Expanding Aspect Queries into Review Sentence Fragments for Product Comparison via LLM-Generated Synthetic Reviews

Naito Yoshihara[1], Takehiro Yamamoto[1][0000−0003−0601−3139], and Yoshiyuki Shoji[1][0000−0002−7405−9270]

[1] Shizuoka University, Hamamatsu, Shizuoka 432–8011, Japan
yoshihara.naito.21@shizuoka.ac.jp, shojiy@inf.shizuoka.ac.jp
[2] University of Hyogo, Kobe, Hyogo 651–2197, Japan
t.yamamoto@sis.u-hyogo.ac.jp

Abstract. This paper proposes a method for retrieving diverse real-world user reviews that refer to a specific Aspect Query representing a user's information need. Given a short Aspect Query, such as "practicality," the system generates a variety of Sentence Fragment queries, *e.g.*, "*able for da*" to retrieve phrases such as "suitable for daily use" or "comfortable for daytime work." These Sentence Fragments act as wildcard-like queries and are particularly effective in languages like Japanese, where inflection and agglutinative structures make exact keyword matching challenging. To construct such fragments, we first use a large-scale language model (LLM) to generate a large number of synthetic Aspect Query–review sentence pairs. These pairs are filtered to retain only high-quality examples, which are subsequently used to fine-tune a lightweight local LLM. The fine-tuned model generates synthetic reviews for arbitrary Aspect Queries, from which Sentence Fragments that are frequent in the synthetic reviews but rare in general reviews are extracted and used as expanded queries. A user study on a real-world review dataset demonstrates that our method enables the retrieval of diverse reviews without compromising accuracy, effectively bridging the lexical gap between abstract Aspect Queries and concrete review expressions.

Keywords: Review Retrieval · Query Expansion · Wildcard.

1 Introduction

Online shopping platforms, such as Amazon and eBay, have made it easy for users to purchase products, resulting in a surge in user-generated reviews. However, reading through all reviews to find information relevant to one's needs has become increasingly complex.

Users often search for products based on specific aspect names, such as "daily use." However, such terms rarely appear verbatim in reviews. For example, a user might enter "daily use" as a query, but reviews seldom say "This is suitable for

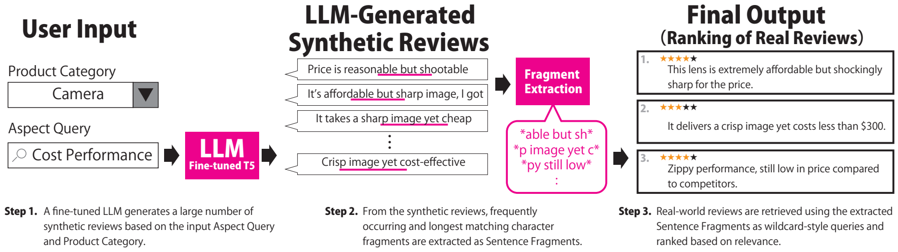
User Input
Product Category
Camera
Aspect Query
Cost Performance
LLM
Fine-tuned T5
LLM-Generated Synthetic Reviews
Price is reasonable but shootable
It's affordable but sharp image, I got
It takes a sharp image yet cheap
⋮
Crisp image yet cost-effective
Fragment Extraction
able but sh
p image yet c
py still low
:
Final Output
(Ranking of Real Reviews)
1. ★★★★★
This lens is extremely affordable but shockingly sharp for the price.
2. ★★★★★
It delivers a crisp image yet costs less than $300.
3. ★★★★★
Zippy performance, still low in price compared to competitors.
Step 1. A fine-tuned LLM generates a large number of synthetic reviews based on the input Aspect Query and Product Category.
Step 2. From the synthetic reviews, frequently occurring and longest matching character fragments are extracted as Sentence Fragments.
Step 3. Real-world reviews are retrieved using the extracted Sentence Fragments as wildcard-style queries and ranked based on relevance.

Fig. 1. Overview of the proposed method. Given a product category and Aspect Query, the system generates synthetic reviews using a fine-tuned LLM. Character-level fragments frequently found in the generated reviews are extracted and used as wildcard-style queries to retrieve and rank real reviews.

daily use." Instead, they may state, "They don't get in the way when worn," implicitly conveying the same concept. As a result, conventional keyword matching fails to retrieve such relevant reviews.

This paper proposes a method for retrieving diverse real-world review sentences that refer to a given Aspect Query, a short keyword-like term representing a specific evaluation aspect, which is often expressed through varied paraphrased expressions in reviews. Figure 1 shows an overview of our algorithm's behavior. Our method leverages large language models (LLMs) to generate synthetic reviews that are likely to implicitly mention the specified aspect. From these synthetic reviews, our method extracts frequently appearing character substrings, called "Sentence Fragments," that serve as expanded queries to retrieve actual user reviews.

