E-ABSA20K: A Dataset and Propose-and-Verify for Aspect-Based Sentiment Analysis in Long E-commerce Reviews

Anonymous ACL submission

Abstract

Aspect-Based Sentiment Analysis (ABSA) is critical for extracting actionable product insights from e-commerce reviews. However, most public ABSA benchmarks are restricted to short texts and a limited range of domains, and therefore underrepresent the challenges posed by real-world reviews—where multiple aspects co-occur, colloquial and noisy expressions are common, and evidence must often be aggregated across sentences in long contexts.

We introduce **E-ABSA20K**, a multi-domain dataset of 20K reviews from four product categories (WomenBags, Dresses, Cosmetics, and Furniture), annotated with review-level sentiment quads. Compared to existing benchmarks, E-ABSA20K contains substantially longer and more aspect-dense reviews, averaging 63.9 words and 6.0 quads per review. We further propose a two-stage *propose-and-verify* framework for review-level quadruple extraction (*target, aspect, opinion, sentiment*). The first stage generates high-recall candidates under strict schema constraints, while the second stage conducts explicit grounding, scope, and modality verification, followed by review-level consolidation to mitigate hallucinations and scope leakage in long reviews. Experiments across multiple Qwen3 model sizes demonstrate that our approach consistently outperforms single-stage prompting (with and without chain-of-thought) as well as competitive ABSA extraction baselines, improving quad-level micro-F1 (μF) and robustness on discourse-hard cases such as comparisons and conditionals.

1 Introduction

E-commerce platforms accumulate massive amounts of user reviews that describe products along many dimensions (e.g., material, craftsmanship, size, odor, logistics, and after-sales service). Traditional sentiment analysis assigns a single polarity to a sentence or a whole review and cannot answer the more actionable question: *which aspects are praised or criticized?* Aspect-Based Sentiment Analysis (ABSA) addresses this by extracting structured aspect-level opinions and their polarities (Zhang et al., 2022). Despite steady

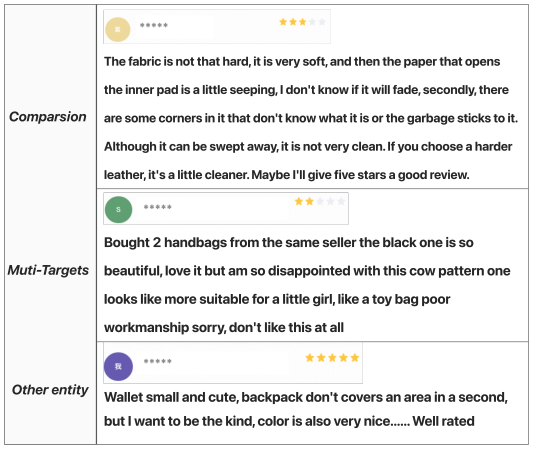

Figure 1: Illustrative examples from E-ABSA20K highlighting the complexities of review-level ABSA.

progress on benchmark datasets, review-level ABSA in real e-commerce settings remains challenging for three reasons. (1) **Long reviews and cross-sentence evidence.** Real reviews often contain transitions, comparisons, and usage narratives, where the target, aspect, and sentiment may be scattered across sentences. (2) **Multi-aspect, fine-grained targets.** A single review can mention many aspects at different granularities (e.g., *strap* vs. *hardware*), which increases omission and mismatch errors. (3) **Hallucinations and scope leakage under single-pass generation.** When prompted to produce structured ABSA outputs in one pass (Ding et al., 2024; Simmering and Huoviala, 2023; Wu et al., 2025; Wang et al., 2025), LLMs frequently generate ungrounded or out-of-scope quads (e.g., extracting opinions about alternative products, hypothetical conditions, or unrelated service entities), especially in long, discourse-rich reviews. While larger

backbones alleviate these issues, smaller and medium-sized models—which are often preferred in deployment—tend to exhibit them more severely. We illustrate these representative challenges in Fig. 1, contrasting the ground-truth quads with common errors (e.g., scope leakage and entity confusion) typically made by single-pass LLM extractors on our E-ABSA20K dataset.

To study these challenges and develop more robust extraction methods, we make three contributions:

1. We release **E-ABSA20K**, a multi-domain e-commerce ABSA dataset with 20K review-level quad annotations across four categories (WomenBags, Dresses, Cosmetics, Furniture), featuring long and aspect-dense reviews.

