QQSUM: A Novel Task and Model of Quantitative Query-Focused Summarization for Review-based Product Question Answering

Anonymous ACL submission

Abstract

Review-based Product Question Answering (PQA) allows e-commerce platforms to automatically address customer queries by leveraging insights from user reviews. However, existing PQA systems generate answers with only a single perspective, failing to capture the diversity of customer opinions. In this paper we introduce a novel task **Quantitative Query-Focused Summarization (QQSUM)**, which aims to summarize diverse customer opinions into representative Key Points (KPs) and quantify their prevalence to effectively answer user queries. While Retrieval-Augmented Generation (RAG) shows promise for PQA, its generated answers still fall short of capturing the full diversity of viewpoints. To tackle this challenge, our model **QQSUM-RAG**, which extends RAG, employs few-shot learning to jointly train a KP-oriented retriever and a KP summary generator, enabling KP-based summaries that capture diverse and representative opinions. Experimental results demonstrate that QQSUM-RAG achieves superior performance compared to state-of-the-art RAG baselines in both textual quality and quantification accuracy of opinions. Our source code is available at: https://anonymous.4open.science/r/QQSUMM-A233

1 Introduction

With the rapid expansion of e-commerce, consumers increasingly rely on product reviews to inform their purchasing decisions. Automatic review-based product question answering (PQA) systems have emerged, leveraging user reviews to provide immediate responses on e-commerce Q&A platforms (McAuley and Yang, 2016; Gupta et al., 2019). However, current PQA systems face a key limitation: they typically generate a single answer (Gupta et al., 2019), overlooking the fact that many subjective e-commerce queries require answers that reflect diverse viewpoints. For example, when comparing camera lenses (Figure 1), some shoppers prioritize versatility and affordability, while others focus on image quality and speed. Recent PQA approaches aim to improve answer quality using retrieval-augmented generation (RAG). These systems first retrieve reviews relevant to the query and then use them as context for large language models (LLMs) to generate answers. Yet, LLMs often struggle to present multifaceted perspectives (Sorensen et al., 2024), leading to answers that primarily reflect dominant opinions from the retrieved reviews (Deng et al., 2020, 2023).

Separately, opinion summarization has made progress through Key Point Analysis (KPA), which summarizes reviews into concise, representative statements called key points (KPs) while also quantifying their prevalence (Bar-Haim et al., 2020a,b, 2021; Tang et al., 2024a,b). However, these KPA methods focus on general summarization rather than answering specific queries. For tasks like product comparison, summarization must incorporate only query-focused KPs, making general KPA approaches insufficient for PQA.

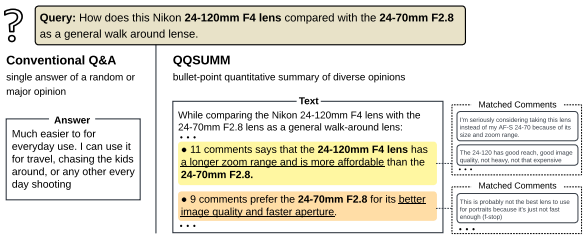

Figure 1: Comparison of conventional Q&A and QQSUM. More details of QQSUM output are in Table 10.

In this paper, we introduce a novel task Quantitative Query-Focused Summarization (QQSUM), which generates comprehensive answers containing diverse KPs along with their quantified relative importance (Figure 1). Our solution, QQSUM-RAG, extends the RAG framework by integrating KP-oriented retrieval and summarization. Specifically, QQSUM-RAG retrieves query-relevant re-

views, clusters them by distinct opinions, and summarizes representative KPs from each cluster. This approach provides broader coverage of key insights, overcoming the single-perspective limitation of conventional RAG-based systems.

A key challenge in implementing this approach is scarcity of training data for such a specialized task. To address this, we develop a co-training strategy that jointly optimizes the retriever and LLM through shared supervision signals, enhancing the alignment between retrieved opinion clusters and generated KPs. This strategy enables robust performance of QQSUM-RAG even with limited training examples. To support few-shot learning, we carefully curated a dataset of queries with KPs and their prevalence quantification, through human-LLM collaboration. Empirical results show that QQSUM-RAG significantly outperforms RAG baselines based on in-context learning and quantitative summarization.

Our main contributions are:

- We introduce a novel task QQSUM. Unlike traditional PQA, QQSUM generates answers that capture diverse customer opinions with their prevalence, addressing queries that require multiple viewpoints.

- We propose QQSUM-RAG, a RAG-based framework with KP-oriented retrieval and summarization. The framework is optimized through a co-training strategy that improves alignment between retrieved opinion clusters and generated KPs in few-shot learning setting. Our experiments show that QQSUM-RAG significantly outperforms baselines with up to 2.11 times improvement in textual similarity with ground-truth KPs and up to 67.12% improvement in quantification performance over state-of-the-art KPA system for reviews (Tang et al., 2024b).

2 Related Work

2.1 Review-based PQA

Unlike domain-specific QA tasks such as biomedical or legal QA focusing on factual answers, review-based PQA seeks to provide answers of consumers' subjective opinions about a product. While extractive PQA approaches retrieve relevant review snippets as answers (Chen et al., 2019a; Yu et al., 2012), it fails to provide precise responses since the review might not be specifically written for answering the given question. Recently, inspired by the advances of seq-2-seq models, abstractive, i.e., generation-based, approaches can generate natural-language answers from reviews (Chen et al., 2019c; Gao et al., 2019). However, these approaches frequently suffer from hallucinations and factual inconsistencies, sometimes generating random answers that misrepresent or contradict the prevalent opinions (Deng et al., 2020, 2023). Existing review-based PQA framework then cannot capture nor quantify faithfully the diverse opinions of reviews in its answer.

2.2 Key Point Analysis

Developed initially to summarize arguments (Bar-Haim et al., 2020a), KPA was later adapted for summarization of reviews (Bar-Haim et al., 2021; Tang et al., 2024a,b). While Bar-Haim et al. (2021) integrates sentiment analysis and collective key point mining to select and match KPs from broader domain with comments, Tang et al. (2024a) integrates aspect-based sentiment analysis (ABSA) into extracting and matching of KPs to comments for more unique KPs and precise quantification. More recent abstractive KPA studies apply abstractive summarization to paraphrase and generate KPs from comments (sentences) (Kapadnis et al., 2021; Li et al., 2023; Tang et al., 2024b). Overall, whether extractive or abstractive approaches, KPA can only produce KPs for general and high-level opinions without catering to specific queries.

2.3 Textual Summarization

Document summarization aims to produce concise textual summaries capturing the salient information in source documents. While extractive review summarization approaches use surface features to rank and extract salient opinions for summarization (Mihalcea and Tarau, 2004; Angelidis and Lapata, 2018; Zhao and Chaturvedi, 2020), abstractive techniques use sequence-to-sequence models (Chu and Liu, 2019; Suhara et al., 2020; Bražinskas et al., 2020b,a; Zhang et al., 2020a) to generate review-like summaries containing only the most prevalent opinions. Recently, prompted opinion summarization leveraging Large Language Models (LLMs) was applied to generate fluent and concise review summaries (Bhaskar et al., 2023). However, existing studies lack focus on presenting and quantifying the diverse opinions in reviews.

3 Quantitative Query-Focused Summarization

3.1 Task Formulation

Let q denote a query, i.e., community question, and $R_e = \{r_j\}_{j=1}^{|R_e|}$ denotes a set of review comments on a product e, QQSUM aims to retrieve relevant comments $\mathcal{D}$ to answer q and generate a KP-based summary $\mathcal{S}$ quantifying viewpoints presented in $\mathcal{D}$. We formulate $\mathcal{S} = \{kp_1, \ldots, kp_n\}$ as a bullet-point summary containing multiple KPs, where each bullet-point represents a KP [1] and its prevalence (Bar-Haim et al., 2021). For instance, with the bullet-point "*23 comments praise that the headphone is very comfortable for long hours*", the KP is "*Comfortable for long hours*", and the prevalence count is 23. Each key point kp_i, is matched to a subset of supporting comments $\mathcal{C}_i = \{c_1, c_2, \ldots\}$ (where $c_i \in \mathcal{D}$), with prevalence being measured as $|\mathcal{C}_i|$.

3.2 The QQSUM-RAG Framework

Figure 2 illustrates the architecture of QQSUM-RAG. QQSUM-RAG is based on the retrieval-augmented generation (RAG) paradigm and consists of 2 stages: ***KP-Oriented Retrieval*** and ***KP Summary Generation***. It utilizes a Retriever to retrieve and cluster query-relevant comments into groups, and the LLM to generate the final KP summary based on the comment clusters. Importantly, the retriever and LLM can be jointly trained with shared supervision signals to ensure comment clusters retrieved match KPs generated.

