but result in dissatisfaction when not fulfilled. In this study, we rank product attributes based on a simple frequency measure of their respective positive and negative opinions. Although simple, this measure is generally consistent with sophisticated importance measure, such as the review appearance rate measure and the local global normalization measure [30]. Therefore, it seems that many repetitive reviews of certain product attributes show their importance to a large extent.

#### 9.2 Sentiment Analysis

First, the proposed method makes use of both affective lexicons and fuzzy SVMs for sentiment analysis. The affective lexicons are constructed based on ANEW and WordNet. The seed list is rated against valence, arousal, and dominance, which have been proved that they are able to measure emotional reactions to stimuli in different contexts [54]. The seeds are then used to expand the affective lexicons using WordNet. Such a list of affective lexicons is domain independent and can be applied to predicting sentiments about reviews of different products.

### Table 5 Example of latent needs elicitation with case adaptation

<table>
<thead>
<tr>
<th>Interpreted needs</th>
<th>Latent needs</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) The tablet is portable</td>
<td>(1) The tablet is portable</td>
</tr>
<tr>
<td>(2) The tablet has good connectivity</td>
<td>(2) The tablet has good connectivity</td>
</tr>
<tr>
<td>(3) The user interface is easy to use and see</td>
<td>(3) The user interface is easy to use and see</td>
</tr>
<tr>
<td>(4) The kindle supports a variety of e-books with lower costs</td>
<td>(4a) The kindle supports a variety of e-books</td>
</tr>
<tr>
<td>(5) The battery life is long enough</td>
<td>(5) The battery life is long enough</td>
</tr>
<tr>
<td>(6) The tablet has a support stand or is light enough to hold</td>
<td>(6) The tablet has a support stand or is light enough to hold</td>
</tr>
<tr>
<td>(7) The tablet has a back facing camera</td>
<td>(7) The tablet takes picture easily</td>
</tr>
</tbody>
</table>

- $C(c_i, c_j) = |c_i - c_j|/(\text{max}_i - \text{min}_i)$, where $\text{max}_i$ and $\text{min}_i$ are the respective maximum and minimum values of the $i$th characteristic. Then, the similarity between $C_i$ and $C_j$ can be denoted as $S_{ij} = \sum_{k=1}^{m} C(c_i, c_j)/m$.

If no similar case is retrieved, we may have to manually elicit the latent customer needs for the given extraordinary case and then save it in the case database for the future use.

\[ C(c_i, c_j) = \frac{|c_i - c_j|}{(\text{max}_i - \text{min}_i)} \]

\[ S_{ij} = \frac{\sum_{k=1}^{m} C(c_i, c_j)}{m} \]
Second, in order to improve the prediction results, we propose fuzzy SVMs with different kernel functions. Using simple features extracted from affective lexicons, the model can achieve the best F-score of 75.1% with balanced recall and precision. As mentioned earlier, one limitation is that a fuzzy SVM is a supervised learning method, which needs a manual labeling process to create a ground truth for training and testing. We capitalize on the user-provided stars in reviews. Those with 1 or 2 stars are considered as negative, and those with 4 or 5 stars are considered as positive. However, this rating information is not always available and reliable. Even a review with 1 or 2 stars, it is still likely that some product attributes are positively reviewed, and vice versa. A possible way to alleviate this problem is to manually label a relatively small number of reviews, and then a bootstrapping strategy can be used to alleviate the laborious labeling and training process for the future work [15].

Third, based on the previous research [16], we extract product attributes using ARM. However, due to the redundancy among the extracted attributes, a similarity-matching method is proposed to refine the extracted product attributes. The shortcoming of this method is that attribute levels are still difficult to obtain or hierarchically organized without human involvement. Thus, how to differentiate product attributes and attribute levels in an automated way is one of the important tasks for customer needs elicitation for future work.

9.3 Latent Customer Needs Elicitation. Latent customer needs can lead to major innovations. Unfortunately, many traditional techniques in marketing research are unable to identify latent customer needs. As an alternative to traditional methods, we propose to make use of CBR for use case analogical reasoning that reuses explicit customer needs from ordinary use cases and adapts them with a domain customized knowledge model for extraordinary use cases. This is enabled by identifying semantic similarities and differences between ordinary use cases and extraordinary use cases. Consistent with Refs. [4], [17], and [18], this idea is inspired by transforming ordinary users into extraordinary users through changing their use cases, including user types, interaction environments, and contextual events. However, unlike these previous studies, our method is based on use case analogical reasoning without directly interviewing users. This is possible because the results from sentiment analysis provide us knowledge to build a case database for case reuse. Then, CBR retrieves the most similar case whenever an extraordinary use case is identified. The adaptation process is implemented by identifying reusable components in the source domain and analogical mapping with substitution semantically in the target domain. Such a process is further enhanced with a domain knowledge model in terms of IF-THEN rule reasoning. This process greatly improves the latent customer needs discovery process and reduces designers’ mental workload.

Unlike the unexpected delimiters in the Kano model, when latent customer needs are not met, they will not result in dissatisfaction. The latent customer needs elicited by the proposed method will delight the customers if they are met or disgust them if they are not. However, this is only happening in extraordinary use cases, such as kids reading in the sunlight on a trip. In ordinary use cases, customers may not even be aware of these needs. Unlike the lead users who are consistent with the innovators in the product diffusion process, the extraordinary users can be any adopters (e.g., late majority) in the product diffusion process. Nevertheless, they will only experience these needs in extraordinary use cases.

Finally, some precautions must be taken. First, the reuse behavior may be difficult when there are only a small number of cases in the database. However, CBR is capable of learning new cases and can progressively increase its database. To finalize latent customer needs, effective evaluation strategies are needed and may be influenced directly by reuse behavior and prior experience of expert designers. However, for inexperienced designers, this system offers an opportunity for them to learn requirements elicitation and analysis.

Acknowledgment

The authors would like to express their sincere gratitude to Editors, Dr. Wei Chen and Dr. Shapour Azarm and anonymous reviewers for providing constructive comments to help significantly enhance the quality of this paper.

Nomenclature

ant(\(w_k\)) = antonym set of \(w_k\)
\(C^g\) = the extraordinary case
\(C^r\) = case database
\(C^{gjk}, C^{gjk}, C^{g^*}\) = retrieved case with four, three, two, and one matched variables
\(M_R\) = the current total number of refined cases
\(N_{wd}\) = the number of words that denotes adverse relations in a sentence
\(N_{n}\) = the number of negation words in a sentence
\(N_{fa}\) = the number of frequent attributes
\(N_{mfa}\) = the number of matched attributes in the frequent attributes discovered by ARM
\(N_{pos}, N_{neg}\) = number of words with positive, negative valence in a sentence
\(p_{a}, p_{n}\) = a term attribute in the user predefined set, frequent attributes
\(R_d\) = domain customized knowledge model
\(S_{ij}\) = similarity measure between two cases \(C_i\) and \(C_j\)
\(s^m_i(s^m_i) = m\)th (normalized) score vector
\(\text{syn}(w_k)\) = synonym set of \(w_k\)
\(V_{ave}, A_{ave}, D_{ave}\) = the average of valence, arousal, and dominance of affective words in a sentence
\(V_{max}, V_{min}\) = the maximum and minimum valence of the words in a sentence
\(w_k\) = kth word in ANEW

References


ceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Seattle, pp. 168–177.


ceedings of the Ninth International Conference on Electronic Commerce, ACM, Minneapolis, MN, pp. 311–318.