2. We propose a **two-stage propose-and-verify** framework that decouples high-recall candidate proposal from explicit *grounding/scope/modality* verification, followed by review-level consolidation enforcing a unique (t, a) key.

3. We complement standard quad-level F1 with a **discourse-hard evaluation** tailored to long reviews, reporting performance on trigger-based subsets for comparisons, conditionals, uncertainty, and other-entity mentions.

Overall, our approach is not merely a stronger extractor; rather, it improves robustness by *externalizing* grounding and scope decisions and enforcing review-level consistency through explicit verification and consolidation.

2 Related Work

ABSA Datasets. The development of Aspect-Based Sentiment Analysis (ABSA) datasets has steadily progressed towards greater complexity and realism. Early benchmarks, such as those from SemEval-2014 (Pontiki et al., 2014), primarily offered sentence-level annotations within narrow domains like restaurants and laptops. Subsequent efforts, notably SemEval-2016 Task 5 (Pontiki et al., 2016), expanded the scope to review-level contexts and multiple languages, establishing a more comprehensive annotation framework.

Recent datasets continue this trend by targeting more complex, joint extraction of sentiment elements. For instance, OATS (Chebolu et al., 2024a) and ROAST (Chebolu et al., 2024b) focus on review-level tuples and introduce metrics for annotation consistency. Shoes-ACOSI (Peper et al., 2024) advances the task for long e-commerce texts by proposing quintuples that include tags for implicit opinions. In parallel, works like M-ABSA address the scarcity of parallel multilingual resources. Despite this progress, a gap remains for large-scale datasets that specifically capture the challenges of long, colloquial e-commerce reviews with high aspect density and cross-sentence dependencies. Our E-ABSA20K dataset is designed to fill this void.

LLM-based Aspect-Based Extraction. The advent of Large Language Models (LLMs) (Brown et al., 2020; Touvron et al., 2023) has fundamentally reshaped approaches to structured information extraction, including ABSA. Two primary application paradigms have emerged.

The first is single-pass generation, where an LLM is prompted to directly output all structured quads from a review in one go (Ding et al., 2024; Simmering and Huoviala, 2023). While straightforward, this approach often struggles with the complexities of long, discourse-rich texts. It is prone to generating ungrounded information (hallucinations) and suffering from scope leakage—erroneously extracting opinions about compared products, hypothetical scenarios, or unrelated service entities (Wang et al., 2025). These issues are particularly severe for smaller, deployment-friendly models, motivating the need for more robust extraction frameworks.

To mitigate these challenges, a second paradigm has emerged: multi-step decomposition and verification. This includes methods like Chain-of-Thought (CoT) prompting, which encourages step-by-step reasoning before the final output. More explicitly, propose-and-verify pipelines first generate a set of high-recall candidates and then employ a second model or a verification procedure to filter unsupported outputs, reduce hallucinations, and enforce task-specific constraints. Our work builds upon this direction but makes several key contributions tailored to the challenges of review-level ABSA. Specifically, our second stage performs explicit verification of grounding, scope, and modality. Furthermore, it introduces a crucial review-level consolidation step that enforces a unique key for each (target, aspect) pair. This final consolidation is critical for ensuring consistency and resolving conflicting opinions scattered across long reviews—a nuance often overlooked by general-purpose verification methods.

