**A TERM PAPER REPORT**

**ON**

**PRUS: Product Recommender System Based on**

**User Specifications and Customer Reviews**

**Submitted in the partial fulfillment of requirements to**

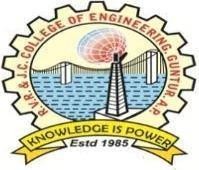
**CS- 363 - Term Paper**

**By**

**Batch No. 19**

**ORUGANTI MONIK PAPARAO (Y22CS139)**

**PENDYALA SKANDA BHAGAVAN (Y22CS145)**

**TULAM SAI SUDHEER (Y22CS184)**

**R.V.R. & J.C. COLLEGE OF ENGINEERING (Autonomous)**

**(NAAC ‘A+’ Grade)**

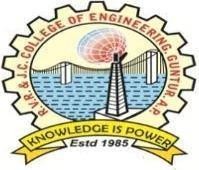
**Approved by AICTE :: Affiliated to Acharya Nagarjuna University**

**Chandramoulipuram::Chowdavaram**

**GUNTUR – 522 019,Andhra Pradesh, India.**

**R.V.R & J.C. COLLEGE OF ENGINEERING (Autonomous)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

****

This is to certify that this Term Paper titled “PRUS: Product Recommender System Based on User Specification and Customer Reviews" is the study conducted by **ORUGANTI MONIK PAPARAO (Y22CS139) , PENDYALA SKANDA BHAGAVAN (Y22CS145), TULAM SAI SUDHEER (Y22CS184)** submitted in partial fulfillment of the requirements to CS 363- Term Paper during the Academic Year 2024-2025.

**Smt. N. Zareena Dr. S J R K Padminivalli V Dr. M. Sreelatha**

Term Paper Guide Term Paper In- Charge Prof. &Head,CSE

# ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without proper suggestions, guidance and environment. Combination of these three factors acts like backbone to our Term Paper titled**“PRUS: Product Recommender System Based on User Specification and Customer Reviews”.**

We are deeply grateful to our guide, **Smt. N.Zareena** , for her unwavering guidance, insightful feedback and encouragement. Her expertise and dedication have been instrumental in shaping the direction and quality of this paper.

We would like to express our sincere gratitude to **Dr. S J R K Padminivalli V,** In-Charge for our term paper. Her expertise, guidance and support were instrumental in the successful completion of this paper.

We express our sincere thanks to **Dr.M.Sreelatha**, Head of the Department of Computer Science and Engineering for her encouragement, support, commitment to enhance paper experience.

We are very much thankful to **Dr. Kolla Srinivas,** Principal, for proving this supportive environment and to engage in paper activities.

Finally we submit our heartfelt thanks to lab staff in the Department of Computer Science and Engineering for their cooperation, support for providing administrative support and technical assistance during selection.

**ORUGANTI MONIK PAPARAO (Y22CS139)**

**PENDYALA SKANDA BHAGAVAN (Y22CS145)**

**TULAM SAI SUDHEER (Y22CS184)**

**ABSTRACT**

The growing popularity of online shopping has led to a continuous influx of product reviews, which play a crucial role in shaping consumer purchasing decisions. Many paper studies have focused on ranking products using these reviews. However, most methodologies tend to overlook the impact of negative sentiments when evaluating products based on customer preferences. This paper aims to address this gap by incorporating both positive and negative polarity in product ranking. To achieve this, the proposed method first breaks down reviews into individual sentences, allowing for a more granular analysis of sentiment at the phrase level. By extracting key elements from these reviews, the approach effectively determines the polarity of different product features. The next step involves linking these polarities to sentence-level attributes within the review. To ensure that product rankings align with user preferences, relative importance is assigned to each polarity based on specific needs.The effectiveness of this approach has been tested using the Amazon review dataset, where experimental evaluations were conducted using rank score (RS) and normalized discounted cumulative gain (nDCG) score. Results demonstrate that the proposed ranking system, PRUS, enables users to customize their recommended product lists by prioritizing either positive or negative aspects based on their preferences.

**Contents Page. No.**

Title Page i

Certificate ii

Acknowledgement iii

Abstract iv

Contents v

List of Tables vi

List of Figures vii

List of Abbreviations viii

1. Introduction 1

1.1 Background 1

1.2 Problem Statement 2

1.3 Objectives 2

1.4 Limitations of Existing Systems 3

2. Literature Survey 6

2.1 Opinion Based Entity Ranking (Spearman’s) 6

2.2 Aspect Based Opinion Ranking 7

2.3 Opinion Based Entity using Learning to Rank 9

2.4 A two fold role based model for Aspect Extraction 10

2.5 Aspect Term Extraction and Opinion Target Extraction 12

2.6 Extracting Prominent Review Aspects from Feedbacks 14

2.7 Association Rule Mining by Fuzzy Logic and Whale Optimization 16

2.8 Opinion Fraud Detection in Online Reviews by Network Effects 19

3. Methodologies Used 22

3.1 Architecture 22

3.2 Datasets Used 24

4. Proposed Model 26

4.1 Rank-ify Algorithm 28

4.2 Benefits of Rank-ify 30

5. Discussions on the Proposed Output 31

5.1 Experiments and Results 31

5.2 Performance Analysis of the PRUS Method 34

6. Conclusion and Future Works 37

6.1 Conclusion 37

6.2 Future Works 38

7. References 39

## LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Name** | **Page No.** |
| **3.2.1** | **Amazon Dataset** | **24** |
| **5.2.1.1** | **Performance of PRUS** | **34** |
|  |  |  |
|  |  |  |

## LISTOF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Name** | **Page No.** |
| **1.1.1** | **Rise of Online Shopping** | **1** |
| **2.1.1** | **Ideal Recommender System** | **6** |
| **2.3.1** | **Opinion Based Entity Ranking** | **9** |
| **2.5.1** | **ATE and OTE Recommender System** | **14** |
| **3.1.1** | **PRUS Architecture** | **23** |
| **4.1.4.1** | **Rank-ify Algorithm** | **30** |
| **5.2.4.1** | **Experimentation Graph** | **36** |

**LIST OF ABBREVIATIONS**

PRUS - Product Recommender System Based on User Specification

AAS - Average Aspect Score

RS - Rank Score

QAM - Query Aspect Modeling

NLP - Natural Language Processing

SVM - Super Vector Machine

GP - Genetic Programming

NLTK - Natural Language Toolkit

DCG - Discounted Cumulative Gain

nDCG - Normalized Discounted Cumulative Gain

IDCG - Ideal Discounted Cumulative Gain

**1. INTRODUCTION**

* 1. **Background**

The rapid expansion of the e-commerce industry has significantly transformed the way people shop. With the increasing convenience of online platforms, consumers are shifting from traditional in-store purchases to digital shopping experiences. The ability to browse a wide range of products, compare prices, read detailed descriptions, and access customer reviews from the comfort of their homes has made online shopping the preferred choice for many. The dominance of online sales has created an environment where customer feedback plays a crucial role in shaping purchasing decisions.

One of the key outcomes of the e-commerce boom is the proliferation of customer reviews on various platforms. Consumers frequently share their experiences by posting feedback on e-commerce websites (e.g., Amazon, eBay, Flipkart), social media platforms, and specialized review websites. These reviews typically highlight both the positive and negative aspects of products, offering valuable insights to potential buyers. For instance, a review of a smartphone might praise its battery life and camera quality but criticize its screen resolution. Similarly, a review of a vacuum cleaner might highlight its suction power as a positive but point out issues with its noise level or durability. These detailed reviews help potential buyers make informed purchasing decisions by offering real-world insights that go beyond product descriptions and marketing claims.



Fig-1.1.1

Customer reviews not only benefit potential buyers but also offer valuable data for companies. Businesses use this feedback to identify product strengths and weaknesses, improve existing products or develop new ones based on customer preferences, and enhance customer service by addressing common complaints or issues.

While customer reviews provide valuable insights, the sheer volume of online reviews makes it difficult to extract meaningful information efficiently. As e-commerce platforms grow, the number of reviews for popular products can reach into the thousands, making manual analysis impractical.

* 1. **Problem Statement**

While customer reviews provide valuable insights, they often contain **mixed sentiments** about different product features, which most existing product ranking methods fail to capture. These methods typically focus on the **overall sentiment** of reviews rather than evaluating individual aspects. For example, a review might praise a smartphone’s **camera and battery life** but criticize its **screen resolution**. Despite the negative feedback on a specific feature, the review may still be classified as positive overall, which can mislead potential buyers.

Many current ranking models **overlook feature-specific sentiments**, relying heavily on general polarity. This makes it difficult for users to identify products that align with their **specific preferences**. For instance, a phone with an excellent camera but poor battery life might be ranked higher due to its overall positive sentiment, even though a buyer prioritizing battery performance would find it unsuitable.

Furthermore, reviews often contain **contradictory opinions**, such as praising a product's **build quality** but criticizing its **customer service**. By only considering the overall sentiment, such contrasting feedback is flattened into a **single polarity**, making the review less informative.

To address this, **aspect-based sentiment analysis (ABSA)** offers a more precise approach by identifying and extracting **sentiments related to specific product features**. This allows for more **detailed and transparent product evaluations**, helping buyers filter products based on the aspects that matter most to them. By integrating ABSA, e-commerce platforms can offer **more accurate and personalized product recommendations**, enhancing the shopping experience.