To generate high-quality synthetic reviews, we adopt a two-stage LLM framework. First, a general-purpose large-scale language model (*e.g.*, ChatGPT) is used to identify plausible aspect terms mentioned in real reviews, creating aspect-review training pairs through majority voting. Then, a lightweight local LLM (*e.g.*, OpenCALM) is fine-tuned on these pairs to generate synthetic reviews that reflect both the given Aspect Query and the product category.

By extracting Sentence Fragments that are frequent in the synthetic reviews but rare in general reviews, we construct expanded queries that help uncover a wide range of semantically related real-world reviews, even when the original aspect term is not explicitly mentioned.

Specifically, we extract Sentence Fragments that match the most prolonged and most frequent character sequences across the generated reviews. For instance, from the Aspect Query "daily use", we may obtain Sentence Fragments like "*able for da*" or "*ent during r*." These wildcard-like fragments can retrieve a wide range of expressions in real reviews, such as "suitable for daily use," "comfortable for daytime work," "consistent during routine work," or "convenient during regular use."

This technique is especially effective in agglutinative languages, such as Japanese, where words frequently undergo inflection, and semantically similar expressions often share overlapping character sequences rather than identical word forms (section 3.1 provides a detailed description). In such cases, conventional keyword matching often fails, while our fragment-based approach remains robust. This makes our method particularly well-suited to information retrieval tasks in morphologically rich languages.

To evaluate whether the proposed method can retrieve a diverse set of review sentences relevant to a given Aspect Query, we conducted a subject experiment. We developed an application that searches a large-scale review dataset for real-world sentences mentioning the specified Aspect Query. Participants were asked to assess the relevance and quality of the sentences retrieved by our method and a baseline approach. The results demonstrate that, by leveraging large language models for fine-tuning, query expansion, and ranking, the proposed method can effectively retrieve diverse and relevant review sentences corresponding to user-specified Aspect Queries.

2 Related Work

This study proposes a method for expanding queries using large language models (LLMs) to search user reviews. This section reviews prior work on review-based retrieval, query expansion, and LLM-assisted search, and positions our approach within this context.

2.1 Review-Based Retrieval and Summarization

In response to the rapid growth of user-generated reviews, numerous studies have explored methods for summarizing, searching, and utilizing review content. For example, Hu *et al.* [9] proposed a method to extract product features from a large number of reviews and generate summaries organized around those features. Fang *et al.* [6] proposed a method for sentiment explanation ranking, based on information-content ranking of sentences and structured sentiment analysis. Yang *et al.* [16] developed a method to extract product features from opinionated sentences retrieved via Online Customer Reviews (OCR), presenting them to users to assist in product selection.

Other studies have used reviews for recommendation purposes. Zheng *et al.* [17], for example, proposed a deep learning-based product recommendation method that jointly learns user and item features extracted from reviews. These studies demonstrate the usefulness of user reviews for supporting product decision-making and building recommender systems.

In contrast, our work focuses on generating synthetic reviews from Aspect Query inputs, and extracting partial expressions from them to serve as expanded queries.

2.2 Query Expansion

Query expansion (QE) has long been studied in information retrieval (IR) as a means to alleviate the vocabulary mismatch problem between user queries and relevant documents. Early work by Maron *et al.* [11] laid the foundation for this area, followed by a variety of dictionary- and co-occurrence-based approaches [5, 8, 10, 12]. While these classical methods introduced essential techniques for term expansion, they often showed inconsistent performance improvements across tasks.

Later studies introduced more data-driven and discriminative techniques. For example, Cui *et al.* [3] utilized query logs to extract term associations and demonstrated effectiveness using the TREC benchmark [7]. Other works employed co-occurrence statistics either from retrieved documents [1] or from external corpora [2], achieving better precision in selected retrieval tasks. However, such approaches often rely heavily on domain-specific logs or structured corpora and may fail to generalize in low-resource or open-ended settings.

More recently, the emergence of large language models (LLMs) has enabled the generation of expanded queries. Wenjun *et al.* [13], for instance, trained an LLM to generate rewrite candidates based on query relationships and relative rankings, showing improvements in downstream engagement metrics such as purchases and site visits. These studies demonstrate that generative models can capture richer semantic associations compared to traditional QE techniques.

Our proposed method follows this generative direction, focusing specifically on bridging the gap between abstract Aspect Query queries and concrete user reviews by generating Sentence Fragments. Unlike previous work, which primarily expands queries at the word or phrase level, we aim to extract and utilize character-level fragments commonly found in generated pseudo-reviews, enabling finer-grained matching in review retrieval scenarios.

2.3 LLM-Based Retrieval

Large Language Models (LLMs), originally developed for natural language understanding and generation tasks, have recently been adopted in information retrieval (IR). These models have been used to enhance various IR components, such as query rewriting, retrieval, re-ranking, and reading comprehension.

