

Supplementing Product Reviews: Retrieving Opinions from Products with Similar Attributes

Marino Fujii¹[0009–0000–2335–7430], Takehiro Yamamoto¹[0000–0003–0601–3139],
and Takayuki Yumoto¹[0009–0005–1854–0448]

University of Hyogo, Kobe, Japan
ad24f053@guh.u-hyogo.ac.jp
t.yamamoto@sis.u-hyogo.ac.jp
yumoto@sis.u-hyogo.ac.jp

Abstract. In this study, we propose a method to help understand products that don’t have many reviews. Specifically, we use an LLM to retrieve reviews of similar products. This helps users get more useful information when they are thinking of buying a product. First, when the user inputs a product and a question, the system uses the LLM to find the attributes that are related to the user’s question. Next, the system calculates the similarity of the attributes and find similar products. After that, by using an LLM, the system retrieves opinions from reviews of the similar products that are related to these attributes. Finally, the system ranks the opinions that were judged to be related to the attribute queries and shows them as the search results. This adds more useful information to support the target product. In the experiment, we compared the similarity between the opinions retrieved by the proposed method and the actual opinions that the target product has. As a result, opinions retrieved from similar products were slightly more similar to the actual opinions than those from randomly selected products.

Keywords: information retrieval · LLM · review supplement.

1 Introduction

As the internet has grown, more and more people are shopping on e-commerce websites [14]. When buying something on an e-commerce site, reviews are important to help people decide what to buy. By reading reviews, people can learn more about the actual use of the product.

However, there are some products, such as new or less popular ones, that have very few reviews. In such cases, users may not be able to find enough information they want. For example, suppose a user wants to know how much space a new TV called *A* takes up by reading reviews. Even if the user searches for reviews about screen size of *A*, there may be very few reviews mentioning it, or none at all. The user might end up buying a TV that is not the right size for the room.

To solve this problem, we supplement the missing opinions by using reviews from similar products, as shown in Figure 1. First, the user selects the product

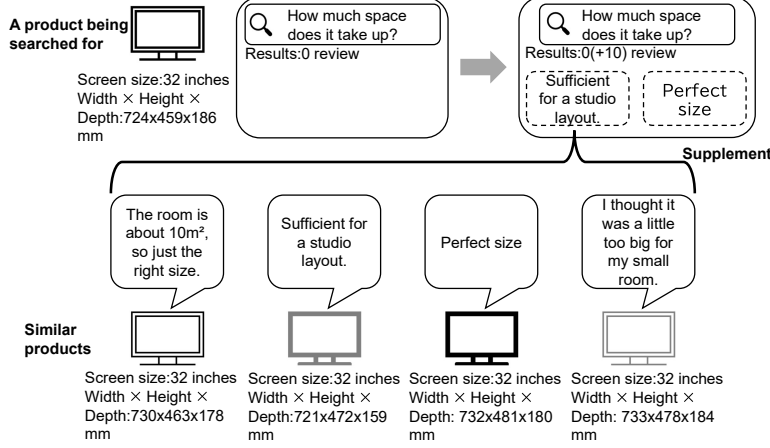


Fig. 1. Examples of supplementing product opinions with reviews from similar products.

they want to research. Next, the user types a question, such as “How much space does it take up?” or “Easy to carry.” Using a Large Language Model (LLM), the system reads this question and selects the related attribute (like screen size, weight, or sound quality) from the product’s specification. In this case, screen size and width × height × depth are selected. Then, using the LLM again, the system finds reviews from similar products that mention the same attribute, such as a review saying “*The room is about 10m², so just the right size.*” These reviews are shown to the user. In this way, even if the target product has very few reviews, the user can get opinions on what people think about it. For example, suppose a review of TV *B*, which has the same screen size as TV *A*, says “*The room is about 10m², so just the right size.*” This opinion also applies to TV *A*.

Studies that use reviews from similar products to help understand the product the user is searching for have been done before [2] [9] [11]. However, no past study has used an LLM. We thought that using the LLM could make the user’s input and the system’s answer more flexible and more accurate. In our study, our goal is to find reviews from similar products that mention the attributes the user wants to know by using the LLM.

