Engineering Applications of Artificial Intelligence 156 (2025) 111110

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Automatic requirements elicitation from user-generated content: A review of data, methods, and representations

Mengsi Cai [*], Wenchuan Yang, Yonghao Du, Yuejin Tan, Xin Lu [**]

College of Systems Engineering, National University of Defense Technology, Changsha, 410073, China

ARTICLE INFO

Keywords:
Requirements elicitation
User-generated content
Requirement representations
Natural language processing
Product aspects

ABSTRACT

Requirements elicitation (RE) is the first and most important phase in requirements engineering. In the era of big data, as online platforms have become one of the primary channels for users to provide product feedback, extensive research has been conducted to explore how customer requirements can be elicited from user-generated content (UGC) through artificial intelligence technologies such as text clustering, topic modeling, and machine learning. We systematically review the existing literature in the area of UGC-based RE by categorizing research according to three dimensions: UGC data sources for RE, RE methods, and requirement representations. Furthermore, we propose a comprehensive research framework for UGC-based RE and identify key directions for future research. The findings of this study can contribute to the development of more effective and intelligent RE processes, while helping to establish a systematic framework for UGC-based RE research.

1. Introduction

Requirements elicitation (RE) is the first and most important phase of requirements engineering, as it helps uncover customers' demand for products and services. The success of product development and improvement depends on the effectiveness of RE (Pohl, 2010). However, due to the fuzziness, implications, dynamics, and variety of customer/user demands, RE is a challenging task. Traditional RE involves conducting interviews, questionnaires, focus groups, workshops, and consultations with field experts, as well as quality function deployment (QFD) and house of quality (HoQ) (Dieste and Juristo, 2010). These approaches rely primarily on direct communications between stakeholders and product developers or requirements engineers and are thus time-consuming and not sufficiently scalable.

User-generated content (UGC) refers to the public media content created by web users, including content from online customer reviews, online discussions, blogs, and social media (Wang et al., 2011). These UGC instances contain valuable reports and feedback on potential requirements, bugs, feature shortcomings, and feature requests, and have thus become new avenues for RE and, in turn, for improving the quality of products and services (Qi et al., 2016), (Zhang et al., 2021). For example, mobile application storefronts such as Google Play Store and Apple App Store allow users to share their opinions on downloaded applications (Tavakoli et al., 2018); Automobile Home, one of the most popular car review websites in China, generates over 40K reviews on various automobiles every day (Cai et al., 2022); and software repositories such as GitHub and Redmine allow developers and users to submit new feature requests and bug reports for software improvement (Li et al., 2020).

In the era of mobile internet and big data, customers are increasingly publishing their experiences and feedback regarding product usage on various online platforms, such as app stores, social media, and e-commerce websites (Xiao et al., 2016). Considerable effort has been directed towards automatically extracting user requirements from the vast pool of online UGC, given the inherent ambiguity in natural language. The key techniques employed for this purpose include textual analysis, named entity recognition, text clustering, and sentiment analysis (Dollmann and Geierhos, 2016). Although some researchers have reviewed the specific methods used in RE research and practices, none of them have paid comprehensive attention to the characteristics of UGC data and target products or the representations of user requirements. For instance, Sonbol et al. (2022) and Meth et al. (2013) omitted discussions regarding the data sources used for RE, while Tavakoli et al. (2018), Necmiye and Alain (2017), and Martin et al. (2017) predominantly concentrated on the requirements engineering challenges associated with software products, with data derived solely from one UGC

[*] Corresponding author.
[**] Corresponding author.
E-mail addresses: caimengsi18@163.com (M. Cai), xin.lu.lab@outlook.com (X. Lu).

https://doi.org/10.1016/j.engappai.2025.111110
Received 8 January 2024; Received in revised form 26 December 2024; Accepted 7 May 2025
Available online 21 May 2025

category, namely app reviews.

Therefore, in this paper, we present a comprehensive review of the literature on automatic RE using UGC (hereafter referred to as UGC-based RE). The UGC-based RE problem is described from three perspectives: UGC data sources, RE methods, and requirement representations. The objective of this review is to answer the following three research questions: (1) What categories of UGC data are utilized for automated RE? (2) Which methodologies and techniques are employed for automated RE? (3) What forms of requirement representations are derived from UGC? To address these questions, we have selected relevant papers from scientific databases and performed a quantitative analysis of these papers to summarize the research topic. The structure of this paper is shown in Fig. 1. The overall contributions of this paper are as follows:

- A comprehensive review of UGC-based RE is provided from three interrelated, cohering perspectives: UGC data sources, RE methods, and requirement representations. Specifically, the *UGC data* used for RE are categorized as online review data, social media data, and online forums data; *RE methods* are categorized as rule-based methods, text clustering, topic modeling, machine learning, and other methods; and *requirement representations* formats are categorized as requirement-related data, requirement categories, and requirement statements at the macro, meso, and micro levels, respectively.
- A V-shaped framework for UGC-based RE research is provided by restructuring and dividing RE-related processes into eight stages: data collection, data preprocessing, data assessment, requirement data identification, requirement category classification, requirement statement extraction, requirement ranking, and product design. These can help both researchers and requirement analysts gain an in-depth understanding of the current status of UGC-based RE research.

The rest of this paper is structured as follows: The literature search process and bibliometric analysis results are presented in Section 2. An overarching framework for UGC-based RE research is summarized in Section 3. The review results and a discussion on the UGC data sources, applied RE methods, and elicited requirement representations are presented in Section 4. The research conclusions and directions for future work are given in Section 5 and Section 6, respectively.

2. Methodology

To lay a solid foundation for a comprehensive survey of UGC-based RE research, we conducted a two-step process. In the first step, we designed general guidelines for the collection of research articles (Section 2.1). The second step involved a bibliometric analysis of research articles (Section 2.2). For this purpose, we predefined the categories for UGC data sources, RE methods, and requirement representations and then analyzed the relationships between these categories. Based on a bibliometric analysis of the relevant literature, we established a V-shaped UGC-based RE research framework, this is presented in Section 3.

2.1. Collection of research articles

To obtain the most relevant literature in the area of automated UGC-based RE, we used the following criteria: (1) We searched nine scholarly databases: the Web of Science, ACM Digital Library, IEEE Xplore, EI Compendex, Elsevier ScienceDirect, Springer, Scopus, EBSCOhost, and ProQuest. (2) The publications were restricted to journal and conference papers that were in English and had undergone peer review. (3) The search keywords were split into two components. The first component was related to requirement elicitation and included these keywords: “requirement elicitation”, “requirement extraction”, “requirement acquiring”, “requirement classification”, “requirement mining”,

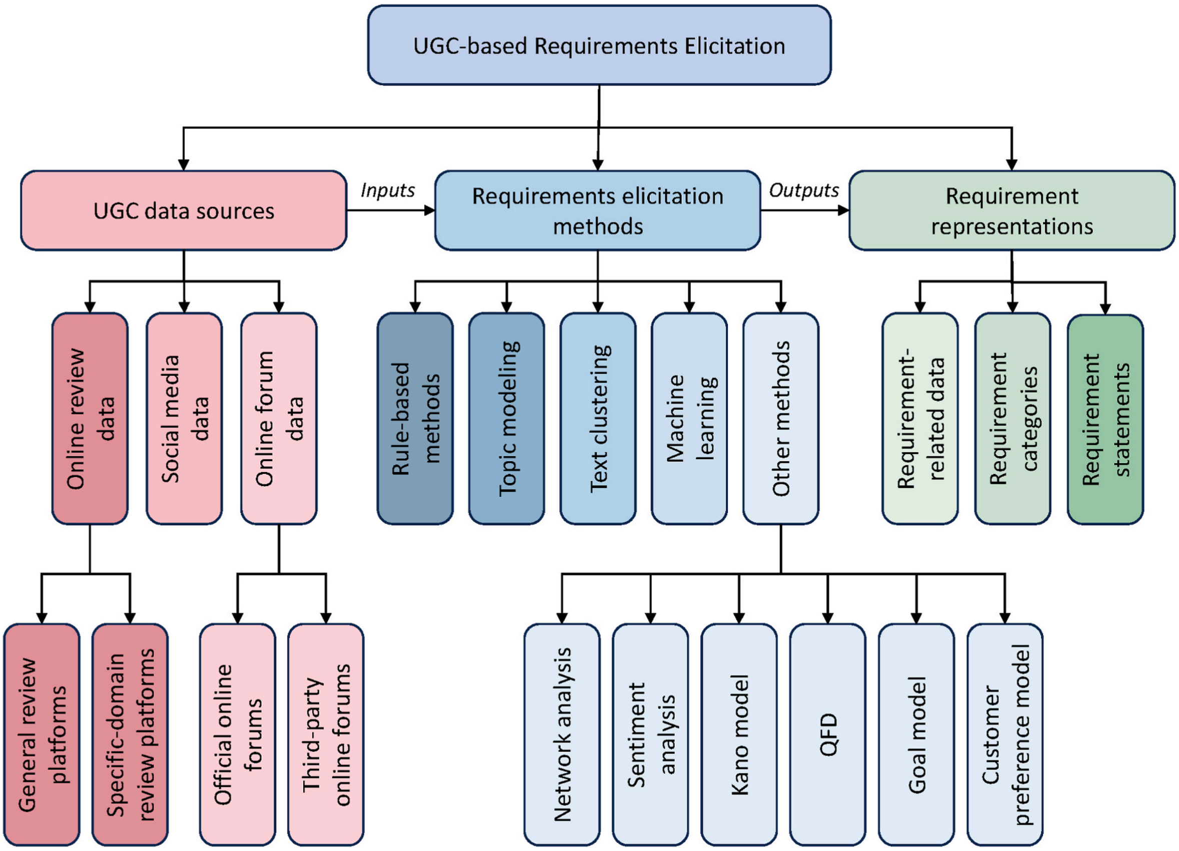

Fig. 1. The review structure.

"requirement analysis", "requirement discovery", "requirement identification", "requirement collection", "requirement gathering", "requirement engineering", "requirement determination", "product requirement", "user requirement", "user preference", "system requirement", "customer demand", "user demand", "customer need", and "user need". The second component was related to UGC sources and analytics, and it included the following keywords: "user-generated content", "online review", "customer review", "user review", "product review", "big data", "app reviews", "app store", "forum", "website", "Twitter", "Facebook", "data analysis", "data mining", "data science", "data driven", "data oriented", "customer oriented", "user driven", "machine learning", "natural language processing", "artificial intelligence", "data processing", and "text mining". Terms within each component were connected by OR operators, while the different components in the search strings were connected by AND operators. For example, a search string can be: ("requirements elicitation" OR "requirements extraction") AND ("user-generated content" OR "machine learning").

We then removed the duplicates and conducted a comprehensive two-round screening process to identify relevant studies. In the first round of screening, the studies were classified into one of three categories – (1) included, (2) excluded, or (3) uncertain – based on an evaluation of their titles, abstracts, and keywords. The studies that fell into the first and third categories were assessed for their eligibility via full-text screening. In each screening process, the delimitations were as follows:

- Only publications employing (semi-)automated RE processes were considered.
- Only publications using UGC data instead of previously obtained requirements were included.
- Only publications that involved sufficient experimental evaluations of methodologies were considered.

As a result, a total of 232 papers were included in this review.

2.2. Bibliometric analysis

To meet the research objectives and for statistical convenience, we classified each publication into a specific category according to the UGC data sources, RE methods, and requirement representations. Specifically, the UGC data source in each paper was identified as the major data source used for the validation or case study; the RE method was identified as the method used for obtaining the final requirement representation; and requirement representation was identified as the final formulation of the elicited user requirements. Each UGC data source, RE method, and requirement representation category is explained below.

2.2.1. Categories of UGC data sources

Based on the characteristics of online platforms that publish and display user-created content, we classified the UGC data into three main types: online review data, social media data, and online forum data.

1) ***Online review data*** include feedback, comments, and reviews that are posted by users on online review platforms, such as product reviews and app reviews. Based on the target products that are usually reviewed, we future classified online review platforms as general and specific-domain review platforms. *General review platforms* provide reviews on full categories of products and services; these include amazon.com (Amazon) (Zhang et al., 2021), jd.com (JD) (Qi et al., 2016), yelp.com (Yelp) (Mukherjee et al., 2013), ebay.com (eBay), taobao.com (Taobao), and dianping.com (Dianping). *Specific-domain review platforms* are generally established for one or several specific product categories; these include app stores, autohome.com.cn (Autohome) (Zhao et al., 2023), g2crowd.com (G2 Crowd) (Buchan et al., 2018), airbnb.cn (Airbnb), tripadvisor.com (Tripadvisor), Booking.com (Hsiao and Hsiao, 2020), and grubhub.com (Grubhub) (Xu, 2021).
2) ***Social media data*** are microblogs and posts that are typically published on social media platforms (e.g. Twitter, Facebook, and Weibo) in different content formats, including short texts, audio, video, and images. These social media instances contain valuable insights into user experiences and recommendations, reflecting the needs of customers.
3) ***Online forum data*** comprise messages sent in online communities for convenient communication and discussion between users, developers, and product owners. We further classified online forums as official online forums and third-party online forums according to their establishers. *Official online forums* are usually established by product owners (companies) and mainly provide services to their own customers; these online forums include official communities and mailing lists. *Third-party online forums* are usually created by third-party companies or persons and can be publicly accessed by various users to discuss products and services; these include question-answer (Q&A) sites, software repositories, and issue tracking systems.

2.2.2. Categories of RE methods

Based on the techniques and methodologies adopted to (semi-)automatically obtain user requirements from freely expressed UGC data, we classified the applied RE methods into five categories: rule-based methods, text clustering, topic modeling, machine learning, and other methods.

1) ***Rule-based methods*** consist of a list of linguistic rules that are used to automatically retrieve user requirements. In the early stages of UGC-based RE research, user requirements were reflected by the keywords of product features (properties/attributes), and sets of rules, including word frequencies, part-of-speech (POS) tagging, keyword vocabularies, and language rules, were precodified by human experts to retrieve feature words from unstructured textual content. For example, several researchers have considered nouns, verbs, or adjectives as words that describe product attributes (Zhou et al., 2015).
2) ***Text clustering*** involves dividing user-generated review texts into different groups/clusters based on the distance (or similarities) between the word embedding vectors of these texts. A review text can be a review document formed by a series of sentences, a review sentence formed by a series of words, a feature phrase, or a feature word. Popular text clustering algorithms include K-means, K-means++, X-means, density-based spatial clustering of applications with noise (DBSCAN), hierarchical DBSCAN (HDBSCAN), hierarchical clustering, and spectral clustering. Clustering algorithms classify similar texts into the same cluster. Each cluster then represents a group of similar or related user requirements, and the subject of each group is usually named using the high-frequency words in the group.
3) ***Topic modeling*** is a process that involves internal semantic knowledge extraction models that are widely used in text mining tasks such as retrieval, summarization, and clustering. Generally, each text is associated with different topics, and these topics are, in turn, associated with different words with certain probability levels. In an RE task, a topic can be a set of words that reflect a user's requirements or concerns with product features; for example, the topic {*crash, update, frustrated, newest, version, help, bug*} describes users' experiences when updating a faulty app (Guzman and Maalej, 2014). Latent Dirichlet allocation (LDA), non-negative matrix factorization (NMF), and biterm topic model (BTM) are some widely used topic models. Using UGC text corpora as inputs, topic models can automatically generate a series of main topics mentioned in a piece of text and visualize the associations between the input texts and topics in the form of a probability matrix. The extracted topics are often regarded as product aspects of high concern among customers and are named using the most relevant/appropriate words.
4) ***Machine learning (ML)*** is a widely used method for solving classification and sequence labeling problems in the RE field. ML models

are trained based on various features learned from UGC data (called feature engineering) and are then used to classify input texts into different classes. ML methods used in UGC-based RE studies can be classified into two types: classification methods and NER (named entity recognition) methods. *ML-based classification methods* are used to identify requirement-related data and to classify user requirements into different categories. Commonly used ML-based classification models include support vector machine (SVM), naïve Bayes (NB), random forest (RF), decision tree (DT), logistic regression (LR), multinomial naïve Bayes (MNB), multivariate logistic regression (MLR), eXtreme gradient boosting (XGBoost), convolutional neural network (CNN), and long short-term memory (LSTM). *ML-based NER methods* are used to extract requirement-related entities from unstructured UGC texts. In an RE task, requirement-related entities, including feature attributes and users' opinions, are often extracted by sequence annotation models such as conditional random field (CRF) (Jin et al., 2016), bidirectional long short-term memory (BiLSTM) (Cai et al., 2022), bidirectional encoder representation from transformers (BERT) (de Araújo and Marcacini, 2021), BiLSTM-CRF (Zhou et al., 2020a), and BERT-BiLSTM-CRF (Xiao et al., 2022).

5) ***Other methods*** include network analysis, sentiment analysis, the Kano model, QFD, goal models, and customer preference models. Among these, the first three methods are used to identify important/priority requirements, and the last three methods are used to determine the product design characteristics that meet user requirements. *Network analysis* is conducted on semantic networks, such as word occurrence networks, to identify the important underlying customer requirements, understand relationships among features, and detect feature groups (Park and Lee, 2011). *Sentiment analysis* is usually used to evaluate customer satisfaction with products or product attributes by calculating the emotional polarity and intensity of online reviews (Cai et al., 2022). Three popular sentiment analysis methods are rule-based methods (e.g. VADER (Kumari and Memon, 2022)), lexical-based methods (e.g. SentiStrength (Wang and Wang, 2014)), and ML-based methods (e.g. SVM (Cai et al., 2022)). The *Kano model* is used to evaluate the impacts of product attributes on customer satisfaction based on questionnaires, with product attributes classified into five levels: must-be, one-dimensional, attractive, indifferent, and reverse (Qi et al., 2016). *QFD* is a famous customer-oriented product development tool that translates customer requirements into a design language (e.g. design requirements, part features, or production plans) through a series of transformations (Liu et al., 2021a). The *goal model* is an effective way to describe requirements, capture interactions between requirements, and further support the software development process (Svee and Zdravkovic, 2016). The *customer preference model* is used for product design selection problems, specifically to define design variables and generate design alternatives, estimate the costs for each design alternative, and finally select the design that can maximize profits.

It is worth noting that, since we paid particular attention to the functions and application scenarios of the RE methods, there might be some overlaps among these method categories. For example, many neural network models can be used for both classification and NER tasks.

2.2.3. Categories of requirement representations

In previous studies, multiple definitions of user requirements have been formed throughout the life cycle of the product design process. For example, in 1996, Hazelrigg (1996) defined user requirements as design decisions that are made by high-level stakeholders or at a high level of design abstraction. In the 2000s, Young (2001) defined user requirements as statements on the critical attributes, characteristics, capabilities, or functions of a design, aimed at improving the understanding and focus of the designer involved. Diev (2007) defined user requirements as conditions that a system or product must conform to. According to Dieter and Schmidt (2009), user requirements are the requirement lists formed in the earliest phases of product design. More recently, the International Council on Systems Engineering (INCOSE) (International Council on Syetems Engineering (INCOSE), 2015) defined user requirements as statements that identify the system or product constraints necessary for stakeholder acceptance.

In this paper, we focus on customer requirements obtained from user-generated content. Based on the user requirements elicited from the UGC data in the reviewed papers, we define user requirements as the needs or expectations of a web user, including preferences, concerns, complaints, feature requests, feature shortcomings, bugs, and defects, that direct the motive for developing a new product or improving a current product or service. In short, user requirements are the feedback or problems reflected in online user-generated content for future product releases (Guzman et al., 2017a).

In addition, due to the diversity of UGC data sources and target products/services, user requirements differ in their formulation and expression. In this paper, we propose that the concept of *requirement representations* indicates the formulation and expression of user requirements obtained from UGC data. We classify the requirement representations as requirement-related data, requirement categories, and requirement statements at the macro, meso, and micro levels, respectively.

1) ***Requirement-related data*** comprise reviews or review sentences that contain requirement-related information, providing a coarse-grained description of user demand at the *macro* level. When eliciting user requirements, researchers often classify UGC data as requirement-related data (informative) or non-requirement-related data (uninformative). A review that contains user requirement-related information would be regarded as informative (Gao et al., 2020). For example, Chen et al. (2014) classified user reviews from Google Play into the unique classes "informative" and "uninformative", with the former implying that the reviews contain information that app developers are looking for and is potentially useful for improving app quality or user experience.
2) ***Requirement categories*** describe the specific requirement types (e.g. feature requests and bug reports) that requirement-related data mention and are thus *meso*-level representations of user requirements. In this regard, software requirements are usually classified into functional and non-functional requirements (Kumari and Memon, 2022), and many researchers have classified software or app reviews into the categories of feature requests, problem discovery, information giving, and information seeking (Al-Hawari et al., 2021), (Tizard et al., 2019).
3) ***Requirement statements*** refer to users' fine-grained opinions or sentiments on specific product aspects (e.g. product features, user concerns, and expectations). These consist of in-depth, detailed descriptions of customer demands at the *micro* level. A product aspect is defined as an attribute or property of a product or service that users pay attention to (Kovacs et al., 2021). For example, automobile product aspects include space, power, manipulation, energy consumption, comfort, exterior, and interior (Cai et al., 2022). A requirement statement for a specific product aspect relates to a specific feature (Wang et al., 2011), a group of features (Shi and Peng, 2021), a specific feature with sentiment (Wang and Wang, 2014), a specific feature with rank (Cai et al., 2022), or other forms of statements, such as "submit form button should send form data" (Vlas and Robinson, 2012) and "learning curve" (Suryadi and Kim, 2019).