Our work is most closely related to query rewriting using LLMs. Dai *et al.* [4] proposed a method to train retrieval models using synthetic queries generated by LLMs in a few-shot setting. Their model produced queries for unlabeled documents, creating effective training data that matched the quality of human-labeled datasets. Similarly, Wang *et al.* [15] generated document-query pairs using LLMs to improve retrievers, showing significant performance gains over traditional methods.

While prior work typically focuses on generating queries or training data at the document level, our approach instead leverages LLMs to generate pseudo-reviews based on Aspect Query queries and extracts recurring Sentence Fragments as wildcard-style expansion queries. This fragment-level generative expansion enables precise and diverse review retrieval that is difficult to achieve with conventional rewriting approaches.

3 Query Expansion and Review Retrieval via Aspect Query-Based Synthetic Review Generation

This section describes our method for retrieving diverse real-world user reviews that refer to a given Aspect Query—a short keyword-like term representing a specific evaluation aspect, such as "sound quality" or "walking suitability." In actual reviews, such queries are rarely stated verbatim; instead, they are often paraphrased through expressions like "the bass is punchy" or "they don't fall out when jogging." We refer to such abstract but user-intended expressions as Aspect Queries throughout this paper.

Our approach addresses the mismatch between abstract Aspect Queries and their varied surface realizations in user reviews by combining three stages:

- (1) generating diverse synthetic reviews using two types of LLMs,
- (2) extracting representative Sentence Fragments as expanded queries, and
- (3) ranking real reviews based on these Sentence Fragments.

We leverage large language models (LLMs) to generate synthetic reviews from an input Aspect Query and product category, from which we extract character-level substrings (called Sentence Fragments) that frequently appear in the synthetic reviews but are rare in general reviews. These Sentence Fragments serve as expanded queries to retrieve diverse real reviews implicitly referring to the original aspect.

Before detailing the technical components, Section 3.1 explains the linguistic motivation behind our fragment-based query expansion strategy, particularly its importance in Japanese as an agglutinative language. We then describe the method step by step, including LLM fine-tuning, Sentence Fragment query generation, and review ranking.

3.1 Linguistic Motivation: Why Sentence Fragment

Japanese is an agglutinative language, where word endings (suffixes) carry grammatical meaning and often appear in semantically similar expressions. This linguistic trait makes fragment-based query expansion particularly effective.

For example, given the aspect term "cost performance," users may not write it explicitly in reviews. Instead, they may use expressions like:

- よく写る割に安い(Shoots well, yet inexpensive),
- 撮れる割に安価だ(Can take pictures, but affordable).

These expressions share the Sentence Fragments "*る割に安*", which reflects the underlying intent. By extracting such fragments from synthetic reviews, our method captures diverse real reviews that traditional keyword-based methods often miss. While partially useful in English (*e.g.*, "*tion is g*"), this technique is especially effective in agglutinative languages, *e.g.*, Korean, or Turkish.

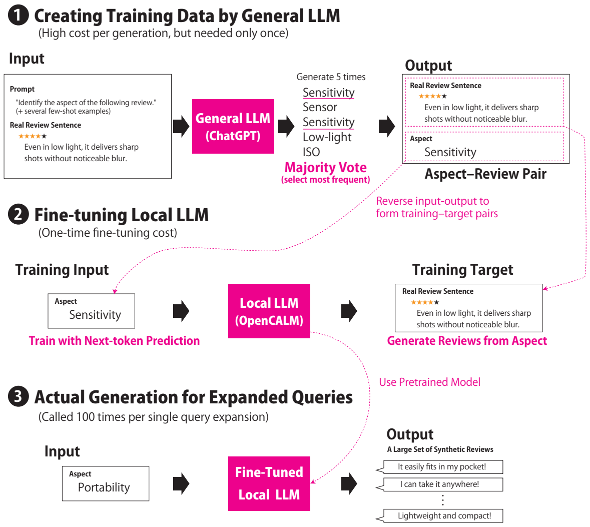
1 Creating Training Data by General LLM
(High cost per generation, but needed only once)
Input
Prompt
"Identify the aspect of the following review."
(+ several few-shot examples)
Real Review Sentence
★★★★★
Even in low light, it delivers sharp shots without noticeable blur.
General LLM (ChatGPT)
Generate 5 times
Sensitivity
Sensor
Sensitivity
Low-light
ISO
Majority Vote
(select most frequent)
Output
Real Review Sentence
★★★★★
Even in low light, it delivers sharp shots without noticeable blur.
Aspect
Sensitivity
Aspect–Review Pair
Reverse input-output to form training–target pairs
2 Fine-tuning Local LLM
(One-time fine-tuning cost)
Training Input
Aspect
Sensitivity
Train with Next-token Prediction
Local LLM (OpenCALM)
Training Target
Real Review Sentence
★★★★★
Even in low light, it delivers sharp shots without noticeable blur.
Generate Reviews from Aspect
Use Pretrained Model
3 Actual Generation for Expanded Queries
(Called 100 times per single query expansion)
Input
Aspect
Portability
Fine-Tuned Local LLM
Output
A Large Set of Synthetic Reviews
It easily fits in my pocket!
I can take it anywhere!
⋮
Lightweight and compact!