The following general loss function describes every training step of QQSUM-RAG, whose parameters are updated at the cluster-KP level rather than the query level:

$$\mathcal{L} = (1-d) \cdot (\mathcal{L}_{\text{clus}} + \text{gold\_score}) + d \cdot \mathcal{L}_{\text{gen}} \quad (1)$$

where $\mathcal{L}_{clus}$ is the retrieval loss for each comment cluster, $\mathcal{L}_{gen}$ is the LLM's generation loss computed for the KP generated from the respective cluster, and d is a damping factor to balance between the two. Notably, $gold\_score$ represents the Perplexity Distillation loss (Izacard et al., 2023), which transforms the supervisory signals of the LLM to improve the Retriever. The intuition is that within a cluster, comments that better contribute to helping the LLM generate the KP with lower perplexity should be ranked higher.

[1] unique and non-overlapping opinion at high level

3.2.1 KP-Oriented Retrieval

Given a query $\mathbf{q}$, the Retriever should retrieve relevant review comments R_q that emphasize opinions focused on $\mathbf{q}$. We utilize a shared encoder $\mathbf{E}$ that can encode both the input query $\mathbf{q}$ and each review comment $\mathbf{r}_j \in R_e$. Comments are ranked by the similarity score $s(\mathbf{x}, \mathbf{r}_j) = \mathbf{E}_c(\mathbf{x})^\top \mathbf{E}_d(\mathbf{r}_j)$ that is calculated by taking the dot product of the embeddings of the query $\mathbf{x}$ and the comment $\mathbf{r}_j$. Only comments with $s(\mathbf{x}, \mathbf{r}_j) \geq 1$ is selected for R_q. [2]

Different from standard RAG where generation is based on the direct retrieval result, to ensure diverse and representative opinions for generation, we enhance the Retriever with the clustering objective to produce distinctive comment groups that conceptually match KPs for generation.

KP-Oriented Retrieval Loss Starting with an empty list of clusters $\mathbf{C}$, and iterate through every comment in R_q, for every comment, we further iterate through every existing cluster $\mathbf{c}_i \in \mathbf{C}$ and calculate its average cosine similarity score to all comments of the cluster. Finally, we add the comment to any clusters with average cosine similarity score above a threshold ($\lambda = 1.2$), [3] otherwise, a new cluster is created. Importantly, a comment can be mapped to multiple clusters.

To train the retriever for KP-oriented retrieval, we align predicted comment clusters $\mathbf{C}$ with annotated clusters $\mathbf{P}$, where $\mathbf{P}$ groups comments matched to the same KP (annotation details in §3.3). Since $\mathbf{P}$ and $\mathbf{C}$ may differ in size at runtime, we map each $\mathbf{c}i \in \mathbf{C}$ to the most similar $\mathbf{p}i \in \mathbf{P}$ by computing the semantic similarity between cluster centroids. The centroid embedding of a cluster is the mean embedding of its comments: $\bar{\mathbf{E}}c(\mathbf{c}i) = \frac{1}{M}\sum k = 1^M \mathbf{E}(\mathbf{s}k)$. We determine $\mathbf{p}_i$ by selecting $\mathbf{p}j \in \mathbf{P}$ that maximizes the similarity score $sim(\mathbf{c}i, \mathbf{p}j) = \bar{\mathbf{E}}c(\mathbf{c}i)^\top \bar{\mathbf{E}}c(\mathbf{p}_j)$. The training objective minimizes the mean-squared-error (MSE) loss between each comment in $\mathbf{c}_i$ and the center of the most similar cluster $\mathbf{p}_i$.

$$\mathcal{L}_{clus} = \frac{1}{|\mathbf{c}_i|} \sum_{k=1}^{|\mathbf{c}_i|} ||\bar{\mathbf{E}}_c(\mathbf{p}_i) - \mathbf{E}(\mathbf{s}_k)||_2^2. \quad (2)$$

3.2.2 KP Summary Generation

A key limitation of previous KPA studies is that KPs may contain redundant opinions, due to that review comments, possibly containing multiple opin-

[2] the similarity threshold 1 is set empirically

[3] set empirically based on cluster quality

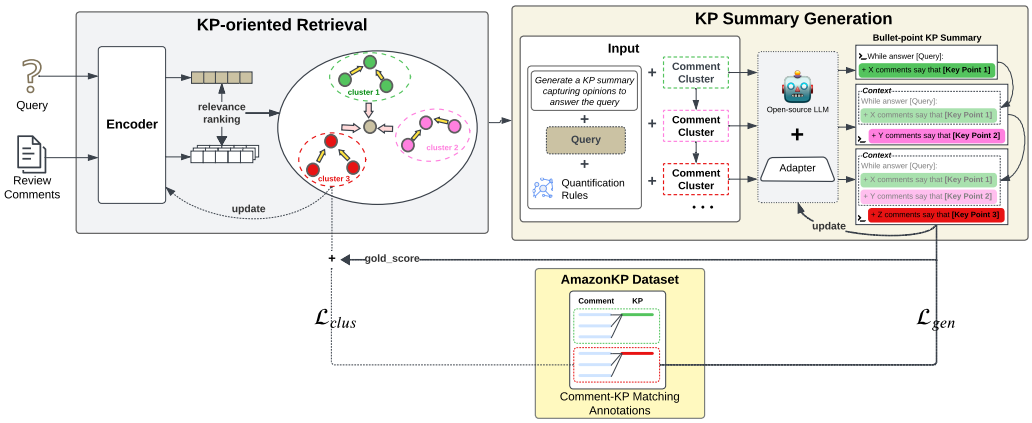

Figure 2: The training architecture of the QQSUM-RAG framework.

ions, are mapped to individual KPs locally (Bar-Haim et al., 2021; Tang et al., 2024b). To address this limitation, we propose to generate KPs at the global level, where the goal is to generate an overall KP-based summary without redundancy. Our main idea is that generated KPs are used as the context for the LLM to better reason and generate the next KP, which should be a unique, non-overlapping opinion statement.

Prompting Strategies Following OpenAI's prompt engineering guidelines[4], we format query-relevant comment clusters from the Retriever into a structured prompt with four parts (detailed in Listing 3, Appendix F): **1)** Context and input structure, **2)** Task definition and output requirements, **3)** Summarization steps for identifying representative KPs per cluster and generating the final KP-based summary, and **4)** Commonsense quantification rules to prioritize clusters by size and prevent overlapping KPs. To minimize ambiguity and hallucination, we encode predicted clusters $\mathbf{C}$ as JSON objects and assign each a unique ID, requiring the LLM to label generated KPs accordingly.

Next-KP-Generation Training During training, generating multiple KPs in a summary lacks alignment with $\mathcal{L}_{clus}$, which is computed per comment cluster. To address this, we introduce a *Next-KP-Generation* objective, inspired by Next-Token Prediction in LMs (Brown et al., 2020), to enhance the generation of salient, non-overlapping KPs. This approach fine-tunes the LLM to iteratively generate KPs within the summary. Specifically, let the final KP-based summary $\mathcal{S} = \{kp_1, \ldots, kp_i, \ldots, kp_n\}$, each kp_i is generated with preceding KPs $\{kp_1, \ldots, kp_{i-1}\}$ as the context, prompting the LLM to iteratively complete $\mathcal{S}$. The generation loss for each kp_i of $\mathbf{c}_i \in \mathbf{C}$ is computed as the negative log-likelihood (NLL) against the reference KP, annotated for the most similar $\mathbf{p}_i \in \mathbf{P}$ identified during retrieval,

$$\mathcal{L}_{gen} = -\frac{1}{T}\sum_{t=1}^{T} \log P(x_t|x_{<t}) \quad (3)$$

where $P(x_t|x_{<t})$ represents the probability assigned by the model to the correct token x_t, given the preceding tokens $x_{<t}$.

3.3 Human-LLM Key Point Annotation

From Section 3.2, to train our QQSUM-RAG framework in the few-shot setting, annotation of KPs for queries and relevant comments are necessary. Prior KPA studies only include annotations matching comments to KPs without queries (Bar-Haim et al., 2020a,b). No datasets exist for matching comments to KPs in PQA.

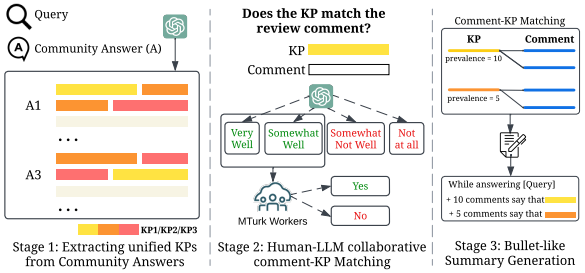

Figure 3: Illustration of the human-LLM collaborative annotation pipeline for AMAZONKP.