Here is an example of an online review: "Nice application, but lacks some important features like support to move on SD card. So, I am not giving five star rating" (see Fig. 2). At the macro level, this review is a piece of requirement-related data; at the meso level, this review can be

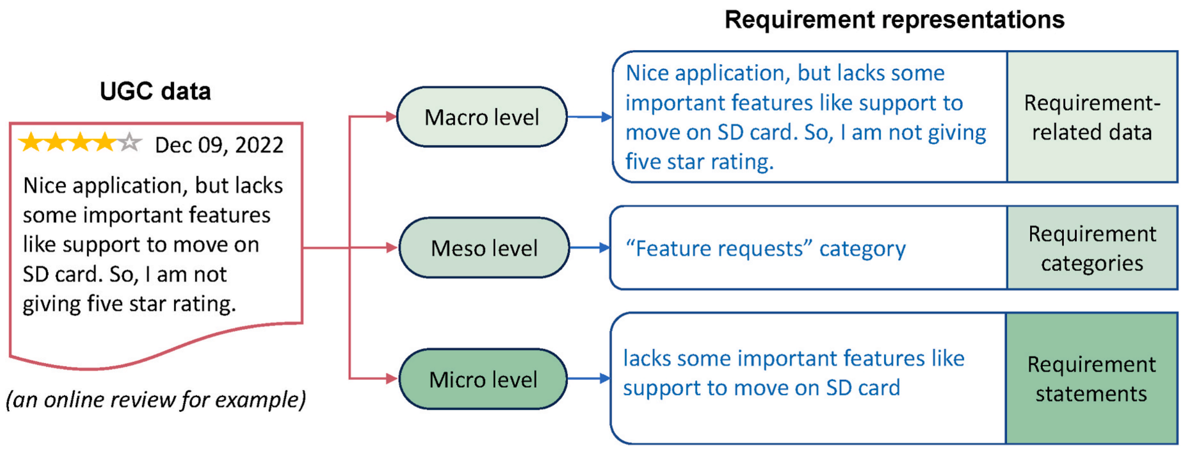
Requirement representations
UGC data
Dec 09, 2022
Nice application, but lacks some important features like support to move on SD card. So, I am not giving five star rating.
(an online review for example)
Macro level
Meso level
Micro level
Nice application, but lacks some important features like support to move on SD card. So, I am not giving five star rating.
Requirement-related data
"Feature requests" category
Requirement categories
lacks some important features like support to move on SD card
Requirement statements

Fig. 2. Diagram of the three levels of requirement representations.

classified into the requirement category of feature requests; and at the micro level, the statement “lacks some important features like support to move on SD card” highlights the product attribute of “support to move on SD card” and the reviewer’s negative sentiment towards this attribute.

2.3. Statistics analysis

Of the 232 publications we collected, 127 were journal papers, and 105 were conference papers. As shown in Fig. 3 (a), the number of studies exhibited an upward trend, with half of the included papers being published between 2021 and 2023 (118 out of 232). In the early stages, rule-based methods were the major approaches used for UGC-based RE, after which topic modeling and text clustering gradually became popular. Since 2015, machine learning models have been widely used and developed rapidly.

The relationships between the UGC data sources (inputs), RE methods, and requirement representations (outputs) of UGC-based RE are shown in Fig. 3 (b). It is obvious that online review data constitute the most popular data source for RE. The majority of RE studies have focused on classifying online review data into different requirement categories through machine learning algorithms, and rule-based methods, topic modeling, and text clustering have usually been used to identify micro-level requirement statements.

3. A research framework for UGC-based RE

Although the methods applied for different RE purposes vary, the underlying logical framework is similar. By summarizing the RE processes mentioned in relevant studies, we finally determine a V-shaped framework for UGC-based RE research, as shown in Fig. 4. We divide RE-related processes into eight sequential stages: data collection (DC), data preprocessing (DP), data assessment (DA), requirement data identification (RDI), requirement category classification (RCC), requirement statement extraction (RSE), requirements ranking (RR), and product design (PD). Among these, the first four stages are directed against *user-generated data*, the fifth to seventh stages are conducted to obtain *user requirements*, and the last stage is aimed at mapping user requirements into *product structures*. In addition, the outputs of requirement data identification (stage 4), requirement category classification (stage 5), and requirement statement extraction (stage 6) are requirement-related data (macro level), requirement categories (meso level), and requirement statements (micro level), respectively, which correspond to the three types of requirement representations mentioned above. Each process in the UGC-based RE research framework is explained below.

1) *Data collection (DC)*

In this stage, the aim is to determine the target products/services and select appropriate data sources. In general, UGC can be collected from online review platforms, social media, and online forums through web crawlers, APIs, and open datasets.

2) *Data preprocessing (DP)*

A raw dataset typically contains many duplicate and redundant data. To obtain high-quality data inputs for RE, several preprocessing steps are conducted, such as dropping duplicates, converting raw user reviews

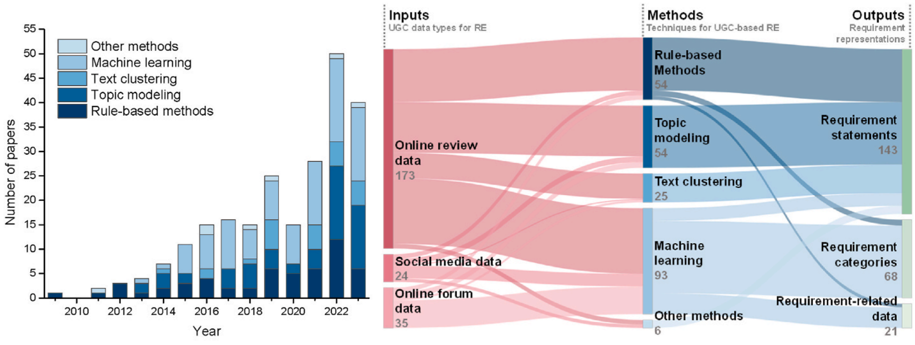

Fig. 3. (a) Statistics of the reviewed papers. (b) Alluvial diagram of the attributes and their relationships. Each rectangle represents a category, each number indicates the number of papers in each category, and the thickness of the edge between two rectangles indicates the number of papers.

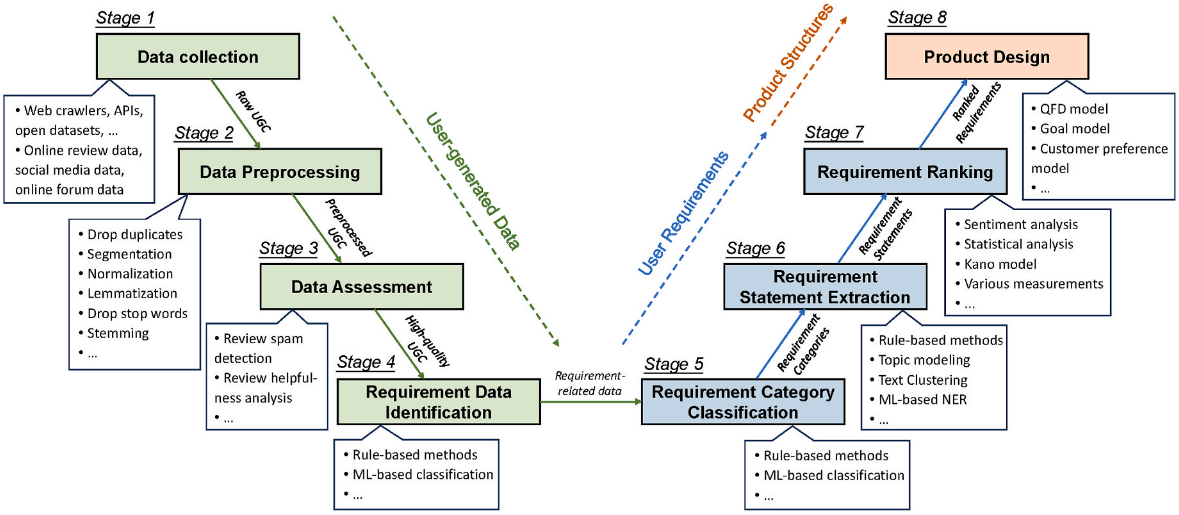

Fig. 4. Overall framework for UGC-based RE research.

into sentence-level review instances, converting text into lower cases, removing stop words, stemming the texts, and lemmatization.

3) *Data assessment (DA)*

Due to the low reliability of online UGC data, it is necessary to filter out spam reviews and select helpful reviews before eliciting user requirements. However, review spam detection has rarely been carried out before requirements elicitation in previous studies, except in (Guzman et al., 2017b), (Williams and Mahmoud, 2017), (Man et al., 2016), (Spada et al., 2023). In addition, to ensure that minimal fake reviews were present in their dataset, Zhang et al. (2023a) filtered out the duplicate reviews that appeared more than once or consisted of similar content. Review helpfulness analysis has rarely received attention in the RE field (Qi et al., 2016), although it has been widely studied in the review analysis field (Yang et al., 2021).

4) *Requirement data identification (RDI)*

Considering the low-value density of online reviews – that not all online reviews contain meaningful information about user requirements – it is necessary to filter out uninformative reviews. A popular way of doing so is to identify requirement-related data from massive amounts of UGC via ML-based binary classification methods (Gao et al., 2020), (Chen et al., 2014). While requirement-related data are macro-level requirement representations, they are still UGC data (e.g. online reviews) in the strict sense, not user requirements.

5) *Requirement category classification (RCC)*

In this stage, the requirement-related UGC data are classified into different requirement categories, such as functional requirements (FRs), non-functional requirements (NFRs), feature requests, bug reports, and information seeking. These meso-level requirement categories are usually obtained through ML-based classification methods, in which the number of requirement categories is predefined based on the target product, requirements, and concerns.

6) *Requirement statement extraction (RSE)*

In this stage, the micro-level requirement statements describing detailed product aspects are usually extracted from unstructured UGC text through rule-based methods, text clustering methods, topic models, and ML-based NER models.

7) *Requirements ranking (RR)*

Requirements ranking, also known as requirement prioritization, can help developers understand which user requirements are the most important and urgent to implement. The importance and urgency of user requirements can be measured using a list of indexes, such as attention degree (Han et al., 2019), satisfaction degree (Cai et al., 2022), (Zhang et al., 2018), importance level (Ireland and Liu, 2018), redesign index (Zhang et al., 2019a), opportunity score (Jeong et al., 2019), and influence power (Park and Lee, 2011), (Kim and Noh, 2019). These indexes are often calculated based on the number and sentiment score of reviews, frequency of noun phrases, semantic consistency score of topics, and review metadata (e.g. forward, like, comments, votes, and ratings) (Kovacs et al., 2021), (Gao et al., 2018). In addition to quantitative analyses of requirement importance and urgency, the Kano model can also be used to rank requirements, as it classifies user requirements into five levels: must-be, one-dimensional, attractive, indifferent, and reversal. These rankings reflect the importance and priority levels of different requirements for product improvement.

8) *Product design (PD)*

Product structures refer to the total proportion of and interrelationships between each component of a product; they reflect customers' needs and provide goals for product optimization design (Liu et al., 2021a). After eliciting user requirements, it is important to map user requirements to product design characteristics such as product structures, functions, and variables. QFD (Liu et al., 2022a), goal models (Svee and Zdravkovic, 2016), (Gao et al., 2020), and customer preference models (Zhang et al., 2019a) are usually adopted for this purpose.

As shown in Table 1, we identified several representative studies for most stages of the V-shaped framework for UGC-based RE research (see Fig. 4). Data collection and data preprocessing are the most basic, necessary steps for dealing with unstructured UGC texts. While user requirements were obtained from the identified requirement-related data in many previous studies, RE tasks were conducted directly on raw online reviews in other studies. Requirement ranking methods have been widely applied to evaluate the importance and urgency of user requirements for product improvement or future product development. However, only a small number of studies have mapped user requirements to product structures, as product design is more of an engineering task that is carried out after obtaining user requirements.

Table 1
Application of the UGC-based RE research framework composed of eight sequential stages.

Typical papers	1) DC	2) DP	3) DA		4) RDI	5) RCC	6) RSE	7) RR			8) PD		
	Data collection	Data preprocessing	Review spam detection	Review helpfulness analysis	Requirement Data Identification	Requirement Category Classification	Requirement Statement Extraction	Sentiment analysis	Kano model	Other measurements or methods	QFD	Goal model	Customer preference model
Williams and Mahmoud (2017)	✓	✓	✓			✓		✓					
Fu et al. (2013)	✓	✓	✓				✓	✓					
Qi et al. (2016)	✓	✓	✓	✓			✓	✓	✓				
Man et al. (2016)	✓	✓	✓		✓		✓			✓			
Zhang et al. (2023a)	✓	✓	✓				✓	✓	✓	✓			
Guzman et al. (2017a)	✓	✓			✓		✓	✓		✓			
Zhou et al. (2020b)	✓	✓			✓		✓	✓	✓				
Han et al. (2019)	✓	✓			✓		✓		✓	✓			
(Chen et al., 2014), (Timoshenko and Hauser, 2019)	✓	✓			✓		✓			✓			
(Kühl, 2016), (Khan et al., 2020)	✓	✓			✓	✓							
Panichella et al. (2015)	✓	✓			✓	✓		✓					
Kumari and Memon (2022)	✓	✓			✓	✓	✓	✓					
Gao et al. (2020)	✓	✓			✓		✓					✓	
Al Kilani et al. (2019)	✓	✓				✓		✓					
Villarroel et al. (2016)	✓	✓				✓	✓			✓			
(Gu and Kim, 2015), (Zhang et al., 2012), (Carreño and Winbladh, 2013)	✓	✓					✓	✓					
(Hsiao and Hsiao, 2020), (Li et al., 2021)	✓	✓					✓	✓	✓				
(Cai et al., 2022), (Ireland and Liu, 2018), (Eldin et al., 2021)	✓	✓					✓	✓		✓			
Zhang et al. (2022)	✓	✓					✓		✓				
(Park and Lee, 2011), (Zhang et al., 2018), (Jeong et al., 2019)	✓	✓					✓			✓			
(Zhang et al., 2019a), (Gao et al., 2018)	✓	✓					✓			✓			✓
Wang et al. (2011)	✓	✓					✓						✓
Svee and Zdravkovic (2016)	✓	✓					✓	✓				✓	
Liu et al. (2021a)	✓	✓					✓	✓			✓	✓	
Liu et al. (2022a)	✓	✓			✓			✓	✓	✓	✓		
Song et al. (2018)	✓	✓					✓	✓	✓		✓		

4. Results and discussion

In this section, we present our analysis of the reviewed publications based on their UGC data, RE methods, and requirement presentations. We then provide a comprehensive comparison of different UGC data sources (Section 4.1.2) and highlight the strengths and weaknesses of UGC data (Section 4.1.3). In addition, based on the proposed V-shaped framework of UGC-based RE research, the relationships between different RE methods and requirement presentations are summarized to understand how macro-, meso-, and micro-level requirements are obtained from UGC data through various techniques (Section 4.4).

4.1. UGC data for RE

User-generated content published on various online platforms showcase consumers' thoughts, opinions, and feelings regarding products and services, thus reflecting their requirements and preferences. Analyzing these UGC instances can provide product developers the opportunity to better understand customers' requirements (Qi et al., 2016). In the present study, we found that UGC data published on various online platforms could be classified as online review data, social media data, and online forum data, which are usually collected by web crawlers (Wang and Wang, 2014), (Shi and Peng, 2021) and APIs (Svee and Zdravkovic, 2016), (Man et al., 2016) (e.g. Twitter API, Google Play API, etc.) or obtained from publicly available datasets (Al-Hawari et al., 2021), (Zhao and Zhao, 2019).

4.1.1. Utilization of different UGC data

The distribution of the literature involving online review data ($n = 173$, 74.57 %), online forum data ($n = 35$, 15.09 %), and social media data ($n = 24$, 10.34 %) as inputs for UGC-based RE are shown in Fig. 5. A detailed introduction on the use of different UGC data for different products is given in Table 2.

In the majority of online review studies, UGC data were collected from specific-domain review platforms ($n = 96$, 41.38 %), such as app stores, Tripadvisor, Ctrip.com, Autohome, G2 Crowd, capterra.com (Capterra), Booking.com, and Grubhub, while general review platforms such as Amazon, epinions.com (Epinions), JD, bestbuy.com (Bestbuy), Taobao, Rakuten Japan, and Alibaba were considered in 33.19 % ($n = 77$) of the studies. In the majority of online forum studies, third-party online forums ($n = 25$, 10.78 %), such as Sourceforge, Reddit, Github, Stack Overflow, and issue tracking systems, were used as data sources, whereas official online forums, such as the NMS Steam community, Mozilla Firefox forum, Xiaomi forum, and Apache Commons User List were only considered in 4.31 % ($n = 10$) of the studies. For social media studies, Twitter, Weibo, and Facebook were widely used to collect user feedback on software applications and electric products.

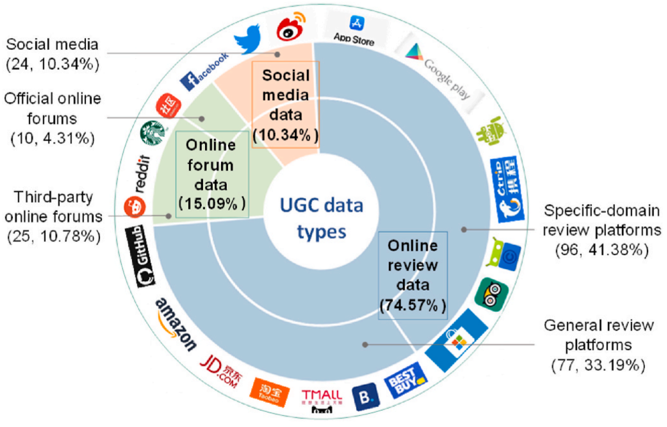

Fig. 5. UGC data types used for requirements elicitation.

Table 2
User-generated content data used for requirements elicitation.

UGC data types	UGC data sources	Num. of papers	Representative platforms and target products
Online review data	General review platforms	77	• Amazon for electronic products (Jin et al., 2016), (Zhang et al., 2019a), (Zhou et al., 2020b), (Zhang et al., 2022), (Jin et al., 2022a), (Wang et al., 2022a), automotive (Nikumanesh and Fathi, 2017), household (Law et al., 2017), daily necessities (Ireland and Liu, 2018), mobile apps (Zhao and Zhao, 2019), software applications (Jiang et al., 2014a), oral care products (Timoshenko and Hauser, 2019), and multiple products (Zhang et al., 2021), (Zhou et al., 2023) • Epinions.com for electronic products (Jin et al., 2016) and software applications (Hedegaard and Simonsen, 2013) • JD for electronic products (Liu et al., 2021a), (Zhang et al., 2023a), (Liu et al., 2022a), (Li et al., 2021), (Zhang et al., 2023b), household (Cong et al., 2023), and fresh food (Zhang and Huang, 2023) • Bestbuy for electronic products (Wang et al., 2011) and household (Law et al., 2017) • Taobao and tmall.com (Tmall) for electronic products (Du et al., 2022), food (Zhang et al., 2023c), and household (Wang, 2022) • Rakuten Japan (Kovacs et al., 2021), Souq.com (Eldin et al., 2021), and Alibaba (Shi and Peng, 2021) for multiple products • Dianping for restaurants (Guo et al., 2022)
	Specific-domain review platforms	96	• App stores: Apple App Store (Chen et al., 2019), (Jha and Mahmoud, 2019), (Alturaief et al., 2021), (Zhang et al., 2023d), Google Play Store (Liu et al., 2021b), (Scalabrino et al., 2017), (Yu et al., 2022), (Das et al., 2023), iOS App Store (Jha and Mahmoud, 2018), (Dalpiaz and Parente, 2019), (Jha and Mahmoud, 2017), Amazon's Software Store (Kurtanović and Maalej, 2017), (Groen et al., 2017), Android Market (Carreño and Winbladh, 2013), (Wen and Chen, 2020), 360 Mobile Assistant (Zhang et al., 2017), (Jin et al., 2022b), Windows Store (Sorbo et al., 2016), f-droid.org (Palomba et al., 2017), (Ciurumelea et al., 2017), for mobile applications; Some studies used UGC from multiple app stores (Man et al., 2016), (Gao et al., 2018), (dos Santos et al., 2021), (McIlroy et al., 2016), (Lu and Liang, 2017), (Guzman et al., 2015), (Binder et al., 2023) • Tripadvisor (Song et al., 2021), (Korfiatis et al., 2019) and Ctrip.com (Bian et al., 2022) for travel-related products/services • Autohome (Cai et al., 2022), pacauto.com and e-car.com (Zeng et al., 2022) for automobiles • Mobile.zol.com.cn (Zhang et al., 2018) for electronic products; G2 Crowd (Buchan et al., 2018) and Capterra (Kamaruddin et al., 2019) for software products; mobilephones

(*continued on next page*)

Table 2 (*continued*)

UGC data types	UGC data sources	Num. of papers	Representative platforms and target products
			urvey.com (MobilephoneSurvey) (Park and Lee, 2011) for mobile phones • Booking.com for hotels (Hsiao and Hsiao, 2020); Grubhub for restaurants (Xu, 2021) • Airbnb for short-term rental (Liu et al., 2022b), (Ji et al., 2023)
Social media data	Social media	24	• Twitter for electric vehicles (Kühl et al., 2020), brands (Gonzalez et al., 2020), companies (Stanik et al., 2019), shoes (Yoon et al., 2018), airline services (Svee and Zdravkovic, 2016), software applications (Guzman et al., 2017a), (Guzman et al., 2017b), (Williams and Mahmoud, 2017), (Kengphanphanit and Muenchaisri, 2020), mobile apps (Henao et al., 2021), mobiles (Arora et al., 2023), and smartwatch (Ali et al., 2019) • Facebook for smartwatch (Ali et al., 2019) and software applications (Kengphanphanit and Muenchaisri, 2020) • Weibo for services (Yan et al., 2022), high-speed rail (Chen et al., 2021), and metro vehicles (Han et al., 2019)
Online forum data	Official online forums	10	• Online communities: NMS Steam community (Tong, 2021) and Mozilla Firefox forum (Tizard et al., 2019) for software applications; Xiaomi forum for mobiles (Liang et al., 2017); Apple's official support community forums for iPod Classic portable music player (Abrahams et al., 2015); Honda-Tech.com, ToyotaNation.com, ChevroletForum.com for automotive (Abrahams et al., 2015); Luce (http://www.luce.com) and Yoka (http://www.yoka.com) for daily necessities (Zhang et al., 2012); SAP community for software applications (Kauschinger et al., 2023) • Mailing list: Apache Commons User List for mobile applications (Takahashi et al., 2015)
	Third-party online forums	25	• Online forums: Sourceforge for software applications (Vlas and Robinson, 2012), (Li et al., 2018); Reddit for mobiles (Jeong et al., 2019), mobile applications (Khan et al., 2019), (Khan et al., 2022) and daily necessities (Kilroy et al., 2022); Patient Opinion website (Tang et al., 2018), Yinling and Keai (Qian and Gui, 2021) for healthcare or medical services; Baidu Tieba for smartphones (Lai et al., 2023) • Repositories: Github for software applications (Li et al., 2020), (Merten et al., 2016), (Mehder and Aydemir, 2022) and mobile apps (Zhang et al., 2019b) • Issue tracking system (ITS): Redmine ITS (Merten et al., 2016), Bugzilla (Zhou et al., 2020a), SEnerCON and Apache OpenOffice ITS (Morales-Ramirez et al., 2019) for software applications • Q&A sites: Stack Overflow for software applications (Khalid et al., 2014)

Among all the investigated studies, we identified several studies in which attempts were made to elicit requirements from different data sources. For example, Henao et al. (2021), Gôlo et al. (2022) and Stanik et al. (2019) used both online review data (Google Play Store and Apple App Store) and social media data (Twitter); Takahashi et al. (2015) used both online review data (Apple App Store) and online forum data (Apache Commons User List); and Devine et al. (2023) used six datasets in previous studies for model evaluation, which involve online review data (f-droid.org, Apple App Store, and Google Play Store), online forum data (VLC media player and Firefox web browser forums), and social media data (Twitter).