Fig. 2. Asymmetric pipeline combining a general LLM for one-time data construction and a fine-tuned local LLM for efficient, repeated review generation to expand Aspect Queries.

3.2 Generating Synthetic Reviews with Two Types of LLMs

To retrieve diverse real-world reviews that mention a given Aspect Query, we aim to expand the query into natural Sentence Fragments commonly found in actual reviews. To discover such Sentence Fragments, we first generate a large number of diverse and plausible synthetic reviews that reflect the input aspect.

To achieve this efficiently and cost-effectively, we design an asymmetric pipeline that leverages two types of LLMs based on task characteristics. Figure 2 illustrates an overview of our pipeline for synthetic review generation. The reverse task, *i.e.*, inferring Aspects from existing reviews, is generic and needs to be performed only once, making it suitable for a general-purpose, large-scale LLM (*e.g.*, GPT-4). In contrast, the forward task, *i.e.*, generating reviews from Aspects, is more specific and must be executed repeatedly at inference time. Therefore, we fine-tune a smaller, local LLM for this purpose.

As the first step, we constructed training data using a general-purpose LLM. To fine-tune a local review generation model, we require training samples consisting of an Aspect, a product category, and a review sentence. While product categories and real review texts are available in existing datasets, the Aspect information (*i.e.*, what the review sentence is referring to) is not provided. Therefore, we used a general-purpose LLM to annotate each review with a plausible Aspect. Specifically, we input a review into the LLM (*e.g.*, GPT-4) and prompted it to generate Aspect candidates.

To annotate each review with a reliable Aspect, we prompted the LLM five times per input. This redundancy addresses the variability in LLM outputs caused by sampling randomness and the lack of confidence-based ranking. For example, when given the review "I love the image quality," the LLM may generate "image quality" as a candidate Aspect. We then selected the most frequent candidate among the five outputs by majority vote. If all five were different, we adopted the first output as a fallback. This process enabled the automatic construction of a review–Aspect dataset suitable for fine-tuning.

As the second step, we fine-tuned the local LLM using the constructed dataset. Each training instance was formatted with a prompt template suitable for autoregressive generation, including a product category, an Aspect, and a corresponding review. The input format was: "`Category: [product category]`, `Aspect: [aspect]`, `Review: [review text]`." For instance, the following prompt–output pair was used for training: "`Category: Camera`, `Aspect: Portability`, `Review: It's so light and easy to carry around on trips`."

As the third step, we used the fine-tuned local LLM to generate diverse review sentences for any given product category and Aspect. Although the model only predicts the next text token, the task effectively becomes generating a review reflecting the given Aspect, since we provide a truncated prompt like "`Category: Camera, Aspect: Cost Performance, Review:`" (the prompt ends abruptly with a colon), which the model then completes with a plausible sentence such as "`The image quality is excellent for the price`."

3.3 Extracting Sentence Fragments from Synthetic Reviews for Query Expansion

After generating a diverse set of synthetic reviews that reflect a given Aspect, we aim to extract Sentence Fragments that can serve as expanded queries for retrieving real-world reviews referring to the same Aspect. To ensure the extracted Sentence Fragments are truly representative and useful for retrieval, we define three requirements that such Sentence Fragments must satisfy:

1. **Intra-Category Frequency**: The fragment should appear frequently in synthetic reviews generated for the given category and Aspect,
2. **Descriptive Length**: The fragment should be sufficiently long to capture meaningful expressions, and
3. **Category- and Domain-Specificity**: The fragment should rarely appear in general reviews or in reviews from the same category overall.

For example, given the input `Category: Camera, Aspect: Cost Performance`, synthetic reviews often contain expressions such as "The image is sharp yet cheap" or "Crisp resolution for the cost." Among these, a Sentence Fragments like `*p image yet c*` emerges frequently across multiple generated reviews, satisfying the first criterion of high intra-category frequency. This Sentence Fragments is also longer and more specific than a single keyword such as "image," which would match too many irrelevant sentences, thereby fulfilling the second criterion of descriptive length. Finally, it is unlikely to appear in generic camera-related reviews, since terms like "camera body" or "very good" frequently occur regardless of cost performance and would not serve as effective queries, thereby meeting the third criterion of category- and domain-specificity. Here, the suffix `p` captures adjectives such as *sharp*, *crisp*, or *top*, while the prefix `c` matches *cost* or *cheap*, illustrating how the Sentence Fragments effectively bridges both quality and cost aspects (especially in agglutinative languages like Japanese, where partial matches are particularly effective).