We leverage the popular PQA dataset AmazonQ&A (Gupta et al., 2019) for our QQSUM task, focusing on only *answerable*, *subjective* (non-factual) questions that have multiple answers. Out of 17 product categories (e.g., Electronics, Video

[4] https://platform.openai.com/docs/guides/prompt-engineering

Statistic	Train	Test
# Product Categories	17	17
# Instances (queries) Per Category	2	148
Total Instances	34	2516
# Reviews Per Query	71.18	72.70
# Review Comments Per Query	452.03	431.62
# Answers Per Query	7.53	6.45
# KPs Per Query (Stage 1)	9.26	6.90
# Relevant Comments Per Query (Stage 2)	24.50	–
# Comments (Prevalence) per KP (Stage 2)	6.37	–
Summary Length (Stage 3)	101.29	–

Table 1: Core statistics of the AMAZONKP dataset.

Games), we only include businesses with 50-100 reviews, and sampling top 150 questions per category based on answer count. For ease of reference we name this curated dataset AMAZONKP. Details on question classification for AMAZONKP are in Appendix A, and their taxonomy in Appendix B. Notably, the dominance of "*Scenario-based*" questions underscore the importance of QQSUM for generating KP summary to answer user questions on preferences and scenarios.

Manually summarizing and quantifying opinions from comments is laborious and time-consuming, if not impossible. Research shows LLM's strong annotation capabilities (He et al., 2024), and so we design a three-stage human-LLM collaborative annotation pipeline, shown in Figure 3.

Stage 1: KP Extraction from Gold Community Answers Given a query q_i, the AmazonQ&A dataset provides multiple answers, i.e. responses, from online users $A_i = \{a_1, a_2, .\}$, serving as ideal approximation of gold opinions. However, these responses can contain overlapping opinions. We therefore zero-shot prompted `GPT-4-o-mini` to extract distinctive and non-overlapping KPs from A_i. Empirical validation with human annotators confirms that the extracted KPs are of high quality, covering up to 97.2% of the opinions expressed in community answers (recall), while 96.7% of the extracted KPs are verified as valid (precision). Further details are in Appendices C and D.

Stage 2: LLM-based and Manual Comment-KP Matching Based on the annotation process in the literature (Bar-Haim et al., 2020a), we further integrate LLMs to reduce human effort and time. Using KPs extracted from gold answers (Stage 1), we prompt `GPT4-o-mini` to annotate pairwise matches between comments and KPs from all available reviews of the product. LLM-matched pairs are then validated by three Amazon Mechanical Turk (MTurk) workers. Finally, comments from validated pairs are grouped by similar KPs, with KP prevalence determined by the number of matching comments. Further details on KP Matching annotations are provided in Apppendix E.

Stage 3: KP-based Summary We utilize KPs and their prevalence counts, discovered for every query, to manually compose a bullet-point KP-based summary, where each bullet point corresponds to a KP and is annotated as "$|kp_i|$ *comments say that* kp_i".

The number of pairwise comment-KP matching annotations required per query can be up to 2K-3.5K. **For training**, to control annotation costs, we conducted Stages 1, 2 and 3 annotations on a small subset of 34 instances for few-shot training of QQSUM-RAG, randomly selecting two queries per product category for supervised labeling. **For evaluating the KP-based summary**, the remaining examples with only Stage 1 annotations serve as the test set. The core statistics of AMAZONKP are shown in Table 1.

4 Experiments

We employ Atlas (Izacard et al., 2023), a pre-trained efficient RAG model, as our backbone model for QQSUM-RAG. We utilized Contriever (Izacard et al., 2022) as the retriever while replacing the original language model with open-source LLMs (e.g., `Vicuna-7B` [5], `Mistral-7B` [6]) for generation. For computational feasiblity, we apply Low-Rank Adaptation (LoRA) (Hu et al., 2021), which adds trainable parameters while freezing the model's original weights.

4.1 Baselines

We benchmark QQSUM-RAG against 3 RAG baselines.

(Retriever + LLM)$_{\text{co-train}}$ We few-shot trained Atlas (Izacard et al., 2023), with the standard RAG architecture and Retriever-LLM generator co-training, for the QQSUM task. The Retriever (Contriever) retrieve relevant comments, while letting the LLM implicitly infer KPs' matching comments and their quantities during KP summary generation. For training, we aggegrated matching comments across KPs, per query, as the retrieval ground truth.

Frozen Retriever + Prompt LLM To assess in-context learning (ICL) for QQFS, we use a

[5] https://huggingface.co/lmsys/vicuna-7b-v1.5

[6] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

frozen Contriever as the Retriever and `Vicuna-7B`, `Mistral-7B`, and `GPT-4-Turbo` as the LLM for ICL. Few-shot training instances are concatenated with test instances, with shot numbers optimized for context length and cost: 4-shot for `Mistral-7B` and `GPT-4-Turbo`, and 2-shot for `Vicuna-7B`.

Frozen Retriever + KPA We replace the LLM of a standard RAG with existing KPA review summarization systems to adapt KPA to the QQSUM task. With comments retrieved by a frozen Contriever (Izacard et al., 2022), **RKPA-Base** (Bar-Haim et al., 2021) leverages a quality ranking model (Gretz et al., 2020) to extract KP candidates, and integrates sentiment analysis and collective key point mining into matching comments to the extracted KPs. **PAKPA** (Tang et al., 2024b) clusters comments by aspect and sentiment before generating aspect-oriented KPs.

All experiments were conducted at the KP level, focusing on KPs in the summary outputs of QQSUM-RAG and baselines for fair comparison. We post-process the output KP-based summary into KPs as JSON objects, where each object covers the KP information of a bullet point in the summary. [7] The baselines were implemented using either the PyTorch module or the Huggingface transformers framework, and were trained on a NVIDIA GeForce RTX 4090 GPU.

4.2 Evaluation Dimensions

We conducted experiments on the test set of AMAZONKP (§3.3), consisting of questions from 17 product categories. For reasonable cost, we sample 8 questions from each category for evaluation.

KP Quality Extracted KPs from gold community answers (Stage 1 in §3.3) serve as the reference KPs. We employed metrics `Rouge` (Lin, 2004), `BLEURT` (Sellam et al., 2020), `BERTScore` (Zhang et al., 2020b), soft-Precision/Recall/F1 (Li et al., 2023), as well as LLM-based metric G-EVAL-4 (Liu et al., 2023) to measure lexical and semantic similarity between KPs in the generated summary and the ground truth. For comparability, G-EVAL scores are scaled from 1-5 to 0-1. To fit our evaluation, we customized the evaluation prompt of G-EVAL, with details presented in Appendix G.

[7] We use a simple LLM-based post processor, prompting `gpt-4-o-mini` with `'Format all key points and their prevalences mentioned in the above bullet-point summary in a JSON list, where each JSON object format as: {'key_point': <key point of a bullet>, 'prevalence': <key point's prevalence>}'`

	P@5	P@10	P@20	P@all
QQSUM-RAG (Ours)				
+ Mistral	**0.668**	**0.633**	**0.601**	**0.535**
+ Vicuna	0.567	0.527	0.526	0.367
$(\textbf{Retriever} + \textbf{LLM})_{\text{co-train}}$ (Izacard et al., 2023)				
+ Mistral	0.479	0.430	0.418	0.320
+ Vicuna	0.386	0.378	0.329	0.173
frozen Retriever				
	0.457	0.398	0.366	0.278

Table 2: Retrieval performance of the Retriever

Redundancy (RD, lower the better) aims to measure the redundancy of KPs, which compute the highest semantic similarity based on set-level soft-Precision/Recall/F_1 between each generated KP and its nearest neighbor in answering a query:

$$RD = \frac{1}{n} \times \sum_{\alpha \in \mathcal{A}} \max_{\theta \neq \alpha \in \mathcal{A}} f(\alpha_i, \theta_j) \quad (4)$$

where f measures similarity between two individual KPs, $\mathcal{A}$ and $\mathcal{B}$ are generated and reference KP sets, with $n = |\mathcal{A}|$ and $m = |\mathcal{B}|$.

KP Quantification We evaluate the KP quantification performance of different systems for KP-comment matching and factual alignment.

KP-comment matching We first assess the accuracy of the KP comment matching by extending Bar-Haim et al. (2021) to measure both *precision* (correctness of predicted matches) and *recall* (coverage of ground-truth matches). For each system, we compute precision and recall by prompting `gpt-4-o-mini` to annotate pairwise *match/non-match* between generated KPs and retrieved comments R_q. Additionally, leveraging annotated comment-KP pairs, we introduce *QuantErr*, which measures the mean absolute error between predicted and actual KP prevalence count. Empirical validation shows `gpt-4-o-mini` annotations highly correlated with MTurk workers' judgement (Pearson's r = 0.647) (Appendix H).