4.1.2. *Differences between UGC data types*

The online review data, social media data, and online forum data obtained in the above studies differed widely in terms of data format, value, language, and target products.

First, regarding the format and value of these UGC data types, online review data and online forum data were mainly represented by plain text, while social media data were obtained from microblogs in textual, photo, video, and audio formats. In particular, online review data consist of ratings (usually 1 star to 5 stars) for users to score their purchased products or services (see Fig. 2), which can help quantify users' satisfaction with products. However, since online reviews can be published by all users without authentication or content restrictions, there are massive spam (fake) reviews on online review platforms (Ott et al., 2013), (Cai et al., 2023), and the proportion of spam data in online review platforms is larger than those on social media and online forums.

Second, different languages are often used on different online platforms based on their audience. Compared to internationalized online platforms that support multiple languages, such as Twitter, Amazon, and GitHub, a majority of online platforms are designed for users who use one specific language, such as JD, Taobao, and Weibo in China. Most of the reviewed studies were conducted using UGC data in English or Chinese, and only some studies were conducted with other languages such as Japanese (Kovacs et al., 2021), Arabic (Eldin et al., 2021), German (Kühl, 2016), (Kühl et al., 2020), Indonesian (Dewi and Mulyani, 2022), and Spanish (Gonzalez et al., 2020). Henao et al. (2021) and Stanik et al. (2019) developed RE methods for English text and Italian text, and Kumari and Memon (2022) translated five languages (France, Russia, Korea, Turkey, and Spain) into a standard language – English.

Third, different online platforms differ in their target products and services (see Table 2 for details). General online review platforms and social media platforms usually consist of reviews on various product categories. For example, Amazon covers a variety of products, including electronic products, cosmetics, clothes, books, daily necessities, and household electrical appliances, while Twitter attracts hundreds of millions of users who actively participate in providing feedback on feature requests and feature shortcomings as well as bug reports for various products in a dynamic, seamless manner. Relatively speaking, specific-domain review platforms and third-party online forums are usually established for a single product category or one specific product. For example, app stores such as Apple App Store, Google Play Store, and iOS App Store only support commenting on mobile apps; Stack Overflow is the largest Q&A site for promoting software development (Khalid et al., 2014); and project repositories and issue tracking systems such as GitHub and Redmine support massive issue reports about open-source projects and software (Li et al., 2020), (Nyamawe et al., 2019). In addition, official online forums are usually designed to enable improvements in the products and services of the establisher. For example, discussions.apple.com is Apple's official support community for Apple products such as the iPhone, iPad, Macbook, Apple Watch, and AirPods (Abrahams et al., 2015).

Fig. 6 shows the UGC data sources of the top five widely studied product categories: mobile apps (n = 75, 32.3 %), electronic products (n = 51, 22.0 %), software applications (n = 31, 13.4 %), automobiles (n =

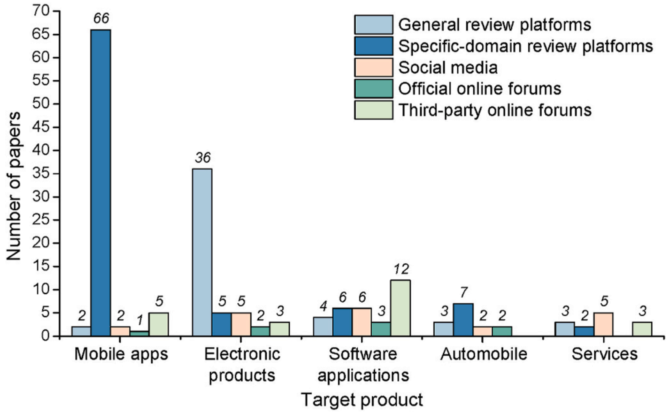

Fig. 6. UGC data source platforms for the top five studied products.

14, 6.0 %), and services (n = 13, 5.6 %). The findings displayed reveal that user requirements on mobile apps and electronic products are more likely to be found on online review platforms rather than other UGC data source platforms and that users often use third-party online forums such as GitHub, Stack Overflow, Sourceforge, and Reddit to publish feedback on mobile/software applications. Therefore, considering the constraints and characteristics of different online platforms, combining UGC data from multiple online platforms is an alternative option for obtaining a larger amount of data (Xiao et al., 2022), (Zhang et al., 2022), (Song et al., 2018), (Law et al., 2017). It is also necessary to select appropriate UGC data types and source platforms for RE according to the research objects (target products and services).

4.1.3. Strengths and weaknesses of UGC data

Various online platforms make direct communications between product owners and customers accessible and convenient. Compared to the small-sized requirement-related data collected via questionnaire surveys or interviews, UGC data are firsthand materials containing user feedback and have outstanding advantages in data volume (see Table 3), data collection cost, and data collection efficiency. However, due to the open availability of online platforms and the flexibility of user-created content, several limitations also exist.

Table 3
Several RE studies using plenitude UGC data.

Ref.	Year	Data sources	Products	Data Volume
Guzman et al. (2017a)	2017	Twitter	Software	68,108
Xiao et al. (2022)	2022	Taobao, Tmall, JD	Air conditioner	108,276
Iacob et al. (2014)	2014	Google App Store	Mobile apps	136,998
Song et al. (2018)	2018	JD, Tmall	Mobile	150,000
Wang and Wang (2014)	2014	Amazon	Camera	155,927
Tong (2021)	2021	NMS Steam community	Software	158,083
Gao et al. (2018)	2018	App Stores	Mobile apps	164,026
Suryadi and Kim (2019)	2019	Amazon	Laptops	218,570
Han et al. (2019)	2019	Weibo	Metro vehicle	452,298
Kim and Noh (2019)	2019	Bestbuy	Washing machines	496,866
Qi et al. (2016)	2016	JD	Mobile	679,422
Zhao and Zhao (2019)	2019	Amazon	Mobile apps	752,937
Liu et al. (2021b)	2021	Google Play Store	Mobile apps	4.48 million
Man et al. (2016)	2016	App Stores	Mobile apps	4.66 million
Fu et al. (2013)	2013	Google Play Store	Mobile apps	13 million
Khalid et al. (2014)	2014	Stack Overflow	Mobile	13.2 million
Kovacs et al. (2021)	2021	Rakuten Japan	Multi types	64 million

First, UGC data only includes feedback from internet users who are active online and tends to lack reviews from customers who do not use the internet or do not post comments, leading to sampling bias. Since requirements representations elicited from UGC can only reflect some online customers' needs (Cai et al., 2022), it is necessary to research the representativeness of UGC data. When there are not enough online reviews for requirement analysis, other complementary methods, such as survey-related methods, should be used.

Second, due to the anonymity and freedom of online users, the reliability of UGC data is hard to guarantee. For example, advertisements, irrelevant comments, and fake reviews may be mixed with genuine user requirements expressed in UGC. Before eliciting user requirements, preprocessing should be conducted on the collected UGC data, such as deleting advertisements, dropping duplicates, and assessing review usefulness.

Third, the inherent ambiguity and complexity of natural language in UGC can make it challenging to accurately extract and interpret user requirements from massive unstructured textual content, necessitating the use of advanced natural language processing methods.

4.2. Requirements elicitation methods

In this section, we introduce the techniques and methods used for UGC-based RE in the literature. When introducing a type of RE method, we highlight its ability to obtain user requirements from UGC data and its inputs (UGC data types) and outputs (requirement representations), to show the relationships between UGC data types, RE methods, and requirement representations.

4.2.1. Rule-based methods

We identified 54 studies that used rule-based methods to elicit requirements from UGC data. The rule-based methods could be divided into three subcategories: word frequency and POS tagging-based methods (WFPT), keyword vocabulary-based methods (KVoc), and language rule-based methods (LRule). The relationships between different rule-based methods and the obtained requirement representations are presented in Table 4. While most of them (n = 46, 85.2 %) were performed to extract requirement statements such as product feature words, a small number of studies were aimed at identifying requirement-related data (n = 3, 5.6 %) (Liu et al., 2021b) and requirement categories (n = 5, 9.2 %). In two studies (Zhou et al., 2015),

Table 4
Typical UGC-based RE studies using rule-based methods.

Method	Requirement-related data	Requirement categories	Requirement statements
WFPT	/	Yang and Liang (2015)	(Hsiao and Hsiao, 2020), (Zhou et al., 2015), (Zhang et al., 2018), (Ireland and Liu, 2018), (Liu et al., 2022a), (Zhang et al., 2012), (Chen et al., 2019), (Dalpiaz and Parente, 2019), (Kamaruddin et al., 2019), (Chen et al., 2021), (Jian et al., 2016)
KVoc	/	(Man et al., 2016), (Mercado et al., 2016)	(Qi et al., 2016), (Spada et al., 2023), (Wen and Chen, 2020), (Jiang et al., 2014b)
LRule	(Gao et al., 2020), (Iacob et al., 2014), (Liu et al., 2021b)	/	(Vlas and Robinson, 2012), (Gu and Kim, 2015), (Eldin et al., 2021), (Yu et al., 2022), (Groen et al., 2017), (Wang et al., 2022b)

(Zhang et al., 2012), the product features were distinguished as explicit features (usually represented by feature words like “appearance” and “price”) and implicit features (usually reflected by emotional words or words other than feature words, such as “ugly” and “expensive”). However, in most cases, researchers only paid attention to explicit features, while implicit features were extracted based on the explicit features through the collocation selection method (Zhang et al., 2012) or analogical reasoning method (Zhou et al., 2015).

1) Word frequency and POS tagging-based methods

Product features were identified based on word frequency, keywords extraction, or term frequency-inverse document frequency (TF-IDF). Furthermore, many researchers reported that feature words are likely to be described by nouns or noun phrases (Jian et al., 2016), verbs or verb phrases (Zhou et al., 2015), (Dalpiaz and Parente, 2019) or adjectives (Dalpiaz and Parente, 2019), which can then be extracted by POS tagging-based methods (Kilroy et al., 2022).

For example, Dalpiaz et al. (Dalpiaz and Parente, 2019) constructed a set of collocation patterns to identify popular user-concerned app features; for example, the word pairs “photo editing” and “edit photo” fell into the (noun, noun) pattern and (verb, noun) pattern, respectively. Chen et al. (2019) summarized a rule for Chinese noun phrase extraction as follows: (adj|noun)(adj|noun|”的”)*noun. The following is an example of a segmented and tagged review sentence: “这个/pron 软件/n 应该/v 添加/v 更多/adv 搞笑/adj 的/u 视频/n” (meaning “this application should add more funny videos”). “软件” (meaning “application”) and “搞笑的视频” (meaning “funny videos”) were chosen as opinion target candidates according to the extraction rule. However, this simple rule only led to an F1 score of 65.0 %.

2) Keyword vocabulary-based methods

Retrieving product feature words based on word frequency and POS information would easily output a large number of candidate feature words that are irrelevant to user requirements. Therefore, many researchers predefined keyword vocabularies, such as a feature word lexicon (Wen and Chen, 2020), sentiment (or emotion) lexicon (Mu et al., 2021), and degree lexicon (Mu et al., 2021), to correctly match relevant product features and related user opinions. The feature word lexicon contained feature expressions of product attributes, specifically nouns, noun phrases, and verb nouns. The sentiment lexicon contained positive words, negative words, and neutral words, which were usually adjectives and were assigned values of 1, -1, and 0, respectively. Based on the polarities of sentiment words, the degree lexicon was used to calculate the degree of emotional intensity, with degree adverbs divided into four levels: insufficient, moderately, high, and extreme.

For example, Spada et al. (2023) constructed a feature lexicon containing 64 quality attributes to discover user needs related to software applications. Qi et al. (2016) constructed an attribute lexicon and a sentiment lexicon to extract cell phone features from online reviews and then calculated consumers’ satisfaction with these features. Jiang et al. (2014b) designed SRPA+ (an expanded version of the syntactic relation-based propagation approach) to identify users’ opinions on software products, and it could extract target-sentiment pairs represented by <noun, adj> (such as “correct version”) and <verb, adv> (such as “update quickly”) through a seed sentiment lexicon and a syntactic relation-based propagation approach (Qiu et al., 2011). The SRPA + achieved precisions of 78.0 % and 73.0 % for Kaspersky Internet Security (KIS) 2011 and TuneIn 3.6, respectively. However, these keywords lexicons are usually constructed manually, leading to high costs and poor scalability.

3) Language rule-based methods

To reduce the cost of manually building keyword dictionaries, researchers have established sets of language rules to automatically extract user requirements from unstructured review sentences, including linguistic rules (Eldin et al., 2021), (Iacob et al., 2014), grammar rules (Gao et al., 2020), (Vlas and Robinson, 2012), and semantic rules (Liu et al., 2021b). For example, Iacob and Harrison (2013) designed MARA (Mobile App Review Analyzer) using 237 keyword-based linguistic rules and 3 grammar-based rules to extract feature requests for app products from reviews, obtaining a precision level of 85.0 %. A linguistic rule can be “Please add (NN) to”, in which (NN) indicates a requested app feature (e.g. “Next update please add a search feature to find photos quickly”). A grammar rule can be “V + N (the child nodes of VP include V and NP)” in which V ∈ {VB, VBD, VBN, VBG, VBP, VBZ} and N ∈ {NN, NNS, NNP, NNPS}; when given the sentence “I cannot share photos with friends on Instagram”, this rule will extract “share photos” as an app feature, as shown in Fig. 7 (a). Iacob and Harrison (Iacob et al., 2014) extended their method to MARA 2.0, which outperformed MARA with a precision of 91.0 %.

Gu et al. (Gu and Kim, 2015) proposed SUR-Miner (Software User Review Miner) for identifying aspect-opinion pairs for apps from review sentences based on 26 predefined semantic templates, and it obtained an F1 score of 85.0 %. For example, the resulting aspect-opinion pairs for the review sentence “The prediction is accurate, but the auto-correct is annoying” were <prediction, accurate> and <auto-correct, annoying>. A semantic template could be “JJ [nsubj-NN,cop-VBZ]”, which involves a root node with a POS tag of JJ and two children: a noun subjection (nsubj) with a POS tag of NN and a copula (cop) with a POS tag of VBZ. As shown in Fig. 7 (b), when given the sentence “The UI is beautiful”, this template will extract “nsubj-NN” (i.e. “UI”) as an aspect word and “JJ” (i.e. “beautiful”) as an opinion word.

In addition, Liu et al. (2021b) integrated 381 keyword-based linguistic rules and 24 semantic rules to extract app features and their updated details from review sentences. They achieved an average precision of 83.05 % for extracting features in four app categories (education, navigation, photograph, and social) on Google Play. Vlas and Robinson (2012) proposed a grammar-based strategy, a delimiter-based strategy, and a hybrid strategy for recognizing user requirements for software applications described by subject-action-object (SAO) triples, which resulted in an F1 score of 83.0 %. Groen et al. (2017) defined a set of 16 language patterns regarding the usability of apps and obtained a high precision level of 92.4 %. For example, the language pattern “(?< EN_Negations)(that |)(easy)(to |)(navigate|customize)” could identify usability statements such as “The app is easy to navigate” and “This thing works like champ, easy to customize and configure”. Yu et al. (2022) proposed the ReviewSolver, which could extract verb phrases and noun phrases by using a parse tree and typed dependency relations. The ReviewSolver was found to outperform ChangeAdvisor (Palomba

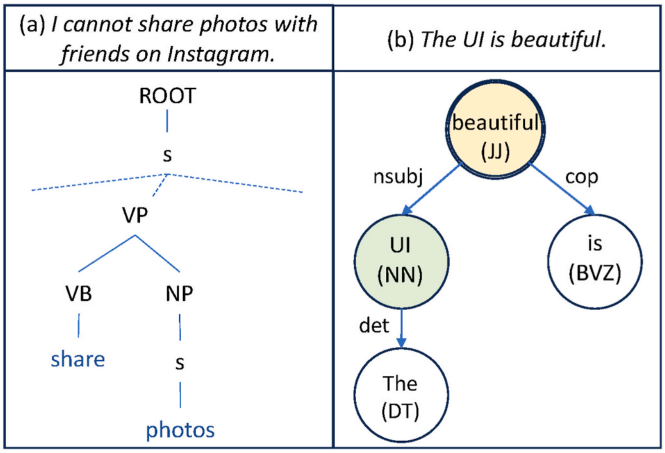

Fig. 7. Examples of (a) linguistic and (b) semantic rules.

et al., 2017) and Where2Change (Zhang et al., 2019b) for automatically localizing function errors in app reviews.

In sum, by defining massive language patterns, rule-based methods can generally result in precision improvements. However, these rules-based methods can only cover a limited portion of all relevant statements and are difficult to expand to other corpora.

4.2.2. *Text clustering*

We identified 25 studies that applied clustering methods for UGC-based RE; the main papers are listed in Table 5. The most widely used types of clustering methods include partitioning clustering (e.g. K-means (Han et al., 2019), (Ji et al., 2023), (de Lima et al., 2022), K-means++ (Jiang et al., 2014a), and X-means (Suryadi and Kim, 2019), (Yang et al., 2023)), density-based clustering (e.g. DBSCAN (Villarroel et al., 2016), (Scalabrino et al., 2017) and HDBSCAN (Park and Kim, 2022)), hierarchical clustering (Cai et al., 2022), (Zhang et al., 2019a), (Jin et al., 2022a), and graph-based clustering (e.g. spectral clustering (Park and Kim, 2022) and affinity propagation clustering (Shi and Peng, 2021)). Other well-known clustering methods include the expectation-maximization (EM) algorithm (Tang et al., 2018) and fuzzy clustering (Liu et al., 2021a). In addition, Jin et al. (2016) used co-clustering to group similar product aspects (i.e. properties or parameters of the product features, such as the meter of a battery) and customer detailed reasons (i.e. explanations written by consumers to support their arguments regarding sentiment polarity, such as "misleading") from online reviews, aiming to provide concise descriptions of customers' mobile phones requirements. Unlike K-means, K-means++, and spectral clustering, which require a predefined cluster number *K*, other clustering algorithms can automatically determine the optimal number of clusters and can thus achieve better performance (Suryadi and Kim, 2019).

Clustering algorithms can be used with input UGC texts at the document, sentence, and word/phrase levels, as shown in Table 5. Among the 25 identified studies in which clustering methods were applied for RE, most ($n = 20$, 81.25 %) were conducted at the word/phrase level. For example, in the MAPP-Reviews (Monitoring App Reviews) method (de Lima et al., 2022), software requirements were extracted from app reviews through the pre-trained RE-BERT (requirements engineering using BERT) model, and then similar requirements were clustered together to reflect the same product feature.

Table 5
Text clustering methods used for UGC-based RE.

Methods	Inputs		
	Document level	Sentence level	Words/Phrases level
K-means	Lobo et al. (2023)	/	(Han et al., 2019), (Jiang et al., 2014a), (Ji et al., 2023), (de Lima et al., 2022)
K-means++	/	/	Jiang et al. (2014a)
X-means	/	/	(Suryadi and Kim, 2019), (Yang et al., 2023)
DBSCAN	(Villarroel et al., 2016), (Scalabrino et al., 2017)	/	Harth et al. (2023)
HDBSCAN	/	/	Park and Kim (2022)
Hierarchical clustering	Tong (2021)	Timoshenko and Hauser (2019)	(Cai et al., 2022), (Zhang et al., 2019a), (Jin et al., 2022a)
Spectral clustering	/	/	Park and Kim (2022)
Affinity propagation	/	/	Shi and Peng (2021)
EM	/	/	Tang et al. (2018)
Fuzzy clustering	/	/	Liu et al. (2021a)
Co-clustering	/	/	Jin et al. (2016)

The P-SRPA (Jiang et al., 2014a) extracted software feature words from Amazon reviews using a rule-based method, then clustered those feature words into 80 clusters through K-means++, and finally obtained 23 system aspects after merging duplicate feature categories; it could achieve an average recall of 86.0 %. The PURA (product-and-user oriented approach) (Cai et al., 2022) first extracted feature words related to automobiles from online reviews using the BiLSTM model, then formed four main clusters and nine subclusters via the agglomerative hierarchical clustering (AHC) model, which outputted a hierarchical structure of the product features in a dendrogram. The model precision (MP) and cluster precision (CP) of the AHC model were 86.1 % and 90.3 %, respectively.

Only three studies were conducted at the document level and two at the sentence level. At the document level, Villarroel et al. (2016) proposed the CLAP (Crowd Listener for releAse Planning), which first classified user reviews as bug reports, suggestions for new features, and others based on random forest, then clustered together related reviews via DBSCAN, and finally obtained all the reviews reporting the same bug and all the reviews reporting the same suggestion for new features; it achieved an overall accuracy of 86.0 %. Scalabrino et al. (2017) extended CLAP to identify fine-grained requirement categories of app reviews, achieving an overall AUC (area under curve) of 96.0 %. At the sentence level, Tong (2021) used Ward's hierarchical clustering to group the embeddings of reviews in the form of sentences for the game No Man's Sky from the NMS Steam community.

Regarding UGC data types, only three studies used online forum data (Tong, 2021), (Tang et al., 2018) and social media data (Han et al., 2019) as data sources. The remaining studies were all conducted using UGC data from online review platforms.

In sum, text clustering methods are good at grouping similar textual content and can help summarize massive user requirements into highlighted results, with the outputs being groups of reviews, sentences, or words/phrases. However, to achieve appropriate clustering results, a predefined cluster number *K* needs to be determined for several clustering algorithms, such as K-means, for which priori knowledge is required. While the majority of previous studies have dealt well with small data and were conducted on sample sizes less than 200 (Cai et al., 2022), (Shi and Peng, 2021), (Han et al., 2019), (Zhang et al., 2019a), (Villarroel et al., 2016), (Tong, 2021), few studies were conducted on data with thousands of feature words/phrases (Park and Kim, 2022), (Harth et al., 2023).

4.2.3. *Topic modeling*

A customer might express multiple requirements for different product aspects in a single review instance. Text clustering methods, which assume that each review instance belongs to exactly one cluster (group), may then become problematic for review instances that exhibit multiple topics (groups) of product aspects. In such cases, topic modeling methods that automatically assign multiple topics to each review instance were used to determine topics and their word distributions related to user requirements.