* 1. **Objectives**

The primary objectives of this paper are:

* **Development of the PRUS framework**: Propose a Product Recommendation based on User Specification (PRUS) framework to generate a customized product list by considering user-specified features and preferences. This framework aims to offer more tailored recommendations by addressing individual needs rather than relying on generalized rankings.
* **Multi-polarity sentiment extraction**: Implement a technique to identify and separate positive and negative sentiments within a single review, ensuring that conflicting opinions are not flattened into a single polarity. This allows the system to capture diverse opinions accurately.
* **Weighted sentiment scoring**: Introduce a technique to assign weights to both positive and negative sentiments of product features. This enables the ranking system to prioritize products based on the user's specific preferences, such as giving more importance to certain features (e.g., battery life) over others.
* **User-defined prioritization**: Allow users to define the importance of different product aspects when generating recommendations. This ensures the final ranked list reflects the user's unique needs and preferences, making the recommendations more relevant and valuable.
* **Customized product ranking**: Implement a dynamic ranking mechanism that reorders products based on the weighted sentiment scores of their aspects. This ensures that products with stronger performance in the features valued by the user are ranked higher.
* **Experimental validation**: Perform extensive evaluations using the Amazon review dataset to validate the effectiveness and accuracy of the PRUS framework. Use performance metrics such as rank score (RS) and normalized discounted cumulative gain (nDCG) to measure the precision and reliability of the system.
* **Improved decision-making**: Enhance the decision-making process for both consumers and businesses by providing more accurate, aspect-based product insights. This will help buyers make better-informed decisions and allow companies to identify product strengths and areas for improvement.
  1. **Scalability and adaptability**: Ensure that the PRUS framework is scalable and adaptable to different product categories and review datasets.
  2. **Limitations of Existing Systems**

Despite the availability of various product ranking systems, several limitations persist in existing approaches:

* **Focus on overall sentiment**: Most existing methods primarily consider the overall sentiment of reviews, overlooking the detailed opinions on individual product features. By only capturing the general polarity of a review, these systems fail to account for feature-specific strengths and weaknesses. This can lead to inaccurate and generalized recommendations, preventing users from identifying products that align with their specific needs.
* **Lack of user-specific customization**: Many systems do not allow users to specify the importance of particular features when generating product recommendations. As a result, all product aspects are treated with equal significance, regardless of the user's priorities. For instance, a customer looking for a laptop with long battery life may receive recommendations that prioritize display quality or performance, which may not match their needs. The absence of user-driven customization reduces the relevance and effectiveness of the recommendations.
* **Inefficient sentiment extraction**: Existing models often struggle to accurately extract and link sentiments to specific product aspects. Sentiments expressed in reviews can be ambiguous or multi-faceted, making it challenging to associate the correct polarity with the appropriate feature. For example, a review mentioning, "*The phone's display is stunning, but the battery drains quickly*" contains contrasting sentiments about different aspects. However, many ranking systems fail to distinguish between them, resulting in less precise and unreliable recommendations.
* **Limited consideration of negative polarity**: Most product ranking systems prioritize positive sentiment while neglecting the significance of negative opinions. This creates a bias toward favourable reviews, leading to misleading recommendations. Negative feedback often highlights critical product flaws, such as poor durability or customer service issues, which are valuable for potential buyers. By underweighting or ignoring negative polarity, existing systems provide incomplete and overly optimistic product evaluations, making it harder for users to make informed decisions.
* **Inability to handle mixed or conflicting sentiments**: Many reviews contain conflicting opinions about different product features, but existing methods often fail to distinguish between them.
* This simplified interpretation reduces the accuracy of the recommendations, as the nuanced user experience is lost.
* **Failure to capture context-specific sentiments**: Sentiment analysis models often miss contextual nuances, such as sarcasm, irony, or subtle expressions of dissatisfaction. For instance, the sentence "*The product works... when it feels like it*" expresses negative sentiment with sarcasm, but many models misinterpret it as neutral or positive. This leads to inaccurate sentiment classification.
* **Limited adaptability across product categories**: Many existing systems are designed for specific product types, making them less adaptable to other categories. A ranking model tailored for electronics may not effectively capture the features and sentiments relevant to clothing or home appliances, limiting its applicability across diverse e-commerce platforms.
* **Static product ranking models**: Most systems generate static product rankings that do not dynamically update based on evolving user preferences or review trends. As customer opinions and product performance shift over time, static models become outdated, providing irrelevant or obsolete recommendations.
* **Lack of multi-polarity analysis**: Existing models often fail to perform multi-polarity sentiment analysis, where both positive and negative opinions within a single review are evaluated independently. For example, if a review praises a product's design but criticizes its functionality, both polarities should be considered separately. However, many systems merge conflicting sentiments, reducing the granularity and accuracy of the analysis.
* **Insufficient consideration of review credibility**: Current systems do not adequately account for fake or biased reviews, which can distort product rankings. Spam reviews or paid promotions often create misleading sentiments, making the recommendations unreliable.
* **Ineffective performance evaluation metrics**: Many existing models rely on basic evaluation metrics, which do not fully capture the effectiveness and accuracy of the ranking system. As a result, the performance of the models in real-world scenarios may not be properly measured, leading to suboptimal recommendations.
* **Reduced user satisfaction**: The inability to personalize rankings based on user preferences, coupled with the incomplete sentiment analysis, often results in irrelevant product recommendations. This reduces user trust in the system and diminishes the overall shopping experience.

1. **LITERATURE REVIEW**

**2.1. Opinion-based entity ranking by Kavita Ganesan and Cheng Xiang Zhai et. al.**

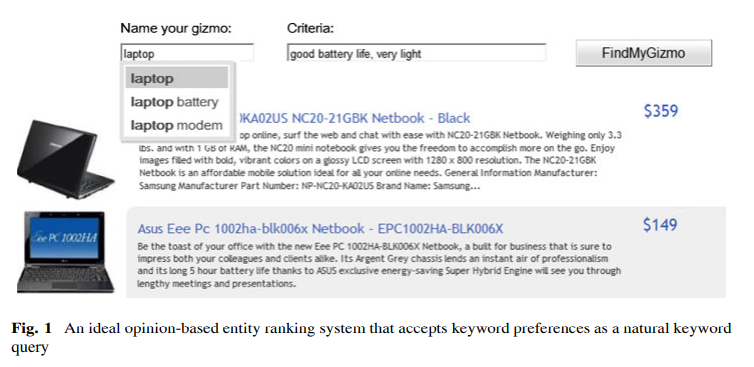
This paper introduces a novel method to rank entities, such as hotels and cars, based on user preferences expressed through keyword queries. Unlike existing methods that focus on summarizing opinions, the proposed approach uses the text from all reviews of an entity to represent it. Given a user’s query, which specifies desired features, the method ranks entities by how well the reviews align with the user’s preferences. The paper explores several text retrieval models, including both standard and extended versions, to address this task. Experiments are conducted in two domains—hotels and cars—demonstrating that the extended models improve ranking accuracy. The method focuses on directly matching user preferences with the content of reviews, offering a more personalized ranking. This allows users to discover entities that best match their individual needs, rather than relying on general ratings or aggregated opinions. The results show that the extended models outperform standard retrieval models for ranking based on opinionated content. The study highlights the potential of opinion matching for better entity discovery. Ultimately, the paper suggests that leveraging opinionated content in this way can be the most significantly improve ranking precision and user satisfaction.

Fig. 2.1.1

**2.2. Aspect-based opinion ranking framework for product reviews by a Spearman’s rank correlation coefficient method. Kumar and S. Abirami**

This paper introduces a new framework for ranking products based on aspects, using opinion mining, or sentiment analysis, to analyze social media content. Opinion mining detects emotions and opinions from structured, semi-structured, and unstructured content at various levels, including document, word, sentence, and aspect levels. The aspect level refers to specific attributes or parts of an entity, highlighting users' likes and dislikes. The proposed system first identifies the aspects of products from reviews. Next, it visualizes the aspects and their associated opinion words using a Harel–Koren fast multi scale layout. A network visualization of these aspects is created, and a Spearman’s rank correlation coefficient is used for opinion ranking, sorting products based on positive and negative feedback. The system also employs supervised learning methods like Naïve Bayes, Maximum Entropy, and Support Vector Machine for aspect-based sentiment classification. These models help classify the sentiments towards the identified aspects of products. The effectiveness of the proposed framework is evaluated through experimental results. Ultimately, the paper demonstrates the potential of aspect-based opinion mining for improving product ranking and decision-making.

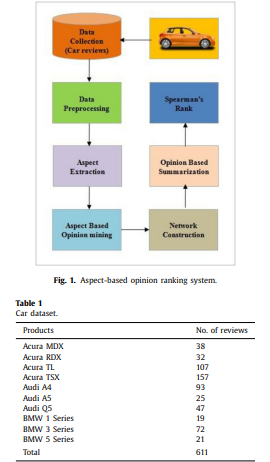


Fig. 2.2.1

**2.3. Opinion-Based Entity Ranking using learning to rank by S. Bashir, W. Afzal, and A. R. Baig et. al.**

As social media and e-commerce continue to grow, opinions have become a vital source of information for decision-making. However, the overwhelming volume of opinions makes it difficult for individuals to evaluate them effectively. Opinion-Based Entity Ranking (OpER) addresses this challenge by directly ranking entities based on how well opinions about them align with a user's preferences, provided as query keywords. This reduces the need to read countless reviews. Previous OpER models primarily compute entity relevance by matching query keywords across all opinions collectively, without considering the subjectivity or importance of individual opinions. This overlooks the fact that entities with positive judgments and strong keyword relevance should be ranked higher than those with negative or weak relevance. To enhance OpER effectiveness, the paper proposes a new framework that matches individual opinions to query keywords, creating a more intuitive ranking model. The authors use genetic programming (GP) to combine various ranking features into a more effective retrieval model. Experimental results demonstrate that the proposed model significantly outperforms the standard OpER approach.