KP-comment factual alignment We further employed `AlignScore` (Zha et al., 2023) for automatic evaluation of factual alignment between generated KPs and their corresponding comments.

4.3 Results

4.3.1 Comment Retrieval

We evaluated the performance of retrievers for all models, by prompting `gpt-4-o-mini` to annotate the relevance of retrieved comments to queries. Table 2 reports the retrieval Precision@k (P@k), measured at different levels of top-k-ranked retrieved comments ($[5, 10, 20, all]$). Results from Table 2 show that the trained Retriever of QQSUM-RAG,

	ROUGE			BERTScore				BLEURT				G-Eval-4			
	R-1	R-2	R-L	sP	sR	sF1	RD↓	sP	sR	sF1	RD↓	sP	sR	sF1	RD↓
QQSUM−RAG (Ours)															
+ Mistral	**0.256**	0.061	**0.220**	**0.39**	**0.29**	**0.33**	**0.37**	**0.51**	**0.41**	**0.46**	**0.49**	**0.88**	**0.82**	**0.85**	**0.36**
+ Vicuna	0.222	**0.078**	0.204	0.38	0.26	0.31	0.53	0.49	0.39	0.44	0.54	0.87	0.81	0.84	0.36
(Retriever + LLM)$_{\text{co-train}}$ (Izacard et al., 2023)															
+ Mistral	0.209	0.057	0.194	0.37	0.28	0.32	0.43	0.49	0.40	0.44	0.55	0.81	0.82	0.81	0.41
+ Vicuna	0.174	0.041	0.161	0.37	0.26	0.31	0.48	0.48	0.38	0.42	0.58	0.78	0.78	0.78	0.41
Frozen Retriever + prompt LLM															
+ Mistral	0.210	0.055	0.191	0.33	0.26	0.29	0.51	0.46	0.38	0.42	0.55	0.79	0.80	0.79	0.41
+ Vicuna	0.164	0.059	0.154	0.22	0.20	0.21	0.48	0.46	0.31	0.37	0.59	0.73	0.73	0.73	0.41
+ GPT-4-Turbo	0.197	0.048	0.174	0.32	0.25	0.28	0.44	0.45	0.38	0.41	0.54	0.77	0.77	0.77	0,38
Frozen Retriever + KPA															
+ PAKPA (Tang et al., 2024b)	0.179	0.027	0.162	0.34	0.28	0.31	0.46	0.47	0.41	0.44	0.54	0.79	0.80	0.80	0.36
+ RKPA-Base (Bar-Haim et al., 2021)	0.121	0.016	0.106	0.16	0.14	0.14	0.50	0.43	0.36	0.39	0.61	0.69	0.70	0.69	0.51

Table 3: KP summary textual quality. sP, sR and sF1 refer to Soft-Precision, Soft-Recall, and Soft-F1 respectively based on set-level evaluation method against reference KPs in gold answer.

	KP-Comment Matching				KP-Comment Factual Alignment
	P	R	F1	QuantErr↓	AlignScore
QQSUM-RAG (Ours)					
+ Mistral	0.694	**0.869**	**0.792**	**04.24**	**0.749**
+ Vicuna	0.538	0.684	0.602	07.83	0.630
(Retriever + LLM)$_{\text{co-train}}$ (Izacard et al., 2023)					
+ Mistral	0.567	0.249	0.346	18.10	0.653
+ Vicuna	0.442	0.094	0.154	30.13	0.394
Frozen Retriever + prompt LLM					
+ GPT-4-Turbo	0.746	0.200	0.313	16.63	0.673
+ Mistral	0.498	0.214	0.300	19.14	0.624
+ Vicuna	0.439	0.185	0.260	21.52	0.531
Frozen Retriever + KPA					
+ PAKPA (Tang et al., 2024b)	**0.762**	0.520	0.619	06.68	**0.749**
+ RKPA-Base (Bar-Haim et al., 2021)	0.371	0.314	0.340	15.62	0.354

Table 4: Performance for KP-Comment matching and factual alignment

as being co-trained with the LLM and extended for KP-oriented Retrieval, outperform all baselines.

4.3.2 KP Quality

The quality of KPs produced by different systems, in terms of textual quality, semantic quality and redundancy are reported in Table 3. Overall, scores of all systems are low in general because in reality opinions in product reviews may not cover all opinions from user answers to community questions. From Table 3, QQSUM-RAG outperforms other systems in all quality dimensions. It shows 2.11 times improvement in textual similarity with reference KPs (0.256 vs. 0.121 in ROUGE-1), 0.23 point absolute improvement in semantic similarity (0.39 vs. 0.16 in BERTScore) and 0.14 point absolute reduction in Redundancy (0.37 vs. 0.51 using BERTScore for semantic similarity).

We believe *KP-oriented Retrieval* of QQSUM-RAG effectively contributes to better scores. Specifically, although (Retriever + LLM)$_{\text{co-train}}$ shares the same backbone model and co-training design with QQSUM-RAG, the lack of (1) opinion-level clustering of retrieved comments and (2) limited modeling capability of LLMs makes this model unable to produce KPs as diverse, unique and representative as QQSUM-RAG. Notably, the weak reasoning capability of LLMs for diverse opinion summarization is further exposed in the frozen Retriever + prompt LLMs setting, where LLMs even with strong modelling capability like GPT-4-Turbo struggle to elaborate diverse and distinctive KPs from hundreds of comments.

It is worth noting that `Mistral-7B` broadly exhibits higher performance than `Vicuna-7B` across all systems based on LLM generation and in all KP quality measurement (up to 15.32%), largely due to its stronger modeling capability.

Frozen Retreiver + KPA baselines, despite their high performance for review summarization, is ineffective for QQSUM. Not surprisingly PAKPA, which generates KPs based on aspect-sentiment, broadly shows better performance than RKPA-Base, an extractive KPA system. It is possible that multiple query-relevant opinions on the same aspect are expected to answer a user query, thus leading to the weak performance of PAKPA.

4.3.3 KP Quantification

Table 4 presents the quantification performance for different systems. F_1, combining Recall and Precision, measures the overall performance of KP-comment matching for all systems. QuantErr (lower the better) directly measures KP quantification errors. Overall, QQSUM-RAG shows the best performance in terms of both F_1 (0.792 vs. 0.154) and QuantErr (4.24 vs. 30.13).

Comparing QQSUM-RAG against the Retriever+LLM generation systems, e.g., (Retriever + LLM)$_{\text{co-trained}}$ and Frozen Retriever + prompt LLM, we can see, without explicit comment cluster formation at the retrieval stage, LLMs perform implicit comment-KP matching and KP quantification, showing extremely low Recall (0.185–0.249), in contrast to the high Recall of QQSUM-RAG (0.684-0.869). The poor performance of Retriever+LLM systems can be attributed to two main

factors: (1) LLMs tend to hallucinate when generating KPs from a large set of retrieved comments, and (2) their limited context window restricts their ability to effectively match comments to KPs.

Comparing QQSUM-RAG against Retriever + KPA systems, our model shows up to 67.12% improvement in quantification performance over state-of-the-art KPA system for reviews (PAKPA) (Tang et al., 2024b), with a 36.53% reduction in QuantErr. Note that Frozen Retriever + PAKPA achieves the highest matching precision due to aspect-level opinion quantification. However, it has low recall, possibly due to its reliance on aspect-based sentiment analysis of comments sometimes, which can fail to identify implicit opinions not explicitly including aspects.

From Table 4, results for KP-Comment Factual Alignment show that QQSUM-RAG and Frozen Retriever + KPA (PAKPA) achieve high factual correctness in KP generation, outperforming other systems (0.749 vs. 0.354). This result highlights that QQSUM-RAG generates KPs grounded in the retrieved comments, and similarly PAKPA generates KPs grounded in aspects.

4.4 Ablation Study

We conducted an ablation study to evaluate contribution of the Next-KP-Generation strategy in QQSUM-RAG, with results in Tables 8 and 9 (Appendix I). In particular, we configure a variant QQSUM-RAG $_{\text{Single-KP}}$ that replaces Next-KP-Generation with KP generation for each comment cluster. Not including previously generated KPs as context, QQSUM-RAG $_{\text{Single-KP}}$ struggles to capture the truly representative opinion of the cluster, likely generating KPs with overlapping opinions, especially for comments containing multiple opinions. Note that while its KP quality underperforms RAG baselines, its KP-comment matching and factual consistency remain superior, largely attributed to KP-oriented Retrieval.