To select fragments that satisfy the above criteria, we compute a Sentence Fragments score inspired by the concepts of TF–IDF and longest-common substring matching. We refer to this score as the *Longest-Frequent Occurrence* (LFO). The score is calculated based on the frequency of each n-gram across three sets of review texts, combined with the character length of the fragment.

The three review sets are defined as follows: (1) the set of synthetic reviews generated from the given query condition (category and Aspect), (2) the set of real reviews from the same category, and (3) the set of general reviews without any specific category or Aspect condition. For each n-gram within these sets, we count its frequency to evaluate how representative and distinctive it is. In our experiments, n ranges from 2 to 10 when enumerating and counting possible Sentence Fragments.

Given a query condition $q =$ (Product Category,Aspect Query), the Longest-Frequent Occurrence (LFO) score LFO(s) of a Sentence Fragments s is defined as:

$$\mathrm{LFO}(s) = \left(\frac{\sum\limits_{r \in R(q)} \mathbb{I}[s \in S_r]}{\sum\limits_{r \in R_{\text{common}}} \mathbb{I}[s \in S_r]} \right) \times \left(\frac{\sum\limits_{r \in R(q)} \mathbb{I}[s \in S_r]}{\sum\limits_{r \in R_{\text{category}}} \mathbb{I}[s \in S_r]} \right) \times |s| \tag{1}$$

Here, $R(q)$ denotes the set of synthetic reviews generated under the given query condition, R_{category} the set of reviews from the same product category, R_{common} the set of general reviews without category or Aspect constraints, and $|s|$ the character length of the Sentence Fragment s.

Sentence Fragments with high LFO scores are selected as the expanded queries. By this definition, Sentence Fragments that frequently appear in general or category-specific reviews receive lower LFO values, while those that frequently appear in synthetic reviews generated for the query condition receive higher LFO values.

3.4 Ranking Real-World Review Sentences

The final step is to rank the real reviews retrieved by the expanded queries so that the most representative reviews are presented. A fragment is considered more useful if it is both *indicative of the target Aspect* (high LFO score) and *widely supported by real reviews* (high match count).

For each expanded query s, we count the number of matched reviews $num(s)$ and define the final score as:

$$s_{\text{final}}(s) = \mathrm{LFO}(s) \times \mathrm{num}(s). \tag{2}$$

Table 1. Product categories and Aspect Queries used in the evaluation

Product Category	Aspect Query
Earphones	Deep Bass
	For Walking
Cameras	Cost Performance
	Retro
Laptops	Portability
	Editing
Televisions	Slim Design
	Solo Living
Wristwatches	Stylish
	Business

This balances the quality of the Sentence Fragments (via LFO) with its coverage (via $num(s)$). From the reviews retrieved by each Sentence Fragments, we select only the most representative one to avoid redundancy. We apply LexRank, treating each review as a node and cosine similarity as edge weight, and rank reviews based on the PageRank-derived importance score.

In summary, this ranking combines LFO-based query relevance with LexRank-based representativeness to produce a final, diverse set of reviews.

4 Evaluation

To verify whether our method can effectively retrieve a diverse set of real reviews referring to a given Aspect, we conducted a user study using a real-world review dataset. We implemented our system, prepared baseline and ablation methods for comparison, and evaluated the search results obtained under various query conditions. This section describes the experimental setup, including the queries and dataset, as well as the user study and its results.

4.1 Dataset

We first constructed the dataset and defined the query conditions for the evaluation. We used a real-world product review dataset from Rakuten Ichiba [14], which includes review texts, product names, and category information. For the experiments, we selected five product categories with a large number of reviews, such as earphones, cameras, laptops, televisions, and wristwatches. We used the top 500 products with the most reviews in each category.

For each category, we prepared two Aspect Queries, resulting in a total of ten query conditions. The query–category pairs used in the evaluation are listed in Table 1.

4.2 Comparison Methods

To verify the effectiveness of our approach and to identify which components contribute most to its performance, we prepared both ablation variants of our method and baseline approaches. Note that the baseline methods do not include any diversification of the retrieved results; therefore, while they may achieve high precision, the top-ranked reviews are often filled with near-duplicate sentences.

We compared the following seven methods:

- **Proposed**: Our full method, which performs query expansion using synthetic reviews and ranks fragments based on both LFO score and the number of retrieved reviews,
- **Query Only**: Ranks Sentence Fragments solely by their LFO score, ignoring the retrieval count,
- **Count Only**: Ranks Sentence Fragments solely by the number of reviews retrieved by each query,
- **BM25**: Uses the input Aspect Query as a keyword to retrieve reviews based on BM25 ranking,
- **Co-occurrence**: A classic co-occurrence-based relevance feedback method [1],
- **Dense Query Retrieval**: Retrieves reviews by encoding the input Aspect Query into a vector and ranking by cosine similarity, and
- **Dense Review Retrieval**: Retrieves reviews based on the average vector of multiple synthetic reviews generated for the input condition, ranked by cosine similarity.

