Cloud Computing and Data Science
https://ojs.wiserpub.com/index.php/CCDS/

Research Article

# Personalized Product Ranking Based on Linguistic Requirements and Online Reviews

**Jiayu Chen**, **Hai Wang***

School of Computer Science, Nanjing Audit University, Nanjing, 211815, Jiangsu, China
E-mail: wanghai@nau.edu.cn

**Received:** 7 February 2025; **Revised:** 9 May 2025; **Accepted:** 9 May 2025

**Abstract:** With the rapid development of online shopping, an increasing number of consumers purchase products and share reviews on e-commerce platforms. Various methods have been proposed to assist consumers in making purchasing decisions based on online product reviews. However, most existing methods focus on objective product aspects, general consumer preferences, or historical user preferences, neglecting the current preferences of specific users. This paper proposes a personalized product ranking method based on the linguistic requirements of current users and historical online reviews, under the framework of multi-attribute decision-making. This method extracts attributes and their weights from the linguistic requirements, taking the current preferences of specific users into account, and increasing flexibility. Furthermore, it segments online reviews based on attribute transitions to improve the accuracy of product attribute scores derived from these online reviews. A case study and comparative analysis are conducted to verify the effectiveness of this proposed method, demonstrating its ability to incorporate the current preferences of specific users and increase the accuracy of product attribute scoring based on online reviews.

***Keywords*:** personalized recommendation, preference analysis, linguistic requirements, online reviews

## 1. Introduction

With the widespread adoption of the Internet, online shopping has become a dominant trend in modern consumer behavior [1]. Many consumers now prefer e-commerce platforms over physical stores [2]. Several studies indicate that consumers rely heavily on online reviews when making purchasing decisions on e-commerce platforms [3-5]. However, consumers have difficulty choosing a satisfactory product among numerous products and the vast quantity and diversity of online reviews [6]. Consequently, the issue of how to rank products based on their online reviews to best meet consumer demands has become crucial. Typically, Multi-Attribute Decision-Making (MADM) methods can be used to rank alternative products, with two key factors as input data: the current preferences of specific users and product online reviews.

In recent years, numerous studies have proposed relevant methods based on online reviews, user preferences, or a combination. Methods based on online reviews extract product attribute scores using various methods, including rule-based methods [4], dictionary-based methods [7], machine learning methods [8], and deep learning methods [9]. These methods are recalled in Section 2.1. In contrast, the number of methods based solely on user preferences is relatively small. These methods typically extract product attributes and their weights directly from user preferences using methods


DOI: https://doi.org/10.37256/ccds.6220256525

such as direct user determination [10], collaborative filtering [11], Bayesian inference [12], and learning models [13]. Hybrid methods combine both online reviews and user preferences. They extract product attributes and their weights from user preferences while calculating product attribute scores from online reviews. These hybrid methods can be categorized into direct and indirect approaches based on how they extract user preferences. Direct methods extract product attributes and their weights directly from user preference data [14], while indirect methods analyze user preferences from online reviews and then extract relevant product attributes and their weights [1]. These methods are recalled in Section 2.2.

The development of this paper is motivated by some vital limitations of current studies. Dictionary-based methods face difficulties in capturing contextual information in the sentence. Rule-based methods require human expertise for design and maintenance, resulting in low scalability and making them difficult to apply to other scenarios. Learning-based methods need formatted data and substantial computational resources [15], and it is difficult for online reviews to meet this demand. These methods may require extensive data analysis and feature engineering work as well, yet most online review datasets are unannotated. Furthermore, when considering user preferences, most methods focus on general user preferences rather than specific user preferences, or use objective product aspects as the product attributes [16], leading to limited flexibility. Other methods based on specific user preferences, such as collaborative filtering Bayesian inference, rely on historical user preferences instead of current user preferences. These methods necessitate a large amount of similar data and historical user information, complicating implementation and applicability to current scenarios [12]. Therefore, it is crucial to develop a more effective and flexible product ranking system that can accurately rank products aligned with the current preferences of specific users.

To address these limitations above, this study proposes a method based on the linguistic requirements of current users and historical online reviews to derive product attributes, product attribute weights, and product attribute scores. Both linguistic requirements of current users and historical online reviews are utilized to extract product attributes and their weights. The sentiment scores of online reviews are used as product attributes. For attribute identification, the Biterm Topic Model (BTM) method [17], which is particularly suited for processing large amounts of incoherent short text, is applied to extract topics from online reviews. These topics are considered as the attributes of the product. For attribute weight assignment, this study analyzes linguistic requirements, selects the relevant attributes from topics, and determines their relations to establish the range of attribute weights. Finally, for decision-making, product attribute scores are aggregated using the sentiment scores of online reviews based on the support degree for each attribute. ELimination Et Choix Traduisant la REalite (ELECTRE) III [18] is used to rank products based on attribute weights and product attribute scores, providing consumers with a prioritized list to aid in their decision-making process. Generally, the technical novelty of our proposed method lies in the inclusion of individual linguistic requirements and the formal syntax, in which the former enables users to express their preferences flexibly and the latter exploits the preferences efficiently.

The main contributions of this paper are as follows:

(1) A user preference analysis method is proposed based on the linguistic requirements of current users. In order to consider the current preferences of the specific users for purchasing products, rather than relying on the less relevant historical user preferences, the proposed method constructs rules based on deep learning methods to extract product attributes and their weights from linguistic requirements.

(2) A hybrid scoring method for online reviews is developed through deep learning methods and dictionaries. This method effectively captures both word-level and contextual information from online reviews, enhancing the accuracy of the correspondence between the online reviews and attributes. It inherits the advantages of these types of methods while mitigating their respective weaknesses. Characterized by its simplicity and flexibility, this model does not require extensive computational resources or annotated data, making it suitable for a diverse range of application scenarios.

The remaining structure in this paper is organized as follows: Section 2 provides a comprehensive review of the existing research in the fields related to the proposed method. Section 3 describes the detailed steps of the method. Section 4 presents a case study and comparative analysis to evaluate the performance of the method. Section 5 presents a case study and comparative analysis to evaluate the performance of the method under the large-scale data scenarios. Finally, Section 6 concludes the paper by analyzing the research findings, summarizing the key discoveries and contributions, and offering insights for future research and practical applications.

## 2. Related works

The development of e-commerce has led to a substantial rise in online reviews on various platforms. Consumers often rely on these online reviews when making purchasing decisions. To effectively extract useful information from online reviews of a vast quantity and diversity and combine it with user preferences, it is crucial to investigate appropriate methods based on user preferences and online reviews. Consequently, the focal area of these studies is MADM product ranking methods based on user preferences and online reviews. This section provides a thorough overview of related work in two primary areas: product ranking methods based on online reviews, and methods that combine both user preferences and online reviews.

### 2.1 *Product ranking methods based on online reviews*

These methods use online reviews to ascertain product attributes and their weights, as well as to determine attribute scores. Given that the extraction of attributes and their weights is primarily based on user preferences, this section will concentrate on the methods that extract attribute scores from online reviews. These methods are primarily categorized into two groups when determining product attribute scores: sentiment analysis and mathematical operators.

The sentiment analysis methods include deep learning models, machine learning models, and dictionary-based models. Deep learning models for natural language processing, such as the Recursive Neural Tensor Network (RNTN) in [19], Bidirectional Neural Long Short-Term Memory (BN-LSTM) in [9], and Bidirectional Encoder Representations using Transformers (BERT) in [20], are commonly utilized. The sentiment analysis methods do not require manual feature engineering. They can automatically capture complex semantics and contextual relationships, performing well in big data and complex tasks with high accuracy. However, these methods require a large amount of data and computing resources. In addition to deep learning models, machine learning methods like [8, 10, 21, 22] have been applied to sentiment classification tasks, performing well in sentiment classification on some datasets, although a lot of feature engineering and data labeling work is required. Dictionary-based methods remain relevant due to their interpretability and lower computational requirements, but the effectiveness of these methods depends heavily on the dictionary and aggregation methods. These include [4, 7, 19, 20, 22-29], and have been employed extensively in earlier works.

In the mathematical operators-based models, researchers employ various mathematical concepts and methodologies to calculate product attribute scores based on online reviews. For example, [4] uses intuitionistic fuzzy weighted averaging to derive scores, [11] calculates scores using Pearson correlation coefficient, [20] uses $q$-rung orthopair fuzzy mean operator, [7] uses the discrete dynamic intuitionistic fuzzy average weighted aggregation and "vertical projection distance" for aggregation scoring, and [30] applies the interval-valued Pythagorean fuzzy number weighted Heronian mean operator for aggregation scoring. These methods are fast and flexible, but their semantic understanding is limited and cannot capture complex semantics and contextual relationships.

Recently, research on sentiment analysis has made significant progress. Sentiment analysis based on large language models has strong semantic understanding capabilities, strong interpretability, high operational efficiency and flexibility, such as GPT-4 in [31] and XLNet in [32]. Multimodal Sentiment Analysis integrates information from multiple modalities, including text, speech, image, and video, to avoid the limitations of a single modality. These methods reduce errors caused by noise or ambiguity and can adapt to scenarios such as multilingual and real-time interactions, such as UniMSE in [33], G2d in [34], Freeze the backbones in [35], and IMITATE in [36].

In these methods, online reviews are typically analyzed at the word or sentence level. Analyzing the word-level reviews often find it challenging to capture contextual information in sentences. On the other hand, when analyzing sentence-level reviews, a single sentence often describes multiple product attributes, making it difficult to isolate and ascertain the score for each attribute from the sentence.

### 2.2 *Product ranking methods based on user preferences and online reviews*

Preference-based methods are used to extract product attributes and their weights from user preferences, which can be categorized into direct and indirect methods.

Direct methods extract product attributes and their weights straightforwardly from user preference data [14]. The most frequent approaches involve directly deciding attributes and their weights in studies [19, 37, 38]. Other approaches use specific formats of user preference information, such as the Analytic Hierarchy Process (AHP) used in [4, 22, 39], and collaborative filtering in [12]. These methods are intuitive and require less computational resources, and the data can more accurately represent user preferences. However, these methods rely on external labels and are difficult to reflect niche preferences. Some methods based on user historical preferences also encounter the "cold start" problem.

Indirect methods, on the other hand, analyze user preferences derived from online reviews to extract product attributes and their weights. The most common methods are attribute extraction based on word frequency. Studies such as [8, 40] derive attributes and their weights directly from the frequency of words in reviews, while [41, 42] use the Term Frequency-Inverse Document Frequency (TF-IDF) method to obtain high-frequency words and their frequencies from reviews. These methods are simple to implement and computationally efficient, making them suitable for short texts. However, they fail to capture word order and semantic associations and are easily disturbed by high-frequency noise words.

Compared to word frequency-based methods, learning-based approaches can capture more nuanced information. Methods [21, 28, 43-45] use the Latent Dirichlet Allocation (LDA) method to build topic models from reviews. Clustering methods [11] and utilizing attention-based learning methods [9] are applied to derive product attributes. Study [20] uses BERT to recognize and group attributes from online reviews. High Adjective Count (HAC) clustering [29] is used for attribute extraction. Some methods combine the advantages of frequency-based and learning-based approaches, for example, [46] combines TF-IDF and Order Preserving Submatrix (OPSM) biclustering methods to extract features. These methods can automatically learn complex weight relationships. However, they rely on labeled data, have poor model interpretability, and the cost of training is high.