4.5 Case studies

We conducted case studies to evaluate the redundancy and specificity of generated KPs for a query comparing camera lenses, presented in Table 12 (Appendix J). Overall, QQSUM-RAG stands out for generating KPs with minimal redundancy, high informativeness, and alignment with the query. First, QQSUM-RAG reduces redundancy by effectively capturing distinct product features relevant to the user's needs (e.g., faster aperture), whereas (Retriever + LLM)$_{\text{co-train}}$, GPT-4-Turbo Prompt LLM, and PAKPA tend to generate repetitive and generic statements, such as "The 24-70mm f/2.8 is a better lens overall." Furthermore, QQSUM-RAG expands feature coverage, capturing details such as Vibration Reduction (VR) technology, which several baselines fail to mention.

4.6 Error Analysis

By analyzing errors in QQSUMM's KP generation reported in Table 11, we identified systematic patterns. A frequent issue occurs when a KP and its matched comment express similar claims but refer to different targets. For instance, the comment "My only complaint is the price tag: for a lens that is overall a rather mixed bag ... it is very expensive." was matched to "The 24-120mm F4 lens has a longer zoom range and is more affordable than the 24-70mm F2.8.". Since the comment lacks an explicit product reference, it remains unclear whether it critiques the *24-120mm F4* or the *24-70mm F2.8*. Another possible error stems from KP-oriented retrieval operating at the sentence level, whereas product review sentences can contain co-existing opinions in multiple aspects (e.g., affordability and zoom range), making it difficult for the Retriever to separate those into more distinctive clusters. This results in KPs containing opinions on multiple aspects, e.g., The 24-120mm F4 lens has *a longer zoom range* and is *more affordable* than the 24-70mm F2.8.

5 Conclusion

In this paper, we explore a new task Quantitative Query-focused Summarization, namely QQSUM, for capturing and quantifying diverse opinions from online reviews for PQA. We propose QQSUM-RAG, a few-shot summarization model based on retrieval-augmented generation where summary is generated by LLMs from groups of user opinions relevant to a query. QQSUM-RAG addresses the issue of existing RAG frameworks for providing only random or major opinion in the answer. By extending the retriever with opinion-based clustering of relevant comments, our model ensures capturing more diverse and representative opinions in the summary, along with accurate quantification. Experimental results show that our solution greatly enhances both the quality and quantitative performance for KPs generated in the summary.

Limitations

We evaluated the textual quality of generated KPs only on AmazonQ&A, as it is the only (to our best knowledge) public dataset with abundance of online community answers written by online users usable as ground truth for our automatic evaluation.

Since we are leveraging answers from AmazonQ&A to summarize/quantify the prevalence of query-relevant opinions from reviews regarding a query, an inevitable limitation is that KPs extracted from the Q&A answers might not fully ensure to always represent all viewpoints presented in reviews while answering a question. Similarly, opinions in product reviews also may not sufficiently cover all expected opinions from gold answer of given questions.

Ethics Statement

We have applied ethical research standards in our organization for data collection and processing throughout our work.

The AmazonQ&A dataset used in our experiments was publicly crowdsourced and released for the research publication for the review-based product question answering task (Gupta et al., 2019). The dataset was published following their ethical standard, after removing all personal information. The answers to questions do not contain contents that are harmful to readers.

We ensured fair compensation for crowd annotators on Amazon Mechanical Turk. We setup and conducted fair payment to workers on their annotation tasks/assignments according to our organization's standards, with an estimation of the difficulty and expected time required per task based on our own experience. Especially, we also made bonus rewards to annotators who exerted high-quality annotations in their assignments.

References

Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686, Brussels, Belgium. Association for Computational Linguistics.

Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. 2020a. From arguments to key points: Towards automatic argument summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4029–4039, Online. Association for Computational Linguistics.

Roy Bar-Haim, Lilach Eden, Yoav Kantor, Roni Friedman, and Noam Slonim. 2021. Every bite is an experience: Key Point Analysis of business reviews. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3376–3386, Online. Association for Computational Linguistics.

Roy Bar-Haim, Yoav Kantor, Lilach Eden, Roni Friedman, Dan Lahav, and Noam Slonim. 2020b. Quantitative argument summarization and beyond: Cross-domain key point analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 39–49, Online. Association for Computational Linguistics.

Adithya Bhaskar, Alex Fabbri, and Greg Durrett. 2023. Prompted opinion summarization with gpt-3.5. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9282–9300.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020a. Few-shot learning for opinion summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4119–4135, Online. Association for Computational Linguistics.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020b. Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5151–5169, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Kathy Charmaz. 2015. Grounded theory. *Qualitative psychology: A practical guide to research methods*, 3:53–84.

Long Chen, Ziyu Guan, Wei Zhao, Wanqing Zhao, Xiaopeng Wang, Zhou Zhao, and Huan Sun. 2019a. Answer identification from product reviews for user questions by multi-task attentive networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 45–52.

Qibin Chen, Junyang Lin, Yichang Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019b. Towards knowledge-based personalized product description generation in e-commerce. In *KDD 2019*, pages 3040–3050.

Shiqian Chen, Chenliang Li, Feng Ji, Wei Zhou, and Haiqing Chen. 2019c. Driven answer generation for product-related questions in e-commerce. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 411–419.

Eric Chu and Peter Liu. 2019. Meansum: A neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning*, pages 1223–1232. PMLR.

Yang Deng, Wenxuan Zhang, and Wai Lam. 2020. Opinion-aware answer generation for review-driven question answering in e-commerce. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 255–264.

Yang Deng, Wenxuan Zhang, Qian Yu, and Wai Lam. 2023. Product question answering in E-commerce: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11951–11964, Toronto, Canada. Association for Computational Linguistics.

Shen Gao, Zhaochun Ren, Yihong Zhao, Dongyan Zhao, Dawei Yin, and Rui Yan. 2019. Product-aware answer generation in e-commerce question-answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 429–437.

Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Assaf Toledo, Dan Lahav, Ranit Aharonov, and Noam Slonim. 2020. A large-scale dataset for argument quality ranking: Construction and analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7805–7813.

Mansi Gupta, Nitish Kulkarni, Raghuveer Chanda, Anirudha Rayasam, and Zachary C Lipton. 2019. Amazonqa: A review-based question answering task. *arXiv preprint arXiv:1908.04364*.

Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2024. AnnoLLM: Making large language models to be better crowdsourced annotators. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 165–190, Mexico City, Mexico. Association for Computational Linguistics.

Edward Hu, Yelong Shen, Phil Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Manav Kapadnis, Sohan Patnaik, Siba Panigrahi, Varun Madhavan, and Abhilash Nandy. 2021. Team enigma at ArgMining-EMNLP 2021: Leveraging pre-trained language models for key point matching. In *Proceedings of the 8th Workshop on Argument Mining*, pages 200–205, Punta Cana, Dominican Republic. Association for Computational Linguistics.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Hao Li, Viktor Schlegel, Riza Batista-Navarro, and Goran Nenadic. 2023. Do you hear the people sing? key point analysis via iterative clustering and abstractive summarisation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14064–14080, Toronto, Canada. Association for Computational Linguistics.

Piji Li, Zihao Wang, Lidong Bing, and Wai Lam. 2019. Persona-aware tips generation? In *WWW 2019*, pages 1006–1016.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Julian McAuley and Alex Yang. 2016. Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th International Conference on World Wide Web*, pages 625–635.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages

7881–7892, Online. Association for Computational Linguistics.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, et al. 2024. A roadmap to pluralistic alignment. *arXiv preprint arXiv:2402.05070*.

Yoshihiko Suhara, Xiaolan Wang, Stefanos Angelidis, and Wang-Chiew Tan. 2020. OpinionDigest: A simple framework for opinion summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5789–5798, Online. Association for Computational Linguistics.

An Tang, Xiuzhen Zhang, and Minh Dinh. 2024a. Aspect-based key point analysis for quantitative summarization of reviews. In *18th Conference of the European Chapter of the Association for Computational Linguistics*.

An Tang, Xiuzhen Zhang, Minh Dinh, and Erik Cambria. 2024b. Prompted aspect key point analysis for quantitative review summarization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10691–10708, Bangkok, Thailand. Association for Computational Linguistics.

Jianxing Yu, Zheng-Jun Zha, and Tat-Seng Chua. 2012. Answering opinion questions on products by exploiting hierarchical organization of consumer reviews. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 391–401, Jeju Island, Korea. Association for Computational Linguistics.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Chao Zhao and Snigdha Chaturvedi. 2020. Weakly-supervised opinion summarization by leveraging external information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9644–9651.

A Opinionated Question Classification for AMAZONKP Dataset

Existing online product-related questions can be categorized into two groups: subjective (opinionated) or objective (factal). While subjective questions asks about positive/negative feeling or stance (e.g., whether a product is “good" or “bad"), objective questions confirms the actual product details (e.g., products properties, specific use-cases). In E-Commerce, questions are often subjective, i.e., asking for former buyer’s opinion, where different customers often have certain preferences over product aspects or information needs (Chen et al., 2019b; Li et al., 2019), leading to various expectations for the provided answers.