We identified 54 studies that used topic models for UGC-based RE; some typical papers are listed in Table 6. LDA was the most widely used model to extract topics from UGC data at the document level (user reviews) and sentence level (review sentences) for various products, such as mobile phones (Jeong et al., 2019), (Zhang et al., 2022), (Song et al., 2018), mobile apps (Chen et al., 2014), (Fu et al., 2013), (Zhang et al., 2023d), (Khalid et al., 2014), laptops (Zhang et al., 2023a), (Zhang et al., 2022), hotels (Liu et al., 2022b), (Dewi and Mulyani, 2022), food (Zhang and Huang, 2023), (Zhang et al., 2023c), automobiles (Zeng et al., 2022), Amazon product ecosystem (Zhou et al., 2020b), smart speakers (Du et al., 2022), insurance (Yan et al., 2022), and others (Kovacs et al., 2021), (Song et al., 2021). While LDA is good at dealing with long text, BTM and adaptively online latent Dirichlet allocation (AOLDA) are suitable for short texts such as tweets (Guzman et al., 2017a) and app store reviews (Gao et al., 2018). For example, Wang

Table 6
Topic modeling methods used for UGC-based RE.

UGC type	Method	Document level	Sentence level
Online reviews	LDA	(Guzman and Maalej, 2014), (Zhang et al., 2023a), (Fu et al., 2013), (Zhang et al., 2022), (Song et al., 2018), (Zhang and Huang, 2023), (Du et al., 2022), (Zhang et al., 2023c), (Zhang et al., 2023d), (Song et al., 2021), (Dewi and Mulyani, 2022)	(Chen et al., 2014), (Zhou et al., 2020b), (Zeng et al., 2022)
	STM	(Korfiatis et al., 2019), (Gholizadeh et al., 2022)	/
	NMF	Kumari and Memon (2022)	/
	SToC	Luiz et al. (2018)	/
	ASUM	Carreño and Winbladh (2013)	Chen et al. (2014)
	LSA	Xu (2021)	/
	AOLDA	Gao et al. (2018)	/
	MVLDA	Zhou et al. (2023)	/
	BJ-LDA	/	Guo et al. (2022)
	BTM	/	(Zhang et al., 2023b), (Wang et al., 2023)
	HDP	/	Palomba et al. (2017)
	ETM	/	Yilin et al. (2023)
Social media	LDA	Yan et al. (2022)	/
	BTM	Guzman et al. (2017a)	/
Online forums	LDA	(Jeong et al., 2019), (Takahashi et al., 2015), (Khalid et al., 2014)	/
	NMF	Arora et al. (2023)	
	HDP	/	Zhang et al. (2019b)

et al. (2023) and Zhang et al. (2023b) performed the BTM model after segmenting review documents into short sentences. In addition, hierarchical latent Dirichlet allocation (HLDA) can be used to generate a hierarchical requirement feature structure representation (Zhao and Zhao, 2019), and the ASUM (aspect and sentiment unification model) allows for associating topics with sentiments (Carreño and Winbladh, 2013). Other topic models, such as hierarchical Dirichlet processes (HDP) (Palomba et al., 2017), (Zhang et al., 2019b), structural topic models (STM) (Korfiatis et al., 2019), (Gholizadeh et al., 2022), embedded topic models (ETM) (Yilin et al., 2023), NMF (Kumari and Memon, 2022), (Arora et al., 2023), semantic topic combination (SToC) (Luiz et al., 2018), and latent semantic analysis (LSA) (Xu, 2021), have also been used for UGC-based RE. Specifically, Guo et al. (2022) proposed BJ-LDA (brand joint latent Dirichlet allocation) to extract general and specific topics on product aspects and corresponding opinions, and Zhou et al. (2023) proposed MVLDA (multi-view latent Dirichlet allocation) to extract textual topics for motivations and product-related topics for purchase behaviors.

Instead of performing topic models directly on raw reviews, some researchers extracted topics from the reviews after preprocessing to yield more informative topic words related to user requirements. The preprocessing steps mainly included tokenization, stop word removal, stemming, and lemmatization. In addition, before performing topic models, uninformative review instances were filtered out through ML models (Guzman et al., 2017a), (Chen et al., 2014), (Fu et al., 2013), and feature words/phrases were extracted through rule-based methods (Guzman and Maalej, 2014), (Kumari and Memon, 2022). For example, methods such as Where2Change (Zhang et al., 2019b), AR-Miner (App Review Miner) (Chen et al., 2014), ALERTme (A LittlE bird Told me) (Guzman et al., 2017a), WisCom (Fu et al., 2013), and ChangeAdvisor (Palomba et al., 2017) all involved extracting informative review sentences on mobile apps through ML-based methods, then grouping and analyzing these informative sentences using topic models. Further, the GuMa (Guzman and Maalej, 2014) and MARA (Iacob and Harrison, 2013) models both involved extracting app feature words from online reviews based on rule-based methods and then grouping these features through the LDA model, which led to precision levels of 59.0 % and 85.0 %, respectively. In addition, Gao et al. (2018) proposed the IDEA (IDentify Emerging App issues) framework, in which topics were extracted from online reviews through AOLDA, and then emerging topics were identified by employing a typical anomaly detection method called Jensen-Shannon (JS) divergence. This resulted in a precision of 88.9 % for identifying emerging issues related to six mobile apps on Google Play Store and Apple App Store.

Regarding topic model performance, Zhang et al. (2019b) found that HDP had the best clustering results when compared to LDA, K-means, DBSCAN, SentenceLDA, and CopulaLDA for clustering users' feedback on mobile apps from GitHub. The homogeneity, completeness, and V score of the HDP model were 21.2 %, 16.1 %, and 18.2 %, respectively. Further, Carreño and Winbladh (2013) found that the ASUM achieved an F1 score of 88.26 % for extracting topics with sentiments from app reviews of Calorie Tracker in the Android Market. Chen et al. (2014) found that LDA outperformed ASUM in terms of F-measure (65.7 %) for grouping informative data from app reviews on the Google Play Store. Similarly, Kumari and Memon (2022) showed that LDA clusters were more coherent than NMF clusters when extracting the features of mobile apps in the travel industry. In addition, Guzman et al. (2017a) found that BTM outperformed LDA in grouping short texts, such as tweets on software products, achieving an average model precision of 52.0 % and an average topic precision of 88.0 % for three software applications – Spotify, Slack, and Dropbox.

Thus, topic models are good at summarizing large corpora and arranging semantically related texts into meaningful groups. This helps product owners and developers navigate between different granularities of topics extracted from UGC texts. However, many topic models, such as correlated LDA models, exhibit limitations when dealing with sparse data (such as short texts) and detecting infrequently mentioned features (Guzman and Maalej, 2014). In addition, the number of topics is arbitrary and subject to the interpretability of topic solutions by researchers.

4.2.4. Machine learning

We identified 93 studies that fall into the ML category. Unlike text clustering and topic modeling, which group user-generated texts in an unsupervised manner, a large number of labeled records are employed as training data in ML models to predict the classes of unlabeled records. In the UGC-based RE studies, ML-based classification models were mainly used for classifying user requirements into different categories, and ML-based NER models were usually used to extract requirement-related entities (e.g. product features and user opinions) from unstructured UGC texts.

1) ML-based classification models

ML-based classification models are trained on various features obtained from UGC data (called feature engineering) and are then used to classify user reviews into different classes. The number of classes (requirement categories) in a classification task ranges from 2 to 24. The features widely used for training an ML classification model include TF-

IDF, bag of words (BOW), n-grams, sentiment, and review metadata, as listed in Table 7. The TF-IDF is commonly used to calculate the importance of a word across a set of documents; words with high TF-IDF scores are those that occur frequently in the document and provide the most information about that specific document. The BOW uses individual words of review sentences as classification features, which results in a dictionary of all words used in the corresponding data corpus. It then calculates the presence or absence of each word in the dictionary. N-gram features mainly include unigrams, bigrams, and trigrams. An n-gram feature captures a sequence of *n* words in a sentence that are important for extracting meaningful phrases such as “highly recommend” instead of the individual words “highly” and “recommend”. Review metadata, such as star rating, text length, submission time, and so on, reflect the common information across online reviews. In addition, Kurtanović and Maalej (2017) used the height of a sentence syntax tree and the number of sentence syntax subtrees as features. Li et al. (2018) defined seven heuristic properties (HPs) as certain parts that strongly indicate the possible category of user requests. When using HPs as features for training learning-based classifiers, values for each HP can only be “0” or “1”.

ML classification can be divided into the following types based on the number of classes: binary (two-class), multi-class, and multi-label classification. A binary classification task involves classifying user reviews into two classes, and multi-class classification tasks involve classifying user reviews into three or more requirement categories. Considering that a user review might reflect more than one requirement category (e.g. feature requests, functional complaints, and privacy issues) (Jha and Mahmoud, 2019), the multi-label classification task has been proposed for automatically classifying reviews into one or more relevant categories; that is, a user review can be classified using more than one label.

The majority of binary classification studies in the literature aimed to distinguish informative data (i.e. reviews containing requirement-related information) from uninformative data (i.e. reviews not containing requirement-related information) (Zhang et al., 2021), (Abrahams et al., 2015), (Kauschinger et al., 2023), (Song et al., 2020), (Nadeem et al., 2021), (Stahlmann et al., 2023), (Rahman et al., 2023), and few aimed to classify user reviews as containing FRs and NFRs (Jin et al., 2022b), (Yang and Liang, 2015), (Deocadez et al., 2017). In many other studies, the detected informative data were used to classify detailed requirement categories via multi-class classification methods (Jin et al., 2022b), (Nyamawe et al., 2019) or to extract and group fine-grained product features using text clustering methods (Villarroel et al., 2016), (Jin et al., 2016) and topic models (Fu et al., 2013), (Zhou et al., 2020b).

Table 7
Popular features for training ML classification models.

Feature type	Typical papers
TF-IDF	(Al Kilani et al., 2019), (Jin et al., 2022b), (dos Santos et al., 2021), (McIlroy et al., 2016), (Stanik et al., 2019), (Kauschinger et al., 2023), (Li et al., 2018), (Khan et al., 2019)
BOW	(Tizard et al., 2019), (Jha and Mahmoud, 2018), (Jin et al., 2022b), (dos Santos et al., 2021), (Kauschinger et al., 2023), (Song et al., 2020)
n-grams	(Tizard et al., 2019), (Al Kilani et al., 2019), (Hedegaard and Simonsen, 2013), (Gonzalez et al., 2020), (Li et al., 2018), (Khan et al., 2019), (Nadeem et al., 2021)
review metadata	(Sorbo et al., 2016), (Merten et al., 2016), (Song et al., 2020), (Maalej et al., 2016)
sentiment	(Kurtanović and Maalej, 2017), (Gonzalez et al., 2020), (Stanik et al., 2019), (Maalej et al., 2016), (Kunaefi and Aritsugi, 2021), (Panichella et al., 2016), (Maalej and Nabil, 2015)
lexicon	(Law et al., 2017), (Kengphanphanit and Muenchaisri, 2020), (Panichella et al., 2016)
POS tags	(Tizard et al., 2019), (Stanik et al., 2019), (Kunaefi and Aritsugi, 2021)
word2vec	(Niu et al., 2021), (Sangaroonsilp et al., 2023)
topic words/ keywords	(Tizard et al., 2019), (Sorbo et al., 2016), (Niu et al., 2021)
heuristic properties	(Li et al., 2018), (Niu et al., 2021)

Notably, half of the multi-class classification studies conducted were aimed at obtaining user requirements of mobile apps, focusing on requirement categories such as bug reports, feature requests (new features), information seeking, information giving, problem discovery, shortcomings (issues), ratings, and user experiences (Henao et al., 2021), (Das et al., 2023), (Duan et al., 2021), (Alshangiti et al., 2022), (Zhou et al., 2022). In addition, through multi-class ML models, Binder et al. (2023) automatically identified four Kano model factors from online reviews: basic needs, performance factors, delighters, and irrelevant factors. Shi and Yu (2022) identified five classes of Maslow’s hierarchy of needs: self-actualization needs, esteem needs, love and belongingness needs, safety needs, and psychological needs. Further, many researchers mapped their multi-class classification tasks into sets of binary classification tasks (Tizard et al., 2019), (Stanik et al., 2019), (Qian and Gui, 2021), (Mehder and Aydemir, 2022), (Maalej and Nabil, 2015), (Wang and Li, 2020) and developed a binary classifier for each single class.

Among the multi-label classification studies, the binary relevance (BR) method was the widely used multi-labeling solution for UGC-based RE (Guzman et al., 2017b), (Jha and Mahmoud, 2019), (Bakiu and Guzman, 2017). This method involves assuming label independence and decomposing a problem with *n* classes (labels) into *n* binary problems. In the BR method, a binary classifier is trained for each label, and a union operator is applied to the predictions from these independent classifiers to obtain a final classification result (the union of predictions). Kaur and Kaur (2023) proposed an approach MNoR-BERT (Multi-Label Non-Functional Requirements classification using BERT) and showed that it outperformed the BR method in predicting important user concerns about mobile apps. Ciurumelea et al. (2017) developed the User Request Referencer (URR) prototype to automatically perform multi-label classifications according to predefined taxonomy. In multi-label classification models, each example could be associated with several labels simultaneously, so performance evaluation of these methods is much more complicated than traditional single-label classification methods. While the performance of traditional single-label classification models is usually evaluated based on accuracy, precision, recall, F-measure, and AUC metrics, the most common and widely used metrics for multi-label classification models are hamming loss, hamming score, and subset accuracy (Jha and Mahmoud, 2019), (Nyamawe et al., 2019).

The most popular ML-based classification algorithms used for UGC-based RE include SVM, NB, RF, DT, and LR (Guzman et al., 2017a), (Law et al., 2017), (dos Santos et al., 2021), (Kühl et al., 2020), (Kauschinger et al., 2023), (Song et al., 2020), (Nadeem et al., 2021), which can be applied to solve both binary classification and multi-class classification tasks. Other widely used ML algorithms include MNB (Khan et al., 2022), MLR (Abrahams et al., 2015), k-nearest neighbor (kNN) (Nadeem et al., 2021), maximum entropy (ME) (Gu and Kim, 2015), J48 (Lu and Liang, 2017), relevance vector machines (RVM) (Alshangiti et al., 2022), and XGBoost (Kunaefi and Aritsugi, 2021). In addition, many studies have applied deep learning (DL) methods for classification, including multiple layer perceptron (MLP) (Wang et al., 2022a), (Alturaief et al., 2021), CNNs (Li et al., 2020), LSTM (Zhang et al., 2021), BiLSTM (Wang and Li, 2020), BERT (Henao et al., 2021), (Kaur and Kaur, 2023), and BERT variations such as RoBERTa, RemBERT, ALBERT, and DistilBERT (Das et al., 2023), (Binder et al., 2023), (Stahlmann et al., 2023). For example, Zhang et al. (2021) used the LSTM model to identify innovative sentences from Amazon reviews based on the word embeddings of input review sentences, which achieved an AUC score of 91.0 % and an F1 score of 89.0 %, outperforming traditional machine learning models such as SVM, NB, and LR. Niu et al. (2021) found that LSTM and BiLSTM performed better than gated recurrent units (GRU) and BiGRU and that these were all better than CNN in classifying user requests of software applications into seven categories: security, reliability, performance, life cycle, usability, capability, and system interface. In this study, LSTM achieved the best accuracy of 80.45 % with the WinMerge project on Sourceforge.

2) ML-based NER models

ML-based NER models are usually trained on word embedding representations of UGC texts to locate and classify named entities mentioned in unstructured text into predefined categories. In the RE field, an NER task (also called a sequence annotation task or a token classification task) can be considered a special classification task that involves assigning a label for each word (token) in the input tokens. A popular word labeling strategy is to use the BIO (beginning, inside, outside) format (de Araújo and Marcacini, 2021), (Bian et al., 2022), in which classes B and I represent the beginning and inside of a requirement, respectively, and class O indicates that the word is unrelated to a requirement in the sentence. For example, the software review sentence "The app crashes when I try to share photos" can be labeled as "O O O O O O O B I", which indicates that "share photos" is a software requirement (de Araújo and Marcacini, 2021). In this way, fine-grained requirement statements such as bug entities (Zhou et al., 2020a), product features/aspects (de Araújo and Marcacini, 2021), (Xiao et al., 2022), and user opinions (Jin et al., 2016), (Bian et al., 2022) can be extracted from massive unstructured UGC texts. The BIO format can also be used for labeling multiple entities. For example, the software review sentence "Serious Junk characters on the Restart dialog at the end of the installation on RUS OS" can be labeled as "B-CA B-GUI I-GUI O O B-GUI I-GUI O O B-GUI O O B-CV O B-PF B-PF", in which CA (common adjective), GUI (graphical user interface), CV (common verb), and PF (platform) represent four different types of bug-specific entities, and the class B-GUI combined with the class I-GUI represent a complete GUI entity (e.g. "Junk characters" and "Restart dialog") (Zhou et al., 2020a).

The popular ML-based NER models for UGC-based RE include BiLSTM (Cai et al., 2022), BERT (de Araújo and Marcacini, 2021), CRF (Jin et al., 2016), BiLSTM-CRF (Zhou et al., 2020a), (Lai et al., 2023), BERT-BiLSTM-CRF (Xiao et al., 2022), BERT (Li et al., 2023), and RoBERTa (Han et al., 2022), most of which hold promise in learning contextual word embeddings from long-term dependencies between tokens in sentences. For example, Cai et al. (2022) proposed the PURA, which applied BiLSTM for extracting automobile feature words and related emotional words and negative words from online reviews, achieving an average precision of 96.0 %. Zhou et al. (2020a) proposed the DBNER (a deep neural network for bug specific entity recognition), which applied the attention-based BiLSTM-CRF model for extracting 16 types of bug report entities of software applications and obtained a precision of 92.14 %. Xiao et al. (2022) found that their proposed BERT-BiLSTM-CRF outperformed baseline models such as CNN, CNN + CRF, BiLSTM, BiLSTM + CRF, BERT, and BERT + CRF, achieving the best precision of 91.03 % in extracting positive, negative, and neutral sentiment features of a household air conditioner. Jin et al. (2016) applied CRF to extract product features (e.g. "battery"), product aspects (e.g. "meter"), and customers' detailed reasons (e.g. "misleading") related to mobile phones batteries from online reviews on Amazon and Epinions, which resulting in F1 scores of 99.9 %, 89.3 %, and 57.5 %, respectively. Based on the extracted product features (entities), these features were further clustered into different groups (Jin et al., 2016), (Bian et al., 2022) in some studies, or sentiment analysis was conducted on the features or feature groups (Cai et al., 2022), (Li et al., 2021).

In sum, ML methods have been shown to succeed in extracting user requirements by converting RE tasks into classification tasks or NER tasks. Table 8 shows several representative ML methods and their best or average performance levels for UGC-based RE. Generally, deep neural networks achieved better, more stable performances than traditional ML models in NER tasks, while multi-class classification tasks performed better than multi-label classification tasks. However, manually labeling the category of user reviews for supervised learning approaches is labor intensive and time-consuming. To overcome the difficulties of manually annotating a large amount of data, semi-supervised approaches such as self-training and co-training (Deocadez et al., 2017), PRODWeakFinder (Wang and Wang, 2014), and expectation maximization for naive Bayes

Table 8
Representative ML-based binary classification (BC), multi-class classification (MCC), multi-label classification (MLC), and NER methods and their best/average performance for UGC-based RE.

Ref.	Year	UGC data source	Method name	Target product	UGC-based RE task	Num. of classes	Performance (best/average) (%)			
							Prec.	Rec.	F1	Acc.
Chen et al. (2014)	2014	App Store (Google)	AR-Miner	Mobile apps	BC	2	/	/	87.70	/
Abrahams et al. (2015)	2015	Honda-Tech, ToyotaNation, ChevroletForum	SMART	Automobiles	BC	2	/	80.20	/	71.40
Kühl (2016)	2016	Twitter	Needming	Automobiles	BC	5	95.00	70.50	/	83.70
Guzman et al. (2017a)	2017	Twitter	ALERTme	Software	BC	2	77.00	79.00	/	/
Kengphanphanit and Muenchaisri (2020)	2020	Facebook, Twitter	ARESM	Software	BC	2	51.72	81.08	631.5	65.00
Panichella et al. (2016)	2016	App Store (Apple, Google)	ARdoc	Mobile apps	MCC	5	89.00	89.00	89.00	/
Jha and Mahmoud (2017)	2017	App Store (iOS)	MARC	Mobile apps	MCC	3	89.00	94.00	/	/
Jha and Mahmoud (2018)	2018	App Store (Apple, Google, iOS)	MARC 2.0	Mobile apps	MCC	3	94.00	99.00	96.00	/
Khan et al. (2019)	2019	Reddit	CrowdRE-Arg	Mobile apps	MCC	3	74.00	95.00	83.00	/
Li et al. (2020)	2020	GitHub	DEMAR	Software	MCC	7	/	/	/	83.30
Al-Hawari et al. (2021)	2021	App Store (Apple, Google)	ACRM	Mobile apps	MCC	4	80.60	74.50	77.40	79.10
Duan et al. (2021)	2021	App Store (Apple, Google)	RCBERT	Mobile apps	MCC	4	90.00	89.00	89.00	/
Zhou et al. (2022)	2022	App Store (iOS)	FSTL	Mobile apps	MCC	3	97.00	94.00	95.00	/
Khan et al. (2022)	2022	Reddit	CrowdRE-VArg	Mobile apps	MCC	3	84.50	46.20	59.60	/
Cong et al. (2023)	2023	JD	ERNIE	Household	MCC	8	92.48	88.45	89.34	/
Nyamawe et al. (2019)	2019	GitHub, JIRA	FR-Refactor	Software	MLC	14	73.00	52.00	59.00	/
Zhang et al. (2017)	2017	360 Mobile Assistant	CSLabel	Mobile apps	MLC	17	66.50	69.80	69.80	/
Kaur and Kaur (2023)	2023	App Store (Apple, Google)	MNoR-BERT	Mobile apps	MLC	4	94.00	60.00	73.00	/
Wang and Wang (2014)	2016	Amazon	PRODWeakFinder	Camera	NER	11	86.74	85.76	86.25	/
Zhou et al. (2020a)	2020	Bugzilla	DBNER	Software	NER	16	92.14	89.60	90.85	/
de Araújo and Marcacini (2021)	2021	Play Store, Amazon Store	RE-BERT	Mobile apps	NER	3	/	/	81.00	/
Cai et al. (2022)	2022	Autohome	PURA	Automobiles	NER	4	87.00	74.00	80.00	/
Xiao et al. (2022)	2022	Taobao, Tmall, JD	BERT-BiLSTM-CRF	Household	NER	7	91.03	97.43	94.12	/

(EMNB) (Chen et al., 2014) have been applied for classification in some studies. This ensure that only a small amount of labeled data is needed to achieve results similar to classical supervised techniques in which all data are labeled (Deocadez et al., 2017).