Word distribution-based methods are often used for calculating attribute weights. Study [47] calculates attribute weights based on the entropy of attribute words, while [14, 48] consider word frequency, word distribution, review frequency, and review length to determine attribute weights. Additionally, study [49] combines word frequency, degree centrality and K-hop centrality in social networks to compute attribute weights. These methods can capture semantic associations and are more effective for long texts but they rely on prior assumptions with relatively high computational complexity.

Association rule mining methods are capable of extracting relationships within online reviews. Studies such as [1, 24] identify attribute words by analyzing the connections between opinion words and nouns. Apriori methods, as utilized in [23, 50], are commonly used for this purpose. Co-occurrence association rule mining methods [27] are used to discover attribute-opinion word pairs. These methods can uncover implicit relationships between attributes and are suitable for causal reasoning. However, generating rules requires a large amount of data, which can easily lead to redundant rules, and they struggle to handle high-dimensional sparse data.

Recently, research on direct preference analysis methods based on user preference data has made significant progress in human preference optimization. These methods simplify the process, enhance computational efficiency, exhibit strong stability, and have low data dependence, such as direct preference optimization in [51], and Kahneman-Tversky optimization in [52]. Research of indirect preference analysis methods has also made significant progress in multimodal, these methods can reflect the preferences of users more comprehensively with strong robustness, such as MatGBM in [53], and cross-domain recommendation in [54].

In these methods, preference-based methods for extracting product attributes and their weights from user preferences encompass a range of methods, categorized into direct and indirect approaches. These methods include word frequency-based model, word distribution-based model, learning models and association rule mining methods, each providing unique advantages for capturing user preferences from various data sources.

## 2.3 *Summary*

The existing research on product ranking methods based on user preferences and online reviews has made significant progress, but several challenges and limitations still need to be addressed. Currently, most of the researches concentrates on general consumer preferences. These methods either use objective attributes as product attributes [24] or extract attributes from online reviews [7], which may not capture the current preferences of specific users. Additionally, the existing methods primarily rely on online reviews to calculate product attribute scores, leading to low accuracy.

When extracting product attributes and their weights, the existing methods consider user preferences in various formats. Attributes derived from objective data, such as product parameters, are limited in scope. Attributes and their weights extracted from product online reviews represent the general preferences of most consumers, failing to accurately capture the preferences of specific users. Another category of methods [8], such as recommendation methods, calculates preferences based on user data. While these methods cater to personalized preferences, they only consider historical preferences rather than the current preferences of the decision-maker.

When calculating product scores for each product attribute, the existing researches primarily rely on product online reviews. These studies focus on establishing relationships between online reviews and attributes [25] and calculating product attribute scores based on the characteristics of online reviews and their relationship with attributes [55]. However, some methods calculate product attribute scores only based on one or more sentiment words corresponding to attribute words in online reviews [26], potentially overlooking the overall perspective of consumers within online reviews. Conversely, other methods use entire online reviews as units to calculate product attribute scores [47], but real-world online reviews usually address multiple attributes of a product, creating a contradiction with methods that evaluate a single attribute score of online reviews.

In conclusion, the existing research has some limitations for user preference analysis. It is necessary to consider the current preference of specific users, rather than general user preferences or historical user preferences. Furthermore, appropriate methods should be adopted to extract structured information from unstructured user preferences.

# 3. Methods

In this section, following the problem statement, the study analyzes product attributes and their weights based on online reviews and linguistic requirements. Furthermore, a MADM method is constructed based on attribute weights and product attribute scores, which will be determined later.

## 3.1 *Problem description and framework of the proposed method*

Given the linguistic requirements $s_{req}$, which outline the current requirements of the special user when selecting products, and a set of products $X = \{X_1, X_2, ..., X_n\}$, each product $X_i$ has a collection of online reviews $R_i = \{r_1^i, r_2^i, ..., r_{K_i}^i\}$, where $K_i$ represents the total count of online reviews for product $X_i$. The problem in this study is to rank the products $X_i$ based on linguistic requirements $s_{req}$, and online reviews $R_i$. The main notations used in this study are defined in Table 1.

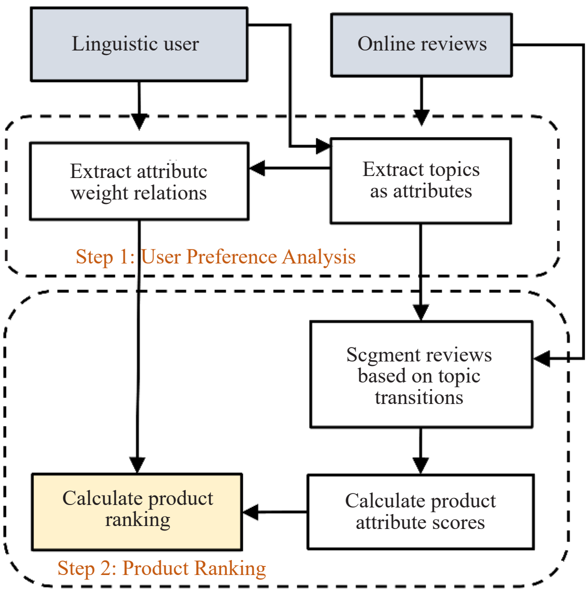


**Figure 1.** The framework of the proposed method which ranking products based on linguistic requirements and online reviews

As is shown in Figure 1, the proposed method consists of two steps: user preference analysis and product ranking. In the user preferences analysis step, topics are extracted as attributes and their weights from the online reviews $R_i$ and the linguistic requirements $s_{req}$. In the product ranking step, product rankings are calculated using ELECTRE III based on attribute weights and product attribute scores. A detailed description of the ELECTRE III model is provided in Appendix A.

**Table 1.** List of main notations

| Notations | Description |
|---|---|
| $s_{req}$ | Linguistic requirements of users |
| $X = \{X_1, X_2, ..., X_n\}$ | Set of products |
| $X_i$ | Product $i$ of set $X$ |
| $R_i = \{r_1^i, r_2^i, ..., r_{K_i}^i\}$ | Set of online reviews of product $i$ |
| $r_k^i$ | Online review $k$ of product $i$ in set $R_i$ |
| $T' = \{T_1', T_2', ..., T_{J'}'\}$ | Set of topics extracted from all the $R_i$ |
| $T_{j'}' = (j', p')$ | Topic $j'$, $p'$ is the topic distribution probability |
| $T = \{T_1, T_2, ..., T_J\}$ | Set of topics used as attributes, $T \subseteq T'$ |
| $T_j$ | Attribute $j$ of set $T$, which is the topic mentioned in $s_{req}$ |
| $w = \{w_1, w_2, ..., w_J\}$ | Set of weight |
| $w_j$ | Weight of attribute $T_j$ |
| $D_r = \{d_1, d_2, ..., d_J\}$ | Set of support degrees of the text $r$ |
| $d_j$ | Support degree for attribute $T_j$ |
| $E = [e_{tj'}]_{len(r) \times J'}$ | Matrix of support degrees of the text $r$ |
| $e_{tj'}$ | Support degree of the $t$-th word in the text $r$ for the topic $T_{j'}'$ |
| $len(r)$ | Count of word for $r$ |
| $v_r^t = (y_1, y_2, ..., y_{J'})$ | Vector of support degrees of the $t$-th word in the text $r$ |
| $y_{j'}$ | Support degree for the topic $T_{j'}'$ |
| $H = \{H_1, H_2, ..., H_Z\}$ | Set of weight relationship constraints |
| $H_z = (D_f, D_g)$ | Weight relationship constraint, denoting the inequality $\sum_{j=1}^{J} d_{fj} \cdot w_j > \sum_{j=1}^{J} d_{gj} \cdot w_j$ |
| $S_i = \{S_{i1}, S_{i2}, ..., S_{iJ}\}$ | Set of attribute scores for product |
| $S_{ij}$ | Score of the product $X_i$ for the attribute $T_j$ |
| $L_i = \{l_1, l_2, ..., l_B\}$ | Set of scores for all the text $u_b$ segmented from $R_i$ |
| $l_b = (d_1, d_2, ..., d_J, s)$ | Vector of support degrees and sentiment score of $u_b$ for the attribute $T_j$ |

## 3.2 *User preferences analysis based on linguistic requirements*

This section analyzes linguistic requirements expressed in natural language based on linguistic rules and deep learning. Prior to this analysis, the Biterm Topic Model (BTM) method is used to extract topics from all the online reviews $R_i$ of each product $X_i$, as well as from the linguistic requirements $s_{req}$. The BTM is a method specifically designed for analyzing and extracting the thematic structure of short texts. This method leverages biterm extraction to capture the co-occurrence information of word pairs within short texts. It combines this information with the Gibbs sampling method for parameter estimation and iterative optimization, ultimately uncovering the underlying topics within a collection of short texts [17].

The extracted topics are denoted as $T' = \{T_1', T_2', ..., T_{J'}'\}$. Among these, topics mentioned in the linguistic requirements are selected as attributes, resulting in a finite set of the attributes $T = \{T_1, T_2, ..., T_J\}$. The extracted topic list $TL$ shows the support degree of all the words for topics, which lists all the words $v$ that support the topic $T_{j'}'$ and the support degree $y_{j'}$.

The preference analysis consists of two steps: Firstly, improved linguistic rules based on [56] are applied to constrain the linguistic requirements according to part-of-speech tags, facilitating the disaggregation of the linguistic requirements into entities and their relationships. Secondly, the semantics of the words within the linguistic requirements are interpreted to ascertain the weight relationships between attributes.

### 3.2.1 *Entity extraction and entity relation extraction*

Given the unstructured linguistic requirements [6], which often involve complex relationships among multiple attributes, it is essential to apply constraints on the linguistic requirements. This constraint ensures that the texts represent either the relationship between two entities or provide a description of a solitary entity. For example, the phrase “A suitable price is most important” can be constrained to describe a solitary entity in the form of <*suitable price, most important*>. In addition, the statement “Product brand is more important than color” can be constrained to describe the relationship between two entities in the form of <*product brand, more important than, color*>.

While applying part-of-speech constraints on the natural language format, the proposed model uses a Bidirectional Long Short-Term Memory-Conditional Random Field (BiLSTM-CRF) [57] to assign parts-of-speech tags to each word in the sentences. Based on these tags, specific rules are applied to structure the sentence structure, ensuring that the linguistic requirements adhere to the specified formats while identifying entities and their relationships. The rules based on [56] include three specific patterns: entity relationship VP1, entity relationship VP2, and entity NP. Detailed descriptions of these rules are shown in Appendix B, C and D.

A. Verifying whether the linguistic requirement text conforms to the pattern “entity NP + entity relationship VP1 + entity NP”, indicating that the beginning and end of the text represent the two entities, with the intervening segment describing the relationship between them.

B. Verifying whether the linguistic requirement conforms to the pattern “entity NP + entity relationship VP2”, where the first segment describes the entity and the latter segment provides a qualifier for that entity.

C. If the conditions in A are met, the text corresponding to the “entity NP” rule at the beginning and end of the linguistic requirement is extracted as the entity, denoted as $NP1$ andc; the text corresponding to the “entity relationship VP1” rule at the middle of the linguistic requirement is extracted as the entity relationship, denoted as $VP1$.