We extract subjective, i.e., opinionated, question from AmazonQ&A by prompting the `Mistral-7B` open-source LLM to analyze the question and its associated answers, published by the online community. In this case, leveraging answers helps to understand the nature of the questions, thereby better reasoning whether the question is seeking for subjective information from users. We present the few-shot prompt for classifying opinionated, i.e., subjective, questions from AmazonQ&A in Listing 1.

B Qualitative Data Analysis of Opinionated Questions’ Categories in AMAZONKP

We further studied the utility of the QQSUM task and our by conducting qualitative data analysis to categorize possible opinionated question’s type in AMAZONKP. Based on the grounded theory methodology (Charmaz, 2015), our analysis employ human-LLM collaborative annotation to iteratively code the fine-grained categories from opinionated questions. We sampled a subset of 100 questions from AMAZONKP for data coding and intepretation. On the subset, we start by prompting ChatGPT to identify potential categories of opinionated questions, including the categories’ name and their definitions (Step 1). Importantly, the data coding process involves human validation, in which we iteratively a human annotator iteratively evaluate the representative of generated categories while interacting with ChatGPT, and manually refine the categories where possible [8] (Step 2). Then, we

[8] On categories requiring more fine-grained categorization, we further conduct another analysis cycle on the particular coarse-grained category, by selecting questions and answers

Listing 1: Few-shot prompt (2 examples) for prompting `Mistral-7B` on opinionated question classification.

```
You will be provided with a question and multiple answers to that question, delimited by triple quotes.
The question was taken from a Question Answering dataset of product reviews, and can either be an opinionated or factual question.

You were tasked to classify whether the given question is an opinionated or factual question.
Factual questions ask for objective data, specifications, or information that can be definitively answered based on product facts, manual, user experience, or specifications. Factual question tends to covers unique and consistent opinions/fact in its answers.
Opinionated questions are subjective and seek insights that are based on personal use, feelings, preferences, judgments, or evaluations about a product. Opinionated question has multiple and diverse opinions in its answers.

Formally, you should perform the following steps:
1. Identify unique opinions from the answers of the given question
2. Based on the question content and the amount of opinions in the question's answer, identify the question's type.

Note that you must briefly explain why the question is opinionated or factual before giving the final decision.

Below are some few-shot examples:

Questions: How well does it work with wireless charging
Answers: ['Unfortunately with this case installed it will not hold the phone vertically.', 'I use the case with the official wireless charger and have had no problems at all.', 'Works great. Not a fan of the dimensions.']
Type: 'Opinionated Question'

Questions: Are the shelves removeable?
Answers: ['yes, they are removeable..', 'Yes they are, you can arrange them for the size of the shot glass.']
Type: 'Factual Question'
```

prompted a `gp4-o-mini` to annotate the labels of entire questions in the subset, before asking human annotator again to validate the representative and suitability of the candidate categories on questions. Categories with abnormal distribution, e.g., 5 times higher than others, or with high unmatching cases will be passed back to Step 2 for another iterative analysis cycles.

As a result, our analysis reported 5 categories commonly representative of question in AMAZONKP, namely, *Performance*, *Quality*, *Recommendation*, *Comparative* and *Controversial*, with each the stating clearly the purpose of the users asking the questions and expected answers. Finally, We prompted `gpt-4-o-mini` to annotate such categories on AMAZONKP's opinionated questions, and reported their taxonomy and statistics in Table 5. Notably, the dominance of "*Scenario-based*" questions underscore the importance of QQSUM for generating KP summary to answer user questions on preferences and scenarios.

from the specific category for analysis.

C Validating GPT4's Key Point Extraction from Gold Community Answer of AmazonQ&A

In this experiment, we empirically validate `gpt-4-o-mini`'s performance and credibility in extracting KPs from gold community answers for AmazonKP (Stage 1 of §3.3). We specifically sampled 2 questions, i.e., queries, from each product categories of AmazonKP, totaling 34 questions, and hired workers to annotate whether the extracted KPs matches original gold community answers of the sampled questions, which is inspired by the KP Matching evaluation of Bar-Haim et al. (2021) More specifically, for a given query, we asked workers to perform pairwise annotation between extracted KPs and the query's respective community answers. While *Precision* calculates the fraction of KPs matched to at least one gold answer, i.e., out of all extracted KPs how many are correctly mapped, *Recall* shows the fractions of gold answers matched to at least one KP, i.e., out of all answers how many are covered by KPs. We then macro-averaged Precision/Recall computed for every question to obtain the final values.

For human annotation, we employed 3 MTurk crowd workers on every answer-KP pair, selecting only those with an 80% or higher approval rate and

Category	Description	Example	# Query
Performance	Ask how well a product performs or functions in general.	How well does it work on carpet?	376
Quality	Ask about the overall or aspect-specific quality of the product.	Is this product worth the money?	265
Scenario-based	Ask whether a product fits specific use cases, sizes, or other products.	Does this item really stop the glare at night even in rain or snow?	1402
Recommendation	Ask for suggestions tailored to specific issues or use cases.	What do you use to spray this stuff on your lawn?	156
Comparative	Seeks opinions about the relative advantages or disadvantages of a product compared to others.	Would a wired keyboard/mouse be better than wireless?	227
Controversial	Reflect dissatisfaction or complaint about a product, likely to provoke debate or controversy.	Why does this need adjustment screws? If I have to align the laser then what's the point?	124

Table 5: A taxonomy of opinion questions AMAZONKP

at least 10 approved tasks. Following Bar-Haim et al. (2021), we exclude annotators with Annotator-$\kappa < 0$ for quality control. This score averages all pairwise Cohen's Kappa (Landis and Koch, 1977) for a given annotator, for any annotator sharing at least 50 judgments with at least 5 other annotators. For labelling correct matches, at least 67% (2 out of 3) of the annotators had to agree that the match was correct. Otherwise, it is incorrect.

Precision	96.7%
Recall	97.1%
# Matched Answer Per KP	2.53
# Matched KP Per Answer	3.12

Table 6: Performance validation of `gpt-4-o-mini`'s KP extraction from gold community answer. While precision calculates the fraction of KPs matched to at least one gold answer, recall shows the fractions of gold answers matched to at least one KP.

Table 6 presents the fraction of extracted KPs matched to at least one gold answer (Precision) and vice versa (Recall). Overall, the experiment confirms that the extracted KPs are of high quality, covering up to 97.2% of the opinions expressed in community answers (recall), while 96.7% of the extracted KPs are verified as valid (precision).

Below are the match annotation guidelines for (extracted KP, gold answer) pairs:

In this task you are presented with a question on a product, a key point extracted from community answers answering the question, and a community answer for answering the query of that product.

You will be asked to answer the following question:"Does the key point match, i.e., represent an opinion in the community answer?"

A community answer might express opinions on multiple aspects. A key point matches a community answer if it captures the gist of the answer, or is directly supported by a point made in the community answer.

The options are:

- Not At All
- Somewhat Not Well
- Somewhat Well
- Very Well

D Prompt for Key Point Extraction from Gold Community Answer of AmazonQ&A

We present the few-shot prompts for extracting key points (KPs) from gold online community answers of AmazonKP in Listing 2.

E Annotation Details of KP Matching for AMAZONKP Dataset

We offer `GPT-4-o-mini` with 4 options for labelling the matching status of given comment-KP pairs. Pairs annotated as *Very Well* or *Somewhat Well* by LLM then becomes *candidate matching pairs*, which will be further validated by human annotation for their correctness. For human annotation, we employed 3 MTurk crowd workers per comment-KP pair, selecting only those with an 80% or higher approval rate and at least 10 approved tasks. Following Bar-Haim et al. (2021), we exclude annotators with Annotator-$\kappa < 0$ for quality control. This score averages all pairwise Cohen's Kappa (Landis and Koch, 1977) for a given annotator, for any annotator sharing at least 50 judgments with at least 5 other annotators. For labelling correct matches, at least 60% of the annotators had to agree that the match is correct, otherwise, it is incorrect. Comments from final