4.2.5. Other methods

Besides the four main categories of methods mentioned above, we identified several other methods used for UGC-based RE, including network analysis, sentiment analysis, the Kano model, QFD, goal models, and customer preference models.

1) Network analysis

Some researchers have addressed the feature words clustering problem by detecting communities in a semantic network, in which nodes represent words/phrases of product features (or user opinions) and edges represent the relationships (e.g. co-occurrence) among these features (or user opinions) (Park and Lee, 2011), (Jiang et al., 2014b). For example, Jiang et al. (2014b) constructed a semantic network of important keywords related to the KIS 2011 software from Amazon and then clustered the network nodes into groups using the Girvan-Newman (GN) algorithm. Yoon et al. (2018) extracted user opinions (i.e. feature-sentiment pairs) about shoes from Twitter for network construction and then used the VOSviewer tool to find communities. Park and Lee (2011) developed a map of customer needs for mobile phones to visualize relationships among them. This map was constructed based on the co-occurrence frequencies of all pairs of keywords extracted from online reviews on MobilephoneSurvey. In addition, Sun et al. (2023) proposed a Network Analysis-Importance Performance Analysis (NA-IPA) model to improve product design by mapping customer requirements to configuration plans.

2) Sentiment analysis

Sentiment analysis has been widely used to calculate user sentiment polarity (positive, negative, or neutral) associated with product features or feature groups (Fu et al., 2013), (Carreño and Winbladh, 2013), as these data can help analysts understand customer satisfaction and prioritize user requirements. For example, Zhang et al. (2012), (Nikumanesh and Fathi, 2017) declared that pressing issues and customer dissatisfaction are more likely to be associated with a negative sentiment polarity. The process of assigning a quantitative value to a piece of text expressing a mood can, in general, be addressed by rule-based methods such as VADER (Lobo et al., 2023), lexical sentiment analysis methods such as SentiStrength (Guzman et al., 2017b), (Zhang et al., 2012), (Nikumanesh and Fathi, 2017), and ML models such as SVM (Ji et al., 2023), NB (Ireland and Liu, 2018), MNB (Al Kilani et al., 2019), J48 (Tang et al., 2018), LSTM (Du et al., 2022), GRU (Mu et al., 2021), Transformer (Das et al., 2023), and BERT (Li et al., 2023).

Wang and Wang (2014) proposed the PRODWeakFinder, an opinion-aware analytical framework, to identify the product weaknesses of digital cameras through a sentiment analysis of Amazon reviews. In the PRODWeakFinder, the final evaluation score of each product aspect is calculated based on the authority score for comparative opinions (e.g. "the shutter of this camera is much slower than that of the D60") and the sentiment score for noncomparative opinions (e.g. "the shutter is slow"); features with evaluation scores less than a given threshold are considered product weaknesses. Cai et al. (2022) calculated the customer satisfaction scores for product features of automobiles based on their sentiment scores (attitude towards a product aspect) and attention scores (degree of attention that a user paid to a product aspect); product features with low satisfaction scores were then viewed as improvement requirements.

3) The Kano model

The Kano model has been used in many studies to understand the product attributes that customers need and to clarify their importance for customer satisfaction, thereby helping designers acquire customer preference information and develop appropriate product improvement strategies (Qi et al., 2016), (Zhang et al., 2022), (Song et al., 2018), (Jin et al., 2022a), (Wen and Chen, 2020). By evaluating the impact of product attributes on customer satisfaction, the Kano model maps product attributes into five categories: must-be requirements (also called basic requirements), which are necessary product features; one-dimensional requirements (also called performance requirements), which refer to features for which customer satisfaction has a positive linear relationship with that of the overall product; attractive requirements (also called excitement requirements), which are features that exceed user expectations; indifferent requirements (also called non-differentiated requirements or neutral requirements), which are features whose availability/existence has no impact on users' satisfaction with the product; and reverse requirements, which are the features that users don't need and are contrary to one-dimensional attributes in that the relationship between feature satisfaction and overall product satisfaction is negatively linear. Among the above five categories, attractive, one-dimensional, and must-be requirements are desirable attributes for product development, whereas reverse and indifferent requirements are less desirable.

The Kano model is usually conducted through offline questionnaire surveys (Zhang and Huang, 2023), which are time-consuming and inefficient. In the internet era, based on the product features and user opinions extracted from massive online UGC text in an automated manner, many researchers have built sets of attribute classification rules based on the sentiment orientation of review texts to map the product features onto Kano requirements (Zhang et al., 2023a), (Liu et al., 2022a), (Zhang et al., 2023b), (Zhang et al., 2023c). For example, Qi et al. (2016) measured the impact of the product attributes of mobile phones (extracted from online reviews from JD) on consumer satisfaction (calculated by sentiment analysis) via conjoint analysis and then proposed several rules for categorizing these product attributes as must-be quality, one-dimensional quality, attractive quality, and reverse quality attributes based on the consumer preference levels of positive and negative sentiments. Hsiao and Hsiao (2020) considered that the positive or negative user-generated reviews of hotel attributes from Booking.com reflected customers' satisfaction or dissatisfaction, respectively. On this basis, properties that appeared only in positive reviews were determined to be attractive quality characteristics; properties that were mentioned only in negative reviews were considered must-be characteristics; and properties that appeared in both positive and negative reviews, mostly with positive words and negative words, respectively, were determined to be one-dimensional characteristics.

4) QFD, goal models, and customer preference models

In RE tasks, QFD is used for mapping customer requirements onto product functions and structures (Liu et al., 2021a), (Song et al., 2018). For example, Liu et al. (2021a) mapped and analyzed the product structures and customer needs of mobile phones via QFD. Further, Liu et al. (2022a) proposed a new QFD method based on the extended Kano model and social network analysis by considering the characteristics of online shopping reviews and the social relationships between an enterprise's departments, following which a case study of mobile phone design was used to verify model practicality.

Goal models are used for describing product features and their relationships to thereby support the subsequent product development process (Ren et al., 2022). For example, Svee et al. (Svee and Zdravkovic, 2016) mapped conceptualized consumer preferences in the airline industry from Twitter onto high-level system requirements by completing a goal model that represented the system requirements supporting the important values of end users.

Customer preference models are used to model customer preferences from UGC for product design selection; the product specifications serve as useful links between customer preferences and product design (Zhang et al., 2019a), (Gao et al., 2018). For example, Wang et al. (2011) developed a customer preference model from web-based UGC while accounting for the heterogeneity of customer preferences and used this model for customer-driven smartphone product design selection.

The above three types of methods can be applied to map product attributes onto engineering features, which corresponds to the product design process in the proposed V-shaped UGC-based RE research framework.

4.3. Requirement representations

In this section, we present three types of requirement representations obtained from UGC data: macro-level requirement data, meso-level requirement categories, and micro-level requirement statements. We introduce each type of requirement presentation and also highlight the relationships between different requirement representations.

4.3.1. Requirement-related data

Considering that not all user-generated reviews are relevant to user requirements, online reviews have been classified as informative and uninformative reviews in some studies so that high-quality data related to user requirements could be obtained. “I hope the size of the cup is larger” is an example of a typical requirement-related review.

In most previous studies, the researchers identified whether user-generated reviews were requirement-related or not (de Araújo and Marcacini, 2021), (Kühl, 2016), (Stahlmann et al., 2023). Specifically, some user-generated reviews were identified as being related to a specific requirement category, such as privacy requirements (Sangaroonsilp et al., 2023), product defects (Abrahams et al., 2015), feature/enhancement requests (Kauschinger et al., 2023), (Song et al., 2020), updated features (Liu et al., 2021b), suggestive intent (Jhamtani et al., 2015), innovative ideas (Zhang et al., 2021), quality attributes (Ali et al., 2019), user rationale (Khan et al., 2020), and software change requests (Nadeem et al., 2021).

However, these requirement-related contents were coarse in granularity and thus had limitations in reflecting the detailed demands of users. Product designers need look through such reviews to understand customer needs. Therefore, many researchers first identified requirement-related data and then classified them into several subcategories (Guzman et al., 2017a), (Kühl, 2016), (Jha and Mahmoud, 2019).

4.3.2. Requirement categories

Researchers have classified requirement-related data into several requirement categories (types). The number of requirement categories was often predefined according to the studied products or services, ranging from 2 to 24. The most popular requirement categories investigated in previous studies are summarized in Table 9.

For mobile applications and software products, researchers have usually classified the related UGC data into FR and NFR categories (Yang and Liang, 2015), (Deocadez et al., 2017). Furthermore, the NFRs can be classified based on usability, security, reliability, portability, supportability, performance, dependability, compatibility, dependability, maintainability, and frequency (Jin et al., 2022b), (dos Santos et al., 2021), (Lu and Liang, 2017), (Kaur and Kaur, 2023). This classification approach is determined by systems functions and constraints, with more attention paid to the product itself. From the perspective of user concerns and review content, software/app reviews have been classified as feature/improvement requests, bug/problem reports, problem discovering, information giving, information seeking/inquiry, feature shortcomings, user experiences, and ratings in the majority of previous studies (Henao et al., 2021), (Jha and Mahmoud, 2017), (Sorbo et al., 2016), (Alshangiti et al., 2022). For example, Araujo et al. (2022) and Duan (Duan et al., 2021) classified mobile app reviews as feature

Table 9
Explanations for the popularly investigated requirement categories.

Requirement category	Explanation and examples
Functional requirements	• Functions that a system must be able to perform (Yang and Liang, 2015) (e.g. “at least give me the option of how I would prefer it to look.”) • Requirements that describe the intended actions of the system (Ali et al., 2019)
Non-functional requirements	• A set of specific qualities other than functionality, but the requirements which users expect the application to meet, such as performance, usability, reliability, and security, etc. (Yang and Liang, 2015) (e.g. “the loss of the bookshelf look, the boring and ugly flat design plus the stark white background make it extremely difficult to read anything on this app.”) • Requirements that define the overall behavior and constraints of the system (Ali et al., 2019)
Feature requests	• Comments that suggest/ask for/request for a new feature or functionality for a product or express preferences/innovative ideas for the re-design of features that already exist (Guzman et al., 2017a), (Henao et al., 2021), (Iacob and Harrison, 2013) or describe the way the users would like the product to behave (Tizard et al., 2019) (e.g. “I want to put a time line to some events in the movie.”) • Comments that express ideas, suggestions, or needs for improving or enhancing the product or its functionalities (Panichella et al., 2015), (Panichella et al., 2016) (e.g. “If you add separate Tabs for video and photo we’ll be very happy.”) • Comments that ask for missing functionality (e.g. provided by other apps) or missing content (e.g. in catalogs and games), share ideas on how to improve the product in future releases by adding or changing features (Maalej et al., 2016), (Maalej and Nabil, 2015) (e.g. “Please add the feature of adding video clip.”)
Bug reports	• Comments that report on bugs, errors, flaws, failures, faults, potential problems, or bad experiences of a software/app (Guzman et al., 2017a), (Williams and Mahmoud, 2017) (e.g. “I have never ever seen the auto-update function of chrome work on any of all my computers.”) • Comments that describe problems with the app that should be corrected, such as a crash, an erroneous behavior, or a performance issue (Maalej et al., 2016), (Maalej and Nabil, 2015) (e.g. “I liked very much the upgrade to pdfs (divisions and search). However, they aren’t displaying anymore. Fix it and it will be perfect.”)
Problem reports	• User comment that describes a concrete problem or bug in the app (Henao et al., 2021) (e.g. “Since the recent update I cannot upload new files to my account.”) • User feedback that states a concrete problem related to a software product or service (Stanik et al., 2019) (e.g. “Since the last update the app crashes upon start.”)
Problem discovery	• Comments that express dissatisfaction or describe issues, unexpected behaviors with the app/software (Panichella et al., 2015), (Panichella et al., 2016) (e.g. “I developed a Class IV allergic reaction to the wrist band.”) • Comments that reflect problems during the phase of installation and updating, or bugs and issues while using the app/software (Tizard et al., 2019) (e.g. “I keep getting an input error after downloading the update.”)
Information giving	• Comments that express satisfaction or inform/update other users, developers, or sellers about the aspect/characteristic/functionality of the product (Panichella et al., 2015), (Panichella et al., 2016), including application guidance, user setup, praise for application, dispraise for application, application usage, attempted solution, acknowledgment of resolution (Tizard et al., 2019) (e.g. “Alexa does not answer general questions as Google Home seems to”, “It’s simple the desktop app is great too” and “It’s so useful and fast and I just love the dark theme”)
Information seeking	•Comments that express the users’ wants to get information or assistance/help from developers or other users (Panichella et al., 2015), (Panichella et al., 2016), including questions on the application, help-seeking, requesting more information, and questions on background (Tizard et al.,

(continued on next page)

Table 9 (*continued*)

Requirement category	Explanation and examples
	2019) (e.g. "I want to know that how to add and delete text and pictures.")
Inquiry	• User feedback that asks for either new functionality, an improvement, or requests information or help for support (Stanik et al., 2019) (e.g. "It would be great if I could invite multiple friends at once.")
Feature shortcomings	• Unsatisfying aspects of an existing feature (Guzman et al., 2017a) (e.g. "It makes me extremely uncomfortable when people I don't know poke me on Facebook.")
User experiences	• Comments that combine *helpfulness* and *feature information* content, which reflects the experience of users with the product and its features in certain situations (Maalej et al., 2016), (Maalej and Nabil, 2015) (e.g. "This is a great little app; especially for those with hectic schedules, it keeps you in like for visual people like me.")
Ratings	• Simple text reflections of the numeric star rating, which are less informative as they only include praise, dispraise, a distractive critique, or a dissuasion (Maalej et al., 2016), (Maalej and Nabil, 2015) (e.g. "Very nice app")
User requirements	• Comments that mainly include requests for new features, or express that a recently added feature is undesirable (Williams and Mahmoud, 2017) (e.g. "pls make it to where I can see an individual score with someone so I know how many snaps we've sent back & forth!")
Claims	• Supporting and attacking arguments for an issue, an alternative, a feature, or a topic under discussion, including supporting claims, attacking claims, and neutral claims (Khan et al., 2020), (Khan et al., 2019) (e.g. "Google app can afford two tabs and google assistant is a different team product, it's impossible for Google to have a good integration across all products.")
Issue	• A question/challenge in response to the main discussion topic, a certain design alternative, or a new feature proposed by others; or a problem that needs further discussion and elaboration (Khan et al., 2020), (Khan et al., 2019) (e.g. "Why does Google Maps have a tab in the first place?")

requests, bug reports, user experience, and ratings. Al-Hawari et al. (2021) and Tizard et al. (2019) identified feature requests, problem discovery, information giving, and information seeking (or inquiry) requirements from online reviews for mobile applications and software products, respectively. In addition, Guzman (Guzman et al., 2015) classified mobile app reviews as bug reports, feature strengths, feature shortcomings, user requests, praises, complaints, usage scenarios, and noises. Kurtanovic and Maalej (Kurtanović and Maalej, 2017) defined five user rationale concepts for software applications, namely issues, alternatives, criteria, decisions, and justifications, to describe the reasons for human decisions, opinions, and beliefs. On this basis, Kunaefi and Aritsugi (2021) further classified user decisions as acquire, recommend, rate, request, and relinquish decisions.

For electronic products and other products, researchers have paid particular attention to user requirements for specific product attributes. For example, Wang et al. (2022a), (Wang and Li, 2020) classified the product specifications mentioned in Amazon reviews of laptops as processors, hard disks, random-access memory (RAM), monitors, and graphics processors. Further, Kühl et al. classified tweets containing the consumers' needs for electric vehicles into four major categories: cost-related, car-related, charging-related, social-related, and other categories (Kühl, 2016). They then expanded these into seven categories: price, car characteristics, charging infrastructure, range charging technology, environment and health, society, and others (Kühl et al., 2020). Qian and Gui et al. (Qian and Gui, 2021) classified the health information needs associated with medical systems into four categories: coping with aging, dietary nutrition, physical exercise, and mental health.

Many researchers have focused on a specific requirement type and identified several subcategories of requirements. For example, Guzman et al. (2017a) classified improvement requests for three software applications (Spotify, Dropbox, and Slack) as bug reports, feature shortcomings, and feature requests. Law et al. (2017) classified the defects of dishwashers into the categories of performance defects, safety defects, and no defects. Iacob et al. (2014) classified the bugs of mobile apps as major, medium, and minor bugs. Khan et al. (2019) classified Google Maps claims as supporting, attacking, and neutral claims. Li et al. (2018) and Niu et al. (2021) classified user requests into the categories of security, reliability, performance, lifecycle, usability, capability, and system interface.

In addition, many researchers have identified requirement categories based on the functions and features of target products. For example, Man et al. (2016) identified seven types of app issues from online reviews: battery, crash, memory, network, privacy, spam, and UI. Cong et al. (2023) identified the appearance, function, and emotion requirements associated with a smart cat feeder from online reviews; the appearance class included shape, color, and material, the function class included interactive operation, intelligent feeding, installation, and maintenance, and the emotion class included value for money and practicality. Alturaief et al. (2021) classified app reviews from three domains (games, productivity, and social networking) into 12 aspect categories. McIlroy et al. (2016) and Zhang et al. (2017) identified 14 and 17 types of app issues, respectively, while Zhou et al. (2020a) identified 16 types of software bug entities. Further, Bakiu and Guzman (2017) and Hedegaard and Simonsen (2013) both classified the usability and user experience (UUX) of software applications into 24 dimensions.

Notably, we found three studies in which data assessments were addressed during requirement category classification. Das et al. (2023) used some specific properties of uninformative reviews as criteria for eliminating such reviews from online review data. Williams and Mahmoud (2017) classified tweets regarding software systems into three categories: bug reports, user requirements, and miscellaneous and spam. In addition, Gonzalez et al. (2020) classified tweets about the Miniso brand into four categories: positive comments towards the brand (POS), negative comments towards the brand (NEG), Miniso advertising (PUB), and customer requests to open new branches (SUC). Among all these tweet categories, miscellaneous and spam and Miniso advertising could help filter out useless and unrelated online reviews for RE tasks.

In sum, the various requirement categories mentioned above can help product designers focus on specific aspects of product design and improvements in requirements engineering. These requirement categories can provide developers with a straightforward understanding of how to prioritize the concerned products. However, the standards for requirement classification have not been unified yet, and there are overlaps in the definitions of some requirement categories due to differences in research objects and objectives. As listed in Table 9, the categories "bug reports", "problem reports", "problem discovery", "feature shortcomings", "claims", and "issue" are almost similar; they all reflect users' dissatisfaction with products. The categories "feature requests", "inquiry", and "user requirements" are also similar in revealing user demand for new features or feature improvements.

4.3.3. Requirement statements

Researchers have previously extracted users' opinions and sentiments (positive/advantage or negative/disadvantage) regarding specific product aspects (feature words/phrases) or aspect clusters (a group of features) from unstructured user-generated reviews. Users' opinions and sentiments on these aspects have been used to help understand and prioritize user requirements (Cai et al., 2022). Considering this, we classified micro-level requirement statements into four types: product aspects, aspect clusters, ranked aspects (clusters), and aspects (clusters) with opinions and sentiments.

1) Product aspects

Many researchers have regarded the feature words/phrases, topic words, and key terms from UGC data as user requirements (Xu, 2021), (Liu et al., 2022a), (Zhou et al., 2023), (Dalpiaz and Parente, 2019), (Song et al., 2021), (Korfiatis et al., 2019), (Zeng et al., 2022), (Chen et al., 2021), (Khalid et al., 2014), (Iacob and Harrison, 2013), such as "camera", "call quality", "email", "OS", "price", "screen", "ease of use" for a mobile phone, which were usually obtained using rule-based methods and topic models. Specifically, Jin et al. (2022a) and Wang (2022) identified the Kansei (a Japanese vocabulary that expresses users' psychological feelings and imaginations of new products) words for electronic products and smart capsule coffee machines, respectively. Vlas and Robinson (2012) represented software product requirements using SAO triples. For example, the statement "the submit form button should send the form data to the processing component" could be represented by the SAO-triple (*submit form button – should send – form data*). Wang et al. (2022b) decomposed a POS tagged review sentence for electronic products into a subject-verb-object (SVO) triple composed of a prefix, a need, and a suffix. For example, the sentence "I tend to take a lot of shots indoors" could be represented by an SVO whose prefix is "I tend to" and need is "take a lot of shots indoors". Suryadi and Kim (2019) identified a usage context for laptops using bigrams (word pairs) such as "docking station", "learning curve", and "playing games".

2) Aspect clusters

Many aspects extracted from massive amounts of UGC data describe the same or similar product features. For example, phrases like "GB memory", "storage capacity", "internal memory", "extra memory", "additional storage", and "storage space" all refer to the memory of a smartphone; and "waiter", "waitress", "server", "bartender", and "chef" all refer to the staff in a restaurant. Therefore, many researchers reduced the dimensionality of the extracted aspects using text clustering methods (Cai et al., 2022), (Yoon et al., 2018), (Tong, 2021), (de Lima et al., 2022), (Yang et al., 2023), (Park and Kim, 2022), network analysis (Ji et al., 2023), goal models (Svee and Zdravkovic, 2016), (Ren et al., 2022), and topic models (Palomba et al., 2017). This allowed similar product aspects to be grouped into the same aspect cluster. For instance, Timoshenko and Hauser (2019) first automated the identification of informative sentences for oral care products using a CNN classifier and then clustered them into different groups using Ward's hierarchical clustering method. Finally, the customer needs were manually extracted by screening the selected sentences from different groups. These kinds of aspects or aspect clusters can help product developers summarize the key information on user requirements instead of screening every online review.