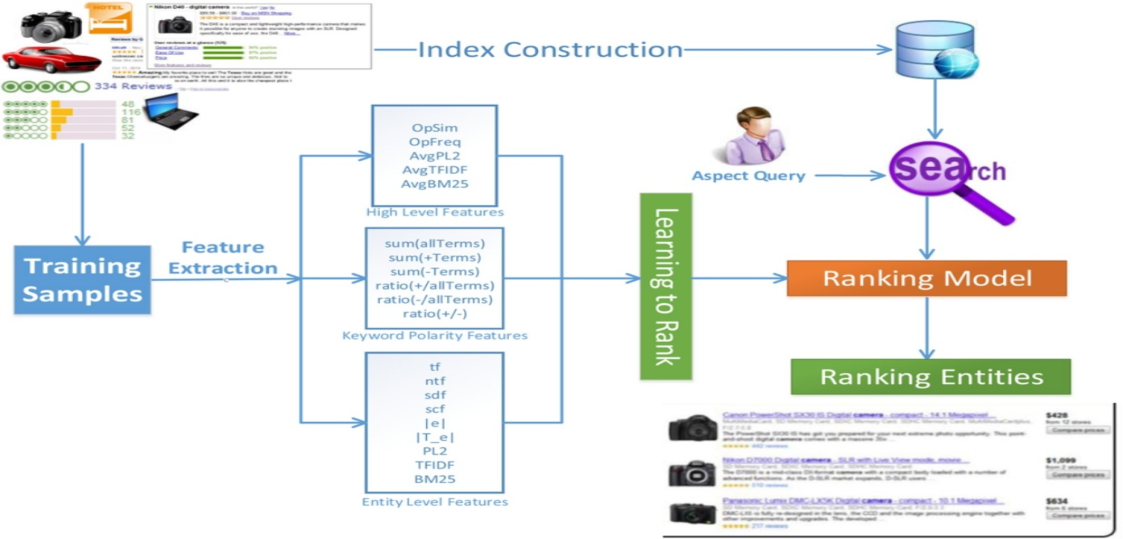


Fig. 2.3.1

**2.4. A Two-Fold Rule-Based Model for Aspect Extraction** **by T. A. Rana and Y.-N. Cheah et. al.**

Opinion target extraction plays a vital role in aspect-based sentiment analysis by identifying the specific targets of users' opinions in online reviews. While syntactic patterns-based approaches have shown promising results in recent years, they come with limitations. These methods rely on dependency parsers that generate syntactic relations according to grammatical rules and language structures. However, in real-world scenarios, online reviewers often use informal language, slang, and fragmented sentences, making it difficult for parser-dependent methods to accurately extract aspects. To overcome this challenge, the paper introduces a two-fold rules-based model (TF-RBM) designed to enhance aspect extraction accuracy. The first fold identifies aspects linked to domain-independent opinions, capturing general sentiments across different contexts. The second fold targets domain-dependent opinions, which are specific to particular products or services. Furthermore, the authors incorporate frequency- and similarity-based techniques to refine the extraction process, ensuring more precise and reliable results. Experimental evaluations demonstrate that the proposed model outperforms existing state-of-the-art approaches, making it a more robust and effective solution for aspect extraction in sentiment analysis.

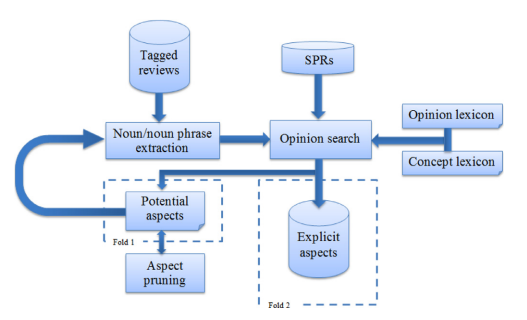


Fig. 2.4.1

**2.5. Aspect Term Extraction (ATE) and Opinion Target Extraction (OTE)**

**Model Proposed by C. Wu, F. Wu, S. Wu, Z. Yuan, and Y. Huang**

In the field of fine-grained sentiment analysis, **Aspect Term Extraction (ATE)** and **Opinion Target Extraction (OTE)** play a crucial role in identifying and analyzing specific components of opinions. ATE focuses on extracting the particular **features or attributes** of an entity being discussed. For example, in a product review, the sentence “The battery life is impressive” contains the aspect term **“battery life”,** as it highlights the specific feature being evaluated. By identifying these aspects, businesses can understand which features customers find satisfactory or problematic, allowing them to make targeted improvements.

On the other hand**, Opinion Target Extraction (OTE)** identifies the **targets of opinions,** which are the objects or entities receiving the sentiment. For instance, in the sentence “The camera is fantastic”, the term **“camera”** is the opinion target, as it is the object being praised. OTE is essential for linking opinions to their corresponding targets, enabling more precise sentiment classification. Without OTE, it would be difficult to determine whether a sentiment refers to the entire product or a specific component, leading to inaccurate or vague insights.

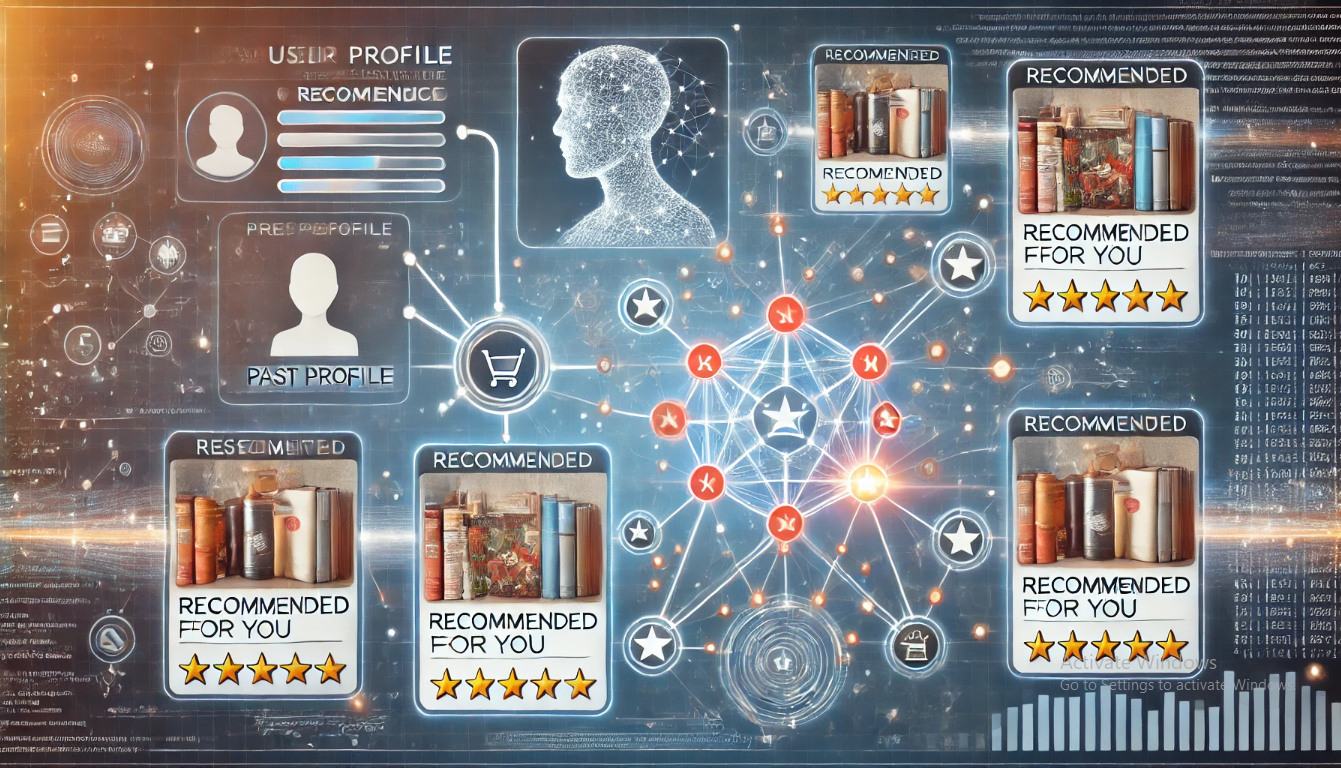


Fig. 2.5.1

**2.6. Extracting prominent review aspects from customer feedback by Z. Luo, S. Huang, F. F. Xu, B. Y. Lin.**

Many existing systems for analyzing and summarizing customer reviews focus on identifying prominent review aspects. Traditionally, these aspects are manually determined, which is both costly and inefficient. This approach cannot scale to large platforms like Amazon, Taobao, or Yelp, where countless product types and new products are frequently introduced. To overcome this challenge, the paper proposes a novel framework called **ExtRA** for automatically extracting the most prominent aspects from textual reviews. ExtRA identifies the **K** most significant aspect terms or phrases without supervision, ensuring there is no semantic overlap between the extracted aspects. The framework is designed to be scalable, making it suitable for large and diverse product types. It uses unsupervised methods, eliminating the need for manual effort. Extensive experiments show that ExtRA achieves state-of-the-art performance. It effectively extracts meaningful and distinct aspects from large review datasets. This makes ExtRA a robust solution for large-scale sentiment analysis tasks.

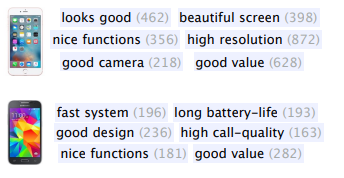


Fig. 2.6.1

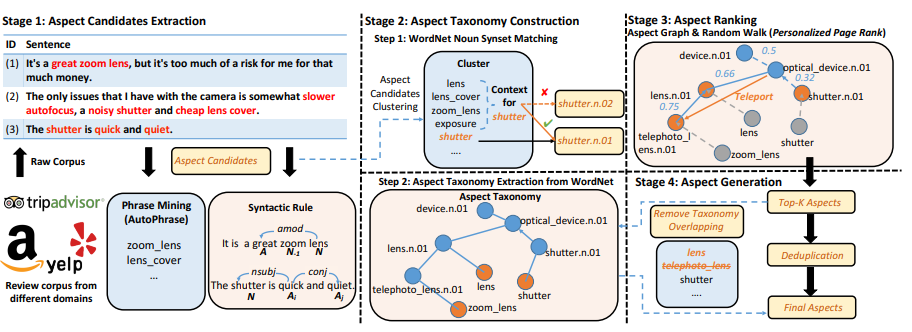


Fig. 2.6.2

**2.7. Association rule mining using fuzzy logic and whale optimization**

**Model Proposed by S. Sharmila and S. Vijayarani**

Association rule mining (ARM) is a widely recognized and essential data mining technique, commonly used for uncovering patterns or relationships within large transactional datasets. The primary goal of ARM is to discover frequent item sets—sets of items that appear together frequently in transactions—and generate meaningful association rules based on these item sets. These association rules are pivotal in various domains, such as retail, healthcare, and market basket analysis, where businesses use them to derive actionable insights like cross-selling opportunities or customer purchasing patterns. In the process of association rule mining, two critical steps are involved: frequent item recognition and association rule generation. Both these steps rely on measures such as minimum support and confidence to ensure that the discovered patterns and rules are statistically significant and useful.