Listing 2: One-shot prompt (1 example) for prompting `GPT-4-o-mini` on KP Extraction from community answers.

```
You will be provided with an opinionated question and multiple answers to that question, delimited by triple quotes.
An opinionated question seek insights of user opinions that are based on personal use, feelings, preferences, judgments, or
    evaluations about a product, and was taken from a Question Answering dataset of product reviews.

You were tasked to extract a list of unique and concise key points from the list of answers to given opinionated question.
Key points are short and high quality sentences that expresses the main claims/viewpoints of users answering the opinionated
    question
Note that the final extracted list of key points must directly relevant and can answer the input opinionated question.

Formally, you should perform the following steps:
1. In every answer from the list, extract all possible key point candidates.
2. From the extracted list of key point candidates, generate a list of only general and non-overlapping key points that are
    relevant and can answer the input opinionated question.

Below are some few-shot examples:

Questions: Can I use these for running/working out? Do they handle sweat?
Answers: ['I have seen other people using these for running/working out. These are very comfortable in your ears for long
    hours. As long you clean them after working out, you should be fine. These are built to last a long time.', 'I use them in
    the gym and on the stair climber machine. They are fine. Not sure about running but would think they would work ok.', "
    I don't know if I'll be any help, but I'll tell you about my experience nevertheless. I used it everyday in the gym & while I
     go for work on my bike inside my helmet. In both cases, the sweat doesn't seem to have any effect. Even during long
    rides, and when it rained heavily, the IE80 held up fine. The only issue you will have to worry about is the cable.
    Though the cables are good quality, rough usage may affect the balance in volume levels between the two channels.
    Though this doesn't affect the clarity, the balance can be disturbed. After a year of really rough usage, the IE80 right
    volume was 1-2% lower than the left [I got mine replaced for free soon after]. But, this is an issue which affects every
    IEM, and nothing to sweat over, since we can replace the cables if necessary. So if you don't give it a hard time, it
    should hold up fine.[I can't even count the times it has fallen down or swung down and taken a hit against the gym
    equipment, or when my phone/DAP slipped and yanked the cable]"]
Key Points: ['Comfortable for long hours', 'Built to last a long time', 'Suitable for gym and stair climber machine', 'Sweat
    resistant during workouts', 'Cables may be affected by rough usage']
```

matching pairs, after confirmed by human, will then be grouped by similar KPs, where the amount of matching comments per KP is the prevalence of the respective KP.

Below are the matching prompt for LLM and the annotation guidelines for workers validating (sentence, KP) pairs:

In this task, you are presented with a question on a product, a key point taken from the summary answering the question, and a sentence taken from a review of that product.

You will be asked to answer the following question: "Does the key point match, i.e, represent an opinion in the review sentence?"

A review sentence might express opinions on multiple aspects. A key point matches a sentence if it captures the gist of the sentence, or is directly supported by a point made in the sentence.

The options are:

- Not At All
- Somewhat Not Well
- Somewhat Well
- Very Well

F Prompts for KP Summary Generation of QQSUM-RAG

We present the instruction-finetuning prompts for KP Summary Generation of QQSUM-RAG in Listing 3.

G Prompts for G-EVAL Evaluation

For implementation of G-EVAL in our KP quality evaluation dimension (§4.2), we specifically customize the model's original prompt for evaluating summary's *relevance* and *redundancy*. While the *relevance* evaluation prompt is customized for evaluating sP/sF/sF1 (Li et al., 2023) between individual generated KPs and the reference KPs, *redundancy* is customized for evaluating RD among generated KPs. We presented our relevance evaluation prompt in Listing 4 and the redundancy evaluation prompt in Listing 5

Listing 3: Prompt for instruction-finetuning QQSUM-RAG's LLM for KP Summary Generation. Please refer to our released code for full prompts.

```
You will be provided with a question and a JSON list of relevant review comments, delimited by triple quotes.
The question asks the opinions of user reviews about a product, and can be answered by the list of comment clusters in the
    provided JSON list. Each element in the JSON has been has been clustered to represent a common opinion answering
    the question, accompanied by the quantity.

You were tasked to generate a quantitative summary that covers all opinions captured in the JSON list in answering the
    questions.

Perform the following actions to solve this task:
- For every element in the JSON list, find the key point that represent the common opinion across the comments of the cluster
- Generate a long-form quantitative summary including all extracted key points and the cluster size, following the below
    template:
'While answering about [Question]:
+ [Cluster size] of comments believe that [Key Point 1]
+ [Cluster size] of comments believe that [Key Point 2]
...'

Below are fundamental rules:
+ Larger cluster means higher support for the key point and with a bigger cluster size, the quantity must be higher
+ Only use number to report the cluster size for each key point, avoiding vague terms (e.g., some, most)
+ Ensure that each key point extracted from a cluster is distinctive and doesn't redundantly cover aspects mentioned in larger
    clusters
```

Listing 4: Zero-shot prompt for G-EVAL relevancy evaluation between generated KPs and reference KPs, supporting sP/sR/sF1 calculation.

```
You will be given one key point, short salient sentence, written to describe user opinion on a product.

Your task is to rate the summary on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and
    refer to it as needed.

Evaluation Criteria:

Relevance (1-5) - selection of important content from the source. The summary should include only important information
    from the source document. Annotators were instructed to penalize summaries which contained redundancies and excess
    information.

Evaluation Steps:

1. Read the key point and the source key point carefully.
2. Compare the key point to the source key point and identify the main points.
3. Assess how well the key point covers the main points of the source key point, and how much irrelevant or redundant
    information it contains.
4. Assign a relevance score from 1 to 5.
```

Listing 5: Zero-shot prompt for G-EVAL redundancy evaluation of generated KPs, supporting RD calculation.

You will be given one key point, short salient sentence, written to describe user opinion on a product.

Your task is to rate the summary on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Redundancy (1–5) – overlapping opinion with the source. The summary should not include semantically similar opinion with the source document. Annotators were instructed to penalize summaries which contained overlapping opinion with the source.

Evaluation Steps:

1. Read the key point and the source key point carefully.
2. Compare the key point to the source key point and identify the main points.
3. Assess how much redundant opinion and information the key point covers that overlap with the source key point
4. Assign a redundancy score from 1 to 5.

H GPT4's Comment-KP Matching Annotation against Human Judgement

To validate `gpt-4-o-mini`'s annotation performance and credibility, we conduct an experiment to measure LLM annotation judgement, as utilized for the KP-comment matching evaluation in our main experiment, in agreement with human (gold) preference. We sampled a subset of 5 queries from the test set in our main experiment and hired workers to annotate the correctness of comment-KP pairs produced as the results of our framework's quantification outcome. Note that these sampled pairs are part of the our main test set and have already been annotated for LLM's labels in our main experiment. For human annotation, we employed 6 MTurk crowd workers on every comment-KP pair, selecting only those with an 80% or higher approval rate and at least 10 approved tasks. Following Bar-Haim et al. (2021), we exclude annotators with Annotator-$\kappa < 0$ for quality control. This score averages all pairwise Cohen's Kappa (Landis and Koch, 1977) for a given annotator, for any annotator sharing at least 50 judgments with at least 5 other annotators. For labelling correct matches, at least 60% of the annotators had to agree that the match is correct, otherwise, it is incorrect. In this experiment, we measured the accuracy, and conducted a Pearson correlation (r) test of `gpt-4-o-mini`'s annotation performance against human judgement, with results reported in Table 7. For r test, we set the null hypothesis as `gpt-4-o-mini`'s and Mturk annotated labels are independent.

From Table 7, we saw signficant small p-value, which indicates strong evidence against the null hypothesis. Importantly, we also recorded Spearman's rank correlation coefficient to be relatively closed to 1. This implies that there is a statistically significant positive correlation between `gpt-4-o-mini` and Mturk annotated labels, which substantiates our decision of using `gpt-4-o-mini` for comment-KP matching evaluation.

Pearson correlation (r)	0.647
p_value	5.342e-16
Accuracy	0.807

Table 7: Performance valiation of GPT4's comment-KP matching annotation against human judgement

Below are the match annotation guidelines for (sentence, KP) pairs:

In this task, you are presented with a question on a product, a key point taken from the summary answering the question, and a sentence taken from a review of that product.

You will be asked to answer the following question: "Does the key point match, i.e, represent an opinion in the review sentence?"

A review sentence might express opinions on multiple aspects. A key point matches a sentence if it captures the gist of the sentence, or is directly supported by a point made in the sentence.

The options are:

- Not At All
- Somewhat Not Well

- Somewhat Well
- Very Well

I Ablation Study Results

We conducted an ablation study to evaluate the impact of KP Summary Generation on QQSUM-RAG, with KP quality and KP-comment matching and factual consistency performance presented in Table 8 and 9 respectively. To this end, we configure QQSUM-RAG $_{\text{Single-KP}}$, a variant that generates one KP at a time for each comment cluster formed by KP-oriented Retrieval.

J Example output of QQSUM-RAG and Baselines

We report the example output of query-relevant comment clusters and KP summary produced by QQSUM-RAG in Table 10 and 11, and further compare top 5 key points, extracted from the summary of QQSUM-RAG and the baselines in Table 12. Overall, QQSUM-RAG stands out for generating KPs with minimal redundancy, higher informativeness, and better alignment with the query.