3) Ranked aspect (clusters)

Ranked aspects or aspect clusters can direct product development and improvement (Kovacs et al., 2021). For this purpose, various measurements have been proposed for requirements prioritization. The commonly used ranking indexes include attention degree (Cai et al., 2022), (Han et al., 2019), (Zhang et al., 2018), satisfaction degree (Jeong et al., 2019), (Dewi and Mulyani, 2022), (Harth et al., 2023), and importance level (Ireland and Liu, 2018), (Du et al., 2022), (Chen et al., 2019), (Yang et al., 2023). Attention degree reflects users' interest or reaction to an aspect, and it is calculated as the number of reviews (or sentences) mentioning this aspect divided by the total number of reviews (Cai et al., 2022), (Zhang et al., 2018), or based on a statistical analysis of social network service (SNS) information such as forwards, likes, and comments (Han et al., 2019). Satisfaction degree is determined by the number of positive reviews (Zhang et al., 2018) or the sentiment score (Jeong et al., 2019) for a product aspect, or it is calculated by a weighted function of the attention degree and sentiment score (Cai et al., 2022). The importance level of an aspect or aspect cluster is measured by the frequency of a noun phrase (Ireland and Liu, 2018), or as the number of reviews divided by the average review rating (Chen et al., 2019).

Jeong et al. (2019) proposed the opportunity score as a way to direct further product development; it is evaluated by applying an opportunity algorithm based on the importance of and customers' satisfaction with a product topic. Similarly, Dewi and Mulyani (2022) identified service opportunities based on the importance and satisfaction levels of hotels to determine their underserved, appropriately served, and overserved needs. Zhang et al. (2023b) proposed an opportunity algorithm to explicitly determine the software development priorities of unfulfilled requirements. Zhang et al. (2019a) proposed a redesign index to measure the priority of redesigning different features. Further, Scalabrino et al. (2017) proposed a set of prioritization indicators for prioritizing review clusters for mobile apps. Gao et al. (2018) prioritized various topics based on their semantic scores and sentiment scores. Kim and Noh (2019) and Park and Lee (2011) ranked the factors they extracted based on their influence power, which was calculated using network analysis indexes such as degree centrality, eigenvector centrality, and weights of edges. Eldin et al. (2021) ranked product aspects based on several factors such as feature weight, user opinion importance, and sentiment polarity.

4) Aspect (clusters) with opinions and sentiments

Many researchers have extracted customer opinions (e.g. reviews/sentences/emotional words) on product aspects (clusters) and then determined user sentiments (positive/negative/neural) based on these opinions through sentiment analysis methods (Guzman and Maalej, 2014), (Xiao et al., 2022), (Fu et al., 2013), (Carreño and Winbladh, 2013), (Zhang et al., 2022), (Zhang et al., 2023b), (Du et al., 2022), (Zhang et al., 2023c), (Bian et al., 2022), (Jian et al., 2016), (Luiz et al., 2018). Product aspects with low sentiment scores or low user satisfaction levels were regarded as product weaknesses or improvement requirements (Cai et al., 2022), (Zhang et al., 2012).

Based on fine-grained analyses of product features and their corresponding user opinions and sentiments, many researchers have further categorized user requirements into must-be, one-dimensional, attractive, indifferent, and reverse requirements based on the Kano model (Qi et al., 2016), (Hsiao and Hsiao, 2020), (Zhang et al., 2023a), (Liu et al., 2022a), (Zhang et al., 2022), (Zhang et al., 2023b), (Zhang et al., 2023c), or they mapped user requirements onto engineering characteristics of product structures via QFD (Liu et al., 2021a) and intuitionistic fuzzy quality function deployment (IFQFD) (Song et al., 2018).

In sum, micro-level requirement statements about specific product aspects can provide detailed, refined information about user requirements and be grouped into several clusters using text clustering methods for understanding and managing high-level user requirements. When combined with corresponding user opinions and sentiments, these product aspects can be further prioritized according to their importance and urgency for product design improvements. Specifically, based on these fine-grained requirement statements, evolutionary requirements documents (Jiang et al., 2014b) and user comment reports (Carreño and Winbladh, 2013) can be summarized to obtain detailed, meaningful, and comprehensive feedback on customer requirements. Thus, requirement statements can help product designers save time by searching only for important user requirements instead of reading massive user reviews.

4.4. Relationships between RE methods and requirement representations

The number of studies in which the five types of RE methods were used to obtain the different requirement representations is shown in Fig. 8. As can be seen, macro-level requirement-related data were mainly identified using ML-based binary classification methods, and meso-level requirement categories were often identified using multi-class or multi-label classification methods based on ML algorithms. Both requirement-related data and requirement categories describe user

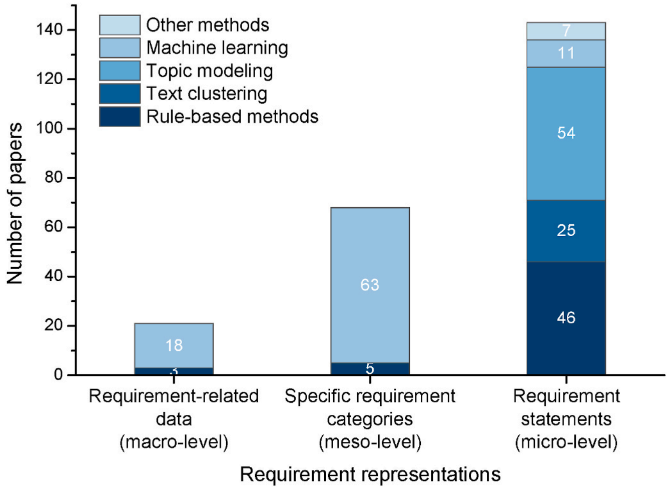

Fig. 8. RE methods used to obtain different requirements representations.

requirements at the review or sentence level, but in the requirement categories, each review or sentience is clearly classified into one specific type of user requirement. For example, software or app reviews are usually classified into several categories such as feature requests, bug reports, user experience, and ratings. At the micro level, the majority of studies have extracted fine-grained product features along with corresponding user opinions and sentiments from unstructured UGC texts through rule-based methods, topic models, and ML-based NER models. These requirement statements were usually obtained for products or services that had complex and varied product structures or characteristics and lacked strict boundaries or guidance for requirement classification, such as automobiles, electronic products, IoT products, hotels, and restaurants.

The cross application of multiple methods in RE processes has allowed for obtaining different levels of requirement representations, although ML methods have usually been used for requirement data identification and requirement category classification, and rule-based methods, text clustering methods, and topic models have often been applied for requirement statement extraction. For example, ML and DL methods have been widely used to obtain requirement-related data (de Araújo and Marcacini, 2021), (Guzman et al., 2017a), (Chen et al., 2014), (Kengphanphanit and Muenchaisri, 2020), requirement categories (Gu and Kim, 2015), (Li et al., 2021), (Jha and Mahmoud, 2017), (Panichella et al., 2016), and requirement statements (Cai et al., 2022), (Xiao et al., 2022), (Li et al., 2021). In many studies, informative sentences were first detected using an ML algorithm and then classified into several groups via topic models such as BTM (Guzman et al., 2017a) and LDA (Chen et al., 2014), (Fu et al., 2013), or clustering methods such as Ward's hierarchical clustering (Timoshenko and Hauser, 2019), co-clustering (Jin et al., 2016) and DBSCAN (Villarroel et al., 2016).

5. Conclusions

In this paper, we presented a comprehensive review of UGC data inputs, applied RE methods, and requirement representation outputs for UGC-based RE. By analyzing 232 relevant publications, we identified three main UGC data sources – online review data, social media data, and online forum data – and five primary RE methods – rule-based methods, topic modeling, text clustering, machine learning, and other methods. To show how the user requirements elicited from UGC data are formulated, we summarized three types of requirement representations according to their formulation and granularity: requirement-related data, requirement categories, and requirement statements. The results of our study showed that online review data constitute the most widely used UGC data source for eliciting customer requirements. Further, most of the obtained user requirements in previous studies were aspect-level requirement statements that provide detailed, refined information about users' demands for specific product attributes. Based on these fine-grained requirement statements, requirement documents and user comment reports were generated to enable developers to carry out product design and improvement. By analyzing the cross relationships between the UGC data (inputs), RE methods, and requirement representations (outputs), we found that ML methods have been widely used for identifying requirement-related data and classifying the data into different requirement categories, while rule-based methods, topic modeling, and text clustering methods have mainly been applied to extract requirement statements from massive unstructured UGC text.

In addition, we established an overall research framework for UGC-based RE; it forms a "V" shape and consists of eight sequential RE-related processes. We included all the UGC data types, RE methods, and requirement representations in this framework to clarify the inputs and outputs involved in the whole RE-related process. The contribution of the UGC-based RE research framework is twofold. First, it summarizes the UGC data sources, RE methods, and requirement representations in the UGC-based RE field and can thus provide guidance for product developers to collect and preprocess UGC data, extract and prioritize user requirements, and obtain product structure characteristics for product development and improvement. Second, the framework is in a "V" shape that is intuitive and clear for beginners and practitioners aiming to gain an in-depth understanding of current research topics and the status of the UGC-based RE field from both macro and micro perspectives.

6. Future work

UGC-based RE has attracted growing attention and has been applied in various domains in recent years. However, there are still many important topics and challenges that require further research due to the limited representativeness and reliability issues of UGC data, the inherent ambiguity and complexity of natural language, and the ever-changing demands of customers.

1) *Data representativeness*

Due to a lack of representativeness, there is a serious level of sampling bias in user requirement data sources obtained from the internet. To carry out RE based on UGC data, it is necessary to study the reliability of internet data sources, the coverage of the target customers, and the representativeness of collected data.

2) *Multisource data fusion*

With the development of e-commerce, it is common to find users' feedback on the same product on multiple internet platforms. Therefore, when selecting data sources for eliciting user requirements, the characteristics of both the products and internet platforms should be taken into consideration. It is also necessary to collect as much related UGC data as possible from different sources. Multisource heterogeneous data fusion methods should be developed to integrate data with different types, characteristics, and qualities. In addition, the UGC data obtained from photos, videos, audio, and other formats can be combined and analyzed to gain a more comprehensive understanding of user requirements and preferences.

3) *Data credibility and review helpfulness*

In the era of big data, more and more users tend to publish comments on various products through online platforms (Cai et al., 2022). However, a substantial number of reviews involve deceptive opinions to promote or demote certain products (Heydari et al., 2015); these are called spam or fake reviews. Therefore, before conducting RE based on UGC data, it is necessary to filter out spam reviews. In addition, due to the low-value density of UGC data – that is, because not every review contains valuable information about user requirements – the helpfulness of UGC data should be assessed to improve data quality for RE. Although review spam detection methods and review helpfulness analysis methods have been widely researched in the information science field, they have rarely been studied for RE tasks. In the future, to improve data quality for RE, the latest review spam detection models, such as ABCM (aspect-based classification method) (Cai et al., 2023), and the state-of-the-art review helpfulness prediction methods proposed by Yang et al. (2021), Malik (2020), and Ren and Hong (2018) should be explored.

4) *Implicit requirements elicitation*

Product features can be classified as explicit features and implicit features. Implicit product features include words that describe product features or functional attributes that do not appear clearly in user comments but can be determined by understanding the semantics. For example, "expensive" refers to the product price. Recognizing implicit product features can lead to detailed, comprehensive information about the product features that users are concerned about and can improve the accuracy of sentiment analysis. However, the natural language used in online reviews is very complex, and there is a lack of high performance implicit RE models.

5) *Dynamic requirement analysis*

The existing literature is mainly focused on RE based on slice data for a certain time period, with which it is difficult to predict the future needs of users. However, user requirements are dynamic and constantly change during a product life cycle, and few studies have focused on mining the dynamic evolution of customer requirements (Cai et al., 2022), (Fu et al., 2013), (Tong, 2021), (de Lima et al., 2022). To guide product improvement iterations, various methods for the analysis, tracking, and prediction of dynamic requirements should be studied. In addition, the characteristics and general laws of requirements evolution need to be further analyzed for different products and customer groups.

6) *Exploration of novel methods*

Requirements elicitation involves multiple stakeholders, such as customers, product owners, and product developers and designers, as well as internet platforms that manage massive amounts of UGC data. The research and development of requirements engineering are system engineering problems. So far, machine learning and deep learning methods have been widely used to extract informative content related to requirements, but there are still other methods for extended research and application, such as complex network analysis, knowledge graph analysis (Wang et al., 2022b), ontology-based approaches (Zhao et al., 2023), (Zeng et al., 2022), (Cao et al., 2022), and generative pretrained transformer (GPT)-based models (Das et al., 2023). Specially, ChatGPT-based RE could be interesting and worth exploring.

7) *Application scenarios expansion*

Researchers have studied user requirements for various products and services, such as software projects, mobile apps, electronics, automobiles, daily necessities, furniture and household, hotels, restaurants, and travel products. In this paper, we declare that the RE techniques used can be also applied to other scenarios, such as understanding the requirements and issues of people in disasters like COVID-19 and earthquakes and the psychological demands of people under quarantine.

Abbreviations

ABCM	aspect-based classification method
ACRM	associative classification approach for review mining
AHC	agglomerative hierarchical clustering
ALERTme	a little bird told me
AOLDA	adaptively online latent Dirichlet allocation
AR-Miner	app review miner
ARdoc	app reviews development oriented classifier
ARESM	automatic requirements elicitation from social media
ASUM	aspect and sentiment unification model
AUC	area under curve
BC	binary classification
BERT	bidirectional encoder representation from transformers
BiLSTM	bidirectional long short-term memory
BIO	beginning, inside, outside
BJ-LDA	brand joint latent Dirichlet allocation
BOW	bag of words
BR	binary relevance
BTM	biterm topic model
CLAP	crowd listener for release planning
CNN	convolutional neural network
CONCOR	convergence of iterated correlations
CP	cluster precision
CR	common requirements
CRF	conditional random field
CrowdRE-Arg	crowd-based requirements engineering approach by argumentation
CrowdRE-VArg	crowd-based requirements engineering by valuation argumentation
DBNER	a deep neural network for bug specific entity recognition
DBSCAN	density-based spatial clustering of applications with noise
DA	data assessment
DC	data collection
DEMAR	deep multitask learning for requirements discovery and annotation
DL	deep learning
DP	data preprocessing
DT	decision tree
EM	expectation-maximization
EMNB	expectation maximization for naive Bayes
ERNIE	enhanced representation through knowledge integration
ETM	embedded topic models
FR	functional requirement
FR-Refactor	feature request-based refactoring
FSTL	frame semantics and transfer learning
GN	Girvan-Newman
GPT	generative pretrained transformer
GRU	gated recurrent units
GUI	graphical user interface
HDBSCAN	hierarchical density-based spatial clustering of applications with noise
HDP	hierarchical Dirichlet processes
HLDA	hierarchical latent Dirichlet allocation
HoQ	house of quality
HPs	heuristic properties
IDEA	identify emerging app issues
IFQFD	intuitionistic fuzzy quality function deployment
INCOSE	International Council on Systems Engineering
ITS	issue tracking system
JS	Jensen-Shannon
KIS	Kaspersky Internet Security
kNN	k-nearest neighbor
KVoc	keyword vocabulary-based methods
LDA	latent Dirichlet allocation
LR	logistic regression
LRule	language rule-based methods
LSA	latent semantic analysis
LSTM	long short-term memory
MAPP-Reviews	monitoring app reviews
MARA	mobile app review analyzer
MARC	mobile application review classifier
MCC	multi-class classification
ME	maximum entropy
ML	machine learning
MLC	multi-label classification
MLP	multiple layer perceptron
MLR	multivariate logistic regression

(*continued on next page*)

(*continued*)

MNB	multinomial naïve Bayes
MNoR-BERT	multi-label non-functional requirements classification using BERT
MP	model precision
MVLDA	multi-view latent Dirichlet allocation
NA-IPA	network analysis-importance performance analysis
NB	naïve Bayes
NER	named entity recognition
NFR	non-functional requirement
NMF	non-negative matrix factorization
PD	product design
POS	part of speech
PURA	product-and-user oriented approach
QFD	quality function deployment
Q&A	question and answer
RAM	random-access memory
RCBERT	review classification using BERT model
RCC	requirement category classification
RDI	requirement data identification
RE	requirements elicitation
RE-BERT	requirements engineering using BERT
RF	random forest
RR	requirements ranking
RSE	requirement statement extraction
RVM	relevance vector machines
SAO	subject-action-object
SMART	social media analytic framework using text
SNS	social network service
SRPA	syntactic relation-based propagation approach
STM	structural topic models
SToC	semantic topic combination
SUR-Miner	software user review miner
SVM	support vector machine
SVO	subject-verb-object
TF-IDF	term frequency-inverse document frequency
TP	Topic precision
UGC	user-generated content
UR	user requirement
URR	user request referencer
UUX	usability and user experience
VADER	valence aware dictionary for sentiment reasoning
VR	variable requirements
WFPT	word frequency and POS tagging-based methods
XGBoost	eXtreme gradient boosting

CRediT authorship contribution statement

Mengsi Cai: Writing – review & editing, Writing – original draft, Methodology, Investigation, Funding acquisition. **Wenchuan Yang:** Visualization, Investigation. **Yonghao Du:** Writing – review & editing, Methodology. **Yuejin Tan:** Supervision, Conceptualization. **Xin Lu:** Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (72025405, 72301285, 92467302, 72421002, 72474223, 72201272, 72301288, 72025405, 72401289, 72401039); the National Social Science Fund of China (22ZDA102); the Hunan Science and Technology Plan Project (2023JJ40685, 2024JJ6069); and the National Postdoctoral Program for Innovative Talents of China (BX20230475). The authors declare that they have no conflict of interest.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.engappai.2025.111110.

Data availability

No data was used for the research described in the article.

References

Abrahams, A.S., Fan, W., Wang, G.A., Zhang, Z., Jiao, J., 2015. An integrated text analytic framework for product defect discovery. Prod. Oper. Manag. 24 (6), 975–990. https://doi.org/10.1111/poms.12303.

Al Kilani, N., Tailakh, R., Hanani, A., 2019. Automatic classification of apps reviews for requirement engineering: exploring the customers need from healthcare applications. International Conference on Social Networks Analysis, Management and Security 541–548. https://doi.org/10.1109/snams.2019.8931820.

Al-Hawari, A., Najadat, H., Shatnawi, R., 2021. Classification of application reviews into software maintenance tasks using data mining techniques. Softw. Qual. J. 29 (3), 667–703. https://doi.org/10.1007/s11219-020-09529-8.

Ali, N., Hwang, S., Hong, J.E., 2019. Your opinions let us know: mining social network sites to evolve software product lines. KSII Transactions on Internet and Information Systems 13 (8), 4191–4211. https://doi.org/10.3837/tiis.2019.08.021.

Alshangiti, M., Shi, W., Lima, E., Liu, X., Yu, Q., 2022. Hierarchical bayesian multi-kernel learning for integrated classification and summarization of app reviews. Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering 558–569. https://doi.org/10.1145/3540250.3549174.

Alturaief, N., Aljamaan, H., Baslyman, M., 2021. AWARE: aspect-based sentiment analysis dataset of apps reviews for requirements elicitation. IEEE/ACM International Conference on Automated Software Engineering Workshops, pp. 211–218. https://doi.org/10.1109/ASEW52652.2021.00049.

Araujo, A.F., Gôlo, M.P., Marcacini, R.M., 2022. Opinion mining for app reviews: an analysis of textual representation and predictive models. Autom. Softw. Eng. 29, 1–30. https://doi.org/10.1007/s10515-021-00301-1.

Arora, I., Chaudhari, A., Madachane, S., 2023. Analysis of Twitter data for business intelligence. Intelligent Systems and Human Machine Collaboration: Lecture Notes in Electrical Engineering 69–81. https://doi.org/10.1007/978-981-19-8477-8_7.

Bakiu, E., Guzman, E., 2017. Which feature is unusable? Detecting usability and user experience issues from user reviews. IEEE International Requirements Engineering Conference Workshops, pp. 182–187. https://doi.org/10.1109/REW.2017.76.

Bian, Y., Ye, R., Zhang, J., Yan, X., 2022. Customer preference identification from hotel online reviews: a neural network based fine-grained sentiment analysis. Comput. Ind. Eng. 172 (Part A), 108648. https://doi.org/10.1016/j.cie.2022.108648.

Binder, M., Vogt, A., Bajraktari, A., Vogelsang, A., 2023. Automatically classifying Kano model factors in app reviews. International Working Conference on Requirements Engineering: Foundation for Software Quality 245–261. https://doi.org/10.1007/978-3-031-29786-1_17.

Buchan, J., Bano, M., Zowghi, D., Volabouth, P., 2018. Semi-automated extraction of new requirements from online reviews for software product evolution. 2018 25th Australasian Software Engineering Conference, pp. 31–40. https://doi.org/10.1109/ASWEC.2018.00013.

Cai, M., Tan, Y., Ge, B., Dou, Y., Huang, G., Du, Y., 2022. PURA: a product-and-user oriented approach for requirement analysis from online reviews. IEEE Syst. J. 16 (1), 566–577. https://doi.org/10.1109/JSYST.2021.3067334.

Cai, M., Du, Y., Tan, Y., Lu, X., 2023. Aspect-based classification method for review spam detection. Multimed. Tool. Appl. 1–22. https://doi.org/10.1007/s11042-023-16293-x.

Cao, E., Jiang, J., Duan, Y., Peng, H., 2022. A data-driven expectation prediction framework based on social exchange theory. Front. Psychol. 12, 783116. https://doi.org/10.3389/fpsyg.2021.783116.

Carreño, L.V.G., Winbladh, K., 2013. Analysis of user comments: an approach for software requirements evolution. International Conference on Software Engineering 582–591. https://doi.org/10.1109/ICSE.2013.6606604.

Chen, N., Lin, J., Hoi, S.C., Xiao, X., Zhang, B., 2014. AR-miner: mining informative reviews for developers from mobile app marketplace. Proceedings of the 36th International Conference on Software Engineering 767–778. https://doi.org/10.1145/2568225.2568263.

Chen, R., Wang, Q., Xu, W., 2019. Mining user requirements to facilitate mobile app quality upgrades with big data. Electron. Commer. Res. Appl. 38, 100889. https://doi.org/10.1016/j.elerap.2019.100889.

Chen, Z.S., Liu, X.L., Chin, K.S., Pedrycz, W., Tsui, K.L., Skibniewski, M.J., 2021. Online-review analysis based large-scale group decision-making for determining passenger demands and evaluating passenger satisfaction: case study of high-speed rail system in China. Inf. Fusion 69, 22–39. https://doi.org/10.1016/j.inffus.2020.11.010.

Ciurumelea, A., Schaufelbühl, A., Panichella, S., Gall, H.C., 2017. Analyzing reviews and code of mobile apps for better release planning. IEEE International Conference on Software Analysis, Evolution and Reengineering 91–102. https://doi.org/10.1109/SANER.2017.7884612.