For each method, we retrieved and ranked reviews, and all results were evaluated manually by human assessors.

4.3 Implementation

We implemented a system that generates synthetic reviews, retrieves real reviews, and ranks them according to our method. For training data, we used GPT-3.5 Turbo with few-shot prompting to extract Aspect terms, sampling 10 examples from 200 manually prepared review–aspect pairs. For each review, five candidate aspects were generated, and the majority vote was used to select the final label.

Synthetic reviews were generated by fine-tuning the OpenCALM-3B model[3] with the constructed dataset. For dense retrieval baselines and representative review selection, we used Sentence-BERT[4] to embed review texts and applied cosine similarity with LexRank to identify the most representative review for each query.

[3] Hugging Face OpenCALM 3b: `https://huggingface.co/cyberagent/open-calm-3b`

[4] Hugging Face sentence-bert-base-ja-mean-tokens-v2: `https://huggingface.co/sonoisa/sentence-bert-base-ja-mean-tokens-v2`

Table 2. Average scores of each method on evaluation metrics (** p<0.01, * p<0.05 compared with Proposed; ILAD excluded).

Method	p@1	p@5	p@10	p@50	p@100	nDCG@100	ILAD
Proposed	1.00	1.00	1.00	0.91	0.61	0.87	0.53
Query Only	1.00	1.00	1.00	0.93	*0.63	*0.70	0.51
Count Only	0.86	0.94	0.92	0.86	**0.59	**0.72	0.57
BM25	1.00	1.00	1.00	*1.00	**0.92	*0.94	0.44
Co-occurrence	0.86	*0.20	**0.10	**0.02	**0.08	**0.40	0.63
Dense Query	1.00	0.94	0.94	0.91	**0.54	*0.83	0.39
Dense Review	1.00	1.00	1.00	0.93	*0.58	*0.83	0.27

4.4 Evaluation Metrics and Subject Experiment

To assess the usefulness of the proposed method in retrieving reviews that support purchase decisions, we conducted a user study with real-world product reviews. We evaluated the top-ranked reviews retrieved by each method under the 10 query conditions defined in Section 4.1 (two Aspect Queries for each of the five product categories). For each method, the top 100 reviews were retrieved per query, resulting in 7,000 reviews for evaluation.

In our method, only the single most representative review per expanded query is presented to avoid redundancy, whereas BM25 presents all top-ranked reviews directly. Two participants independently rated all retrieved reviews, and the average score was used for evaluation.

As the evaluation criteria, participants rated each review on a five-point scale (1 = not useful, 5 = highly useful) based on how informative it would be when searching for a product with the given Aspect. If a review did not mention the Aspect at all, it was assigned a score of 1. We computed Precision@k and nDCG using these relevance scores.

To assess diversity, we also calculated Intra-List Average Distance (ILAD), which measures the average pairwise distance among retrieved items:

$$\mathrm{ILAD}(R_u) = \frac{1}{|R_u|} \sum_{i \in R_u} \frac{1}{|R_u| - 1} \sum_{j \in R_u \setminus \{i\}} d(i, j), \tag{3}$$

where R_u is the set of retrieved reviews and $d(i, j)$ is the cosine distance between reviews i and j, computed on Sentence-BERT embeddings.

4.5 Results

Table 2 summarizes the average scores of all methods across the 10 query conditions, including Precision@k, nDCG@100, and ILAD. Statistical significance was tested against the Proposed method using a t-test, with * and ** indicating p<0.05 and p<0.01, respectively.

Overall, the Proposed method achieved consistently high Precision and nDCG, significantly outperforming Query Only, Count Only, Co-occurrence, and Dense-based methods on most metrics. While BM25 showed slightly higher Precision in some ranks, it lacks diversification and tended to retrieve redundant reviews. In terms of diversity (ILAD), the Proposed method ranked third, following Co-occurrence and Count Only, but still outperformed the remaining baselines.

5 Discussion

This section discusses the experimental findings from three perspectives, such as the trade-off between precision and diversity compared to conventional baselines, the contribution of our ranking and query expansion strategies, and the stability of performance across different types of queries.

We first examine why the Proposed method occasionally falls behind the baselines in terms of precision and diversity. The slightly lower precision compared to BM25 is largely due to our diversification strategy. To avoid redundancy, we used the LexRank algorithm to select a single representative review for each expanded query, whereas BM25 directly ranks all reviews containing the Aspect term without any filtering. As a result, the reviews evaluated under BM25 almost always included the input Aspect term verbatim, leading to higher precision scores.