To generate these association rules, ARM typically involves multiple iterations of the dataset to identify frequent item sets that meet the predefined support threshold. The rules themselves are then formed by evaluating the strength of relationships between different item sets based on their co-occurrence. However, as datasets grow in size and complexity, the computational cost associated with identifying frequent item sets and generating association rules becomes prohibitively high. Specifically, the number of database scans required increases significantly as the size of the dataset expands. This challenge is further compounded by the fact that not all transactions and items in the dataset contribute equally to the discovery of meaningful patterns. Thus, the need for techniques that reduce the dataset's dimensionality and computational complexity becomes increasingly important.

To address these issues, the first step in the proposed paper focuses on dimensionality reduction techniques to significantly reduce the size of the dataset. This step is crucial because it helps in eliminating irrelevant items and transactions that do not contribute meaningfully to the mining process, thus improving the overall efficiency of the algorithm. The dimensionality reduction technique implemented in this paper utilizes two primary methods: low variance and hash table techniques. Low variance methods focus on identifying features or items with low variance that are less likely to influence the association rules, allowing for their removal from the dataset. Meanwhile, the hash table method helps in mapping items and transactions in a more compact manner, further reducing the computational burden. By applying these techniques, the algorithm effectively narrows down the data to include only the most significant transactions and items, ensuring that subsequent analysis is focused on the most relevant parts of the dataset.

A key challenge that arises in dimensionality reduction is dealing with datasets that have a high number of dimensions or features, particularly when the number of items in the transactional database exceeds what the algorithm can efficiently handle. This issue becomes even more pronounced when dealing with large-scale, high-dimensional datasets where irrelevant or less significant items are abundant. The proposed algorithm tackles this challenge by systematically reducing both the dimensionality of the items and the transactions in the dataset, removing unnecessary or less relevant data from the analysis. The algorithm is designed to retain only those items and transactions that are likely to contribute to the discovery of frequent item sets, thereby reducing the size of the dataset while maintaining the integrity of the analysis. In the proposed method, the dimensionality reduction process is evaluated and compared against other techniques, such as extended frequent pattern (EFP) and intersection set theory, as well as dimensionality reduction using frequency count. These comparisons focus on various performance metrics, including item reduction, transaction reduction, execution time, and memory usage, all of which are critical factors in assessing the efficiency of the algorithm.

In the second step of the proposed paper, a fuzzy logic-based approach is integrated with a whale optimization algorithm for frequent item set identification and association rule generation. Fuzzy logic is particularly useful in this context because it allows for the representation of uncertain or imprecise relationships between items, which is often the case in real-world transactional data. This addition makes the algorithm more flexible and capable of handling fuzzy, ambiguous relationships that might otherwise be overlooked using traditional crisp-set methods. Whale optimization is then used to further enhance the process of frequent item identification and rule generation by optimizing the search for relevant item sets and improving the overall quality of the generated rules. Whale optimization is inspired by the social behavior of humpback whales and is known for its strong performance in solving optimization problems. By combining fuzzy logic with whale optimization, the proposed algorithm is able to efficiently identify significant item sets and generate highly accurate association rules, even in complex and noisy datasets.

The performance of the proposed algorithm is rigorously evaluated and compared with several well-established optimization techniques, such as particle swarm optimization (PSO), genetic algorithms (GA), and fuzzy frequent item set-Miner (FFIM). These comparisons focus on several key performance metrics, including the number of frequent items identified, the number of association rules generated, execution time, and memory requirements. Experimental results from these evaluations show that the proposed algorithm outperforms the competing methods in all aspects. Specifically, the algorithm identifies a higher number of frequent item sets, generates more accurate association rules, and does so in a significantly shorter amount of time while consuming less memory. This demonstrates the effectiveness of the proposed fuzzy and whale optimization-based approach, especially in the context of large-scale datasets where computational efficiency and scalability are of paramount importance.

In conclusion, the proposed algorithm represents a significant advancement in the field of association rule mining. By incorporating dimensionality reduction techniques and leveraging fuzzy logic and whale optimization for efficient frequent item set identification and association rule generation, the algorithm is able to handle large, high-dimensional datasets effectively. The results of extensive experiments highlight the algorithm's superior performance in terms of both accuracy and efficiency, making it a promising tool for real-world applications that require the analysis of complex transactional data. With its ability to reduce dimensionality, improve computational efficiency, and generate high-quality association rules, the proposed approach has the potential to revolutionize how businesses and paperers analyze transactional datasets, leading to more actionable insights and better decision-making.

**2.8. Opinion fraud detection in online reviews by network effects**

**Model Proposed by L. Akoglu**

User-generated online reviews have become an essential component of the decision-making process for consumers when evaluating products, services, and establishments like hotels or restaurants. Positive reviews can significantly boost a product's reputation and sales, while negative reviews can cause harm to a business’s credibility. However, the authenticity of online reviews is often compromised by opinion spammers—individuals or automated bots who create fraudulent reviews to either artificially inflate the reputation of a product or service or damage the reputation of a competitor. These fraudulent reviews distort the perceived quality of products, leading to a misleading representation of their true value. To counteract the impact of these fraudulent activities, it is crucial to develop reliable methods for identifying and filtering out fake reviews. In this regard, we propose a fast and effective framework, FRAUDEAGLE, designed to spot fraudsters and fake reviews in online review datasets.

The main advantage of FRAUDEAGLE lies in its innovative approach to detecting fraudulent reviews. Unlike most existing methods that rely solely on analyzing review text or user behavior patterns, FRAUDEAGLE leverages the network effect among reviewers and products. This network effect can reveal the underlying connections between users and products, allowing the framework to identify suspicious patterns that may not be evident when only considering individual reviews or users in isolation. For example, a fraudulent reviewer may repeatedly post fake reviews for a specific product or may be linked to other suspicious users or products, forming a pattern that can be detected using network-based methods. By focusing on these connections, FRAUDEAGLE is able to uncover fraudulent activity that traditional text analysis or behavior-based methods might miss.

The framework is structured in two complementary steps. The first step involves scoring users and reviews for fraud detection. Here, FRAUDEAGLE assigns a fraud score to both the users posting the reviews and the reviews themselves, based on the identified patterns in the network of interactions. This allows the system to detect both fake reviews and users who engage in fraudulent activities. The second step focuses on grouping—this is the process of clustering suspicious reviews and reviewers together for visualization and sense making. By grouping potentially fraudulent reviews and users, the framework allows analysts to easily spot clusters of suspicious activity, providing valuable insights into patterns of fraudulent behavior. This approach not only enhances the detection process but also aids in understanding the context and scale of the fraudulent activities.

One of the standout features of FRAUDEAGLE is that it operates in an unsupervised fashion, meaning it does not require any labeled data to train the model. This is particularly advantageous because acquiring labeled data for fraud detection is both labor-intensive and expensive. Most existing fraud detection systems rely on manually labeled examples of fraudulent and genuine reviews, which can be time-consuming and may not scale well. In contrast, FRAUDEAGLE requires no prior knowledge about which reviews or users are fraudulent, making it highly adaptable to different datasets and easily deployable across various platforms. Furthermore, FRAUDEAGLE can incorporate side information if available, such as the reputation or activity level of users, which can help improve the accuracy of fraud detection without requiring a complete labeled dataset.

Another significant benefit of FRAUDEAGLE is its scalability. The framework is designed to handle large datasets efficiently, with its runtime growing linearly with the size of the network. This makes it suitable for large-scale applications, such as analyzing reviews on major online platforms where the volume of reviews can reach millions. As the amount of data continues to increase, the ability to scale becomes crucial, and FRAUDEAGLE’s linear time complexity ensures that it remains efficient even as the dataset grows. This scalability also means that the framework can be applied to real-time data, allowing businesses and platforms to detect and respond to fraudulent activities as they occur.

To validate the effectiveness of FRAUDEAGLE, we conducted experiments on both synthetic and real datasets, including a large-scale online app review database. The results demonstrated that FRAUDEAGLE was able to successfully identify fraud-bots—automated systems designed to post fake reviews—in a real-world scenario. By analyzing the network of interactions among users and products, FRAUDEAGLE was able to uncover suspicious patterns that were indicative of fraudulent activities. These findings highlight the robustness and real-world applicability of the framework in detecting fraud on large platforms where traditional fraud detection methods might struggle due to the sheer volume of data.

In conclusion, FRAUDEAGLE represents a significant advancement in the field of fraud detection for online reviews. Its innovative use of network-based methods, unsupervised operation, and scalability makes it an ideal solution for tackling the growing problem of fraudulent reviews in large datasets. By focusing on the relationships between users and products, FRAUDEAGLE is able to uncover hidden patterns of fraudulent activity that other methods may overlook. Its ability to operate without labeled data and scale efficiently with increasing dataset size further enhances its practicality for deployment on large-scale review platforms. As online reviews continue to play an integral role in consumer decision-making, frameworks like FRAUDEAGLE are essential for maintaining the integrity and trustworthiness of these review systems.

1. **METHODOLOGIES USED**

**3.1 Architecture**

The PRUS framework (Product Recommendation based on User Specification) is an advanced system designed to deliver personalized product recommendations by leveraging aspect-level sentiment analysis. Unlike traditional product ranking methods that rely on overall review polarity, PRUS focuses on individual product features, ensuring that the recommendations align with user-specified preferences. The framework consists of several key stages, each contributing to the accurate extraction, evaluation, and ranking of products based on their aspect-level sentiments.