In the experiment, we evaluated the system from three points of view. The first point is how well the LLM can select the correct attribute related to what the user wants to know. We compared the results of the author’s judgments and the LLM’s judgments, and calculated precision, recall, and F_1 -score. The result was that effectiveness was sufficiently high. The second point is how well the LLM can find reviews related to the selected attribute. Again, we compared the results of the author’s judgments and the LLM’s judgments, and calculated precision, recall, and F_1 -score. The result was that effectiveness was not high.

The third point is how similar the opinions found by the proposed method are to the actual opinions of the target product. We used both automatic similarity calculation and human judgment to evaluate this. As a result, our method, which uses opinions from similar products, performed better than the baseline method, which uses opinions from randomly selected products. Therefore, the proposed method can retrieve opinions that are closer to actual opinions.

2 Related Work

2.1 Analyzing Customer Reviews

There have been studies on extracting attributes or aspects and opinions from reviews. Hu et al. collected positive and negative opinions about product attributes using WordNet [5]. Poria et al. used word embeddings to find aspects mentioned in reviews [10]. Wang et al. proposed using a multi-layer coupled attention mechanism to extract attributes and opinions [16]. Ishii et al. used T5 to generate which aspects were discussed in each sentence of the review [6]. Adhikary et al. used an LLM to extract review aspects and their opinions [1].

These previous studies have focused only on finding features and opinions in reviews about products. They do not address the issue of products with few or no reviews. In our study, we use reviews of similar products to fill in the gaps. This allows us to obtain useful information about a particular product that users are interested in, even when there are few reviews of the product.

2.2 Retrieving Opinions from Reviews of Similar Products

There have been studies on methods to supplement opinions of a product by using reviews from similar products when the product does not have enough reviews. Park et al. used a query likelihood model to retrieve reviews from products that have similar attributes to the target product [9]. Du et al. also used similar products and considered both usefulness and coverage when retrieving reviews [2]. Rozen et al. built a model using RoBERTa to answer questions that were not answered in the reviews of the target product [13]. Their model used reviews from similar products to create answers. Pourgholamali et al. used embeddings learned in the same vector space of attribute values and reviews to obtain reviews of similar products [11].

These previous studies often used limited methods for asking and answering questions, such as only allowing simple inputs or yes/no outputs. In our study, we introduce a new approach using a Large Language Model (LLM). This enables users to ask questions more flexibly and receive answers in many different forms, something previous methods could not achieve.

3 Problem Definition

In this section, we define the problem that our study focuses on. A product category consists of k products, written as $\{P^{(1)}, P^{(2)}, \dots, P^{(k)}\}$. Each product

has its attributes that describe its specifications. An attribute is written as a pair (a_x, v_x) , where a_x is the name of the attribute and v_x is its value. For example, in the case of a TV, attributes can be written as (screen size, 32 inches) or (weight, 4.5 kg). If the products in a category have n attributes, the set of attributes for product $P^{(i)}$ can be written as shown in Equation (1).

$$P^{(i)} = \{(a_1^{(i)}, v_1^{(i)}), (a_2^{(i)}, v_2^{(i)}), \dots, (a_n^{(i)}, v_n^{(i)})\} \quad (1 \leq i \leq k) \quad (1)$$

The user inputs the target product $P^{(i)}$ and the question or the requirement q about the product into the system. We call the product $P^{(i)}$ the **target product**, and the question or the requirement q the **text query**. Examples of text queries include “How much space does it take up?” and “Easy to carry.”

The system is given a set of products $\{P^{(1)}, P^{(2)}, \dots, P^{(k)}\}$, a target product $P^{(i)}$, and a text query q . The system then selects the attributes that are related to the text query q . We call these attributes the **attribute queries**. For example, if the text query is “How much space does it take up?”, then attributes like “screen size” and “width $\times$ height $\times$ depth” become the attribute queries. The selected set of m attribute queries is written as $A_q^{(i)} = \{(a_l^{(i)}, v_l^{(i)})\}_{l=1}^m \subseteq P^{(i)}$. For each attribute query $p_l^{(i)} = (a_l^{(i)}, v_l^{(i)}) \in A_q^{(i)}$, the system calculates its similarity to the same attribute in other products, written as $\text{sim}(p_l^{(i)}, p_l^{(j)})$. Using this similarity, the system selects N products that are similar to the target product. This set is written as $S_q = \{P^{(1)}, \dots, P^{(N)}\}$ and is called the **similar products**. Then, the system uses the reviews of these similar products in S_q to find and rank the opinions that are likely to match the target product. These opinions are shown to the user.