Regarding diversity, the Proposed method showed lower ILAD compared to the classical co-occurrence method. This is because the co-occurrence approach produces low-quality expanded queries, such as "コスパ (cos-pa; Abbreviation of Cost Performance)," and even trivial particles like "です (desu)" or "ます (masu)," which are Japanese polite sentence endings similar to adding "is" or "does" at the end of an English sentence, polite but semantically meaningless. Such weak query terms cause the retrieved reviews to be less similar to each other, inflating the ILAD score.

These observations highlight a trade-off between precision and diversity: BM25 excels at precision by focusing on exact term matches but lacks diversity, while the co-occurrence method achieves higher diversity by retrieving loosely related reviews but sacrifices precision. In contrast, our Proposed method strikes a better balance, retrieving reviews that are both relevant and diverse.

We next discuss the impact of our ranking strategy. In the experiment, we compared three ranking methods: the full Proposed method combining LFO scores with the number of retrieved reviews, a variant using only LFO scores, and another variant using only the number of retrieved reviews.

The LFO-only ranking achieved the highest precision among the three. This is because incorporating the number of retrieved reviews sometimes prioritizes popularity over direct relevance to the Aspect, allowing reviews that do not explicitly mention the Aspect to rise in the ranking. This effect is even more pronounced in the Count-only variant, which shows the lowest precision of all.

On the other hand, the Count-only ranking achieved the highest diversity score (ILAD), since prioritizing retrieval frequency leads to a less consistent set of

Table 3. Examples of reviews retrieved by the Proposed method (translated from Japanese)

Query Condition	Retrieved Review
Earphones, Deep Bass	It might be realistic playback in some sense, but bass lovers may find it lacking.
Earphones, For Walking	Suitable for sports and designed to stay in place, so I look forward to using it outdoors.
Cameras, Cost Performance	For this price, the performance and usability are excellent.
Cameras, Retro	A design that evokes the good old days of film cameras.
Laptops, Portability	I wanted something light enough to carry around on business trips.
Laptops, Editing	I was looking for a device that could handle fast video encoding, and I found this one.
Televisions, Slim Design	It doesn't take up much space, the price is reasonable, and the setup was simple.
Televisions, Solo Living	Affordable and compact—just the right size for a small room.
Wristwatches, Stylish	I bought this as a gift; it's simple but very stylish.
Wristwatches, Business	It matches my suits perfectly, so I'm happy with the purchase.

reviews. In contrast, LFO-only ranking, which tightly focuses on Aspect-related fragments, exhibited the lowest diversity. These results illustrate the trade-off between precision and diversity depending on which signal is emphasized.

The full Proposed method achieves a balance by combining both signals. It ranked highest in overall user evaluation, as integrating retrieval counts helps surface reviews that users frequently mention, while LFO ensures that the reviews remain Aspect-relevant. As a result, the Proposed method can retrieve a diverse yet coherent set of reviews that better supports user decision-making than either signal alone.

The Proposed method enabled the retrieval of a wider variety of reviews than the baselines. Table 3 shows examples of reviews retrieved by our approach. For each query condition, the retrieved reviews often used expressions that were semantically related but did not directly repeat the Aspect term.

For instance, for the query (Earphones, Deep Bass), the extracted Sentence Fragments included variations like "with deep bass" or "rich low tones," as well as related terms such as "powerful bass" or "punchy sound," which do not explicitly contain the original keyword. Similarly, for (Cameras, Retro), Sentence Fragments like "retro feel," "classic design," or "vintage style" were discovered. For (Laptops, Portability), Sentence Fragments such as "light enough to carry" or "easy to take on business trips" appeared, while for (Wristwatches, Business), Sentence Fragments like "suitable for business use" or "matches formal suits" were identified.

Moreover, the number of relevant reviews retrieved increased significantly. For example, under the query (Wristwatches, Business), BM25 retrieved 34 valid reviews, Dense Query retrieved 113, and Dense Review retrieved 84, while the Proposed method retrieved 198. These results confirm that our query expansion strategy is effective for discovering a diverse set of relevant reviews. Finally, we compare the results for general queries versus more specific queries, as summarized in Table 4. Among the baselines, Dense Review showed a noticeable drop in precision from 0.67 (general) to 0.50 (specific), indicating its sensitivity to query specificity. In contrast, the Proposed method maintained stable precision—0.60 for general and 0.62 for specific queries—demonstrating robustness regardless of query type.

Table 4. Average evaluation scores for general vs. specific queries (** p<0.01, * p<0.05 compared with Proposed; ILAD excluded).

Method	General Queries			Specific Queries		
	p@100	nDCG@100	ILAD	p@100	nDCG@100	ILAD
Proposed	0.60	0.86	0.53	0.62	0.88	0.53
Query Only	*0.63	*0.91	0.51	0.63	*0.61	0.51
Count Only	*0.57	*0.76	0.56	*0.59	*0.57	0.57
BM25	**0.81	*0.99	0.38	**0.87	*0.90	0.50
Co-occurrence	**0.01	**0.39	0.61	**0.07	**0.29	0.65
Dense Query	*0.49	*0.91	0.41	*0.59	*0.75	0.38
Dense Review	*0.67	*0.85	0.24	*0.50	*0.81	0.30

Overall, these findings confirm that our method consistently achieves a balanced combination of precision and diversity, successfully retrieving a broad range of relevant reviews that other approaches fail to capture.