	ROUGE			BERTScore					BLEURT					G-Eval-4				
	R-1	R-2	R-L	sP	sR	sF1	RD↓	Rel	sP	sR	sF1	RD↓	Rel	sP	sR	sF1	RD↓	Rel
QQSUM-RAG (Ours)																		
+ Mistral	**0.256**	0.061	**0.220**	**0.39**	**0.29**	**0.33**	**0.37**	**0.27**	**0.51**	**0.41**	**0.46**	**0.49**	**0.45**	**4.52**	**4.29**	**4.40**	**2.43**	**4.05**
+ Vicuna	0.222	**0.078**	0.204	0.38	0.26	0.31	0.53	0.25	0.49	0.39	0.44	0.54	0.41	4.47	4.25	4.36	2.45	3.68
QQSUM-RAG $_{\text{Single-KP}}$																		
+ Mistral	0.191	0.035	0.160	0.29	0.22	0.25	0.48	0.22	0.48	0.39	0.43	0.62	0.39	4.21	4.22	4.22	2.51	3.14
+ Vicuna	0.171	0.045	0.154	0.22	0.17	0.19	0.57	0.20	0.48	0.38	0.42	0.66	0.36	4.10	4.12	4.11	2.60	2.87

Table 8: KP-level textual quality evaluation of generated summary between full implementation of QQSUM-RAG and without (w/o) KP Summary Generation. sP, sR and sF1 refer to Soft-Precision, Soft-Recall, and Soft-F1 respectively based on set-level evaluation method against reference KPs in gold answer. G-EVAL-4 asks GPT-4 to score a summary from 1-5.

	KP-Comment Matching				KP-Comment Factual Consistency	
	P	R	F1	QuantErr↓	AlignScore (cluster-level)	AlignScore (retrieval-level)
QQSUM-RAG (Ours)						
+ Mistral	0.694	**0.869**	**0.792**	**04.24**	**0.749**	**0.826**
+ Vicuna	0.538	0.684	0.602	07.83	0.630	0.690
QQSUM-RAG $_{\text{Single-KP}}$						
+ Mistral	0.640	0.520	0.574	17.84	0.682	0.741
+ Vicuna	0.598	0.471	0.527	22.63	0.601	0.660

Table 9: KP-Comment matching performance and factual consistency of generated summary between full implementation of QQSUM-RAG and without (w/o) KP Summary Generation.

Query	How does this *Nikon 24-120mm F4* lens compared with the *24-70mm F2.8* as a general walk around lense?
Query-Relevant Comment Clusters	Cluster1: • I like the 24-70 better but *this lens is a good all around and compact optic for everyday shooting.* • As has been said many times before: "the best lens is the one you will use", and I know *I wouldn not use the 24-70mm F2.8 because itś too heavy and bulky to take on backpacking/camping trips and when traveling abroad.* • This is *the one lens which could replace 24-70 / 2.8, 70-200 2.8 VR II (up to some extent) for "everyday" use.'* • … Cluster2: • I have an upcoming stay in Spain, and *I'm seriously considering taking this lens instead of my AF-S 24-70 because of its size and zoom range.* • My only complaint is the price tag: for a lens that is overall a rather mixed bag (depending on what you're looking for you might be very happy with it, or very disappointed) *it is very expensive*. • *The 24-120 has good reach, good image quality, not heavy, not that expensive for what it can do* (constant f/4 in a zoom is very respectable) and it's also the only usable medium-telephoto FX zoom from Nikon with the VR technology. • *For a 5x zoom to be able to compete with a 3x zoom costing over $500 more(the Nikkor 24-70mm F2.8) should only mean that the 5x zoom is a remarkable lens.* • … Cluster3: • *For one thing, 24 70 is know to have better quality than this one.* • *The range from 70 to 120 is not as important as a better overall quality.* • *This is probably not the best lens to use for portraits because it's just not fast enough (f-stop)*, but for travel, chasing you kids around, or any other every day shooting this lens is perfect. • *The biggest pro for the 24-70mm is the extra 1 stop of light, slightly quicker autofocus speed, and of course the corresponding softer bokeh due to the 1 stop aperture opening.* • …
KP Summary	While comparing the Nikon 24-120mm F4 lens with the 24-70mm F2.8 lens as a general walk-around lens: + 135 of comments believe that the Nikon 24-120mm F4 lens is relatively lightweight and compact, making it easy to carry around and use for extended periods of time. + 11 of comments suggest that the 24-120mm F4 lens has a longer zoom range and is more affordable than the 24-70mm F2.8. + 9 of comments prefer the 24-70mm F2.8 for its better image quality and faster aperture. …

Table 10: Example output of query-relevant comment clusters and KP summary produced by QQSUM-RAG, given a query, i.e., question, from AmazonQ&A. Comment clusters to a particular KP are marked in the same color as the corresponding bullet in the summary. The *relevant opinion* in each comment that directly support the corresponding KP is *italicized*.

Query: How does this *Nikon 24-120mm F4* lens compared with the *24-70mm F2.8* as a general walk around lense?		
Key Point	**Prevalence**	**Matching Comments**
The Nikon 24-120mm F4 lens is relatively lightweight and compact, making it easy to carry around and use for everyday shooting.	135	I like the 24-70 better but *this lens is a good all around and compact optic for everyday shooting.*
		As has been said many times before: "the best lens is the one you will use", and I know *I wouldn not use the 24-70mm F2.8 because itś too heavy and bulky to take on backpacking/camping trips and when traveling abroad.*
The 24-120mm F4 lens has a longer zoom range and is more affordable than the 24-70mm F2.8.	11	I have an upcoming stay in Spain, and *I'm seriously considering taking this lens instead of my AF-S 24-70 because of its size and zoom range.*
		My only complaint is the price tag: for a lens that is overall a rather mixed bag (depending on what you're looking for you might be very happy with it, or very disappointed) *it is very expensive.*
Prefer the 24-70mm F2.8 for its better image quality, faster aperture and better for wide shot.	9	*For one thing, 24 70 is know to have better quality than this one.*
		The range from 70 to 120 is not as important as a better overall quality.

Table 11: Top 3 key points mentioned in the KP summary produced by QQSUM-RAG for answering a query from AMAZONKP. For each key point, we show the prevalence, i.e., number of matching comments (with similar aspects of the same cluster), and two top matching comments. The *relevant opinion* in each comment that directly support the corresponding KP is *italicized*.

Query: How does this *Nikon 24-120mm F4* lens compared with the *24-70mm F2.8* as a general walk around lense?				
QQSUM-RAG	**(Retriever+ LLM)$_{\text{co-trained}}$**	**Contriever + GPT-4-Turbo**	**Contriever + PAKPA**	**Contriever + RKPA-Base**
The Nikon 24-120mm F4 lens is relatively lightweight and compact, making it easy to carry around and use for everyday shooting	The 24-120mm f/4 offers more reach and versatility than the 24-70mm f/2.8.	The 24-120mm lens offers good versatility and value for general use	The 24-120 lens is preferred over the Nikkor 24-70mm F2.8. due to its lighter weight.	The 24-120 is finally at a stage where you can carry it around on your FX camera and have no regrets.
The 24-120mm F4 lens has a longer zoom range and is more affordable than the 24-70mm F2.8.	The 24-120mm f/4 is lighter and more affordable than the 24-70mm f/2.8.	The 24-70mm lens has superior image quality and performance	Best 4+ star walk-around lens on the market.	If you want a 4+ star walk-around lens that covers a great range, this is the best on the market.
Prefer the 24-70mm F2.8 for its better image quality, faster aperture and better for wide shot.	The 24-70mm f/2.8 is a better lens overall.	the 24-70mm lens is preferred for its optical superiority.	The 24-70mm lens is highly recommended for wide shots.	The 24-70mm lens is more expensive but buy it if you need to shoot wide.
The 24-120mm F4 lens has good image quality, with sharpness and contrast that is comparable to the 24-70mm f/2.8	The 24-120mm f/4 is too heavy.	The 24-120mm lens is a more practical choice for everyday use.	The Nikon 24-120 lens has good contrast compared to the Nikon 24-70 lens.	I briefly considered the 24-70, but the extra reach, vibration reduction, and lower price point sold me on this lens.
The 24-120mm F4 lens has good Vibration Reduction (VR) technology that helps to reduce camera shake when taking handheld shots.	The 24-120mm f/4 has image stabilization, which is a significant advantage for handheld shots.	The 24-120mm f/4 has image stabilization for handheld shots.	N/A	N/A
...				

Table 12: Top 5 key points, extracted from the summary of QQSUM-RAG and the baselines, ranked by their prevalence on an example query from AMAZONKP. Overlapping opinions across KPs are highlighted red. KPs lacking of informativeness are highlighted yellow