4 Methodology

4.1 Overview

An overview of the proposed method is shown in Figure 2. This method uses an LLM to supplement opinions of the target product by retrieving opinions about attribute queries from similar products. First, the user inputs a target product and a text query. Next, the system uses an LLM to select attribute queries from the text query. Then, the system calculates the similarity of the values of these attribute queries between the target product and other products to find similar products. After that, by using an LLM, the system retrieves opinions from reviews of the similar products that are related to the attribute queries. Finally, the system ranks the opinions that were judged to be related to the attribute queries and shows them as the search results.

4.2 Transformation of Text Query into Attribute Queries

We use an LLM to judge which attributes are related to a text query. In our study, we limit the text queries to those that ask about the features of the product

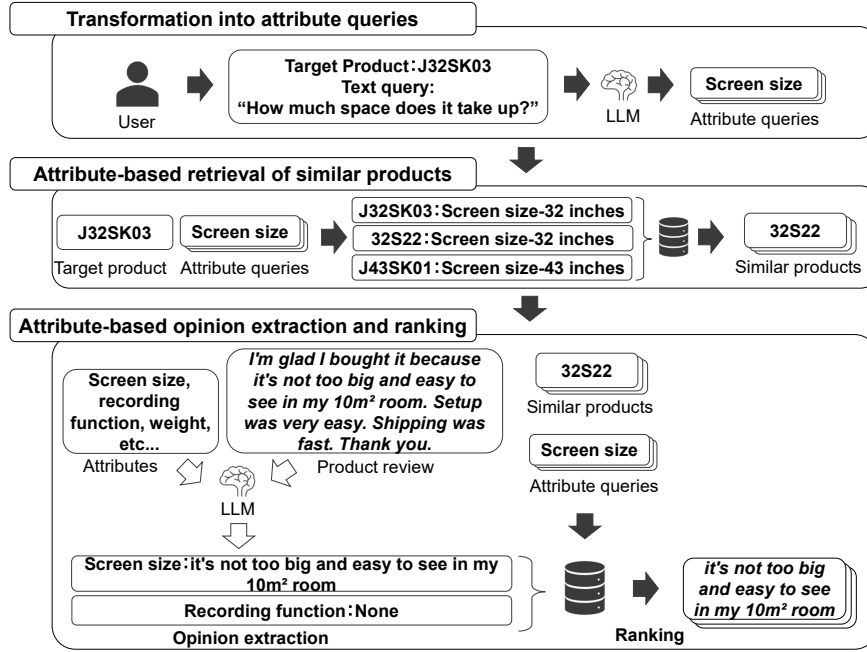


Fig. 2. Overview of the proposed method.

because most of the product attributes used in the supplement are related to features. Although price is also important when reading reviews, we exclude it from our study because it can change depending on when the user buys.

The following is the prompt used when judging the attributes related to TV:

You are now trying to search for reviews of a {product category}.
 Given the input "{text query}", does the attribute "{attribute query candidate}" match the query? Think step-by-step, and write the answer as True or False.
 ###OUTPUT EXAMPLE:
 True or False

In this case, {product category} is "TV", {text query} can be something like "How much space does it take up?" or "Good image quality", and {attribute query candidate} can be attributes such as "screen size", "number of pixels", and others. Attributes that output True at this prompt are used as attribute queries.

4.3 Attribute-Based Retrieval of Similar Products

We calculate the similarity between the values of attribute queries of the target product and those of other products to select similar products. At this time, we

use only the attributes that were judged as True in Section 4.2 as the attribute queries $A_q^{(i)}$. For a product $P^{(i)}$ with an attribute $p_l^{(i)} = (a_l^{(i)}, v_l^{(i)})$, we calculate its similarity with the same attribute $p_l^{(j)} = (a_l^{(j)}, v_l^{(j)})$ of another product $P^{(j)}$. If the attribute is quantitative data, the similarity is calculated using the following formula:

$$\text{sim}(p_l^{(i)}, p_l^{(j)}) = 1 - |v_l^{(i)} - v_l^{(j)}| \quad (2)$$

Here, $v_x^{(i)}$ is a normalized value so that the maximum value becomes 1. If the attribute is categorical data, the similarity is calculated using the following formula.

$$\text{sim}(p_l^{(i)}, p_l^{(j)}) = \begin{cases} 1 & \text{if } v_l^{(i)} = v_l^{(j)} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

In this case, the similarity is 1 only when the values of attribute queries match exactly.