D. If the conditions in A are not met, but the conditions in B are met, the text corresponding to the “entity NP” rule at the beginning of the linguistic requirement is extracted as an entity, denoted as $NP3$; the text corresponding to the “entity relationship VP2” at the end of the linguistic requirement is extracted as the entity relationship, denoted as $VP2$.

E. If the conditions in A and B are not met, the linguistic requirement is shown to the user to modify it, and repeat the above steps for the modified text.

### 3.2.2 *Calculation of attributes and their weight constraint*

After applying linguistic rules to constrain and analyze the linguistic requirements, this step proceeds with semantic constraints and the analysis of entity relationships using the three rules: Entity NP, Entity Relationship VP1, and Entity Relationship VP2. In this phase, the text corresponding to these rules is processed, including $NP1$, $NP2$, $NP3$, $VP1$, and

$VP2$, and uncertain expressions are transformed into definite preference relations.

(1) *Generation of the attribute vector of entity*.

The attribute vector is extracted from the entity based on the topic list $TL$. Given the text $NP1$, $NP2$, and $NP3$, the extracted information is represented as the normalized support degrees of the text for all attributes in the form of $D = \{d_1, d_2, ..., d_J\}$, where $d_j$ denotes the support degree of the text for attribute $T_j$. The calculation of the support degree for attributes is performed below.

To compute the support degrees for attributes, a support matrix $E = [e_1, e_2, ..., e_{J'}]$ is generated from the text, with the formula as follows:

$$E = \left[e_{tj'}\right]_{len(NP)\times J'} \tag{1}$$

where text $NP$ represents the text for Entity NP, $len(NP)$ denotes the number of words in the text $NP$. The element $e_{tj'}$ in the matrix represents the support degree of the $t$-th word in the text $NP$ for the topic $T'_{j'}$, $y_{j'}$ signifies the support degree of the $t$-th word in the text $NP$ for the topic $T'_{j'}$. If the word does not support any topic, lemmatize the word. If the lemmatized word does not support any topic $T'_{j'}$, then $e_{tj'}$ equals 0.

Next, the support degrees for each column of the support matrix $E$ are summed to obtain the support degree for the topic $T'_{j'}$, with the formula as follows:

$$E_{j'} = \sum_{t=1}^{len(NP)} e_{tj'} \tag{2}$$

From this, the support degree for the attributes is extracted as:

$$d'_j = \begin{cases} E_{j'}, & T_j = T'_{j'} \\ 0, & \text{Other} \end{cases} \tag{3}$$

where $E_{j'}$ denotes the support degree for the topic $T'_{j'}$, $d'_j$ represents the support degree of the online review $r_k^i$ for the attribute $T_j$.

Finally, the normalized the support degree $d_j$ is calculated as:

$$d_j = \frac{d'_j}{\sum_{j=1}^{J} d'_j} \tag{4}$$

where all the $d_j$ constitute the support degrees of the entity text for all attributes, represented as $D = \{d_1, d_2, ..., d_J\}$.

(2) *Generation of the attribute weight constraint*.

This step describes a process for constraining and analyzing the extracted texts corresponding to the two types of rules: Entity Relationship VP1 and Entity Relationship VP2. Utilizing the constructed preference dataset, the texts are compared with all the texts in the dataset that match the entity relationship types through Sentence-Bidirectional Encoder Representations from Transformers (SBERT) [58] similarity comparisons. Entity relationships associated with texts that exhibit the highest similarity, exceeding the predefined similarity threshold $\omega$, are selected as the entity relationships. If no suitable entity relationships can be identified, domain experts will annotate the entity relationships accordingly.

The entity relationships for Entity Relationship VP1 are categorized into two types: "better than" and "worse than". For example, the text "be better than" indicates the relationship "prior to". Conversely, Entity Relationship VP2 evaluates the importance on a scale from 0% to 100%, divided into $M$ equal intervals, represented as $c_0, c_1, ..., c_{M-1}$. The importance of the $m$ level is calculated as:

$$c_m = \frac{m}{M-1} \times 100\% \quad (m = 1, 2, \ldots, M-1) \tag{5}$$

The data in the preference dataset contains three elements: the type of entity relationship, the entity relationship itself, and the natural text. The detailed construction steps are as follows:

A. Initialization: Language experts provide common texts that correspond to Entity Relationships VP1 and VP2.

B. Annotation: A group of experts annotates these texts with corresponding relationships. In cases where there is a disagreement, the relationship of that with the highest annotation count is selected. If more than one relationship has the same highest count, the annotations are reviewed to all experts for reevaluation until a clear, most-frequent relationship is identified. The natural text, along with its corresponding entity relationship and type, is then added to the preference dataset.

C. Enhancement: When users utilize this method, phrases and entity relationship types from their linguistic requirements that match the dataset format are collected and annotated by experts following the above steps.

After extracting the entity relationship, attribute weight constraints are extracted from the linguistic requirements based on the corresponding entities and entity relationships. Given the attribute vector of the entities *NP*1, *NP*2, and *NP*3, and the entity relationship of *VP*1, and *VP*2. The various segments of the text can be directly converted into their corresponding entities and relationships, allowing for the establishment of attribute weight relationships. The detailed steps are outlined as follows:

A. If the linguistic requirements conform to the structure "Entity NP + Entity Relationship VP1 + Entity NP", the support degrees for all attributes by the two entities are represented as $D_1 = \{d_{11}, d_{12}, \ldots, d_{1J}\}$ and $D_2 = \{d_{21}, d_{22}, \ldots, d_{2J}\}$. When the entity relationship corresponding to VP1 is "better than", the attribute weight relationship is expressed as: $\sum_{j=1}^{J} d_{1j} \cdot w_j > \sum_{j=1}^{J} d_{2j} \cdot w_j$. Conversely, if the relationship indicates "worse than", the relationship is represented as: $\sum_{j=1}^{J} d_{2j} \cdot w_j > \sum_{j=1}^{J} d_{1j} \cdot w_j$. Here, $w_j$ denotes the weight of attribute $T_j$. This inequality is then added to the set of weight entity relationships, denoted as $H_z$, an element of $H$, formatted as $(D_1, D_2)$ where the weighted sum of $D_1$ is larger than that of $D_2$.

B. If the linguistic requirements conform to the structure "Entity NP + Entity Relationship VP2", the support degrees for the attributes are represented as $D = \{d_1, d_2, \ldots, d_J\}$, and the entity relationship is defined as $c_m$. For all pairs of inconsistent entity relationships, the noun phrase with higher importance level will have its support degrees denoted as $D_f = \{d_{f1}, d_{f2}, \ldots, d_{fJ}\}$, while the noun phrase with lower importance level will be denoted as $D_g = \{d_{g1}, d_{g2}, \ldots, d_{gJ}\}$. The attribute weight relationship is expressed as $H_z : \sum_{j=1}^{J} d_{fj} \cdot w_j > \sum_{j=1}^{J} d_{gj} \cdot w_j$. This inequality between $D_f$ and $D_g$, denoted as $H_z$ in the form of $(D_f, D_g)$ is added to the set of the weight relationships $H$.

C. A feasible region is constructed from the inequalities within the entity relationship set. If no feasible region exists to satisfy all inequalities, the linguistic requirements corresponding to all subsets of the weight constraint set that have feasible domains will be shown to the user. The user will have the option to modify these requirements or select a subset to make the feasible region exist.

## 3.3 *Product ranking method*

This section proposes a MADM method designed to leverage attribute weight relationships and online reviews for product ranking. Firstly, to ensure that online reviews accurately evaluate each attribute, the method segments the online reviews and performs sentiment scoring on the segmented parts. Secondly, based on the support degrees of the attributes derived from the segmented online reviews, the method aggregates the sentiment scores from online reviews to calculate the ratings for each product for each attribute. Lastly, the method conducts an aggregation of the product attribute scores using the ELECTRE III model, considering the established attribute weight relationships, leading to product ranking.

### 3.3.1 *Online reviews segmentation and sentiment scoring*

Based on the coherence of natural language, continuous text typically focuses on a limited set of attributes, with shifts in support degrees for different topics occurring between segments describing different attributes. By leveraging this characteristic of natural language, online reviews can be segmented based on topic transitions. This step segments all the online reviews $R_i$ for product $X_i$ according to the attributes they support, facilitating a more accurate calculation of sentiment scores for each attribute. The detailed steps are as follows:

(1) All the online reviews will be used to generate the support matrix $E$ according to Equation (2), and the support degrees for all the attributes $D = \{d_1, d_2, ..., d_J\}$ will be calculated based on the entity NP derived from the semantic constraints and analysis in Section 3.2.

(2) Given the similarity threshold $\omega$. For each online review, the cosine similarity $Sim(v_k^{i,t}, v_k^{i,t+1})$ between the two adjacent words $v_k^{i,t}$ and $v_k^{i,t+1}$ will be calculated. If $Sim(v_k^{i,t}, v_k^{i,t+1}) < 1 - \omega$, this indicates a significant shift in semantic, prompting a segmentation between $v_k^{i,t}$ and $v_k^{i,t+1}$. Based on these segmentations, all the segment text $u_b$ will be extracted from the online reviews $R_i$. The formula for calculating cosine similarity is as follows:

$$Sim\left(v_k^{i,t}, v_k^{i,t+t'}\right) = \frac{v_k^{i,t} \cdot v_k^{i,t+t'}}{\left|v_k^{i,t}\right| \cdot \left|v_k^{i,t+t'}\right|} \tag{6}$$

(3) The analysis results of the segmented online reviews are organized into $l_b$, which in the segmented online reviews set $L_i$. Each analysis result of the text $u_b$ is represented as $l_b = \{d_1, d_2, ..., d_J, s\}$, where $u_b$ is the $b$-th part of segmented online reviews $R_i$; $d_j$ represents the support degree of $u_b$ for the attribute $T_j$ as calculated in the step (1); and $s$ is the sentiment score of $u_b$, using the LSTM method.

### 3.3.2 *Product attribute scoring*

In this step, the online reviews are aggregated using the ELECTRE III model based on their support degrees for the attributes, resulting in the ranking of products. The specific method is as follows:

For all $l_b$ in the scores set $L_i$ of the product $X_i$, the sentiment scores for each attribute $T_j$ are extracted and aggregated. This process yields the product attribute scores $S_i = \{s_{i1}, s_{i2}, ..., s_{iJ}\}$ for the product $X_i$. The detailed steps are outlined in Algorithm 1.

**Algorithm 1** Sentiment Aggregation ($L_i$, $\omega_s$)
**Input**: $L_i = \{l_1, l_2, ..., l_B\}$, $\omega_s$
**Output**: $S_i = \{s_{i1}, s_{i2}, ..., s_{iJ}\}$

**Begin**
  **for** $j \leftarrow 1$ to $J$ **do**
    $count_j = 0$, $sum_j = 0$
    **for** $b \leftarrow 1$ to $B$ **do**
      **if** $l_b \cdot d_j > \omega_s$ **then**
        $sum_j += l_b \cdot d_j * l_b \cdot s$
        $count_j += 1$
      **end if**
    **end if**
    $s_{ij} = \dfrac{sum_j}{count_j}$
  **end for**
**end**

In this context, $\omega_s$ represents the threshold for determining whether the score of $l_b$ for the attribute $T_j$ should be included. The variable $s_{ij}$ denotes the sentiment score of the product $X_i$ for the attribute $T_j$. Additionally, $l_b \cdot s$ indicates the sentiment score of the text $u_b$, while $l_b \cdot d_j$ represents the support degree of $u_b$ for the attribute $T_j$. Finally, $B$ signifies the total number of $l_b$ in the set $L_i$.