The process begins with the data retrieval phase, where the system collects customer reviews from a review dataset. These reviews form the input data for the recommendation system. Instead of analyzing the reviews as a whole, PRUS breaks them down into individual sentences, allowing the system to conduct granular sentiment analysis. This segmentation is crucial because customer reviews often contain mixed opinions. For instance, a single review may praise a smartphone’s camera but criticize its battery life. By analyzing sentences individually, the framework ensures that both positive and negative sentiments associated with different features are captured separately, preventing misleading overall sentiment interpretations.

In the feature extraction phase, each sentence is processed to identify specific product features and the corresponding sentiment polarity (positive, negative, or neutral). To enhance the accuracy and relevance of the recommendations, the framework employs information gain (IG) as an evaluation method. Information gain, based on entropy calculation, measures how much predictive value a particular feature contributes to the overall classification. Features with higher information gain are deemed more informative and influential, increasing their importance in the ranking process. For example, in a laptop review, features such as battery performance or processing speed may carry more weight if they are frequently mentioned and have a significant impact on the overall user experience. This weighted evaluation ensures that the most relevant and influential product features are prioritized.

Once the features and their associated sentiments are extracted, the system appends the sentiment polarity to the respective feature. The next step is to match these extracted features with the user-specified preferences. This enables the system to prioritize products that align with the user's desired attributes.

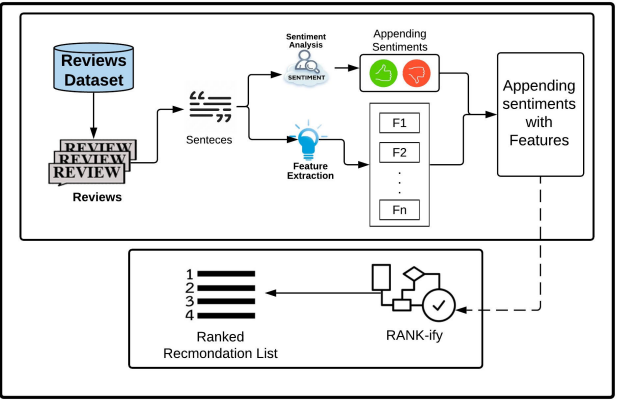


Fig. 3.1.1

The ranking phase is executed using the **RANK-ify algorithm**, which plays a crucial role in assigning weights based on sentiment polarity. The algorithm increases the weight of features with positive sentiments, making them more influential in the final ranking. Conversely, features with negative sentiments are assigned lower weights, decreasing their influence. This weighted sentiment scoring ensures that products with more favourable feedback on the desired features appear higher in the recommendation list. By considering both positive and negative sentiments, the system offers a balanced and realistic evaluation of products, preventing bias towards only positive opinions.

Finally, the ranked recommendation list presents the products along with their features and corresponding sentiment scores, weighted according to the user’s preferences. This output provides a personalized, sentiment-driven ranking, making it easier for users to identify products that best match their specific needs. For example, a user seeking a smart phone with a good camera and long battery life would receive a list where products excelling in these areas are ranked higher, regardless of their overall review sentiment.

The PRUS framework offers several key advantages:

* **Granular sentiment analysis:** By breaking reviews into sentences, it captures feature-specific opinions instead of general sentiments.
* **Enhanced precision with information gain:** The use of information gain ensures that the most relevant and informative features carry greater influence.
* **Personalized recommendations:** The framework prioritizes products based on user-defined preferences, ensuring tailored recommendations.
* **Balanced sentiment weighting:** By applying the **RANK-ify algorithm**, the system balances positive and negative sentiments, offering fair and accurate product rankings.

**3.2 Datasets Used**

The dataset used in this study consists of 413,840 reviews, primarily collected through a data crawler from Amazon, specifically focusing on mobile phone reviews from various brands and companies. The dataset contains three key attributes:

* **Product title**: The name of the mobile phone being reviewed.
* **Brand**: The company that manufactures the mobile phone.
* **Review text**: The actual customer feedback or opinion.

**3.2.1 Data Preprocessing**

To ensure the quality and consistency of the data, a thorough preprocessing phase was conducted. During this step:

* Reviews with missing product titles or brand names were removed, as they were considered incomplete and irrelevant.
* Reviews containing fewer than five words were also eliminated, as they were unlikely to offer any meaningful feature or sentiment information.
* The remaining review text underwent cleaning and normalization processes to reduce noise and enhance the quality of the data. This included:
  + Removing stop words and punctuation to eliminate irrelevant terms.
  + Converting integers into words for consistency.
  + Stemming and lemmatizing using the WordNet Lemmatizer, which reduced words to their base or root form. This helped minimize word variations and standardize different writing styles, making the feature extraction process more effective.



Table No. 3.2.1

**3.2.2 Refined Dataset**

After the preprocessing phase, the dataset was refined and reduced to 316,811 reviews, which were used for the experimental evaluation of the proposed PRUS framework. The cleaning process ensured that the dataset contained high-quality, meaningful reviews by eliminating incomplete, short, or irrelevant entries. The reduced dataset offers a substantial and diverse sample for analysis, making the evaluation process more reliable and robust.

The cleaned dataset covers 371 different mobile phone brands, representing a broad spectrum of manufacturers. This diversity allows the framework to be tested across a variety of products, ensuring that the evaluation is not limited to a few popular brands. The inclusion of both major and lesser-known brands adds depth to the dataset, making the analysis more comprehensive.

Upon further examination, it was found that approximately 40% of the reviews were related to just three major brands: Samsung, BLU, and Apple. This indicates the dominance and popularity of these brands in the dataset. Their higher review volume reflects their market influence and customer base, making them significant players in the analysis. The concentration of reviews around these three brands also suggests that they attract more customer attention and engagement, possibly due to their widespread availability, brand reputation, and extensive marketing efforts.

The dataset spans reviews for approximately 4,300 different mobile phone models, showcasing a large and varied product sample. This diversity ensures that the PRUS framework is tested on a wide range of products with different specifications, performance levels, and customer experiences. Analyzing such a large number of models helps the system generalize better, making it applicable to a variety of products and not just a limited subset.

The analysis further revealed that the top ten most reviewed products in the dataset received between 800 and 1000 reviews each, indicating their popularity and high customer engagement. The large volume of reviews for these products highlights their frequent customer feedback, making them ideal candidates for detailed sentiment analysis. The presence of multiple reviews allows the PRUS framework to accurately identify trends, patterns, and recurring sentiments associated with these products. The higher review count also makes the sentiment analysis more reliable, as a larger sample size reduces the impact.

1. **PROPOSED MODEL**

The PRUS framework (Product Recommendation based on User Specification) follows a systematic process to extract aspect-level sentiments and generate a ranked product recommendation list.

* **Sentence Segmentation**

The process begins by breaking the review into individual sentences. This step is performed using the Python Natural Language Toolkit (NLTK) library, which enables efficient sentence tokenization. The purpose of this segmentation is to analyze each sentence independently, as reviews often contain mixed sentiments about different product features. By splitting the review into sentences, the system can separately evaluate each feature's sentiment, rather than relying on the overall review polarity.

* **Sentiment Analysis and Feature Extraction**

In the second phase, sentiment analysis and feature extraction are applied to the segmented sentences. This is implemented using the Python TextBlob library, which performs polarity classification by categorizing each sentence as positive, negative, or neutral. Simultaneously, TextBlob extracts the features mentioned in each sentence. For example:

* In the sentence, *"The phone has an excellent camera but poor battery life"*, the features "camera" and "battery life" are extracted.
* The sentiment polarity is positive for the camera and negative for the battery life.
  + This process is applied to all the product reviews in the dataset. Each sentence is analyzed to extract both the features and their associated sentiments, which are appended together.
* Labelling Features with Sentiment
  + Once the features and their sentiments are extracted, they are labelled accordingly:
* If a sentence expresses positive sentiment, the associated features are marked as positively reviewed.
* If a sentence has negative sentiment, its features are labelled as negatively reviewed.
* Sentences classified as neutral result in the features being labelled as neutrally reviewed.
  + This labelling enables the system to quantify the sentiment distribution for each product feature, providing a detailed sentiment breakdown.
* Sentiment Scoring
  + The system then calculates sentiment scores by counting the number of times each feature is labelled as positive, negative, or neutral across all reviews for a specific product.
* The total positive, negative, and neutral counts for each feature are aggregated.
* The sentiment score for a feature is computed using the following formula:
  + *Score = (Positive count−Negative count) / Total Count*
* This formula ensures that features with more positive occurrences receive higher scores, while those with frequent negative mentions receive lower scores.
* The neutral mentions balance the score by preventing extreme polarity shifts.
  + For instance, if the feature "camera" appears 100 times in the reviews, with 60 positive mentions, 30 negative mentions, and 10 neutral mentions, the score would be:

Score = (60-30) / 100 = 0.3

* + This indicates a moderately positive sentiment for the camera.
* **Ranking with RANK-ify Algorithm**
  + Finally, the system applies the RANK-ify algorithm to generate the ranked product recommendation list. The algorithm uses the calculated sentiment scores to rank the products:
  + Products with higher positive sentiment scores for the user-specified features are ranked higher.
  + Products with more negative mentions or lower sentiment scores are ranked lower.
  + The RANK-ify algorithm ensures that the recommendations reflect both user preferences and sentiment-based feature evaluations, providing a personalized and accurate ranking.
  + Key Benefits of the PRUS Process
  + Granular Sentiment Analysis: By analyzing sentence-level sentiments, the system captures aspect-specific opinions, preventing misleading overall polarity judgments.
  + Comprehensive Feature Scoring: The system quantifies the positivity and negativity of each feature, making the ranking more accurate and meaningful.
  + Balanced Recommendations: By considering both positive and negative sentiments, the framework provides realistic and reliable product recommendations.
  + Efficient Ranking: The use of RANK-ify ensures that the final ranked list prioritizes products with strong positive sentiments for the features specified by the user.