In our study, we calculate the similarity for each attribute in the attribute queries $A_q^{(i)}$ using the formulae above. We then select the top N products that have the largest number of similar attributes as the similar products. If two or more products have the same total number of similar attributes, we give higher priority to the product that has more reviews.

4.4 Attribute-Based Opinion Extraction and Ranking

We use an LLM to extract the opinions of product reviews that mention specific attribute queries.

When using TV reviews, the following is an example of the prompt:

From the {product category} review “{review of the product in the product category}”, please get the opinion related to “{one attribute query}.” Follow the OUTPUT_EXAMPLE format.

If there is no related description, output the value of “attribute” as “{one attribute query}” and the value of “sentence” as “none.”

{instructions for JSON output}

###INPUT_EXAMPLE:

(Example for a hair dryer review)

This hair dryer has strong power and dries my hair quickly.

####OUTPUT_EXAMPLE:

When getting an opinion related to airflow from the hair dryer review

{{attribute: airflow, sentence: has strong power and dries my hair quickly}}

INPUT: {review of the product in the product category}

JSON-OUTPUT:

To generate more accurate outputs, we use a few-shot prompt that includes examples. At that time, to prevent the examples from affecting the output,

Table 1. Example of output when the input is “How much space does it take up?”

How much space does it take up?
1. Size
2. It is just the right size for watching from 1.5 to 2 meters away. It might be too small for those who sit farther.
3. I mounted it on the wall (installed 5 to 7 cm from the wall behind the TV).
4. Perfect size
5. I searched for a 32-inches TV because my room is $10m^2$.
6. Sufficient for a studio layout.
7. Because it is lightweight, you don’t need strong furniture to place it.
8. I thought it was a little too big for my small room.
9. The room is about $10m^2$, so just the right size.
10. This size

we use attributes and reviews from a different product category than the one currently being searched.

For example, suppose we want to extract information about “screen size” or “recording function” from the following TV review: “*I’m glad I bought it because it’s not too big and easy to see in my $10m^2$ room. Setup was very easy. Shipping was fast. Thank you.*” In this case, {product category} is “TV”, {review of the product in the product category} is “*I’m glad I bought it because it’s not too big and easy to see in my $10m^2$ room. Setup was very easy. Shipping was fast. Thank you.*”, and {one attribute query} is either “screen size” or “recording function”. When we run the prompt using “screen size” as the attribute query, the output includes the phrase “*it’s not too big and easy to see in my $10m^2$ room*”, since it relates to screen size. On the other hand, since this review does not mention anything related to the “recording function” attribute, running the prompt with “recording function” as the attribute query will return “none” as the output.

By applying the above prompt, we can extract the opinion that corresponds to each attribute in a review. Note that this extraction step is independent of the user’s text query; therefore, the entire review corpus can be processed offline before search time, and the attribute-opinion pairs can be extracted in advance.

Then, we present to the user the opinions related to the attribute query that were found in the reviews of similar products. To rank the opinions, we calculated the similarity between the text query and these opinions. Specifically, the text query and these opinions were converted into embeddings and the cosine similarity was calculated.

We describe an example of the actual output. When the target product is a TV with model number “J32SK03” and the text query is “How much space does it take up?”, the top 10 results ranked by similarity to the text query are shown in Table 1. For example, opinions like “*Sufficient for a studio layout*” and “*I searched for a 32-inches TV because my room is $10m^2$* ” were successfully retrieved from reviews related to the size of a 32-inches TV.

Table 2. List of attributes used for cameras and TVs.