### 3.3.3 *Product ranking*

In this step, the product attribute scores, denoted as $S_i = \{s_{i1}, s_{i2}, ..., s_{iJ}\}$, are combined with the attribute weight relationships derived from the $H$, which is used to compute the product ranking by the ELECTRE III method. To increase the accuracy of ranking, the method applies linear programming to determine the feasible region for the product attribute weights. Based on this feasible region, Monte Carlo simulations based on ELECTRE III method are used to approximate the product ranking.

The specific method for calculating the range of product attribute weights is outlined as follows:

$$
\begin{aligned}
&Max\ w_j\ (j = 1, 2, \ldots, J) \\
&\text{s.}t.: \begin{cases} \sum_{j=1}^{J} d_{fj} \cdot w_j > \sum_{j=1}^{J} d_{gj} \cdot w_j, \ \left(D_f, D_g\right) \in H \\ \sum_{j=1}^{J} w_j = 1, \ \ w_j \geq 0 \end{cases} \\
&Min\ w_j\ (j = 1, 2, \ldots, J) \\
&\text{s.}t.: \begin{cases} \sum_{j=1}^{J} d_{fj} \cdot w_j > \sum_{j=1}^{J} d_{gj} \cdot w_j, \ \left(D_f, D_g\right) \in H \\ \sum_{j=1}^{J} w_j = 1, \ \ w_j \geq 0 \end{cases}
\end{aligned}
\tag{7}
$$

The first constraint represents all inequalities within the $H$ set, while the second constraint ensures that the sum of all attribute weights equals 1.

The steps for using Monte Carlo simulation are as follows: First, a random set of attribute weights is generated in the feasible region of the $H$ set. Second, using the generated attribute weights and the decision matrix $D(X_i)$, the ELECTRE III method is used to compute the rankings $o'_{hi}$ for all products $X_i$ during the $h$-th Monte Carlo simulation. This process is repeated $\omega_t$ times to calculate the average ranking $o_i'$, which is then sorted in ascending order to establish the final product ranking $o_i$.

The formula for calculating the average product ranking is as follows:

$$
o_i' = \frac{\sum_{h=1}^{\omega_t} o'_{hi}}{\omega_t} \tag{8}
$$

where $o_i'$ represents the average product ranking, $o'_{hi}$ denotes the ranking of the product $X_i$ during the $h$-th Monte Carlo simulation. The final product ranking $o_i$ is the ascending order of $o_i'$.

# 4. Case study 1: accommodation ranking based on public datasets

In this section, a case study is presented to demonstrate the effectiveness of the proposed method. Additionally, this section compares the proposed method with other similar methods.

## 4.1 *Data collection and processing*

The public dataset, *Booking.com Hotel Reviews*, was collected from the popular travel booking website (https://www.booking.com/), comprising hotel online reviews and ratings. This dataset features over 700,000 records detailing customer online reviews and ratings for various hotels in different locations. It provides substantial data support for analysis and research in the tourism sector. For the case study, five hotels meeting certain requirements were selected. These hotels have significant differences in overall ratings and have similar counts of online reviews. The linguistic requirements utilized in this case study were derived from a designed questionnaire, which guided users through common hotel attributes, such as price and service level, using examples and questions. A total of four linguistic requirements was collected and added into a user preferences dataset.

In addition, the data was processed. Given that the method is designed for English text, online reviews were filtered using the Python language detection tool, LangDetect. To enhance topic extraction, both the online reviews and linguistic requirements were text cleaned, which involved removing punctuation, emojis, tokenization, eliminating stop words, and performing lemmatization. Topics are extracted from the processed online reviews and linguistic user preferences.

## 4.2 *Product ranking*

Step 1: User preference analysis was performed using the collected data, following the steps outlined in Section 3.2. Prior to applying the BTM for topic extraction, the user linguistic requirements were repeated several times to improve the representation of user opinions. This text was combined with the online review texts for topic extraction, generating a topic list. This list is used to establish a support matrix for online reviews and linguistic user preferences.

After topic extraction, the method analyzed the linguistic requirements based on three predefined rules. The specified segments from these requirements were compared with the preference dataset to derive the corresponding entity relationships, which facilitated the calculation of the attribute weight relationships. Part of the preference dataset is presented in Table 2, the identified entities and their support degrees for the attributes are shown in Table 3, and the entity relationships are detailed in Table 4.

**Table 2.** The entity relationships and other information extracted by the proposed method from the linguistic requirements

| Entity relationships | Types | Natural texts |
|---|---|---|
| Better than ($\succ$) | VP1 | Better than, superior to, preferable to, surpass, … |
| Worse than ($\prec$) | VP1 | Not as good as, inferior to, less superior than, … |
| 100% | VP2 | Extremely important, crucial, vital, paramount, essential, … |
| 80% | VP2 | Important, significant, key, major, critical, … |
| 60% | VP2 | Moderately important, fairly significant, … |
| 40% | VP2 | Somewhat important, minor, insignificant, peripheral, … |
| 20% | VP2 | Unimportant, negligible, trivial, irrelevant, … |
| 0% | VP2 | Extremely unimportant, completely trivial, … |

**Table 3.** The support degrees of the entities for all the attributes extracted by the proposed method from the linguistic requirements

| Entities | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ |
|---|---|---|---|---|---|
| The management and safety | 0.1936 | 0.3045 | 0.2591 | 0.1367 | 0.5101 |
| Price | 0.1076 | 0.0480 | 0.0426 | 0.0113 | 0.2933 |
| The quality of service | 0.1643 | 0.3554 | 0.0527 | 0.0838 | 0.0047 |
| Condition of the facilities | 0.2993 | 0.1772 | 0.4504 | 0.2562 | 0.0067 |
| Food | 0.2351 | 0.1149 | 0.1952 | 0.5120 | 0.1852 |
| Consumer experience | 0.1936 | 0.3045 | 0.2591 | 0.1367 | 0.5101 |

**Table 4.** The entity relationships and other information extracted by the proposed method from the linguistic requirements

| Requirement texts | Types | Entities | Entity relationship | Similarity |
|---|---|---|---|---|
| The management and safety are the most important | VP1 | The management and safety | 100 | 0.9316 |
| Price is more important than the quality of service | VP1 | Price/the quality of service | better than | 0.9761 |
| Condition of the facilities is less important than food | VP2 | Condition of the facilities/food | worse than | 0.9755 |
| Consumer experience is very important | VP2 | Consumer experience | 80 | 0.9219 |

Step 2: In accordance with Section 3.3, the online reviews were segmented and evaluated for sentiment scores. The proposed similarity threshold $\omega$ is set to 0.8 to achieve the best effect of the online review segmentation. For instance, the sentiment score for the segmented online review "everything was perfect, quite, cozy place to relax" was 0.4. The attribute support degrees for this review were {0.29, 0.32, 0.23, 0.11, 0.05}. These sentiment scores were aggregated and sorted by the proposed method in this paper and a method based on LDA and ELECTRE I [21], which is shown in Table 5 and Figure 2. The average product ranking of [21] is obtained through Monte Carlo simulation in Section 3.3.3.

**Table 5.** The average product rankings of all the products calculated by the proposed method and the LDA and ELECTRE I-based method [21]

| Products | LDA and ELECTRE I [21] | This paper |
|---|---|---|
| $X_1$ | 2.0000 | 2 |
| $X_2$ | 4.0000 | 4 |
| $X_3$ | 4.9485 | 5 |
| $X_4$ | 3.0460 | 3 |
| $X_5$ | 1.0006 | 1 |

Thus, the method can effectively calculate the product ranking based on linguistic requirements and online reviews.

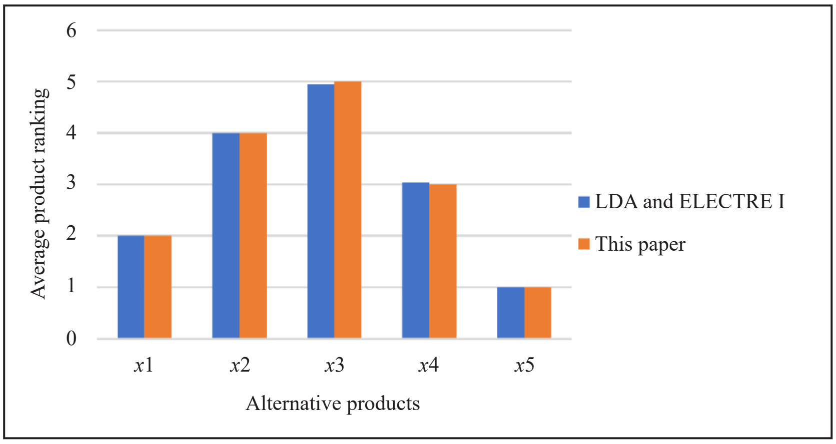


**Figure 2.** The average ranking of the alternatives in this case derived from the LDA and ELECTRE I-based method [21] and the proposed method

## 4.3 *Comparative analysis*

In this section, several similar methods are compared with the proposed method to analyze the effectiveness. This section is divided into five steps. Section 4.3.1 introduces the concept of accuracy as a metric for evaluating the method. Section 4.3.2 analyzes the effectiveness of the user preference method. Section 4.3.3 discusses the effectiveness of the product rating method. Section 4.3.4 shows the effectiveness of the overall performance. Section 4.3.5 discusses the settings of thresholds. The analysis uses the hotel ratings and review data provided by the dataset *Booking.com Hotel Reviews*.

To minimize error, this section selects several pairs of hotels $X_\alpha$ and $X_\beta$ from the *Booking.com Hotel Reviews* dataset, where the number of reviews exceeds the review threshold $\omega_R$, the score of segmented online reviews is more than the score threshold $\omega_s$, and the rating difference exceeds the rating threshold $\omega_x$. For experimental purposes, the score threshold $\omega_s$ is set to 0.4, the review threshold $\omega_R$ is set to 30, and the rating threshold $\omega_x$ is set to 0.7. To ensure that the analysis approximates the real scenario, random sampling is used to obtain data, after which the average accuracy is calculated. Here, the sample size $\omega_t$ is set to 1,000. BTM model used in the method requires three parameters, including the number of topics $Z$ and two hyperparameters $\alpha$, $\beta$. We set $Z$ to 5, $\alpha$ to 0.2 and $\beta$ to 0.2 to minimize the perplexity of the model.

### 4.3.1 *Accuracy analysis*

This section introduces three accuracy metrics to evaluate the performance of the method: Ranking Correctness (AC1), Ranking Distance (AC2), and Ranking Similarity (AC3).

**Definition 1** The metric ranking correctness (AC1) measures the proportion of correctly ranked pairs in a given ranking. The mathematic exception for AC1 is 0.5; a value above this indicates method effectiveness, with higher values suggesting better performance. For a set of $n$ options, the correctness is defined as follows:

$$F(X_\alpha, X_\beta) = \begin{cases} 1, & o_\alpha > o_\beta \\ 0, & \text{other} \end{cases} \tag{9}$$

The accuracy of the method is evaluated by using the accuracy marking method, where the $\omega_t$ distinct pairs of options $X_\alpha$ and $X_\beta$ ($0 < \alpha < \beta < n + 1$) that meet the specified conditions are randomly selected. The formula for calculating accuracy is defined as:

$$\text{AC1} = \frac{\sum_{h=1}^{\omega_t} F\left(X_\alpha, X_\beta\right)}{\omega_t \cdot \binom{n}{2}} \tag{10}$$

where $h$ denotes the $h$-th Monte Carlo simulation.