**4.1 Rank – ify Algorithm**

The RANK-ify algorithm is a core component of the PRUS framework, designed to rank products based on feature scores in accordance with user-specified preferences. Its goal is to generate a personalized, relevance-based product list by prioritizing products that exhibit strong positive sentiment for the features the user specifies in their query. The algorithm considers both positive and negative occurrences of the features, ensuring a balanced and realistic ranking.

**4.1.1. Input Parameters and Initialization**

The RANK-ify algorithm takes three input parameters:

* **k**: Represents the maximum number of products to be returned in descending order of relevance, i.e., from the most relevant to the least relevant according to the user's query.
* **Lp**: Refers to the set of all products in the dataset that have been reviewed. This set contains the features associated with each product along with the positive and negative occurrence counts of these features.
* **Q:** Represents the list of features extracted from the user’s query, specifying the aspects that the user prioritizes in the product search.

**4.1.2. Iterative Feature Matching**

The algorithm iterates over:

* Every feature of each product (Line 4) and
* Each feature in the user query (Line 5).

For each product, the algorithm checks if the user-specified feature exists in the list of features associated with the product (Line 6). If the queried feature is present, the algorithm proceeds to calculate the Feature Score (FeaSco) for that feature using Equation 1 (Line 7):

*FeaSco(f)=(w1\*c(f+)−w2\*c(f−)) / (c(f+)+c(f−))*

Where:

* c(f+) → The **count of positive occurrences** of the feature fff.
* c(f−) → The **count of negative occurrences** of the feature.
* w1 → The **weight assigned to positive occurrences**.
* w2→ The **weight assigned to negative occurrences**.
* **Condition Sum of w1 and w2 should equals to One Always!!!**

**4.1.3. Weighted Sentiment Scoring**

The weight attributes w1 and w2 control the impact of positive and negative occurrences on the final ranking. Since users may value positive and negative opinions differently, these weights allow the system to adjust the influence of each sentiment.

* For example, if the user is more concerned with positive feedback, w1 will be larger than w2.
* Conversely, if the user gives more importance to negative feedback, w2 will be larger.
* To maintain balance and prevent one sentiment from dominating the ranking excessively, the algorithm enforces the condition:

*w1+w2=1*

This ensures that the influence of both sentiment polarities is normalized.

**4.1.4. Rank-Score Calculation**

Once the FeaSco is calculated for each user-specified feature in every product, the algorithm sums these feature scores to compute the overall Rank-Score of the product (Line 10). This Rank-Score represents the combined sentiment impact of all features in relation to the user’s query.

* Products with higher Rank-Scores exhibit stronger positive alignment with the user’s preferences.
* Products with more negative sentiment or fewer matching features receive lower scores.

**Sorting and Final Recommendation**

After calculating the Rank-Score for all products, the algorithm sorts the products in descending order (Line 12), prioritizing those with higher Rank-Scores.

* The top-k products from the sorted list are returned to the user (Line 13), providing a personalized and ranked recommendation list that closely matches their preferences.

Example Execution

For example, if a user queries for a smartphone with good battery life and camera, the algorithm:

* Iterates through all products and checks if they contain these features.
* Calculates the FeaSco for both battery life and camera by considering the positive and negative occurrences.
* Combines the FeaSco values into an overall Rank-Score for each product.
* Sorts the products based on their Rank-Score and presents the top-k ranked products to the user.

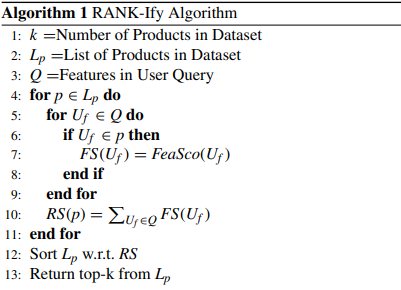


Fig. 4.1.4.1

**4.2. Key Benefits of the RANK-ify Algorithm**

* **Granular Sentiment Evaluation:** By considering both positive and negative occurrences, the algorithm offers a balanced ranking.
* **Weighted Scoring:** The weight parameters allow the system to customize the impact of positive and negative feedback, aligning with user priorities.
* **Efficient and Scalable:** The algorithm efficiently processes large datasets by iterating over relevant features only, making it suitable for scalable e-commerce platforms.
* **Personalized Recommendations:** By prioritizing products based on user-specified features, the algorithm generates accurate and personalized recommendations, enhancing the user experience.

Overall, the **RANK-ify algorithm** is a robust and efficient method for ranking products based on aspect-level sentiment analysis. Its ability to consider both positive and negative occurrences, apply weight-based scoring, and provide personalized recommendations makes it highly effective for e-commerce product ranking systems.

1. **DISCUSSIONS ON THE PROPOSED OUTPUT**

**5.1 Experiments and Results**

The experimental evaluation of the PRUS framework demonstrates its effectiveness in generating personalized product recommendations by ranking products based on aspect-level sentiment analysis. The evaluation was conducted using a preprocessed Amazon dataset containing mobile phone reviews from various brands. The system was tested by extracting features from a user query and generating a ranked recommendation list based on the Discounted Cumulative Gain (DCG) measure. The results highlight the accuracy and reliability of the PRUS framework in providing feature-specific recommendations, overcoming the limitations of overall review-based scoring.

**Sentiment-Driven Feature Scoring**

In the proposed approach, TextBlob was used to perform sentiment analysis, assigning individual sentiment scores to the features of each product. These scores were normalized on a scale of 1 to 10 to ensure consistent evaluation across all products. The framework calculates the Average Aspect Score (AAS) for each product usingthebelow equation:



Here, for a product p, all the features relevant to the user query (Q) are considered. Their sentiment scores are averaged, providing a composite score that reflects the product’s overall feature relevance. This ensures that products are ranked based on the sentiment scores of the features that matter most to the user, rather than relying on an overall rating.

For example, in a review stating:

*“The screen quality is good, but the battery life is poor,”*

The **screen quality** receives a **sentiment score of 8.65**, while the **battery life is assigned score of 3.7**, indicating positive and negative sentiments for different features. Traditional systems would assign a single overall rating, which could be misleading, while PRUS assigns individual scores to each feature, offering more precise and granular sentiment analysis.

**DCG and nDCG-Based Ranking**

Once the AAS values are calculated for all products, the DCG score is computed using the below Equation:



The DCG score quantifies how effectively a product matches the user-specified features and their associated sentiment scores. To ensure a fair and unbiased ranking, the system calculates the Ideal DCG (IDCG) using the below Equation:



The IDCG represents the optimal ranking scenario, where the most relevant products receive the highest scores. Finally, nDCG is calculated using the below Equation:

​ 

The nDCG value ensures that the ranking is normalized, making it more consistent and reliable across different product lists.

**Demonstration of the Output**

To demonstrate the effectiveness of PRUS, the authors conducted an example evaluation using a user query that requested a mobile phone with a good camera and high screen resolution. The system successfully displayed the top 10 ranked products based on:

* Rank Score (RS): Calculated using the RANK-ify algorithm.
* nDCG values: Derived using Equation 5.

The results showed that the highest-ranked products were those that received strong positive sentiment scores for both camera quality and screen resolution. While several products shared the same Rank Score (0.9), the nDCG values varied, allowing the system to further differentiate the products based on their overall sentiment relevance.

* The products with higher nDCG scores appeared at the top of the recommendation list, as they exhibited stronger positive feedback for the features specified by the user.
* The products with lower nDCG scores were ranked lower, reflecting their weaker alignment with the user’s query.
* This two-layer ranking system ensures that the most relevant products are placed higher, even when multiple products have the same Rank Score.

**Key Advantages of the PRUS Output**

The experimental results demonstrate several key benefits of the PRUS framework:

* Feature-Specific Ranking: Unlike traditional models that rely on overall ratings, PRUS assigns individual sentiment scores to product features, ensuring more accurate and personalized recommendations.
* Granular Sentiment Analysis: The system successfully captures both positive and negative opinions about different product features, preventing misleading overall ratings.
* DCG-Based Normalization: The use of nDCG values ensures that products are ranked fairly and consistently, even when their Rank Scores are identical.
* Improved Decision-Making: The feature-based scoring allows users to make better-informed decisions by clearly identifying products that excel in the specific features they care about.
* Efficient Query-Based Personalization: PRUS effectively tailors the recommendations to user preferences, offering a highly customized product ranking.

Discussion on Practical Implications

The PRUS framework offers a significant improvement over traditional product ranking methods, especially for e-commerce platforms. The ability to prioritize features based on user-specified preferences ensures that the recommendations are:

* Highly relevant and tailored to individual needs.
* Accurate, as the system captures fine-grained feature sentiments rather than relying on general product ratings.
* Efficient, by applying DCG and nDCG scoring, making the ranking fair and consistent across products.

For example, a customer searching for a laptop with long battery life and lightweight design would benefit from PRUS, as it ranks products based on aspect-specific sentiment scores rather than general ratings.

Overall, the experimental output of the PRUS framework demonstrates its effectiveness, reliability, and practicality in delivering accurate, feature-specific, and personalized product recommendations.

**5.2 Performance Analysis of the PRUS Method**

The **PRUS framework** was subjected to a series of experiments to evaluate its performance and effectiveness in generating ranked product recommendations based on user-specified features. The experiments were conducted by varying the weights w1w1w1 and w2w2w2, which control the influence of positive and negative occurrences of product features, respectively. The key objective was to observe the impact of these weight variations on the Rank Score (RS) and Normalized Discounted Cumulative Gain (nDCG), which are used as evaluation metrics.