Cong, Y., Yu, S., Chu, J., Su, Z., Huang, Y., Li, F., 2023. A small sample data-driven method: user needs elicitation from online reviews in new product iteration. Adv. Eng. Inform. 56, 101953. https://doi.org/10.1016/j.aei.2023.101953.

Dalpiaz, F., Parente, M., 2019. RE-SWOT: from user feedback to requirements via competitor analysis. International Working Conference on Requirements Engineering. Foundation for Software Quality, pp. 55–70. https://doi.org/10.1007/978-3-030-15538-4_4.

Das, S., Deb, N., Chaki, N., Cortesi, A., 2023. Driving the technology value stream by analyzing app reviews. IEEE Trans. Software Eng. 49 (7), 3753–3770. https://doi.org/10.1109/TSE.2023.3270708.

de Araújo, A.F., Marcacini, R.M., 2021. RE-BERT: automatic extraction of software requirements from app reviews using BERT language model. Proceedings of the 36th Annual ACM Symposium on Applied Computing, pp. 1321–1327. https://doi.org/10.1145/3412841.3442006.

de Lima, V.M.A., de Araújo, A.F., Marcacini, R.M., 2022. Temporal dynamics of requirements engineering from mobile app reviews. PeerJ Comput. Sci. 8, e874. https://doi.org/10.7717/peerj-cs.874.

Deocadez, R., Harrison, R., Rodriguez, D., 2017. Automatically classifying requirements from app stores: a preliminary study. *IEEE International Requirements Engineering Conference Workshop*s, pp. 367–371. https://doi.org/10.1109/REW.2017.58.

Devine, P., Koh, Y.S., Blincoe, K., 2023. Evaluating software user feedback classifier performance on unseen apps, datasets, and metadata. Empir. Softw. Eng. 28 (2), 26. https://doi.org/10.1007/s10664-022-10254-y.

Dewi, L., Mulyani, Y.P., 2022. Analysis of hotel attributes and service opportunities in Indonesia on COVID-19 pandemic era through online reviews. IEEE International Conference on Industrial Engineering and Engineering Management 645–649. https://doi.org/10.1109/IEEM55944.2022.9989748.

Dieste, O., Juristo, N., 2010. Systematic review and aggregation of empirical studies on elicitation techniques. IEEE Trans. Software Eng. 37 (2), 283–304. https://doi.org/10.1109/TSE.2010.33.

Dieter, G.E., Schmidt, L.C., 2009. Engineering Design. McGraw-Hill, New York, USA.

Diev, S., 2007. Requirements development as a modeling activity. Software Eng. Notes 32 (2), 1–3. https://doi.org/10.1145/1234741.1234756.

Dollmann, M., Geierhos, M., 2016. On-and off-topic classification and semantic annotation of user-generated software requirements. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1807–1816. https://doi.org/10.18653/v1/D16-1186.

dos Santos, R., Villela, K.B., Avila, D.T., Thom, L.H., 2021. A practical user feedback classifier for software quality characteristics. Proceedings of International Conference on Software Engineering & Knowledge Engineering 1–6. https://doi.org/10.18293/SEKE2021-055.

Du, Y., Liu, D., Duan, H., 2022. A textual data-driven method to identify and prioritise user preferences based on regret/rejoicing perception for smart and connected products. Int. J. Prod. Res. 60 (13), 4176–4196. https://doi.org/10.1080/00207543.2021.2023776.

Duan, S., Liu, J., Peng, Z., 2021. RCBERT an approach with transfer learning for app reviews Classification. CCF Conference on Computer Supported Cooperative Work and Social Computing, pp. 444–457. https://doi.org/10.1007/978-981-19-4549-6_34.

Eldin, S.S., Mohammed, A., Hefny, H., Ahmed, A.S.E., 2021. An enhanced opinion retrieval approach on Arabic text for customer requirements expansion. Journal of King Saud University-Computer and Information Sciences 33 (3), 351–363. https://doi.org/10.1016/j.jksuci.2019.01.010.

Fu, B., Lin, J., Li, L., Faloutsos, C., Hong, J., Sadeh, N., 2013. Why people hate your app: making sense of user feedback in a mobile app store. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1276–1284. https://doi.org/10.1145/2487575.2488202.

Gao, C., Zeng, J., Lyu, M.R., King, I., 2018. Online app review analysis for identifying emerging issues. Proceedings of the 40th International Conference on Software Engineering 48–58. https://doi.org/10.1145/3180155.3180218.

Gao, S., Liu, L., Liu, Y., Liu, H., Wang, Y., 2020. Updating the goal model with user reviews for the evolution of an app. Journal of Software: Evolution and Process 32 (8), e2257. https://doi.org/10.1002/smr.2257.

Gholizadeh, M., Akhlaghpour, S., Isaias, P., Namvar, M., 2022. Emergent affordances and potential challenges of mobile learning apps: insights from online reviews. Inf. Technol. People 35 (7), 2500–2517. https://doi.org/10.1108/ITP-05-2021-0412.

Gôlo, M.P., Araújo, A.F., Rossi, R.G., Marcacini, R.M., 2022. Detecting relevant app reviews for software evolution and maintenance through multimodal one-class learning. Inf. Software Technol. 151, 106998. https://doi.org/10.1016/j.infsof.2022.106998.

Gonzalez, R.A., Rodriguez-Aguilar, R., Marmolejo-Saucedo, J.A., 2020. Text mining and statistical learning for the analysis of the voice of the customer. Artificial Intelligence and Applied Mathematics in Engineering Problems: Proceedings of the International Conference on Artificial Intelligence and Applied Mathematics in Engineering 43, 191–199. https://doi.org/10.1007/978-3-030-36178-5_16.

Groen, E.C., Kopczyńska, S., Hauer, M., Krafft, T.D., Doerr, J., 2017. Users - the hidden software product quality experts?: a study on how app users report quality aspects in online reviews. Proceedings of 2017 IEEE International Requirements Engineering Conference, pp. 80–89. https://doi.org/10.1109/RE.2017.73.

Gu, X., Kim, S., 2015. What parts of your apps are loved by users? (t). IEEE/ACM International Conference on Automated Software Engineering 760–770. https://doi.org/10.1109/ASE.2015.57.

Guo, Y., Wang, F., Xing, C., Lu, X., 2022. Mining multi-brand characteristics from online reviews for competitive analysis: a brand joint model using latent Dirichlet allocation. Electron. Commer. Res. Appl. 53, 101141. https://doi.org/10.1016/j.elerap.2022.101141.

Guzman, E., Maalej, W., 2014. How do users like this feature? A fine grained sentiment analysis of app reviews. IEEE International Requirements Engineering Conference, pp. 153–162. https://doi.org/10.1109/RE.2014.6912257.

Guzman, E., El-Haliby, M., Bruegge, B., 2015. Ensemble methods for app review classification: an approach for software evolution (n). IEEE/ACM International Conference on Automated Software Engineering 771–776. https://doi.org/10.1109/ASE.2015.88.

Guzman, E., Ibrahim, M., Glinz, M., 2017a. A little bird told me: mining tweets for requirements and software evolution. IEEE International Requirements Engineering Conference, pp. 11–20. https://doi.org/10.1109/RE.2017.88.

Guzman, E., Alkadhi, R., Seyff, N., 2017b. An exploratory study of Twitter messages about software applications. Requir. Eng. 22 (5), 387–412. https://doi.org/10.1007/s00766-017-0274-x.

Han, X., Li, R., Li, W., Ding, G., Qin, S., 2019. User requirements dynamic elicitation of complex products from social network service. International Conference on Automation and Computing 1–6. https://doi.org/10.23919/iconac.2019.8895140.

Han, Y., Moghaddam, M., Suthar, M.T., Nanda, G., 2022. Aspect-sentiment-guided opinion summarization for user need elicitation from online reviews. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, V002T02A007. https://doi.org/10.1115/DETC2022-90108.

Harth, P., Jähde, O., Schneider, S., Horn, N., Buchkremer, R., 2023. From data to human-readable requirements: advancing requirements elicitation through language-transformer-enhanced opportunity mining. Algorithms 16 (9), 403. https://doi.org/10.3390/a16090403.

Hazelrigg, G.A., 1996. Systems Engineering: an Approach to Information-Based Design. Prentice-Hall, Upper Saddle River, N.J.

Hedegaard, S., Simonsen, J.G., 2013. Extracting usability and user experience information from online user reviews. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems 2089–2098. https://doi.org/10.1145/2470654.2481286.

Henao, P.R., Fischbach, J., Spies, D., Frattini, J., Vogelsang, A., 2021. Transfer learning for mining feature requests and bug reports from Tweets and app store reviews. IEEE International Requirements Engineering Conference Workshops, pp. 80–86. https://doi.org/10.1109/REW53955.2021.00019.

Heydari, A., Tavakoli, M., Salim, N., Heydari, Z., 2015. Detection of review spam: a survey. Expert Syst. Appl. 42 (7). https://doi.org/10.1016/j.eswa.2014.12.029, 3634-364.

Hsiao, Y.H., Hsiao, Y.T., 2020. Online review analytics for hotel quality at macro and micro levels. Ind. Manag. Data Syst. 121 (2), 268–289. https://doi.org/10.1108/IMDS-04-2020-0214.

Iacob, C., Harrison, R., 2013. Retrieving and analyzing mobile apps feature requests from online reviews. Working Conference on Mining Software Repositories 41–44. https://doi.org/10.1109/MSR.2013.6624001.

Iacob, C., Harrison, R., Faily, S., 2014. Online reviews as first class artifacts in mobile app development. International Conference on Mobile Computing, Applications, and Services, pp. 47–53. https://doi.org/10.1007/978-3-319-05452-0_4.

International Council on Syetems Engineering (INCOSE), 2015. Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, fourth ed. John Wiley and Sons, Inc, San Diego, CA, USA.

Ireland, R., Liu, A., 2018. Application of data analytics for product design: sentiment analysis of online product reviews. CIRP J. Manuf. Sci. Technol. 23, 128–144. https://doi.org/10.1016/j.cirpj.2018.06.003.

Jeong, B., Yoon, J., Lee, J.M., 2019. Social media mining for product planning: a product opportunity mining approach based on topic modeling and sentiment analysis. Int. J. Inf. Manag. 48, 280–290. https://doi.org/10.1016/j.ijinfomgt.2017.09.009.

Jha, N., Mahmoud, A., 2017. MARC: a mobile application review classifier. International Conference on Requirements Engineering. Foundation for Software Quality, Workshops, pp. 1–15.

Jha, N., Mahmoud, A., 2018. Using frame semantics for classifying and summarizing application store reviews. Empir. Softw. Eng. 23 (1), 3734–3767. https://doi.org/10.1007/s10664-018-9605-x.

Jha, N., Mahmoud, A., 2019. Mining non-functional requirements from app store reviews. Empir. Softw. Eng. 24 (6), 3659–3695. https://doi.org/10.1007/s10664-019-09716-7.

Jhamtani, H., Chhaya, N., Karwa, S., Varshne, D., Kedia, D., Gupta, V., 2015. Identifying suggestions for improvement of product features from online product reviews. International Conference on Social Informatics, pp. 112–119. https://doi.org/10.1007/978-3-319-27433-1_8.

Ji, F., Cao, Q., Li, H., Fujita, H., Liang, C., Wu, J., 2023. An online reviews-driven large-scale group decision making approach for evaluating user satisfaction of sharing accommodation. Expert Syst. Appl. 213, 118875. https://doi.org/10.1016/j.eswa.2022.118875.

Jian, J., Ping, J., Rui, G., 2016. Identifying comparative customer requirements from product online reviews for competitor analysis. Eng. Appl. Artif. Intell. 49, 61–73. https://doi.org/10.1016/j.engappai.2015.12.005.

Jiang, W., Ruan, H., Zhang, L., 2014a. Analysis of economic impact of online reviews: an approach for market-driven requirements evolution. Requir. Eng.: Communications in Computer and Information Science 432, 45–59. https://doi.org/10.1007/978-3-662-43610-3_4.

Jiang, W., Ruan, H., Zhang, L., Lew, P., Jiang, J., 2014b. For user-driven software evolution: requirements elicitation derived from mining online reviews. Pacific-asia Conference on Knowledge Discovery and Data Mining, pp. 584–595. https://doi.org/10.1007/978-3-319-06605-9_48.

Jin, J., Ji, P., Kwong, C.K., 2016. What makes consumers unsatisfied with your products: review analysis at a fine-grained level. Eng. Appl. Artif. Intell. 47, 38–48. https://doi.org/10.1016/j.engappai.2015.05.006.

Jin, J., Jia, D., Chen, K., 2022a. Mining online reviews with a Kansei-integrated Kano model for innovative product design. Int. J. Prod. Res. 60 (22), 6708–6727. https://doi.org/10.1080/00207543.2021.1949641.

Jin, H., Wan, H., Li, Z., Wang, W., 2022b. An empirical study on software requirements classification method based on mobile app user comments. IEEE International Conference on Software Quality, Reliability, and Security Companion, pp. 533–541. https://doi.org/10.1109/QRS-C57518.2022.00085.

Kamaruddin, N., Wahab, A., Bakri, M., Hamiz, M., 2019. Science lab repository requirements elicitation based on text analytics. International Conference on Soft Computing in Data Science 351–360. https://doi.org/10.1007/978-981-15-0399-3_28.

Kaur, K., Kaur, P., 2023. MNoR-BERT: multi-label classification of non-functional requirements using BERT. Neural Comput. Appl. 35 (30), 22487–22509. https://doi.org/10.1007/s00521-023-08833-1.

Kauschinger, M., Vieth, N., Schreieck, M., Krcmar, H., 2023. Detecting feature requests of third-party developers through machine learning: a case study of the SAP community. Proceedings of the 56th Hawaii International Conference on System Sciences, pp. 950–959. https://doi.org/10.24251/HICSS.2023.118.

Kengphanphanit, N., Muenchaisri, P., 2020. Automatic requirements elicitation from social media (ARESM). Proceedings of the 2020 International Conference on Computer Communication and Information Systems, pp. 57–62. https://doi.org/10.1145/3418994.3419004.

Khalid, H., Shihab, E., Nagappan, M., Hassan, A.E., 2014. What do mobile app users complain about? IEEE Software 32 (3), 70–77. https://doi.org/10.1109/MS.2014.50.

Khan, J.A., Xie, Y., Liu, L., Wen, L., 2019. Analysis of requirements-related arguments in user forums. IEEE International Requirements Engineering Conference 63–74. https://doi.org/10.1109/RE.2019.00018.

Khan, J.A., Liu, L., Wen, L., 2020. Requirements knowledge acquisition from online user forums. IET Softw. 14 (3), 242–253. https://doi.org/10.1049/iet-sen.2019.0262.

Khan, J.A., Yasin, A., Fatima, R., Vasan, D., Khan, A.A., Khan, A.W., 2022. Valuating requirements arguments in the online user's forum for requirements decision-making: the CrowdRE-VArg framework. Software Pract. Ex. 52 (12), 2537–2573. https://doi.org/10.1002/spe.3137.

Kilroy, D., Healy, G., Caton, S., 2022. Using machine learning to improve lead times in the identification of emerging customer needs. IEEE Access 10, 37774–37795. https://doi.org/10.1109/ACCESS.2022.3165043.

Kim, H.S., Noh, Y., 2019. Elicitation of design factors through big data analysis of online customer reviews for washing machines. J. Mech. Sci. Technol. 33 (6), 2785–2795. https://doi.org/10.1007/s12206-019-0525-5.

Korfiatis, N., Stamolampros, P., Kourouthanassis, P., Sagiadinos, V., 2019. Measuring service quality from unstructured data: a topic modeling application on airline passengers' online reviews. Expert Syst. Appl. 116, 472–486. https://doi.org/10.1016/j.eswa.2018.09.037.

Kovacs, M., Buryakov, D., Kryssanov, V., 2021. An unsupervised approach for customer need assessment in e-commerce: a case study of Japanese customer reviews. Proceedings of 2021 6th International Conference on Cloud Computing and Internet of Things 41–48. https://doi.org/10.1145/3493287.3493294.

Kühl, N., 2016. Needmining: towards analytical support for service design. International Conference on Exploring Services Science 187–200. https://doi.org/10.1007/978-3-319-32689-4_14.

Kühl, N., Mühlthaler, M., Goutier, M., 2020. Supporting customer-oriented marketing with artificial intelligence: automatically quantifying customer needs from social media. Electron. Mark. 30 (2), 351–367. https://doi.org/10.1007/s12525-019-00351-0.

Kumari, S., Memon, Z.A., 2022. Extracting feature requests from online reviews of travel industry. Acta Sci. Technol. 44 (1), e58658. https://doi.org/10.4025/actascitechnol.v44i1.58658.

Kunaefi, A., Aritsugi, M., 2021. Extracting arguments based on user decisions in app reviews. IEEE Access 9, 45078–45094. https://doi.org/10.1109/ACCESS.2021.3067000.

Kurtanović, Z., Maalej, W., 2017. Mining user rationale from software reviews. IEEE International Requirements Engineering Conference, pp. 61–70. https://doi.org/10.1109/RE.2017.86.

Lai, X., Huang, G., Zhao, Z., Lin, S., Zhang, S., Zhang, H., Chen, Q., Mao, N., 2023. Social listening for product design requirement analysis and segmentation: a graph analysis approach with user comments mining. Big Data 3, 1–22. https://doi.org/10.1089/big.2022.0021.

Law, D., Gruss, R., Abrahams, A.S., 2017. Automated defect discovery for dishwasher appliances from online consumer reviews. Expert Syst. Appl. 67, 84–94. https://doi.org/10.1016/j.eswa.2016.08.069.

Li, C., Huang, L., Ge, J., Luo, B., Ng, V., 2018. Automatically classifying user requests in crowdsourcing requirements engineering. J. Syst. Software 138, 108–123. https://doi.org/10.1016/j.jss.2017.12.028.

Li, M., Shi, L., Yang, Y., Wang, Q., 2020. A deep multitask learning approach for requirements discovery and annotation from open forum. Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering, pp. 336–348. https://doi.org/10.1145/3324884.3416627.

Li, S., Zhang, Y., Li, Y., Yu, Z., 2021. The user preference identification for product improvement based on online comment patch. Electron. Commer. Res. 21 (2), 423–444. https://doi.org/10.1007/s10660-019-09372-5.

Li, Q., Yang, Y., Li, C., Zhao, G., 2023. Energy vehicle user demand mining method based on fusion of online reviews and complaint information. Energy Rep. 9, 3120–3130. https://doi.org/10.1016/j.egyr.2023.02.004.

Liang, R., Guo, W., Yang, D., 2017. Mining product problems from online feedback of Chinese users. Kybernetes 46 (3), 572–586. https://doi.org/10.1108/K-03-2016-0048.

Liu, H., Cui, T., He, M., 2021a. Product optimization design based on online review and orthogonal experiment under the background of big data. Proc. Inst. Mech. Eng. Part E J. Process Mech. Eng. 235 (1), 52–65. https://doi.org/10.1177/0954408920943690.

Liu, H., Wang, Y., Liu, Y., Gao, S., 2021b. Supporting features updating of apps by analyzing similar products in app stores. Inf. Sci. 580, 129–151. https://doi.org/10.1016/j.ins.2021.08.050.

Liu, P., Zhang, K., Dong, X., Wang, P., 2022a. A big data-Kano and SNA-CRP based QFD model: application to product design under Chinese new e-commerce model. IEEE Trans. Eng. Manag. 1–15. https://doi.org/10.1109/TEM.2022.3227094.

Liu, J., Hu, X., Zhong, Q., 2022b. Exploring Airbnb users' concerns with LDA-based topic model and sentiment analysis. International Conference on Data Science and Information Technology, pp. 1–5. https://doi.org/10.1109/DSIT55514.2022.9943877.

Lobo, E.H., Abdelrazek, M., Frølich, A., Rasmussen, L.J., Livingston, P.M., Islam, S.M.S., Kensing, F., Grundy, J., 2023. Detecting user experience issues from mHealth apps that support stroke caregiver needs: an analysis of user reviews. Front. Public Health 11, 1027667. https://doi.org/10.3389/fpubh.2023.1027667.

Lu, M., Liang, P., 2017. Automatic classification of non-functional requirements from augmented app user reviews. Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, pp. 344–353. https://doi.org/10.1145/3084226.3084241.

Luiz, W., Viegas, F., Alencar, R., Mourão, F., Salles, T., Carvalho, D., Gonçalves, M.A., Rocha, L., 2018. A feature-oriented sentiment rating for mobile app reviews. Proceedings of the 2018 World Wide Web Conference, pp. 1909–1918. https://doi.org/10.1145/3178876.3186168.

Maalej, W., Nabil, H., 2015. Bug report, feature request, or simply praise? On automatically classifying app reviews. IEEE International Requirements Engineering Conference 116–125. https://doi.org/10.1109/RE.2015.7320414.

Maalej, W., Kurtanović, Z., Nabil, H., Stanik, C., 2016. On the automatic classification of app reviews. Requir. Eng. 21 (3), 311–331. https://doi.org/10.1007/s00766-016-0251-9.

Malik, M.S.I., 2020. Predicting users' review helpfulness: the role of significant review and reviewer characteristics. Soft Comput. 24 (18), 13913–13928. https://doi.org/10.1007/s00500-020-04767-1.

Man, Y., Gao, C., Lyu, M.R., Jiang, J., 2016. Experience report: understanding cross-platform app issues from user reviews. International Symposium on Software Reliability Engineering 138–149. https://doi.org/10.1109/ISSRE.2016.27.

Martin, W., Sarro, F., Yue, J., Zhang, Y., Harman, M., 2017. A survey of app store analysis for software engineering. IEEE Trans. Software Eng. 43 (9), 817–847. https://doi.org/10.1109/TSE.2016.2630689.

McIlroy, S., Ali, N., Khalid, H., Hassan, A.E., 2016. Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews. Empir. Softw. Eng. 21 (3), 1067–1106. https://doi.org/10.1007/s10664-015-9375-7.

Mehder, Ş., Aydemir, F.B., 2022. Classification of issue discussions in open source projects using deep language models. IEEE International Requirements Engineering Conference Workshops, pp. 176–182. https://doi.org/10.1109/REW56159.2022.00040.

Mercado, I.T., Nuthan, M., Meneely, A., 2016. The impact of cross-platform development approaches for mobile applications from the user's perspective. Proceedings of the International Workshop on App Market Analytics, pp. 43–49. https://doi.org/10.1145/2993259.2993268.

Merten, T., Falis, M., Hübner, P., Quirchmayr, T., Bürsner, S., Paech, B., 2016. Software feature request detection in issue tracking systems. IEEE International Requirements Engineering Conference, pp. 166–175. https://doi.org/10.1109/RE.2016.8.