Category	Attributes
TV	Screen size, Number of pixels, Panel type, HDR format, Image processing engine, Backlight, Quantum dots, High frame rate feature, BS 8K, BS 4K/110°CS 4K, Recording function, Built-in drive, Auto recording, Simultaneous recording of 2 programs, Fast playback, Smart speaker connection, Rotating stand, HDMI terminal, Voice-operated remote control, Number of speakers, Width $\times$ Height $\times$ Depth, Weight
Camera	Self-timer, Selfie function, LCD monitor, Startup time, Number of shots, Shutter speed, ISO sensitivity, Weight, Width $\times$ Height $\times$ Depth, Interface, Time-lapse, Number of pixels, Continuous shooting, Image stabilization, 4K support, Touch panel, Storage media, Recording format, Image sensor

5 Experiment

5.1 Datasets

We conducted the experiment with two categories: TVs and cameras. We selected these two categories because they have a large number of attributes. We used the specification information of TVs and cameras listed on the Japanese website kakaku.com¹ as the product attributes. Kakaku.com is a website in Japan that shows product descriptions, user reviews, and information from online stores. Each product page includes a section called “specification information,” which lists the attributes of the product. If the products belong to the same category, they usually share the same set of attributes. We used 22 attributes for TVs and 19 attributes for cameras. The list of attributes used is shown in Table 2.

For the product reviews, we used the Rakuten Ichiba review data contained in the Rakuten Dataset [12]. Rakuten Ichiba is a Japanese e-commerce site. Specifically, we selected the top 100 products with the most reviews in the TV and camera categories. These reviews were posted between January 1, 2017 and December 31, 2019. From these, we used only the products that had clearly written model numbers and specification information available on kakaku.com. As a result, we used 7,963 reviews for 87 TV products and 1,840 reviews for 55 camera products. For the text queries, we prepared 25 queries for each category by hand. The queries we prepared are shown in Table 3. Note that the experiments were conducted in Japanese, and the attributes and queries listed in Tables 2 and 3 are presented in English translation.

5.2 Model Used

We describe the models used in the proposed method. For judging the attribute queries related to the text query in Section 4.2, we used GPT-4o(2024-0806)². We

¹ <https://kakaku.com/>

² <https://platform.openai.com/docs/models#gpt-4o>

Table 3. List of actual queries prepared.

TV	Camera
Easy to carry	Best for taking still photos of baseball
No delay when playing games	Good for beginners
Good for watching movies	Good for taking pictures of trains
Good sound quality	Vibration resistance (image stabilization)
Good for watching sports	Can it take pictures of stars?
Smooth movement of rotating stand	Good for indoor selfies
Fast response of the remote control	Difference from iPhone
High image quality	Faster shooting than a smartphone
Stable stand	Easy to carry
Easy to see from the side	Long battery life
Can record a lot	Good image quality in high sensitivity
Can be operated while recording two programs	Good for taking pictures of food
Smooth playback of recorded programs	Good printing quality
How much space does it take up?	Difference from the LCD screen image
Accurate voice recognition	Good for shooting in dark places
Does not tire the eyes during long viewing	High-speed continuous shooting
Easy volume adjustment	Easy to edit on a PC
Smooth connection with external HDD	Good for landscape photography
Disadvantage of no high frame rate feature	Easy for children to use
Easy-to-use HDMI port	Smooth time-lapse function
Easy to set recording options	Easy-to-see LCD screen
Smooth video display	Responsive touch panel
Good for use as a PC monitor	Actual number of pictures taken
Good signal reception	Vivid colors
Good for watching concerts	Large storage capacity

selected this model because the number of processes required was relatively small ((number of text queries) $\times$ (number of attributes)), allowing us to prioritize accuracy over computational cost. For identifying the attribute related to product reviews in Section 4.4, we used **Llama-3.1-Swallow- 8B-Instruct-v0.3** [3] [4]. This model is based on Llama 3.1 and has been further pre-trained on Japanese datasets, which improves its performance in Japanese. We selected this model because the number of required processes was large ((number of reviews) $\times$ (number of attributes)), making it necessary to use a lightweight model that can run efficiently at a lower computational cost. In addition, for calculating cosine similarity during ranking in Section 4.4, we used **multilingual-e5-large** [15]. We selected this model because of its high retrieval performance.

5.3 Effectiveness of Judging Attributes Related to Text Queries

We evaluated the effectiveness of judging the attributes related to the text queries, as described in Section 4.2. For 25 text queries each from the TV and camera categories, we manually judged whether each attribute was relevant or not. We treated these manual judgments as the ground truth, and then compared them with the results given by the LLM.

Table 4. Effectiveness of judging attributes related to target products.

Metric	TV	Camera
Precision	0.702	0.534
Recall	0.676	0.761
F_1 -score	0.688	0.629

Table 5. Effectiveness of identifying related attributes in product reviews.