**5.2.1. Experiment Setup and Weight Variations**

In the experiments, the **weights were alternated** in such a way that their **sum equals 1**, i.e.,w1+w2=1

The weights were varied over a range of **0.1 to 0.9**, creating different **combinations of positive and negative influence** on the final product ranking. The **product list (P1 to P10)** represents the **top-ranked products** based on the **RS and nDCG scores** corresponding to the different weight combinations. The **results of these experiments** are summarized in the below Table**.**

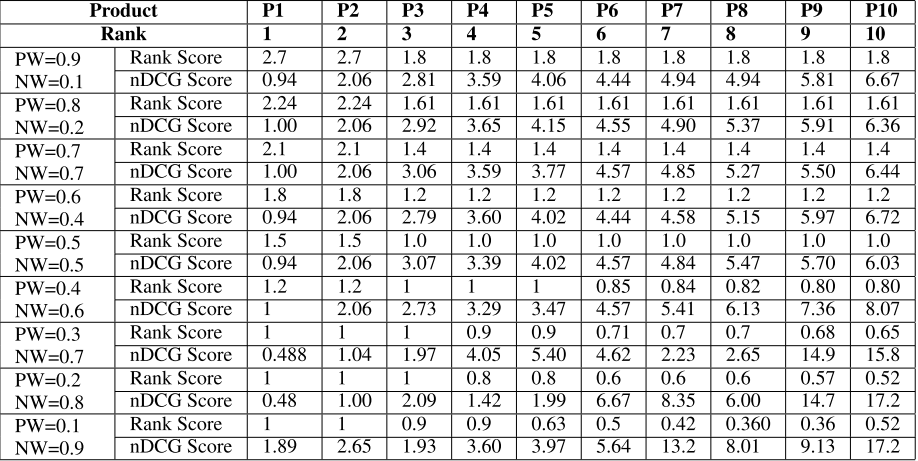


Table No. 5.2.1.1

**5.2.2. Impact of Positive and Negative Weights on nDCG**

The nDCG scores represent the relevance of the ranked products based on the user-specified features and their associated sentiments. The experimental results reveal the following trends:

* When the positive weight (w1) ranges between 0.9 and 0.5, there is no significant change in the overall nDCG scores. This indicates that as long as positive sentiments dominate, the relevance of the product ranking remains consistent.
* However, as soon as the negative weight (w2) increases beyond 0.5, there is a sudden increase in the nDCG scores. This happens because products with more negative mentions gain higher weight, making them more prominent in the ranking.
* This outcome demonstrates that nDCG scores increase when negative feedback is given more importance, reflecting a shift in product relevance towards products with more critical sentiments.

**5.2.3. Impact on Rank Score (RS)**

The **Rank Score (RS)**, calculated using the **RANK-ify algorithm**, represents the overall product ranking based on both positive and negative occurrences of the features. The experimental results show the following trend:

* When the positive weight (w1) is high (greater than 0.5), products with more positive feature sentiments receive higher Rank Scores.
* As the negative weight (w2) crosses 0.5, the RS begins to decline. This is because the increased influence of negative feedback causes products with more negative mentions to be ranked higher, lowering the RS for products with predominantly positive reviews.
* This behaviour aligns with the FeaSco formula (Equation 1), which defines the Feature Score as:

*FeaSco(f)=(w1\*c(f+)−w2\*c(f−)) / (c(f+)+c(f−))*

Where:

* w1 → The positive weight.
* w2 → The negative weight.
* c(f+) → The count of positive occurrences.
* c(f−) → The count of negative occurrences.
* Increasing w2 reduces the overall FeaSco value, leading to lower Rank Scores for products with more positive mentions.

**5.2.4. Influence of Weight Variation on the Product List**

The product list (P1 to P10) varies significantly as the positive and negative weights are adjusted.

* With higher positive weights, the products with strong positive sentiment for the user-specified features dominate the top ranks.
* As the negative weight increases, the list becomes populated with products that have more negative mentions for the specified features, reflecting the increased influence of negative sentiment.
* This highlights the flexibility of the PRUS framework, allowing users to adjust the ranking by controlling the impact of positive and negative occurrences.

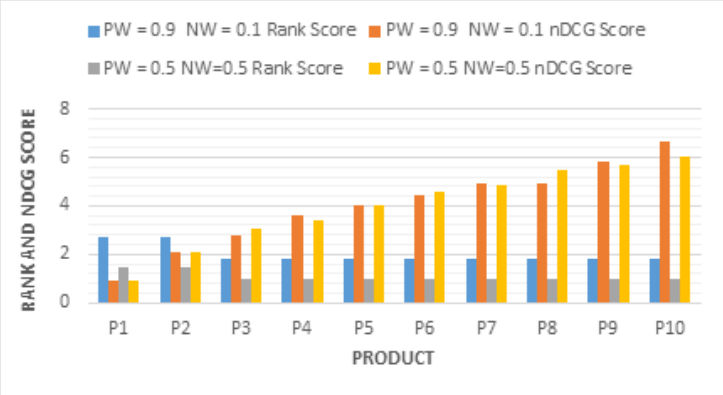


Fig. 5.2.4.1

**5.2.5**. **Key Observations from the Performance Analysis**

The experiments reveal several key insights into the performance of the PRUS framework that was discussed in the paper:

* No Major Change in nDCG with Higher Positive Weight: When the positive weight ranges between 0.9 and 0.5, there is no substantial change in the nDCG scores. This suggests that as long as positive sentiment dominates, the overall relevance of the recommendations remains stable.
* nDCG Increases with Higher Negative Weight: As the negative weight increases beyond 0.5, nDCG scores rise sharply. This is because the system prioritizes products with more negative feedback, making critical reviews more prominent in the ranking.
* Rank Scores Decrease with Increasing Negative Weight: When negative sentiment gains more influence (w2 > 0.5), the Rank Scores decline. This occurs because negative features are weighted more heavily, reducing the overall RS of products with predominantly positive reviews.
* User-Controlled Sentiment Influence: The PRUS framework provides users with the ability to control the degree of positive and negative influence on the final product list. This makes the system highly flexible and customizable, catering to individual preferences.

1. **CONCLUSION AND FUTURE WORK**

**6.1. Conclusion**

In this study, a Product Recommendation based on User Specification (PRUS) framework was proposed to generate a personalized, ranked list of products by analyzing customer reviews. Unlike existing methods, the PRUS approach incorporates both positive and negative sentiments of individual product features, enabling more accurate and personalized recommendations. While traditional ranking models often prioritize products based on overall sentiment, PRUS addresses the granularity issue by performing aspect-level sentiment analysis, ensuring that feature-specific opinions influence the ranking process.

The PRUS framework operates in three phases:

1. **Feature and Sentiment Extraction**: The system extracts features and their associated sentiments from individual sentences of customer reviews, capturing both positive and negative opinions at the sentence level.
2. **Feature-Sentiment Association**: The extracted features are appended with their respective sentiment polarities, allowing the system to quantify the positive and negative occurrences of each feature.
3. **Ranking Phase**: The user-specified features are extracted from the query, and the products are ranked using the RANK-ify algorithm based on their Rank Scores (RS). The RS considers both positive and negative weights, providing a balanced and realistic ranking.

The PRUS framework offers flexibility by allowing users to adjust the weight parameters associated with positive and negative sentiments. This gives users the ability to customize the ranking based on their individual preferences. Through extensive experiments, the PRUS framework demonstrated its effectiveness and accuracy in generating personalized recommendations, showing that:

* Increasing the negative weight results in a higher nDCG score, as products with more negative mentions become more prominent in the ranking.
* Increasing the positive weight causes the Rank Score to increase, prioritizing products with more positive sentiment.

The experimental results validated the reliability and accuracy of the PRUS framework, making it a robust solution for e-commerce platforms and other review-based product ranking systems.

**6.2. Future Work**

While the PRUS framework has demonstrated promising results, several avenues for future paper remain open. These directions aim to enhance the framework’s flexibility, scalability, and applicability in real-world scenarios:

1. **Multilingual Product Review Datasets**:
   * The current PRUS framework is applied to English-language reviews only. A natural extension of this work would be to implement multilingual support.
   * This would enable the framework to process reviews in multiple languages, making it applicable to global e-commerce platforms.
   * Future work could incorporate language translation models or multilingual sentiment analysis techniques to extract feature-level sentiments across diverse languages.
2. **User-Controlled Polarity Customization**:
   * The PRUS framework currently allows users to adjust the weight parameters associated with positive and negative sentiments, offering some flexibility in ranking.
   * Future enhancements could provide more advanced customization options, allowing users to:
     + Specify individual sentiment thresholds for specific features.
     + Define custom polarity preferences, such as favoring features with balanced sentiments or discounting neutral feedback.
   * This would result in more dynamic and user-centric recommendations.
3. **Incorporating Contextual and Temporal Factors**:
   * Future paper could enhance the PRUS framework by incorporating contextual information such as time-sensitive reviews or trending sentiments.
   * This would allow the system to prioritize recent reviews or consider seasonal trends when ranking products, making the recommendations more relevant and timely.
4. **Real-Time Product Recommendation System**:
   * To make the PRUS framework more practical for real-world applications, future work could focus on real-time product ranking.
   * This would enable the system to dynamically rank products as new reviews are added, ensuring that the recommendations remain current and accurate.

**REFERENCES**

[1] P. K. Jain, R. Pamula, and G. Srivastava, ‘‘A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews,’’ Comput. Sci. Rev., vol. 41, Aug. 2021, Art. no. 100413.

[2] S. Moro and P. Rita, ‘‘Data and text mining from online reviews: An automatic literature analysis,’’ WIREs Data Mining Knowl. Discovery, vol. 12, no. 3, May 2022, Art. no. e1448. [3] N. Hussain, H. T. Mirza, G. Rasool, I. Hussain, and M. Kaleem, ‘‘Spam review detection techniques: A systematic literature review,’’ Appl. Sci., vol. 9, no. 5, p. 987, Mar. 2019.

[4] L. He, X. Wang, H. Chen, and G. Xu, ‘‘Online spam review detection: A survey of literature,’’ Hum.-Centric Intell. Syst., vol. 2, nos. 1–2, pp. 14–30, Jun. 2022.