Meth, H., Brhel, M., Maedche, A., 2013. The state of the art in automated requirements elicitation. Inf. Software Technol. 55, 1695–1709. https://doi.org/10.1016/j.infsof.2013.03.008.

Morales-Ramirez, I., Kifetew, F.M., Perini, A., 2019. Speech-acts based analysis for requirements discovery from online discussions. Inf. Syst. 86, 94–112. https://doi.org/10.1016/j.is.2018.08.003.

Mu, R., Zheng, Y., Zhang, K., Zhang, Y., 2021. Research on customer satisfaction based on multidimensional analysis. Int. J. Comput. Intell. Syst. 14 (1), 605–616. https://doi.org/10.2991/ijcis.d.210114.001.

Mukherjee, A., Venkataraman, V., Liu, B., Glance, N., 2013. What Yelp Fake Review Filter Might Be Doing? International AAAI Conference on Weblogs and Social Media, pp. 409–418. https://doi.org/10.1609/icwsm.v7i1.14389.

Nadeem, M., Shahzad, K., Majeed, N., 2021. Extracting software change requests from mobile app reviews. IEEE/ACM International Conference on Automated Software Engineering Workshops, pp. 198–203. https://doi.org/10.1109/ASEW52652.2021.00047.

Necmiye, G.N., Alain, A., 2017. A systematic literature review: opinion mining studies from mobile app store user reviews. J. Syst. Software 125, 207–219. https://doi.org/10.1016/j.jss.2016.11.027.

Nikumanesh, E., Fathi, M., 2017. An indicator for measuring sentiment and polarity: applied knowledge discovery using online customer reviews. IEEE International Conference on Electro Information Technology 472–475. https://doi.org/10.1109/EIT.2017.8053408.

Niu, F., Li, C., Luo, B., 2021. A deep classifier for crowdsourcing user requests. Modern Industrial IoT, Big Data and Supply Chain: Proceedings of the IIoTBDSC 2020 218, 11–22. https://doi.org/10.1007/978-981-33-6141-6_2.

Nyamawe, A.S., Liu, H., Niu, N., Umer, Q., Niu, Z., 2019. Automated recommendation of software refactorings based on feature requests. IEEE International Requirements Engineering Conference, pp. 187–198. https://doi.org/10.1109/RE.2019.00029.

Ott, M., Cardie, C., Hancock, J., 2013. Negative deceptive opinion spam. Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 497–501.

Palomba, F., Salza, P., Ciurumelea, A., Panichella, S., Gall, H., Ferrucci, F., De Lucia, A., 2017. Recommending and localizing change requests for mobile apps based on user reviews. Proceedings of 2017 IEEE/ACM International Conference on Software Engineering, pp. 106–117. https://doi.org/10.1109/ICSE.2017.18.

Panichella, S., Sorbo, A.D., Guzman, E., Visaggio, C.A., Canfora, G., Gall, H.C., 2015. How can I improve my app? Classifying user reviews for software maintenance and evolution. IEEE International Conference on Software Maintenance and Evolution 281–290. https://doi.org/10.1109/ICSM.2015.7332474.

Panichella, S., Sorbo, A.D., Guzman, E., Visaggio, C.A., Canfora, G., Gall, H.C., 2016. ARdoc: app reviews development oriented classifier. Proceedings of the 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, pp. 1023–1027. https://doi.org/10.1145/2950290.2983938.

Park, S., Kim, H.M., 2022. Phrase embedding and clustering for sub-feature extraction from online data. J. Mech. Des. 144 (5), 054501. https://doi.org/10.1115/1.4052904.

Park, Y., Lee, S., 2011. How to design and utilize online customer center to support new product concept generation. Expert Syst. Appl. 38 (8), 10638–10647. https://doi.org/10.1016/j.eswa.2011.02.125.

Pohl, K., 2010. Requirements Engineering: Fundamentals, Principles, and Techniques. Springer, Heidelberg. https://doi.org/10.1007/978-3-642-12578-2.

Qi, J., Zhang, Z., Jeon, S., Zhou, Y., 2016. Mining customer requirements from online reviews: a product improvement perspective. Inf. Manag. 53 (8), 951–963. https://doi.org/10.1016/j.im.2016.06.002.

Qian, Y., Gui, W., 2021. Identifying health information needs of senior online communities users: a text mining approach. Aslib J. Inf. Manag. 73 (1), 5–24. https://doi.org/10.1108/AJIM-02-2020-0057.

Qiu, G., Liu, B., Bu, J., Chen, C., 2011. Opinion word expansion and target extraction through double propagation. Comput. Linguist. 37 (1), 9–27. https://doi.org/10.1162/coli_a_00034.

Rahman, S., Ahmed, F., Nayebi, M., 2023. Mining Reddit data to elicit students' requirements during COVID-19 pandemic. IEEE International Requirements Engineering Conference Workshops, pp. 76–84. https://doi.org/10.1109/REW57809.2023.00021.

Ren, G., Hong, T., 2018. Examining the relationship between specific negative emotions and the perceived helpfulness of online reviews. Inf. Process. Manag. 56 (4), 1425–1438. https://doi.org/10.1016/j.ipm.2018.04.003.

Ren, S., Nakagawa, H., Tsuchiya, T., 2022. Hierarchical user review clustering based on multiple sub-goal generation. Joint Conference on Knowledge-Based Software Engineering, pp. 207–219. https://doi.org/10.1007/978-3-031-17583-1_16.

Sangaroonsilp, P., Choetkiertikul, M., Dam, H.K., Ghose, A., 2023. An empirical study of automated privacy requirements classification in issue reports. Autom. Softw. Eng. 30 (2), 20. https://doi.org/10.1007/s10515-023-00387-9.

Scalabrino, S., Bavota, G., Russo, B., Di, Penta M., Oliveto, R., 2017. Listening to the crowd for the release planning of mobile apps. IEEE Trans. Software Eng. 45 (1), 68–86. https://doi.org/10.1109/TSE.2017.2759112.

Shi, Y., Peng, Q., 2021. Definition of customer requirements in big data using word vectors and affinity propagation clustering. Proc. Inst. Mech. Eng. Part E J. Process Mech. Eng. 235 (5), 1279–1291. https://doi.org/10.1177/09544089211001776.

Shi, P.Y., Yu, J.H., 2022. Research on the identification of user demands and data mining based on online reviews. International Conference on Big Data, Information and Computer Network, pp. 43–47. https://doi.org/10.1109/BDICN55575.2022.00016.

Sonbol, R., Rebdawi, G., Ghneim, N., 2022. The use of NLP-based text representation techniques to support requirement engineering tasks: a systematic mapping review. IEEE Access 10, 62811–62830. https://doi.org/10.1109/ACCESS.2022.3182372.

Song, H., Chen, C., Yu, Q., 2018. Research on Kano model based on online comment data mining. IEEE 3rd International Conference on Big Data Analysis, pp. 76–82. https://doi.org/10.1109/ICBDA.2018.8367654.

Song, R., Li, T., Ding, Z., 2020. Automatically identifying requirements-oriented reviews using a top-down feature extraction approach. Asia-Pacific Software Engineering Conference, pp. 450–454. https://doi.org/10.1109/APSEC51365.2020.00054.

Song, Y., Wang, R., Fernandez, J., Li, D., 2021. Investigating sense of place of the Las Vegas Strip using online reviews and machine learning approaches. Landsc. Urban Plann. 205, 103956. https://doi.org/10.1016/j.landurbplan.2020.103956.

Sorbo, A.D., Panichella, S., Alexandru, C.V., Shimagaki, J., Visaggio, C.A., Canfora, G., Gall, H.C., 2016. What would users change in my app? Summarizing app reviews for recommending software changes. Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering 499–510. https://doi.org/10.1145/2950290.2950299.

Spada, I., Barandoni, S., Giordano, V., Chiarello, F., Fantoni, G., Martini, A., 2023. What users want: a natural language processing approach to discover users' needs from online reviews. Proceedings of the Design Society 3, 3879–3888. https://doi.org/10.1017/pds.2023.389.

Stahlmann, S., Ettrich, O., Kurka, M., Schoder, D., 2023. What do customers say about my products? Benchmarking machine learning models for need identification. Proceedings of the 56th Hawaii International Conference on System Sciences, pp. 2120–2129. https://doi.org/10.24251/HICSS.2023.264.

Stanik, C., Haering, M., Maalej, W., 2019. Classifying multilingual user feedback using traditional machine learning and deep learning. IEEE International Requirements Engineering Conference Workshops 220–226. https://doi.org/10.1109/REW.2019.00046.

Sun, H., Ong, S.K., Nee, A.Y.C., Guo, W., 2023. A customer requirements analysis method of considering product scenarios for improving product design. J. Eng. Des. 34 (5–6), 339–362. https://doi.org/10.1080/09544828.2023.2225843.

Suryadi, D., Kim, H., 2019. Automatic identification of product usage contexts from online customer reviews. Proceedings of the Design Society: International Conference on Engineering Design 1 (1), 2507–2516. https://doi.org/10.1017/dsi.2019.257.

Svee, E.O., Zdravkovic, J., 2016. A model-based approach for capturing consumer preferences from crowdsources: the case of Twitter. IEEE Tenth International Conference on Research Challenges in Information Science 1–12. https://doi.org/10.1109/RCIS.2016.7549323.

Takahashi, H., Nakagawa, H., Tsuchiya, T., 2015. Towards automatic requirements elicitation from Feedback comments: extracting requirements topics using LDA. Proceedings of the 27th International Conference on Software Engineering and Knowledge Engineering, pp. 489–494. https://doi.org/10.18293/SEKE2015-103.

Tang, M., Liu, Y., Li, Z., Liu, Y., 2018. Identifying service gaps from public patient opinions through text mining. Intelligent Computing and Internet of Things: International Conference on Intelligent Computing for Sustainable Energy and Environment and International Conference on Intelligent Manufacturing and Internet of Things, pp. 99–108. https://doi.org/10.1007/978-981-13-2384-3_10.

Tavakoli, M., Zhao, L., Heydari, A., Nenadic, G., 2018. Extracting useful software development information from mobile application reviews: a survey of intelligent mining techniques and tools. Expert Syst. Appl. 113, 186–199. https://doi.org/10.1016/j.eswa.2018.05.037.

Timoshenko, A., Hauser, J.R., 2019. Identifying customer needs from user-generated content. Mark. Sci. 38 (1), 1–20. https://doi.org/10.1287/mksc.2018.1123.

Tizard, J., Wang, H., Yohannes, L., Blincoe, K., 2019. Can a conversation paint a picture? Mining requirements in software forums. IEEE International Requirements Engineering Conference 17–27. https://doi.org/10.1109/RE.2019.00014.

Tong, X., 2021. Positioning game review as a crucial element of game user feedback in the ongoing development of independent video games. Computers in Human Behavior Reports 3, 100077. https://doi.org/10.1016/j.chbr.2021.100077.

Villarroel, L., Bavota, G., Russo, B., Oliveto, R., Penta, M.D., 2016. Release planning of mobile apps based on user reviews. Proceedings of 38th International Conference on Software Engineering 14–24. https://doi.org/10.1145/2884781.2884818.

Vlas, R.E., Robinson, W.N., 2012. Two rule-based natural language strategies for requirements discovery and classification in open source software development projects. J. Manag. Inf. Syst. 28 (4), 11–38. https://doi.org/10.2753/MIS0742-1222280402.

Wang, T., 2022. A novel approach of integrating natural language processing techniques with fuzzy TOPSIS for product evaluation. Symmetry 14 (1), 120. https://doi.org/10.3390/sym14010120.

Wang, Y., Li, X., 2020. Mining product reviews for needs-based product configurator design: a transfer learning-based approach. IEEE Trans. Ind. Inf. 17 (9), 6192–6199. https://doi.org/10.1109/TII.2020.3043315.

Wang, H., Wang, W., 2014. Product weakness finder: an opinion-aware system through sentiment analysis. Ind. Manag. Data Syst. 114 (8), 1301–1320. https://doi.org/10.1108/IMDS-05-2014-0159.

Wang, L., Youn, B.D., Azarm, S., Kannan, P.K., 2011. Customer-driven product design selection using web based user-generated content. ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference 54822, 405–419. https://doi.org/10.1115/DETC2011-48338.

Wang, Y., Li, X., Zhang, L.L., Mo, D., 2022a. Configuring products with natural language: a simple yet effective approach based on text embeddings and multilayer perceptron. Int. J. Prod. Res. 60 (17), 5394–5406. https://doi.org/10.1080/00207543.2021.1957508.

Wang, Z., Tan, L., Yu, X., 2022b. A method for obtaining user requirements based on NLTK and knowledge graph. World Conference on Mechanical Engineering and Intelligent Manufacturing, pp. 276–280. https://doi.org/10.1109/WCMEIM56910.2022.10021488.

Wang, Q., Wang, S., Fu, S., 2023. A sustainable iterative product design method based on considering user needs from online reviews. Sustainability 15 (7), 5950. https://doi.org/10.3390/su15075950.

Wen, P., Chen, M., 2020. A new analysis method for user reviews of mobile fitness apps. International Conference on Human-Computer Interaction 188–199. https://doi.org/10.1007/978-3-030-49065-2_14.

Williams, G., Mahmoud, A., 2017. Mining Twitter feeds for software user requirements. IEEE 25th International Requirements Engineering Conference 1–10. https://doi.org/10.1109/RE.2017.14.

Xiao, S., Wei, C., Dong, M., 2016. Crowd intelligence: analyzing online product reviews for preference measurement. Inf. Manag. 53 (2), 169–182. https://doi.org/10.1016/j.im.2015.09.010.

Xiao, Y., Li, C., Thürer, M., Liu, Y., Qu, T., 2022. User preference mining based on fine-grained sentiment analysis. J. Retailing Consum. Serv. 68 (9), 103013. https://doi.org/10.1016/j.jretconser.2022.103013.

Xu, X., 2021. What are customers commenting on, and how is their satisfaction affected? Examining online reviews in the on-demand food service context. Decis. Support Syst. 142 (3), 113467. https://doi.org/10.1016/j.dss.2020.113467.

Yan, C., Liu, L., Liu, W., Qi, M., 2022. A novel data analytic model for mining user insurance demands from microblogs. Comput. Inf. 41 (3), 689–713. https://doi.org/10.31577/cai_2022_3_689.

Yang, H., Liang, P., 2015. Identification and classification of requirements from app user reviews. Proceedings of 27th International Conference on Software Engineering and Knowledge Engineering, pp. 7–12. https://doi.org/10.18293/SEKE2015-063.

Yang, S., Zhou, C., Chen, Y., 2021. Do topic consistency and linguistic style similarity affect online review helpfulness? An elaboration likelihood model perspective. Inf. Process. Manag. 58 (3), 102521. https://doi.org/10.1016/j.ipm.2021.102521.

Yang, B., Liu, Y., Chen, W., 2023. A twin data-driven approach for user-experience based design innovation. Int. J. Inf. Manag. 68, 102595. https://doi.org/10.1016/j.ijinfomgt.2022.102595.

Yilin, Z., Fayoumi, A., Shahgholian, A., 2023. Understanding online customer touchpoints: a deep learning approach to enhancing customer experience in digital retail. International Conference on Information Technology Trends, 193279. https://doi.org/10.1109/ITT59889.2023.10184269.

Yoon, Y.S., Oh, H.W., Park, K., 2018. Mining Twitter to identify customers' requirements and shoe market segmentation. International Conference on Information and Communication Technology Convergence, pp. 1194–1199. https://doi.org/10.1109/ICTC.2018.8539363.

Young, R.R., 2001. Effective Requirements Practices. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, United States.

Yu, L., Wang, H., Luo, X., Zhang, T., Liu, K., Chen, J., Zhou, H., Tang, Y., Xiao, X., 2022. Towards automatically localizing function errors in mobile apps with user reviews. IEEE Trans. Software Eng. 49 (4), 1464–1486. https://doi.org/10.1109/TSE.2022.3178096.

Zeng, D., Zhao, J., Zhang, W., Zhou, Y., 2022. User-interactive innovation knowledge acquisition model based on social media. Inf. Process. Manag. 59 (3), 102923. https://doi.org/10.1016/j.ipm.2022.102923.

Zhang, X., Huang, Y., 2023. Fresh produce e-commerce user demand mining and analysis based on KANO model. International Symposium on Computer Technology and Information Science 1110–1114. https://doi.org/10.1109/ISCTIS58954.2023.10213192.

Zhang, W., Xu, H., Wan, W., 2012. Weakness finder: find product weakness from Chinese reviews by using aspects based sentiment analysis. Expert Syst. Appl. 39 (11), 10283–10291. https://doi.org/10.1016/j.eswa.2012.02.166.

Zhang, L., Huang, X.Y., Jiang, J., Hu, Y.K., 2017. CSLabel: an approach for labelling mobile app reviews. J. Comput. Sci. Technol. 32 (6), 1076–1089. https://doi.org/10.1007/s11390-017-1784-1.

Zhang, H., Rao, H., Feng, J., 2018. Product innovation based on online review data mining: a case study of Huawei phones. Electron. Commer. Res. 18 (1), 3–22. https://doi.org/10.1007/s10660-017-9279-2.

Zhang, L., Chu, X., Xue, D., 2019a. Identification of the to-be-improved product features based on online reviews for product redesign. Int. J. Prod. Res. 57 (8), 2464–2479. https://doi.org/10.1080/00207543.2018.1521019.

Zhang, T., Chen, J., Zhan, X., Luo, X., Lo, D., Jiang, H., 2019b. Where2Change: change request localization for app reviews. IEEE Trans. Software Eng. 47 (11), 2590–2616. https://doi.org/10.1109/TSE.2019.2956941.

Zhang, M., Fan, B., Zhang, N., Wang, W., Fan, W., 2021. Mining product innovation ideas from online reviews. Inf. Process. Manag. 58 (1), 102389. https://doi.org/10.1016/j.ipm.2020.102389.

Zhang, M., Sun, L., Wang, G.A., Li, Y., He, S., 2022. Using neutral sentiment reviews to improve customer requirement identification and product design strategies. Int. J. Prod. Econ. 254, 108641. https://doi.org/10.1016/j.ijpe.2022.108641.

Zhang, M., Sun, L., Li, Y., Wang, G.A., He, Z., 2023a. Using supplementary reviews to improve customer requirement identification and product design development. Journal of Management Science and Engineering 8, 584–597. https://doi.org/10.1016/j.jmse.2023.03.001.

Zhang, K., Lin, K.Y., Wang, J., Ma, Y., Li, H., Zhang, L., Liu, K., Feng, L., 2023b. UNISON framework for user requirement elicitation and classification of smart product-service system. Adv. Eng. Inform. 57, 101996. https://doi.org/10.1016/j.aei.2023.101996.

Zhang, D., Shen, Z., Li, Y., 2023c. Requirement analysis and service optimization of multiple category fresh products in online retailing using importance-Kano analysis. J. Retailing Consum. Serv. 72, 103253. https://doi.org/10.1016/j.jretconser.2022.103253.

Zhang, J., Hua, J., Niu, N., Chen, S., Savolainen, J., Liu, C., 2023d. Exploring privacy requirements gap between developers and end users. Inf. Software Technol. 154, 107090. https://doi.org/10.1016/j.infsof.2022.107090.

Zhao, L., Zhao, A., 2019. Sentiment analysis based requirement evolution prediction. Future Internet 11 (2), 52. https://doi.org/10.3390/fi11020052.

Zhao, D., Tang, Z., Sun, F., 2023. Research on the weak demand signal identification model of innovative product based on domain ontology construction. Kybernetes. https://doi.org/10.1108/K-05-2023-0850.

Zhou, F., Jiao, R.J., Linsey, J., 2015. Latent customer needs elicitation by use case analogical reasoning from sentiment analysis of online product reviews. J. Mech. Des. 137 (7), 071401. https://doi.org/10.1115/1.4030159.

Zhou, C., Li, B., Sun, X., 2020a. Improving software bug-specific named entity recognition with deep neural network. J. Syst. Software 165 (11), 110572. https://doi.org/10.1016/j.jss.2020.110572.

Zhou, F., Ayoub, J., Xu, Q., Yang, X.J., 2020b. A machine learning approach to customer needs analysis for product ecosystems. J. Mech. Des. 142 (1), 011101. https://doi.org/10.1115/1.4044435.

Zhou, W., Wang, Y., Qu, Y., Li, L., 2022. Automating app review classification based on extended semantic. International Conference on Dependable Systems and Their Applications, pp. 106–115. https://doi.org/10.1109/DSA56465.2022.00022.

Zhou, F., Jiang, Y., Qian, Y., Liu, Y., Chai, Y., 2023. Product consumptions meet reviews: inferring consumer preferences by an explainable machine learning approach. Decis. Support Syst., 114088 https://doi.org/10.1016/j.dss.2023.114088.

Mengsi Cai received the Ph.D. degree in management science and engineering from the National University of Defense Technology, Changsha, Hunan, China, in 2022, where she completed postdoctoral training in 2025. She is currently a Research Associate with the College of Systems Engineering, National University of Defense Technology, China. She was a yearly visiting researcher at University Medical Center Utrecht in the Netherlands. Her research interests include complex networks, data mining, and natural language processing.

Wenchuan Yang obtained the B.E. degree in management science from the Sichuan University, Chengdu, Sichuan, China in 2019. He is currently a Ph.D. student at the College of Systems Engineering, National University of Defense and Technology, Changsha, China. His research interests are recommender systems, complex networks, graph neural networks, and data mining.

Yonghao Du received the B.S. degree from the Southeast University, China, in 2015, and received the M.S. degree and Ph.D. degree from the National University of Defense Technology, Changsha, China, in 2017 and 2021, respectively. He is currently an Associate Professor with the College of Systems Engineering, National University of Defense Technology, China. He was a yearly visiting researcher at Leiden University in the Netherlands. His research interests include intelligent optimization, resource scheduling, and mission planning.

Yuejin Tan is currently a Professor with the National University of Defense Technology of China. He won the first-class post allowance for military excellent professional and technical talents, the National Distinguished Teacher Award. He has led a number of national, provincial or ministry-level research projects, winning two first prizes and 2 s prizes of provincial or ministry-level awards for science and technology progress. He has obtained over 100 papers and 13 research monographs, winning two National Book Awards.

Xin Lu received the Ph.D. degree in medical science from the Department of Public Health Sciences, Karolinska Institutet, Solna, Sweden, in 2013. He is currently a Professor with the National University of Defense Technology, Changsha, China. His research interests include big data analytics, complex networks, human behavior, and emergency management.