**Definition 2** The metric ranking distance (AC2) measures the distance between the method's ranking result and the standard ranking result. The mathematic exception for AC2 is 0.67; values above this indicates method effectiveness, with higher values suggesting better performance. It is defined as:

$$\text{AC2} = 1 - \frac{\sum_{h=1}^{\omega_t} \sum_{j=1}^{n} \left| o_{hi} - i \right|}{\omega_t \cdot n^2} \tag{11}$$

where $o_{hi}$ denotes the ranking of the product $X_i$ during the $h$-th Monte Carlo simulation.

**Definition 3** The metric ranking similarity (AC3) views ranking result as vectors and computes the cosine similarity between the method's ranking result and the standard ranking result. The mathematic exception for AC3 is 0.67; values above this indicates method effectiveness, with higher values suggesting better performance. The similarity uses the formula:

$$\text{AC3} = \frac{\sum_{h=1}^{\omega_t} \sum_{j=1}^{n} (o_{hi} \cdot i)}{\omega_t \cdot \sum_{j=1}^{n} i^2} \tag{12}$$

where $o_{hi}$ denotes the ranking of the product $X_i$ during the $h$-th Monte Carlo simulation.

### 4.3.2 *Effectiveness of user preference analysis*

Attribute weight relationships are defined as inequalities based on the level of support for a topic within reviews. Relationships are disregarded if the difference between two attribute weights is below a specified threshold, denoted as $\omega_w$, which is set to 0.1. For example, if the support degree for the attribute $T_1$ is 0.4 and for attribute $T_2$ is 0.2, there exists a weight relationship inequality $w_1 > w_2$. If the support degrees for the attributes $T_6$ and $T_7$ are 0.12 and 0.11, respectively, and the preference threshold $\omega_w$ is 0.05, then $0.12 - 0.11 = 0.01 < 0.05$, indicating that no weight relationship inequality exists between $w_6$ and $w_7$.

Let the product ranking step uses the ELECTRE III method, the product attribute scores are calculated by Section 3.3.2. The user preference analysis part of this method is compared with the user-direct decision method [47], entropy weighting method [28], feature evaluation and attention-based method [46], Analytic Hierarchy Process (AHP) method [39], large language model [59], and Principal Component Analysis (PCA) method [60], under the condition that other parameters and the steps of the method remain consistent. The results are shown in Table 6, indicating that the performance of this method in user preference analysis surpasses that of the other methods.

**Table 6.** The comparison of the accuracy of the user preference analysis method in this paper with other preference analysis methods

| Method | AC1 | AC2 | AC3 |
|---|---|---|---|
| This paper | 0.7827 | 0.8291 | 0.8498 |
| User-direct decision [47] | 0.7319 | 0.8054 | 0.8341 |
| Large language model [59] | 0.5294 | 0.7402 | 0.5536 |
| Entropy weighting [28] | 0.7034 | 0.7874 | 0.8024 |
| Feature evaluation and attention-based [46] | 0.6823 | 0.7941 | 0.7132 |
| AHP [39] | 0.1690 | 0.6356 | 0.4491 |
| PCA [60] | 0.5802 | 0.7029 | 0.7031 |

The weights of these methods are shown in Figure 3, as can be seen from the figure, the values of $w_2$, $w_3$, and $w_4$ calculated by these sets of methods do not differ significantly, and their impact on the results is minor. The larger the difference between $w_1$ and $w_5$ calculated by other methods compared to the method proposed in this paper, the lower their accuracy is. Notably, the proposed method demonstrates a superior ability to effectively extract attribute weights compared to similar methods.

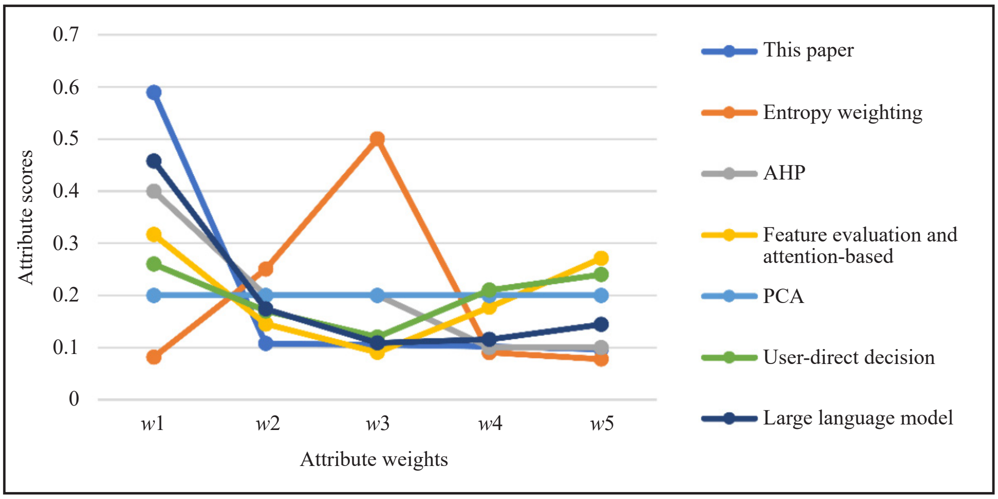


**Figure 3.** Comparison of the attribute weight extraction method in this paper with other similar methods

### 4.3.3 *Effectiveness of product ranking*

Let the attributes are extracted from the linguistic requirements $s_{req}$ and all the online reviews $R_i$. Attribute weight relationships are calculated by the proposed method in Section 3.2.2. The sentiment scoring method is BiLSTM-CRF. The product ranking performance of this method is compared with several other methods, including VlseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR) [28], TOmada de Decisão Interativa e Multicritério (TODIM) [47], Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [46], RankNet, ListNet [9], Compromise Ranking of Alternatives from Distance to Ideal Solution (CRADIS) [61], and Random Forest [62]. The results are presented in Table 7, demonstrating that, under the condition that other parameters and steps of the method remain consistent, the proposed method outperforms the other methods in terms of product ranking performance.

**Table 7.** The comparison of the product ranking method in this paper with other product ranking methods

| Method | AC1 | AC2 | AC3 |
|---|---|---|---|
| This paper | 0.8018 | 0.8308 | 0.8911 |
| CRADIS | 0.6750 | 0.7573 | 0.7761 |
| VIKOR | 0.5217 | 0.7094 | 0.6301 |
| TODIM | 0.5807 | 0.6912 | 0.6585 |
| TOPSIS | 0.4108 | 0.6087 | 0.5505 |
| RankNet | 0.2149 | 0.5179 | 0.4336 |
| ListNet | 0.2976 | 0.5786 | 0.4842 |
| LambdaMart | 0.3558 | 0.5509 | 0.5376 |
| Incremental Reduced Support Vector Machines (IRSVM) | 0.2287 | 0.5076 | 0.5001 |
| Support Vector Machine (SVM) | 0.3890 | 0.5986 | 0.5655 |
| Random Forest | 0.0178 | 0.5221 | 0.3435 |

### 4.3.4 *Effectiveness of the proposed method*

Let the attributes are extracted from the linguistic requirements $s_{req}$ and all the online reviews $R_i$ by BTM method. In this context, the method proposed in this paper is compared with other similar methods, such as entropy weighting and VIKOR method [28], user-directed decision-making and TODIM method [47], feature evaluation and attention-based method with TOPSIS method [46], hierarchical attention network in conjunction with the RankNet method [9], and a method based on LDA and ELECTRE I [21]. Table 8 presents a comparison of the proposed method's overall performance metrics with those of similar methods and learning methods. The results indicate that the proposed method outperforms the others in all evaluated metrics.

**Table 8.** The comparison of the overall performance of this method with other methods

| Method | AC1 | AC2 | AC3 |
|---|---|---|---|
| This paper | 0.7635 | 0.8277 | 0.8434 |
| Entropy weighting method + VIKOR [28] | 0.5210 | 0.6908 | 0.6822 |
| User-direct decision + TODIM [47] | 0.5040 | 0.6824 | 0.6698 |
| Feature evaluation and attention-based + TOPSIS [46] | 0.4144 | 0.6329 | 0.5938 |
| Hierarchical attention network + RankNet [9] | 0.2976 | 0.5786 | 0.4842 |
| LDA + ELECTRE I [21] | 0.5616 | 0.7353 | 0.7314 |

### 4.3.5 *Setting of the thresholds*

This section identifies optimal thresholds that significantly enhance the effectiveness of comparative analysis.

As can be seen, the method includes reviews that do not distinctly support any attribute into the calculation range with low thresholds, thus leading to an excessive number of interfering items in the method's results and making it impossible to calculate an accurate product ranking. On the contrary, the count of reviews filtered out is too small with high thresholds, resulting in greater uncertainty in the outcomes and an inability to reflect the true attribute scores of the products. The proposed method is used to analyze the selection of the thresholds $\omega_s$, $\omega_R$, $\omega_x$ and $\omega_t$.

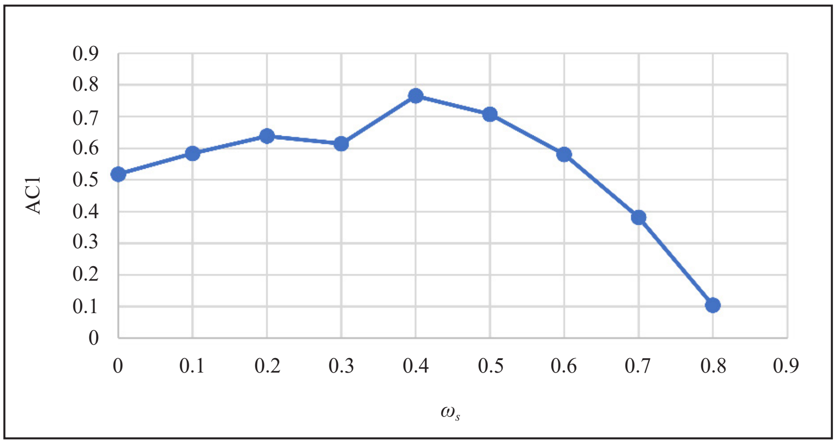


**Figure 4.** The ranking correctness (AC1) of the proposed method with different score threshold $\omega_s$

For the threshold $\omega_s$, which ranges from 0 to 1, this case study takes the proposed method as an example, with other thresholds remaining constant. The threshold $\omega_s$ is incremented by 0.1 intervals, and its appropriateness is determined using AC1 as the criterion, with the analysis results shown in Figure 4.

The range of the review threshold $\omega_R$ is approximately 1 to 1,000. With other conditions remaining constant in this case, the threshold $\omega_R$ is incremented by 100 intervals, and its appropriateness is determined using AC1 as the standard, with the analysis results shown in Figure 5.