Metric	TV	Camera
Precision	0.600	0.441
Recall	0.308	0.650
F_1 -score	0.409	0.526

The results are shown in Table 4. In the TV category, the precision was 0.702, the recall was 0.676, and the F_1 -score was 0.688. In the camera category, the precision was 0.534, the recall was 0.761, and the F_1 -score was 0.629. Compared to the TV category, two of the three values were lower.

5.4 Effectiveness of Judging Attributes Related to Product Reviews

We evaluated the effectiveness of judging the attributes related to product reviews, as described in Section 4.4. We randomly selected 200 reviews each from the TV and camera categories. For each review, we manually judged whether an attribute was relevant or not. These manual judgments were used as the ground truth, and we compared them with the results judged by the LLM.

The results are shown in Table 5. In the TV category, the precision was 0.600, the recall was 0.308, and the F_1 -score was 0.409. In the camera category, the precision was 0.441, the recall was 0.650, and the F_1 -score was 0.526.

The impact of this effectiveness is discussed in Section 6.1.

5.5 Similarity of the Opinions Retrieved by the Proposed Method

To verify the efficacy of the proposed method, we calculated the similarity between the opinions retrieved by the proposed method and the actual opinions of the target product. The purpose of this evaluation is to check whether the opinions retrieved by the proposed method can be used to supplement opinions when the target product has no reviews. If the opinions retrieved from similar products is similar to the actual opinions for the target product, we can say the proposed method can supplement the opinions of the target product.

Evaluation Method In the proposed method, we first selected 10 similar products using the method described in Section 4.3. Then, using the methods of Sections 4.4, we ranked the opinions related to the attribute queries and used the

Table 6. Opinion similarity between target products and similar products retrieved by proposed and baseline methods. Bold indicates a higher metric score.

Metric	TV		Camera	
	Proposed	Baseline	Proposed	Baseline
ROUGE	0.381	0.364	0.311	0.309
BLEU	11.725	11.107	9.553	9.098
BERTScore	0.762	0.761	0.745	0.741

top 10 results. We selected the top 10 results because this number includes a reasonable amount of opinions that are highly related to the text query. As a baseline method, we replaced the 10 similar products with 10 randomly selected products. Target products were prepared for each text query. Specifically, one product was randomly selected from the top five products with the highest number of reviews for each category. We also selected the top 10 actual opinions of the target product using the same methods described in Sections 4.4. We then evaluated whether the opinions retrieved by the proposed method or the baseline method were more similar to the actual opinions of the target product.

Automatic Evaluation We conducted an automatic evaluation of the similarity between the opinions retrieved by the proposed method from similar products and the actual opinions of the target product. We used ROUGE (ROUGE-1) [7], BLEU [8], and BERTScore [17]. These metrics are often used when evaluating the similarity between machine summary or machine translation and the original text. In our study, we considered the opinions in the target product as the original sentences. We used these metrics to evaluate the similarity of the opinions retrieved by the proposed method. We used the text queries show in Table 3. For the top 10 opinions retrieved by both the proposed method and the baseline method, we calculated the maximum similarity for each opinion against the top 10 actual opinions of the target product. The average scores for the proposed method and the baseline method are shown in Table 6. In all evaluation metrics, the proposed method outperformed the baseline method. This suggests that the proposed method can retrieve opinions that are closer to actual opinions.

Manual Evaluation We also conducted a human evaluation of the similarity between the opinions retrieved by the proposed method and those of the target product. Three annotators from our university participated in this evaluation. We used the text queries show in Table 3. Similarly to the automatic evaluation, for the top 10 opinions retrieved by both the proposed method and the baseline method, the annotators judged which method retrieved opinions that were more similar to the actual opinions of the target product. The evaluation was conducted using a five-point Likert scale:

1. The proposed method is much closer to the opinions of the target product.

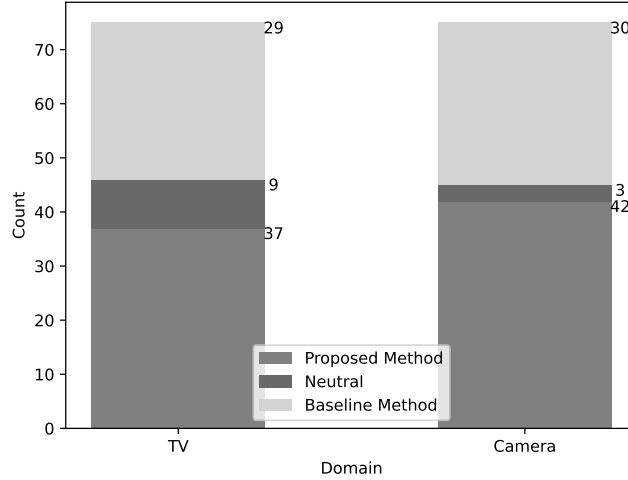


Fig. 3. Number of text queries judged by the annotators to be similar to the target product.