6 Conclusion and Future Work

In summary, our main contributions are as follows:

1. We proposed a novel query expansion method that extracts Sentence Fragments from LLM-generated synthetic reviews to bridge the gap between abstract Aspect Queries and the diverse expressions found in real reviews.
2. We introduced an asymmetric generation pipeline that leverages a large-scale LLM for data construction and a fine-tuned local LLM for cost-effective and diverse review generation.
3. We designed a ranking strategy that combines LFO (Longest Frequent Occurrence) scores with retrieval frequency, achieving a favorable balance between precision and diversity, as demonstrated through user evaluation.

The evaluation results confirmed that our approach retrieves a wide range of aspect-relevant reviews that better support user decision-making compared to conventional baselines.

In future work, we plan to conduct larger-scale user studies and explore more advanced strategies for Sentence Fragments selection and ranking. We also aim to refine synthetic review generation by better controlling content diversity and specificity, further improving the coverage and practical usefulness of retrieved reviews.

Acknowledgements

This work was partially supported by JSPS KAKENHI under Grant Numbers 25K03228, 25K03229, and 24K03228. We used "Rakuten Dataset" [14] provided by Rakuten Group, Inc. via IDR Dataset Service of National Institute of Informatics.

References

1. Buckley, C., Salton, G., Allan, J., Singhal, A.: Automatic query expansion using smart: Trec 3. NIST special publication sp pp. 69–69 (1995)
2. Carpineto, C., de Mori, R., Romano, G., Bigi, B.: An information-theoretic approach to automatic query expansion. ACM Trans. Inf. Syst. **19**(1), 1–27 (2001)
3. Cui, H., Wen, J.R., Nie, J.Y., Ma, W.Y.: Probabilistic query expansion using query logs. In: Proc. of the 11th International Conference on World Wide Web. p. 325–332. ACM (2002)
4. Dai, Z., Zhao, V., Ma, J., Luan, Y., Ni, J., Lu, J., Bakalov, A., Guu, K., Hall, K.B., Chang, M.W.: Promptagator: Few-shot dense retrieval from 8 examples. ArXiv **abs/2209.11755** (2022)
5. Doszkocs, T.: AID, an Associative Interactive Dictionary for Online Bibliographic Searching. University Microfilms (1982)
6. Fang, L., Qian, Q., Huang, M., Zhu, X.: Ranking sentiment explanations for review summarization using dual decomposition. In: Proc. of the 23rd ACM International Conference on Conference on Information and Knowledge Management. p. 1931–1934. ACM (2014)
7. Harman, D.K.: The first text retrieval conference (TREC-1), vol. 500. US Department of Commerce, National Institute of Standards and Technology (1993)
8. Harper, D.J., Van Rijsbergen, C.J.: An evaluation of feedback in document retrieval using co-occurrence data. Journal of documentation **34**(3), 189–216 (1978)
9. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proc. of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 168–177. KDD '04, ACM (2004)
10. Lesk, M.E.: Word-word associations in document retrieval systems. American documentation **20**(1), 27–38 (1969)
11. Maron, M.E., Kuhns, J.L.: On relevance, probabilistic indexing and information retrieval. J. ACM **7**(3), 216–244 (1960)
12. Minker, J., Wilson, G.A., Zimmerman, B.H.: An evaluation of query expansion by the addition of clustered terms for a document retrieval system. Information Storage and Retrieval **8**(6), 329–348 (1972)
13. Peng, W., Li, G., Jiang, Y., Wang, Z., Ou, D., Zeng, X., Xu, D., Xu, T., Chen, E.: Large language model based long-tail query rewriting in taobao search. In: Companion Proc. of the ACM Web Conference 2024. p. 20–28. ACM (2024)
14. Rakuten Group, Inc.: Rakuten dataset (2020), https://doi.org/10.32130/idr.2.1, https://rit.rakuten.com/data_release/
15. Wang, L., Yang, N., Wei, F.: Query2doc: Query expansion with large language models. In: Proc. of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 9414–9423. ACL (2023)
16. Yang, C.S., Wei, C.P., Yang, C.C.: Extracting customer knowledge from online consumer reviews: a collaborative-filtering-based opinion sentence identification approach. In: Proc. of the 11th International Conference on Electronic Commerce. p. 64–71. ACM (2009)
17. Zheng, L., Noroozi, V., Yu, P.S.: Joint deep modeling of users and items using reviews for recommendation. In: Proc. of the Tenth ACM International Conference on Web Search and Data Mining. pp. 425–434. ACM (2017)