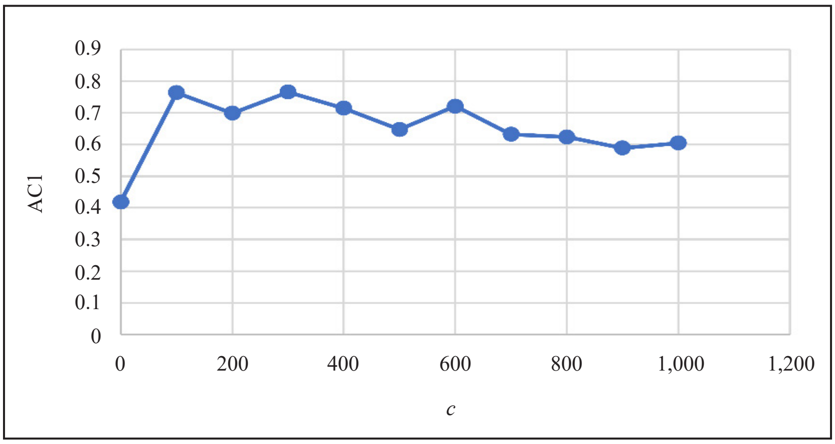


**Figure 5.** The AC1 of the proposed method with different review threshold $\omega_R$ from 1 to 1,000

It is evident that when the review threshold exceeds 100, the accuracy AC1 is the highest. To include as many products as possible within the calculated range of the proposed method, the threshold $\omega_R$ is further incremented from 1 to 100 by 10 intervals, with the analysis results shown in Figure 6. When $\omega_R$ is larger than 30, the accuracy AC1 approaches stability. Therefore, this case selects 30 as the value for the threshold $\omega_R$.

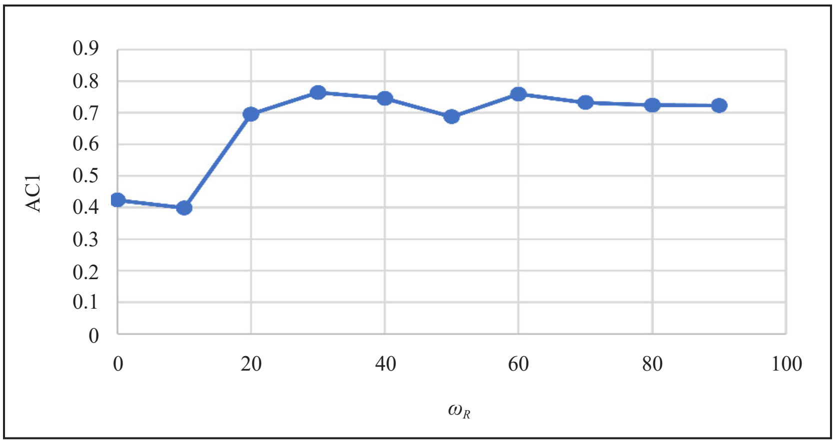


**Figure 6.** The AC1 of the proposed method with different review threshold $\omega_R$ from 1 to 100

Regarding the threshold $\omega_x$ and the number of samples $\omega_t$, a larger $\omega_X$ results in a smaller $\omega_t$. It leads to lower discriminability in product ranking and increases the uncertainty of the calculation results. Once the value of $\omega_x$ is determined, $\omega_t$ can also be established. Given that the standard rating range for products is from 3.8 to 10, and five products are selected for ranking in each experiment, the range of $\omega_x$ is approximately 0 to 1.5. With other conditions remaining constant in this case, $\omega_x$ is incremented from 0.1 to 1.5 by 0.2 intervals, and its appropriateness is determined using AC1 as the criterion, with the analysis results shown in Figure 7.

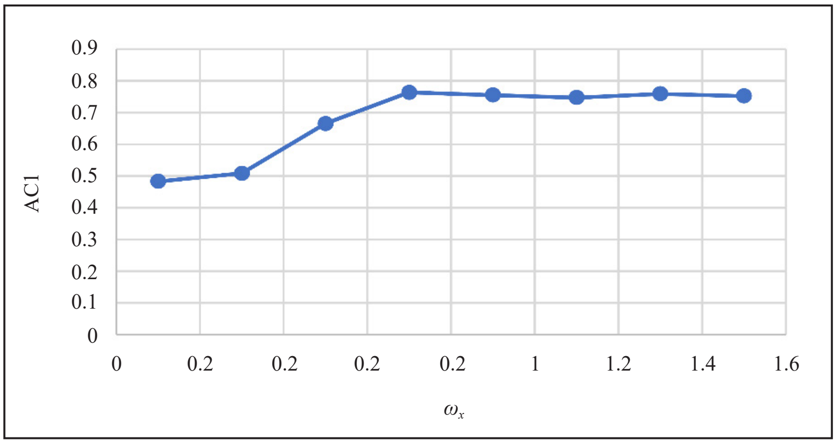


**Figure 7.** The AC1 of the proposed method with different review threshold $\omega_x$ from 0.1 to 1.5

It is evident that when $\omega_x$ is bigger than 0.7, the discriminability of products is relatively high. Let $\omega_x$ be 0.7, the method can select about 1,000 sets of eligible products for ranking in this case, hence $\omega_x$ is set to 1,000.

## 5. Case study 2: kindle book ranking based on public datasets

In this section, a case study is presented to demonstrate the effectiveness of the proposed method under the large-scale data scenarios. Additionally, this section compares the proposed method with other similar methods.

### 5.1 *Data collection and processing*

The public dataset, *Amazon Review Data (2018)*, was collected from the Amazon e-commerce platform (https://www.amazon.com/), including reviews (ratings, text, helpfulness votes) and product metadata (descriptions, category information, price, brand, and image features). This dataset contains over 5,000,000 online reviews and nearly 500,000 product metadata for kindle books. For the case study, five Kindle books meeting certain requirements were selected. These kindle books have significant differences in overall ratings and have similar counts of online review. The linguistic requirements used in this case study were derived from a designed questionnaire, which guided users to have a profound understanding of kindle books and consider their attributes, such as price, reading experience and platform portability. A total of five linguistic requirements was collected and added into a user preferences dataset.

In addition, the data was processed. The ratings of Kindle books are not considered in the datasets, it is necessary to acquire the ratings based on product metadata from Amazon.com. Given that the method is designed for English text, online reviews were filtered using the Python language detection tool, LangDetect, to select the English online reviews. To enhance topic extraction, both the online reviews and linguistic requirements were text cleaned, which involved removing punctuation, emojis, tokenization, eliminating stop words, and performing lemmatization. Topics are extracted from the processed online reviews and linguistic user preferences.

### 5.2 *Comparative analysis*

In this section, several similar methods are compared with the proposed method to analyze the effectiveness. To minimize error, this section selects several pairs of Kindle books $X_\alpha$ and $X_\beta$ from the *Amazon Review Data (2018)* dataset, where the number of reviews exceeds the similarity threshold $\omega$, the review threshold $\omega_R$, the score of segmented online reviews is more than the score threshold $\omega_s$, and the rating difference exceeds the rating threshold $\omega_x$. For experimental purposes, the score threshold $\omega_s$ is set to 0.4, the review threshold $\omega_R$ is set to 100, and the rating threshold $\omega_x$ is set to 0.5. To ensure that the analysis approximates the real scenario, random sampling is used to obtain data, after which the average accuracy is calculated. Here, the sample size $\omega_t$ is set to 10,000. The proposed similarity threshold $\omega$ is set to 0.8. BTM model used in the method requires three parameters, including the number of topics $Z$ and two hyperparameters $\alpha$, $\beta$. We set $Z$ to 10, $\alpha$ to 0.1 and $\beta$ to 0.1 to minimize the perplexity of the model. The accuracy metrics to evaluate the performance of the method, including Ranking Correctness (AC1), Ranking Distance (AC2), and Ranking Similarity (AC3), are given in Section 4.3.1.

While analyzing the effectiveness of the user preference analysis method, attribute weight relationships are defined as inequalities based on the level of support for a topic within reviews. The threshold of the difference between two attribute weights $\omega_w$ is set to 0.05. Let the product ranking step uses the ELECTRE III method, the product attribute scores are calculated by Section 3.3.2. The user preference analysis part of this method is compared with the methods mentioned in Section 4.3.2. The results are shown in Table 9, indicating that the performance of this method in user preference analysis surpasses that of the other methods under the large-scale data scenarios.

**Table 9.** The comparison of the accuracy of the user preference analysis method in this paper with other preference analysis methods under the large-scale data scenarios

| Method | AC1 | AC2 | AC3 |
|---|---|---|---|
| This paper | 0.8069 | 0.8583 | 0.8845 |
| User-direct decision [47] | 0.7794 | 0.8090 | 0.8284 |
| Large language model [59] | 0.4756 | 0.7027 | 0.5832 |
| Entropy weighting [28] | 0.6498 | 0.7556 | 0.7912 |
| Feature evaluation and attention-based [46] | 0.5290 | 0.6037 | 0.6514 |
| AHP [39] | 0.2257 | 0.5148 | 0.4842 |
| PCA [60] | 0.6013 | 0.7173 | 0.7597 |

While analyzing the effectiveness of the product ranking method, attributes are extracted from the linguistic requirements $s_{req}$ and all the online reviews $R_i$. Attribute weight relationships are calculated by the proposed method in Section 3.2.2. The sentiment scores are calculated by BiLSTM-CRF. The product ranking performance of this method is compared with the methods mentioned in Section 4.3.3. The results are presented in Table 10, demonstrating that, under the condition that other parameters and steps of the method remain consistent, the proposed method outperforms the other methods in terms of product ranking performance under the large-scale data scenarios.

While analyzing the effectiveness of the proposed method, the attributes are extracted from the linguistic requirements $s_{req}$ and all the online reviews $R_i$ by BTM method. In this context, the method proposed in this paper is compared with the methods mentioned in Section 4.3.4. Table 11 presents a comparison of the proposed method's overall performance metrics with those of similar methods and learning methods. The results indicate that the proposed method outperforms the others in all evaluated metrics under the large-scale data scenarios.

The result of the case study and its comparative analysis shows that the proposed method performs better than other similar methods under the large-scale data scenarios, demonstrating good scalability. The proposed method demonstrates superior performance compared to the similar methods under the large-scale scenarios when using limited computational resources, as well as outperforming its own performance in small-scale scenarios, indicating relatively minor overall performance bottlenecks.

**Table 10.** The comparison of the product ranking method in this paper with other product ranking methods under the large-scale data scenarios

| Method | AC1 | AC2 | AC3 |
|---|---|---|---|
| This paper | 0.8590 | 0.9086 | 0.9267 |
| CRADIS | 0.7241 | 0.7803 | 0.8023 |
| VIKOR | 0.5888 | 0.6923 | 0.6432 |
| TODIM | 0.6152 | 0.7216 | 0.6973 |
| TOPSIS | 0.4883 | 0.6759 | 0.6003 |
| RankNet | 0.1782 | 0.5162 | 0.4625 |
| ListNet | 0.2547 | 0.5987 | 0.5313 |
| LambdaMart | 0.4031 | 0.5757 | 0.5604 |
| IRSVM | 0.1211 | 0.4039 | 0.4709 |
| SVM | 0.4520 | 0.6118 | 0.5958 |
| Random Forest | 0.1096 | 0.5050 | 0.4459 |

**Table 11.** The comparison of the overall performance of this method with other methods under the large-scale data scenarios

| Method | AC1 | AC2 | AC3 |
|---|---|---|---|
| This paper | 0.8069 | 0.8583 | 0.8845 |
| Entropy weighting method + VIKOR [28] | 0.6198 | 0.7135 | 0.7054 |
| User-direct decision + TODIM [47] | 0.7424 | 0.7571 | 0.7881 |
| Feature evaluation and attention-based + TOPSIS [46] | 0.4941 | 0.6662 | 0.6151 |
| Hierarchical attention network + RankNet [9] | 0.2026 | 0.5392 | 0.5051 |
| LDA + ELECTRE I [21] | 0.6608 | 0.8060 | 0.7744 |

# 6. Conclusion

This paper proposes a method based on both linguistic requirements and online reviews, considering the current preferences of specific users. The method introduces a preference processing model based on deep learning and rules, enabling it to accurately extract the current preferences of specific users from the requirement texts in natural language form. Furthermore, it leverages a combination of deep learning and dictionaries to construct a scoring model for online reviews. To demonstrate its effectiveness, it applies the proposed method to a public dataset as a case study to demonstrate its effectiveness, emphasizing the advantages of this method in terms of ranking accuracy and analyzing the current preferences of specific users. The result of the case study shows that the performance of the proposed method significantly outperforms similar methods in terms of three accuracy metrics: Ranking Correctness (AC1), Ranking Distance (AC2), and Ranking Similarity (AC3). The method demonstrates good scalability and indicates relatively minor overall performance bottlenecks under large-scale data scenarios.