2. The proposed method is slightly closer to the opinions of the target product.
3. Neither method is close to the opinions of the target product.
4. The baseline method is slightly closer to the opinions of the target product.
5. The baseline method is much closer to the opinions of the target product.

To prevent bias, we presented the methods as “Method 1” and “Method 2”, and randomly switched the order of the proposed and baseline methods for each query, so that the annotators did not know which result came from which method. The breakdown of the results from the 25 queries evaluated by 3 human annotators is shown in Figure 3. In this figure, ratings of 1 or 2 are labeled as “closer to the proposed method”, a rating of 3 is labeled as “neutral”, and ratings of 4 or 5 are labeled as “closer to the baseline method”. In both product categories, the number of cases judged to be closer to the proposed method was higher. From the evaluation of the human point of view, we can say that the proposed method retrieves opinions that are closer to actual opinions.

6 Discussion

6.1 Analysis of Results

To discuss why the proposed method outperforms the baseline method, we analyze cases where the proposed method worked well. From the results in Table 6 and Figure 3, we can see that using opinions from similar products helps us retrieve opinions that are closer to the actual reviews. For example, for the text query “ease of viewing the LCD screen” in the camera category, both the similar

products and the target product had negative opinions such as “*the screen is a bit small*” and “*the viewing angle of the LCD monitor is narrow and hard to see*.” However, these opinions were not found in the opinions retrieved from random products. The random products had LCD monitors with many different screen sizes and quality levels, so the retrieved reviews were more general. In contrast, the similar products had LCD monitors with screen size and quality close to the target product, so the retrieved opinions were more relevant. Therefore, using an LLM to retrieve opinions from similar products can be a useful way to supplement opinions for products with few reviews.

However, in Table 6, the automatic evaluation shows only a small difference between the proposed method and the baseline method. There are two reasons for this. First, when the reviews mention the same attribute, they often use similar words. For example, the BERTScore similarity between the pair “*The LCD is hard to see*” and “*The LCD is easy to see*” is 0.893, and between “*The LCD is hard to see*” and “*The LCD is difficult to see*” is 0.899. Since the words used are similar, the similarity score becomes high, even when the meaning is different. This is one reason why there was little difference between the two methods. Second, it is possible that the same products were selected in both the proposed method and the baseline method. In fact, in both TV and camera categories, on average, at least one common product was included per text query. Because of this, similar or the same reviews might have been shown in both methods.

Also, as shown in Table 4, the effectiveness of identifying related attributes from text queries using the LLM is fairly high. On the other hand, Table 5 shows that the effectiveness of identifying the relationship between product reviews and attributes using the LLM is not very high. However, we believe that even if this effectiveness is low, it does not have a large negative effect on the overall performance of the proposed method. This is because the proposed method finally selects only the reviews that have high similarity to the query, and does not use unrelated reviews. Nevertheless, further improving the effectiveness of these two judgment tasks is important for making the proposed method more accurate overall. For example, we could improve the prompts used with the LLM or use another LLM to check and filter the output.

6.2 Limitation

In our study, we did not explore hyperparameters such as the number of similar products selected by the proposed method or the number of reviews shown in the ranking. Therefore, the evaluation may not be conducted in the best performance setting. We need to test which values are appropriate in the future. Moreover, our study was evaluated using only two product categories. Therefore, it is not clear whether the proposed method would also work well in other categories. Also, the amount of knowledge that each annotator had about each category may have affected the results. In the future, we need to test the method on more diverse product categories and increase the number of annotators to conduct a more reliable human evaluation.