[5] S. K. Maurya, D. Singh, and A. K. Maurya, ‘‘Deceptive opinion spam detection approaches: A literature survey,’’ Int. J. Speech Technol., vol. 53, no. 2, pp. 2189–2234, Jan. 2023.

[6] E. Lee, F. Rustam, P. B. Washington, F. E. Barakaz, W. Aljedaani, and I. Ashraf, ‘‘Racism detection by analyzing differential opinions through sentiment analysis of tweets using stacked ensemble GCR-NN model,’’ IEEE Access, vol. 10, pp. 9717–9728, 2022.

[7] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, ‘‘Automatic text summarization: A comprehensive survey,’’ Expert Syst. Appl., vol. 165, Mar. 2021, Art. no. 113679.

[8] M. Umer, I. Ashraf, A. Mehmood, S. Kumari, S. Ullah, and G. S. Choi, ‘‘Sentiment analysis of tweets using a unified convolutional neural network-long short-term memory network model,’’ Comput. Intell., vol. 37, no. 1, pp. 409–434, Feb. 2021.

[9] P. K. Soni and R. Rambola, ‘‘A survey on implicit aspect detection for sentiment analysis: Terminology, issues, and scope,’’ IEEE Access, vol. 10, pp. 63932–63957, 2022. [10] N. Aslam, K. Xia, F. Rustam, A. Hameed, and I. Ashraf, ‘‘Using aspectlevel sentiments for calling app recommendation with hybrid deep-learning models,’’ Appl. Sci., vol. 12, no. 17, p. 8522, Aug. 2022.

[11] G. Brauwers and F. Frasincar, ‘‘A survey on aspect-based sentiment classification,’’ ACM Comput. Surv., vol. 55, no. 4, pp. 1–37, Apr. 2023.

[12] M. Sivakumar and S. R. Uyyala, ‘‘Aspect-based sentiment analysis of mobile phone reviews using LSTM and fuzzy logic,’’ Int. J. Data Sci. Anal., vol. 12, no. 4, pp. 355–367, Oct. 2021.

[13] J. Zhang, X. Lu, and D. Liu, ‘‘Deriving customer preferences for hotels based on aspect-level sentiment analysis of online reviews,’’ Electron. Commerce Res. Appl., vol. 49, Sep. 2021, Art. no. 101094.

[14] K. Ganesan and C. Zhai, ‘‘Opinion-based entity ranking,’’ Inf. Retr., vol. 15, no. 2, pp. 116–150, Apr. 2012.

[15] L. Akoglu, ‘‘Opinion fraud detection in online reviews by network effects,’’ in Proc. ICWSM, vol. 7, no. 1, 2013, pp. 2–11.

[16] E. Marrese-Taylor, J. D. Velásquez, and F. Bravo-Marquez, ‘‘A novel deterministic approach for aspect-based opinion mining in tourism products reviews,’’ Expert Syst. Appl., vol. 41, no. 17, pp. 7764–7775, Dec. 2014.

[17] A. Kumar and S. Abirami, ‘‘Aspect-based opinion ranking framework for product reviews using a Spearman’s rank correlation coefficient method,’’ Inf. Sci., vols. 460–461, pp. 23–41, Sep. 2018.

[18] S. Bashir, W. Afzal, and A. R. Baig, ‘‘Opinion-based entity ranking using learning to rank,’’ Appl. Soft Comput., vol. 38, pp. 151–163, Jan. 2016.

[19] T. A. Rana and Y.-N. Cheah, ‘‘A two-fold rule-based model for aspect extraction,’’ Expert Syst. Appl., vol. 89, pp. 273–285, Dec. 2017.

[20] C. Wu, F. Wu, S. Wu, Z. Yuan, and Y. Huang, ‘‘A hybrid unsupervised method for aspect term and opinion target extraction,’’ Knowl.-Based Syst., vol. 148, pp. 66–73, May 2018.

[21] Z. Luo, S. Huang, F. F. Xu, B. Y. Lin, H. Shi, and K. Zhu, ‘‘ExtRA: Extracting prominent review aspects from customer feedback,’’ in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 3477–3486.

[22] S. Sharmila and S. Vijayarani, ‘‘Association rule mining using fuzzy logic and whale optimization algorithm,’’ Soft Comput., vol. 25, no. 2, pp. 1431–1446, Jan. 2021.

[23] S. Gurumoorthy, B. N. K. Rao, X.-Z. Gao, B. V. Krishna, A. K. Pandey, and A. S. Kumar, ‘‘Feature based opinion mining and sentiment analysis using fuzzy logic,’’ in Cognitive Science and Artificial Intelligence (SpringerBriefs in Forensic and Medical Bioinformatics). Singapore, 2018, pp. 79–89.

[24] M. Venugopalan and D. Gupta, ‘‘An enhanced guided LDA model augmented with BERT based semantic strength for aspect term extraction in sentiment analysis,’’ Knowl.-Based Syst., vol. 246, Jun. 2022, Art. no. 108668.

[25] M. Hu and B. Liu, ‘‘Mining and summarizing customer reviews,’’ in Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2004, pp. 168–177.

[26] A.-M. Popescu and O. Etzioni, ‘‘Extracting product features and opinions from reviews,’’ in Proc. Conf. Hum. Lang. Technol. Empirical Methods Natural Lang. Process. (HLT), 2005, pp. 9–28.

[27] S. Li, L. Zhou, and Y. Li, ‘‘Improving aspect extraction by augmenting a frequency-based method with web-based similarity measures,’’ Inf. Process. Manage., vol. 51, no. 1, pp. 58–67, Jan. 2015.

[28] Z. Li, M. Zhang, S. Ma, B. Zhou, and Y. Sun, ‘‘Automatic extraction for product feature words from comments on the web,’’ in Proc. Asia Inf. Retr. Symp. Sapporo, Japan: Springer, Oct. 2009, pp. 112–123.

[29] S. Raju, P. Pingali, and V. Varma, ‘‘An unsupervised approach to product attribute extraction,’’ in Proc. Eur. Conf. Inf. Retr. Toulouse, France: Springer, Apr. 2009, pp. 796–800.

[30] Ö. F. Güneş, T. Furche, and G. Orsi, ‘‘Structured aspect extraction,’’ in Proc. COLING 2016, 26th Int. Conf. Comput. Linguistics, Tech. Papers, 2016, pp. 2321–2332.

[31] C. Quan and F. Ren, ‘‘Unsupervised product feature extraction for feature-oriented opinion determination,’’ Inf. Sci., vol. 272, pp. 16–28, Jul. 2014.

[32] W. Maharani, D. H. Widyantoro, and M. L. Khodra, ‘‘Clue propagation based on non-adjective opinion words for handling disconnected propagation in product reviews,’’ IEEE Access, vol. 10, pp. 56785–56795, 2022.

[33] X. Luo, ‘‘Efficient english text classification using selected machine learning techniques,’’ Alexandria Eng. J., vol. 60, no. 3, pp. 3401–3409, Jun. 2021.

[34] C.-P. Wei, Y.-M. Chen, C.-S. Yang, and C. C. Yang, ‘‘Understanding what concerns consumers: A semantic approach to product feature extraction from consumer reviews,’’ Inf. Syst. e-Bus. Manage., vol. 8, no. 2, pp. 149–167, Mar. 2010.

[35] B. Ma, D. Zhang, Z. Yan, and T. Kim, ‘‘Anlda and synonym lexicon based approach to product feature extraction from online consumer product reviews,’’ J. Electron. Commerce Res., vol. 14, no. 4, p. 304, 2013.

[36] K. Liu, L. Xu, and J. Zhao, ‘‘Co-extracting opinion targets and opinion words from online reviews based on the word alignment model,’’ IEEE Trans. Knowl. Data Eng., vol. 27, no. 3, pp. 636–650, Mar. 2015. 81296 VOLUME 11, 2023 N. Hussain et al.: PRUS: Product Recommender System Based on User Specifications and Customers Reviews.

[37] C.-H. Lai, D.-R. Liu, and K.-S. Lien, ‘‘A hybrid of XGBoost and aspectbased review mining with attention neural network for user preference prediction,’’ Int. J. Mach. Learn. Cybern., vol. 12, no. 5, pp. 1203–1217, May 2021.

[38] G. Kaur and A. Sharma, ‘‘A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis,’’ J. Big Data, vol. 10, no. 1, pp. 1–23, Jan. 2023. [39] M. Nasir and C. I. Ezeife, ‘‘Semantic enhanced Markov model for sequential E-commerce product recommendation,’’ Int. J. Data Sci. Anal., vol. 15, no. 1, pp. 67–91, Jan. 2023.

[40] S. Bird, E. Klein, and E. Loper, Natural Language Processing With Python: Analyzing Text With the Natural Language Toolkit. Sebastopol, CA, USA: O’Reilly Media, 2009.

[41] S. Loria, Textblob Documentation, document Release 0.15, 2018, vol. 2, no. 8.

[42] D. Valcarce, A. Bellogín, J. Parapar, and P. Castells, ‘‘Assessing ranking metrics in top-N recommendation,’’ Inf. Retr. J., vol. 23, no. 4, pp. 411–448, Aug. 2020.

[43] P. Nitu, J. Coelho, and P. Madiraju, ‘‘Improvising personalized travel recommendation system with recency effects,’’ Big Data Mining Anal., vol. 4, no. 3, pp. 139–154, Sep. 2021. [44] M. K. Singh and O. P. Rishi, ‘‘Event driven recommendation system for E-commerce using knowledge based collaborative filtering technique,’’ Scalable Comput., Pract. Exper., vol. 21, no. 3, pp. 369–378, Aug. 2020.

[45] Z.-P. Fan, G.-M. Li, and Y. Liu, ‘‘Processes and methods of information fusion for ranking products based on online reviews: An overview,’’ Inf. Fusion, vol. 60, pp. 87–97, Aug. 2020.