The proposed method utilizes linguistic requirements to flexibly consider current preferences of specific users. The proposed method improves the existing limitations, which previously only took general user preferences and historical user preferences into account. It combines the advantages of dictionary-based methods, learning-based methods, and rule-based methods in both the preference analysis and product ranking phases, increasing the accuracy of extracting attributes, attribute weights and product attribute scores.

A case study revealed that the method performs poorly in ranking when the discriminability among products is limited. Therefore, it is suitable for products within the same category that have higher discriminability. Additionally, when there are fewer product online reviews, these reviews may not comprehensively evaluate each attribute of the product, leading to greater uncertainty. Hence, when using this method, it is advisable to select products with a larger number of reviews for comparison. Besides, due to the introduction of uncertainty during the process of topic extraction and review segmentation, there are some online reviews that cannot evaluate specific attributes considered by users, which should not be considered. Lastly, there exist misinterpretations of user preferences because users have inaccurate representations of their preferences due to a lack of understanding of the product, their true needs, or limited language expression abilities when giving linguistic requirements.

The proposed approach can be further improved by incorporating large-scale model technologies and question-answering systems. Large-scale model technologies will be used to analyze entities in user requirement texts and their entity relationships, further relaxing the restrictions on users when presenting their requirements. Question-answering systems will guide users to consider their needs thoroughly from multiple attributes, increasing their understanding of the products and guiding them to better express their preferences.

## Conflict of interest

No potential conflict of interest was reported by the authors.

## References

[1] Xu H, Wei R, Degroof R, Carberry J. Evaluating online products using text mining: A reliable evidence-based approach. *International Journal of Semantic Computing*. 2022; 16(4): 585-611. Available from: https://doi.org/10.1142/S1793351X22500064.

[2] Darley WK, Blankson C, Luethge DJ. Toward an integrated framework for online consumer behavior and decision making process: A review. *Psychology & Marketing*. 2010; 27(2): 94-116. Available from: https://doi.org/10.1002/mar.20322.

[3] Zhang KZ, Zhao SJ, Cheung CM, Lee MK. Examining the influence of online reviews on consumers' decision-making: A heuristic-systematic model. *Decision Support Systems*. 2014; 67: 78-89. Available from: https://doi.org/10.1016/j.dss.2014.08.005.

[4] Liu Y, Bi JW, Fan ZP. Ranking products through online reviews: A method based on sentiment analysis technique and intuitionistic fuzzy set theory. *Information Fusion*. 2017; 36: 149-161. Available from: https://doi.org/10.1016/j.eswa.2022.119142.

[5] Mitra S, Jenamani M. Helpfulness of online consumer reviews: A multi-perspective approach. *Information Processing & Management*. 2021; 58(3): 102538. Available from: https://doi.org/10.1016/j.ipm.2021.102538.

[6] Tata SV, Prashar S, Gupta S. An examination of the role of review valence and review source in varying consumption contexts on purchase decision. *Journal of Retailing and Consumer Services*. 2020; 52: 101734. Available from: https://doi.org/10.1016/j.jretconser.2019.01.003.

[7] Yang Z, Xiong G, Cao Z, Li Y, Huang L. A decision method for online purchases considering dynamic information preference based on sentiment orientation classification and discrete DIFWA operators. *IEEE Access*. 2019; 7: 77008-77026. Available from: https://doi.org/10.1109/ACCESS.2019.2921403.

[8] Zha ZJ, Yu J, Tang J, Wang M, Chua TS. Product aspect ranking and its applications. *IEEE Transactions on Knowledge and Data Engineering*. 2013; 26(5): 1211-1224. Available from: https://doi.org/10.1109/TKDE.2013.136.

[9] Lee HC, Rim HC, Lee DG. Learning to rank products based on online product reviews using a hierarchical deep neural network. *Electronic Commerce Research and Applications*. 2019; 36: 100874. Available from: https://doi.org/10.1016/j.elerap.2019.100874.

[10] Kumar A, Abirami S. Aspect-based opinion ranking framework for product reviews using a Spearman's rank correlation coefficient method. *Information Sciences*. 2018; 460: 23-41. Available from: https://doi.org/10.1016/j.ins.2018.05.003.

[11] Liu Z, Wang L, Li X, Pang S. A multi-attribute personalized recommendation method for manufacturing service composition with combining collaborative filtering and genetic algorithm. *Journal of Manufacturing Systems*. 2021; 58: 348-364. Available from: https://doi.org/10.1016/j.jmsy.2020.12.019.

[12] Jalili M, Ahmadian S, Izadi M, Moradi P, Salehi M. Evaluating collaborative filtering recommender algorithms: a survey. *IEEE Access*. 2018; 6: 74003-74024. Available from: https://doi.org/10.1109/ACCESS.2018.2883742.

[13] Yang Z, Ouyang T, Fu X, Peng X. A decision-making algorithm for online shopping using deep-learning–based opinion pairs mining and *q*-rung orthopair fuzzy interaction Heronian mean operators. *International Journal of Intelligent Systems*. 2020; 35(5): 783-825. Available from: https://doi.org/10.1002/int.22225.

[14] Wu C, Zhang D. Ranking products with IF-based sentiment word framework and TODIM method. *Kybernetes*. 2019; 48(5): 990-1010. Available from: https://doi.org/10.1108/K-01-2018-0029.

[15] Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao J. Deep learning--based text classification: a comprehensive review. *ACM Computing Surveys (CSUR)*. 2021; 54(3): 1-40. Available from: https://doi.org/10.1145/3439726.

[16] Zhang C, Tian YX, Fan LW, Li YH. Customized ranking for products through online reviews: a method incorporating prospect theory with an improved VIKOR. *Applied Intelligence*. 2020; 50(6): 1725-1744. Available from: https://doi.org/10.1007/s10489-019-01577-3.

[17] Yan X, Guo J, Lan Y, Cheng X. A biterm topic model for short texts. In: *Proceedings of the 22nd International Conference on World Wide Web*. ACM, Inc.; 2013. p.1445-1456. Available from: https://doi.org/10.1145/2488388.2488514.

[18] Hashemi SS, Hajiagha SHR, Zavadskas EK, Mahdiraji HA. Multicriteria group decision making with ELECTRE III method based on interval-valued intuitionistic fuzzy information. *Applied Mathematical Modelling*. 2016; 40(2): 1554-1564. Available from: https://doi.org/10.1016/j.apm.2015.08.011.

[19] Zhao M, Shen X, Liao H, Cai M. Selecting products through text reviews: An MCDM method incorporating personalized heuristic judgments in the prospect theory. *Fuzzy Optimization and Decision Making*. 2022; 21: 21-44. Available from: https://doi.org/10.1007/s10700-021-09359-8.

[20] Yin S, Wang Y, Shafiee S. Ranking products through online reviews considering the mass assignment of features based on BERT and *q*-rung orthopair fuzzy set theory. *Expert Systems with Applications*. 2023; 213: 119142. Available from: https://doi.org/10.1016/j.eswa.2022.119142.

[21] Darko AP, Liang D, Xu Z, Agbodah K, Obiora S. A novel multi-attribute decision-making for ranking mobile payment services using online consumer reviews. *Expert Systems with Applications*. 2023; 213: 119262. Available from: https://doi.org/10.1016/j.eswa.2022.119262.

[22] Liu Y, Bi JW, Fan ZP. A method for ranking products through online reviews based on sentiment classification and interval-valued intuitionistic fuzzy TOPSIS. *International Journal of Information Technology & Decision Making*. 2017; 16(6): 1497-1522. Available from: https://doi.org/10.1142/S021962201750033X.

[23] Wang W, Wang H, Song Y. Ranking product aspects through sentiment analysis of online reviews. *Journal of Experimental & Theoretical Artificial Intelligence*. 2017; 29(2): 227-246. Available from: https://doi.org/10.1080/0952813X.2015.1132270.

[24] Zhang D, Li Y, Wu C. An extended TODIM method to rank products with online reviews under intuitionistic fuzzy environment. *Journal of the Operational Research Society*. 2020; 71(2): 322-334. Available from: https://doi.org/10.1080/01605682.2018.1545519.

[25] Tayal DK, Yadav SK, Arora D. Personalized ranking of products using aspect-based sentiment analysis and plithogenic sets. *Multimedia Tools and Applications*. 2023; 82(1): 1261-1287. Available from: https://doi.org/10.1007/s11042-022-13315-y.

[26] Guo C, Du Z, Kou X. Products ranking through aspect-based sentiment analysis of online heterogeneous reviews. *Journal of Systems Science and Systems Engineering*. 2018; 27: 542-558. Available from: https://doi.org/10.1007/s11518-018-5388-2.

[27] Gobi N, Rathinavelu A. Analyzing cloud based reviews for product ranking using feature based clustering algorithm. *Cluster Computing*. 2019; 22(Suppl 3): 6977-6984. Available from: https://doi.org/10.1007/s10586-018-1996-3.

[28] Guo C, Du Z, Kou X. Mining online customer reviews for products aspect-based ranking. In: *Knowledge and Systems Sciences: 18th International Symposium, KSS 2017*. Bangkok, Thailand: Springer; 2017. p.150-161.