7 Conclusion and Future Work

In our study, we propose a method using an LLM to supplement opinions on products with few reviews from reviews of similar products. First, when the user inputs a product and a question, the system uses the LLM to find the attributes that are related to the user’s question. Next, the system calculates the similarity of the attributes and find similar products. After that, by using the LLM, the system retrieves opinions from reviews of the similar products that are related to these attributes. Finally, the system ranks the opinions that were judged to be related to the attribute queries and shows them as the search results. We first conducted effectiveness evaluations for the parts where the LLM was used. For identifying the relationship between the text query and the attributes, the effectiveness was found to be fairly high. In contrast, the effectiveness of identifying the relationship between reviews and attributes was not very high, indicating that, to supplement reviews with an LLM, the accuracy of determining the review–attribute relationship must be improved. We then compared the similarity between the opinions retrieved by our method and the opinions about the target product. The result was slightly better than using randomly selected products.

In future work, we would like to build a question-answering system that generates evaluation information based on the opinions from similar products retrieved in our method. In our study, we were able to make the input more flexible compared to previous studies. However, the output is still limited to showing actual opinions, just like in past studies. Therefore, we aim to make the output more flexible by summarizing the collected opinions using an LLM and generating answers to user questions. Also, to make the system more practical, it will be important to improve the effectiveness of the LLM’s judgments. In addition, we need to show which attributes are similar to the text query to make the system more trustworthy.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP24K03228, and JP25K03229. We also used “Rakuten Dataset” (<https://rit.rakuten.com/data.release/>) provided by Rakuten Group, Inc. via IDR Dataset Service of National Institute of Informatics.

References

1. Adhikary, J., Alqudah, M., Udayashankar, A.P.: RevieWeaver: Weaving together review insights by leveraging LLMs and semantic similarity. In: Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track). pp. 431–448 (2025)
2. Du, N.n., Xu, L.t., Huang, H.: MSRM: A novel model to retrieve meaningful opinion sentences for new products. In: Proceedings of the 2017 International Conference on Mathematics, Modelling and Simulation Technologies and Applications. pp. 214–221 (2017)

3. Fujii, K., Nakamura, T., Loem, M., Iida, H., Ohi, M., Hattori, K., Shota, H., Mizuki, S., Yokota, R., Okazaki, N.: Continual pre-training for cross-lingual LLM adaptation: Enhancing japanese language capabilities. In: Proceedings of the 1st Conference on Language Modeling (2024)
4. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., et al: The llama 3 herd of models (2024), <https://arxiv.org/abs/2407.21783>
5. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 168–177 (2004)
6. Ishii, T., Shoji, Y., Yamamoto, T., Ohshima, H., Fujita, S., Dürst, M.J.: Generating fine-grained aspect names from movie review sentences using generative language model. In: Proceedings of the 25th International Conference on Information Integration and Web Intelligence. pp. 217–232 (2023)
7. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text summarization branches out. pp. 74–81 (2004)
8. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. pp. 311–318 (2002)
9. Park, D.H., Kim, H.D., Zhai, C., Guo, L.: Retrieval of relevant opinion sentences for new products. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 393–402 (2015)
10. Poria, S., Cambria, E., Gelbukh, A.: Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems* **108**, 42–49 (2016)
11. Pourgholamali, F., Kahani, M., Noorian, Z., Bagheri, E.: Learning product representations for generating reviews for cold products. *Knowledge-Based Systems* **228**, 107282 (2021)
12. Rakuten Group, Inc: Rakuten Ichiba data. Informatics Research Data Repository, National Institute of Informatics. (dataset). (2020), <https://doi.org/10.32130/idr.2.1>
13. Rozen, O., Carmel, D., Mejer, A., Mirkis, V., Ziser, Y.: Answering product-questions by utilizing questions from other contextually similar products. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 242–253 (2021)
14. United Nations Conference on Trade and Development: Business e-commerce sales and the role of online platforms. https://unctad.org/system/files/official-document/dtlecde2024d3_en.pdf (2024), accessed on March twenty-ninth, 2025
15. Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., Wei, F.: Multilingual E5 text embeddings: A technical report. arXiv preprint arXiv:2402.05672 (2024)
16. Wang, W., Pan, S.J., Dahlmeier, D., Xiao, X.: Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In: Proceedings of the 31st AAAI conference on artificial intelligence. vol. 31, pp. 3316–3322 (2017)
17. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: Evaluating text generation with BERT. In: Proceedings of the 8th International Conference on Learning Representations. pp. 1–43 (2020)