Available from: https://doi.org/10.1007/978-981-10-6989-5_13.
[29] Dahooie JH, Raafat R, Qorbani AR, Daim T. An intuitionistic fuzzy data-driven product ranking model using sentiment analysis and multi-criteria decision-making. *Technological Forecasting and Social Change*. 2021; 173: 121158. Available from: https://doi.org/10.1016/j.techfore.2021.121158.
[30] Fu X, Ouyang T, Yang Z, Liu S. A product ranking method combining the features–opinion pairs mining and interval-valued Pythagorean fuzzy sets. *Applied Soft Computing*. 2020; 97: 106803. Available from: https://doi.org/10.1016/j.asoc.2020.106803.
[31] Özel M, Bozkurt ÖÇ. Sentiment analysis on GPT-4 with comparative models using Twitter data. *Acta Infologica*. 2024; 8(1): 23-33. Available from: https://doi.org/10.26650/acin.1418834.
[32] Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R, Le QV. XLNet: generalized autoregressive pretraining for language understanding. In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc.; 2019. p.5753-5763.
[33] Hu G, Lin TE, Zhao Y, Lu G, Wu Y, Li Y. UniMSE: Towards unified multimodal sentiment analysis and emotion recognition. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics; 2022. p.7837-7851.
[34] Liu C, Ouyang C, Cheng S, Shah A, Bai W, Arcucci R. G2d: From global to dense radiography representation learning via vision-language pre-training. In: *38th Conference on Neural Information Processing Systems (NeurIPS 2024)*. NeurIPS; 2024. p.14751-14773.
[35] Qin J, Liu C, Cheng S, Guo Y, Arcucci R. Freeze the backbones: a parameter-efficient contrastive approach to robust medical vision-language pre-training. In: *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Seoul, Korea: IEEE; 2024. p.1686-1690. Available from: https://doi.org/10.1109/ICASSP48485.2024.10447326.
[36] Liu C, Cheng S, Shi M, Shah A, Bai W, Arcucci R. Imitate: Clinical prior guided hierarchical vision-language pre-training. *IEEE Transactions on Medical Imaging*. 2025; 44(1): 519-529. Available from: https://doi.org/10.1109/tmi.2024.3449690.
[37] Wu X, Liao H. Managing uncertain preferences of consumers in product ranking by probabilistic linguistic preference relations. *Knowledge-Based Systems*. 2023; 262: 110240. Available from: https://doi.org/10.1016/j.knosys.2022.110240.
[38] Zhang D, Wu C, Liu J. Ranking products with online reviews: A novel method based on hesitant fuzzy set and sentiment word framework. *Journal of the Operational Research Society*. 2020; 71(3): 528-542. Available from: https://doi.org/10.1080/01605682.2018.1557021.
[39] Zulfitriansyah F, Sutomo R. Recommendations for selecting houses in the bintaro area using the AHP method for property sales companies. *Journal of Applied Research In Computer Science and Information Systems*. 2023; 1(2): 66-79. Available from: https://doi.org/10.61098/jarcis.v1i2.57.
[40] Rai R. Identifying key product attributes and their importance levels from online customer reviews. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. ASME; 2012. p.533-540. Available from: https://doi.org/10.1115/DETC2012-70493.
[41] Liang X, Guo J, Sun Y, Liu X. A method of product selection based on online reviews. *Mobile Information Systems*. 2021; 2021: 1-16. Available from: https://doi.org/10.1155/2021/9656315.
[42] Zhao H, Liu Z, Yao X, Yang Q. A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach. *Information Processing & Management*. 2021; 58(5): 102656. Available from: https://doi.org/10.1016/j.ipm.2021.102656.
[43] Guo M, Liao X, Liu J, Zhang Q. Consumer preference analysis: a data-driven multiple criteria approach integrating online information. *Omega*. 2020; 96: 102074. Available from: https://doi.org/10.1016/j.omega.2019.05.010.
[44] Joung J, Kim HM. Approach for importance−performance analysis of product attributes from online reviews. *Journal of Mechanical Design*. 2021; 143(8): 081705. Available from: https://doi.org/10.1115/1.4049865.
[45] Zhang C, Xu Z, Gou X, Chen S. An online reviews-driven method for the prioritization of improvements in hotel services. *Tourism Management*. 2021; 87: 104382. Available from: https://doi.org/10.1016/j.tourman.2021.104382.
[46] Chen X, Xue Y, Zhao H, Lu X, Hu X, Ma Z. A novel feature extraction methodology for sentiment analysis of product reviews. *Neural Computing and Applications*. 2019; 31: 6625-6642. Available from: https://doi.org/10.1007/s00521-018-3477-2.
[47] Zhang C, Tian YX, Fan LW, Li YH. Customized ranking for products through online reviews: a method incorporating prospect theory with an improved VIKOR. *Applied Intelligence*. 2020; 50: 1725-1744. Available from: https://doi.org/10.1007/s10489-019-01577-3.

[48] Liang X, Liu P, Wang Z. Hotel selection utilizing online reviews: A novel decision support model based on sentiment analysis and DL-VIKOR method. *Technological and Economic Development of Economy*. 2019; 25(6): 1139-1161. Available from: https://doi.org/10.3846/tede.2019.10766.
[49] Jinghua Z, Haiying R. Multi-attribute decision-making based on data mining under a dynamic hybrid trust network. *Computers & Industrial Engineering*. 2023; 185: 109672. Available from: https://doi.org/10.1016/j.cie.2023.109672.
[50] Zhang Z, Guo J, Zhang H, Zhou L, Wang M. Product selection based on sentiment analysis of online reviews: An intuitionistic fuzzy TODIM method. *Complex & Intelligent Systems*. 2022; 8(4): 3349-3362. Available from: https://doi.org/10.1007/s40747-022-00678-w.
[51] Rafailov R, Sharma A, Mitchell E, Ermon S, Manning CD, Finn C. Direct preference optimization: your language model is secretly a reward model. In: *NIPS'23: 37th International Conference on Neural Information Processing Systems*. New Orleans: Curran Associates Inc.; 2023. p.53728-53741.
[52] Li Q, Du W, Liu J. Detoxifying large language models via Kahneman-Tversky optimization. In: *Natural Language Processing and Chinese Computing. NLPCC 2024*. Singapore: Springer; 2025. p.409-417. Available from: https://doi.org/10.1007/978-981-97-9443-0_36.
[53] Wong LNY, Liu Z. MatGBM: A computer vision-aided triangular mesh generator for high-fidelity grain-based model. *Computers and Geotechnics*. 2025; 177: 106871. Available from: https://doi.org/10.1016/j.compgeo.2024.106871.
[54] Ge C, Wang X, Zhang Z, Qin Y, Chen H, Wu H, et al. Behavior importance-aware graph neural architecture search for cross-domain recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2025; 39(11): 11708-11716. Available from: https://doi.org/10.1609/aaai.v39i11.33274.
[55] Wu X, Liao H. Modeling personalized cognition of customers in online shopping. *Omega*. 2021; 104: 102471. Available from: https://doi.org/10.1016/j.omega.2021.102471.
[56] Fader A, Soderland S, Etzioni O. Identifying relations for open information extraction. In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics; 2011. p.1535-1545.
[57] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv:1508.01991*. 2015. Available from: https://doi.org/10.48550/arXiv.1508.01991.
[58] Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using Siamese BERT-Networks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Hong Kong, China: Association for Computational Linguistics; 2019. p.3982-3992.
[59] Fang C, Li X, Fan Z, Xu J, Nag K, Korpeoglu E, et al. Llm-ensemble: Optimal large language model ensemble method for e-commerce product attribute value extraction. In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. USA: Association for Computing Machinery; 2024. p.2910-2914. Available from: https://doi.org/10.1145/3626772.3661357.
[60] Iftikhar A, Ghazanfar MA, Ayub M, Mehmood Z, Maqsood M. An improved product recommendation method for collaborative filtering. *IEEE Access*. 2020; 8: 123841-123857. Available from: https://doi.org/10.1109/ACCESS.2020.3005953.
[61] Puška A, Stević Ž, Pamučar D. Evaluation and selection of healthcare waste incinerators using extended sustainability criteria and multi-criteria analysis methods. *Environment, Development and Sustainability*. 2021; 24: 11195-11225. Available from: https://doi.org/10.1007/s10668-021-01902-2.
[62] Breiman L. Random forests. *Machine Learning*. 2001; 45: 5-32. Available from: https://doi.org/10.1023/A:1010933404324.

## Appendix A

This section describes the definition of ELECTRE III method.

ELECTRE III is a MADM method developed as an evolution of the ELECTRE series of methods [18]. Different from other methods of the ELECTRE series, ELECTRE III introduces approximate criteria, allowing for interval comparisons rather than precise scores. It also combines the harmony and discordance indices into a single credibility index, providing a nuanced view of the relative merits of each alternative, which is more suitable for decision-making with high uncertainty and ambiguity.

ELECTRE III constructs a decision matrix $D(X_i)$ using the attribute weights $w = \{w_j | j = 1, 2, ..., J\}$ and the alternative attribute scores to determine the alternative ranking. The formula for constructing the decision matrix is as follows:

$$D(X_i) = \left[w_j \cdot s_{ij}\right]_{n\times J} \tag{13}$$

where $w_j$ is the weight of the attribute $T_j$, and $s_{ij}$ is the score of the alternative $X_i$ for the attribute $T_j$.

Giving the attributes and the alternatives, the detailed steps of ELECTRE III are as follows:

(1) Assign Weights: After determining weights of all attributes, this method introduces approximate criteria, allowing for interval comparisons rather than precise scores.

(2) Establish Decision Matrix: Attribute weights and alternative attribute scores are used to establish decision matrix calculated by Equation (8).

(3) Calculate Concordance and Discordance: This method computes the concordance and discordance indices for each pair of alternatives, introducing a credibility index for interval comparisons.

(4) Rank Alternatives: The results from the concordance and discordance analysis are used to rank the alternatives, allowing decision makers to rank alternatives at different levels of ambiguity.

## Appendix B

In this section, Entity relationship VP1 rule constructed in Section 3.2.2 represents the relationship between two entities, and the detailed description is as follows:

$$VP1 = V(W * P)*$$

$$V = <VB.*><RP>?<RB.*>?$$

$$W = <NN>|<JJ.*>|<RB.*>|<DT>|<PRP>|<PRP\$>$$

$$P = <IN>|<RP>|<VBD>|<VBG>|<VBN>$$

In this rule:

(1) <*VB*.*>: Verb.

(2) <*RP*>: Function word.

(3) <*RB*.*>: Adverb.

(4) <*NN*>: Noun.

(5) <*JJ*.*>: Adjective.

(6) <*DT*>: Determiner.

(7) <*PRP*>: Pronoun.

(8) <*PRP*$>: Possessive pronoun.

(9) <*IN*>: A preposition or subordinating conjunction.

(10) <*VBD*>: Past tense verb.

(11) <*VBG*>: Present participle verb.

(12) <*VBN*>: Past participle verb.

The corresponding predicates that indicate the relationship between two entities can be a verb like “surpass”, a verb combined with a preposition like “prevail over”, or a verb followed by a noun, adjective, or adverb ending with a preposition like “hold an advantage over”.

## Appendix C

This section describes the definition of Entity relationship VP2 rule, which signifies the importance level of an entity.

The detailed description is as follows:

$$VP2 = < VBZ >< .* > *(< RB.* > | < JJ.* > | < NN > | < IN >) + < .* > *$$

In this rule:

(1) <*VBZ*>: Verb stands in the third person.

(2) <*RB*.*>: Adverb.

(3) <*JJ*.*>: Adjective.

(4) <*NN*>: Noun.

(5) <*IN*>: A preposition or subordinating conjunction.

(6) <.*>: Any word.

This rule describes the importance level of an entity, often represented by a verb, a linking verb combined with an adjective, or a verb followed by a noun phrase.

## Appendix D

This section describes the Entity NP rule, which represents an entity within a sentence.
This rule is as follows:

$$NP =< DT > ? < JJ.* > *(\wedge((?!(< NNP > KCD >)).)*\$KNN >) + .$$

In this rule:
(1) $<DT>$: Determiner.
(2) $<JJ.*>$: Adjective.
(3) $<NN>$: Noun.
(4) $<NNP>$: Proper noun, including names of people, places, organizations, etc.
(5) $<CD>$: Cardinal number, representing numerical expressions.
This rule denotes noun phrases that do not include the proper nouns or the cardinal numbers.