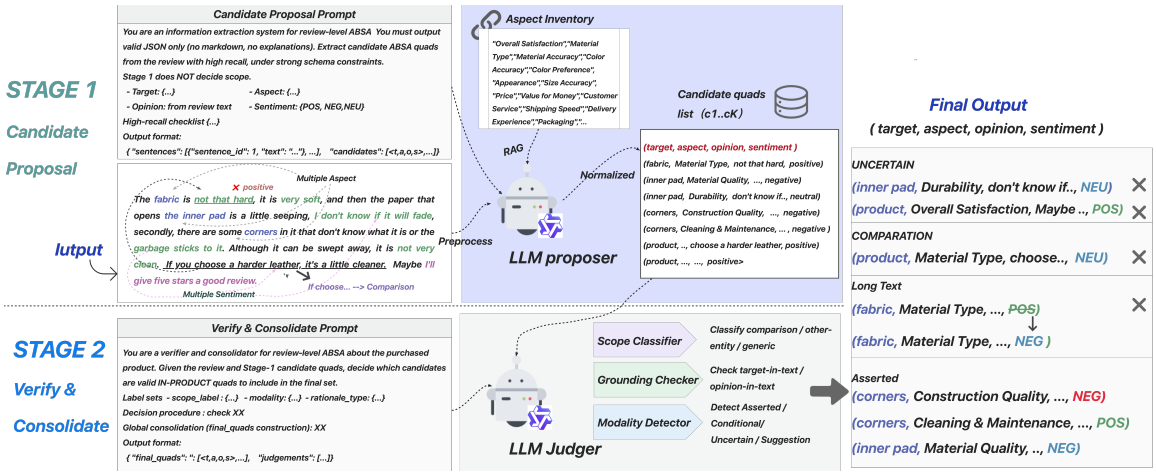

Figure 2: **Overview of the proposed two-stage framework.** Stage 1 proposes high-recall candidates; Stage 2 verifies scope/grounding and consolidates into unique (t, a) keys.

3 E-ABSA20K Dataset

3.1 Task Definition

Given a review text x, the goal is to output a set of quads (Zhang et al., 2021):

$$\mathcal{Y} = \{(t_i, a_i, o_i, s_i)\}_{i=1}^{n}, \tag{1}$$

where t_i is the opinion target, a_i is an aspect from a pre-defined inventory, o_i is the opinion phrase grounded in the review, and $s_i \in \{\texttt{positive}, \texttt{negative}, \texttt{neutral}\}$.

We evaluate both (i) quad extraction accuracy and (ii) faithfulness/consistency under long contexts.

3.2 Data Collection and Domains

We collected raw user reviews from real e-commerce platforms across four product categories: *WomenBags*, *Dresses*, *Cosmetics*, and *Furniture*. We choose these domains because they exhibit distinct writing styles and aspect distributions, and they naturally include both product-focused descriptions and service-related content (e.g., shipping, packaging, and after-sales).

To make the data suitable for annotation and modeling, we apply a unified preprocessing pipeline: (i) deduplication and anomaly filtering to remove duplicate, spam/template-like, and extremely short reviews; (ii) text normalization to standardize encoding and full-/half-width characters and to remove noisy control characters while preserving colloquial expressions and punctuation; and (iii) privacy-preserving anonymization to mask potentially identifying information (e.g., phone numbers, addresses, and order IDs). After preprocessing, E-ABSA20K contains 20,000 reviews as the base corpus.

3.3 Annotation Process

We adopt a weak-supervision annotation pipeline to balance quality and cost, combining a human-labeled seed set, multi-LLM single-stage labeling, voting-based aggregation, and judge-model-based verification. The final annotation target for each review x is a set of grounded quads $\mathcal{Y} = \{(t, a, o, s)\}$, where t is the opinion target, a is an aspect category from a domain-specific inventory, o is the opinion phrase, and $s \in \{\texttt{positive}, \texttt{negative}, \texttt{neutral}\}$.

Aspect inventories and schema constraints. To reflect domain-specific granularity, we define an aspect inventory per category: *WomenBags* (34), *Dresses* (29), *Cosmetics* (42), and *Furniture* (33) aspects. During annotation (human and model), we enforce the following constraints: (i) a must be an exact match from the corresponding inventory; (ii) t and o must be *verbatim spans* from the review text (i.e., substring-grounded); (iii) we cap the number of quads per review to K=20 to limit over-generation and to keep the annotation and verification workload bounded.

Step 1: Human-labeled gold set. We manually annotate 760 reviews to form a gold set $\mathcal{D}_{\text{human}}$, following unified guidelines on target granularity, aspect assignment, and polarity boundaries (especially for neutral vs. weak sentiment). This set is used for calibration and quality reporting.

Step 2: Multi-LLM single-stage labeling. For each remaining review $x \in \mathcal{D}_{\text{auto}}$, we query four strong LLMs (GPT-4-turbo, Gemini-2.5-Pro, Deepseek-v3.1, Deepseek-R1) with the same *single-stage* prompt and the above schema constraints to obtain four candidate quad sets $\{\hat{\mathcal{Y}}^{(m)}\}_{m=1}^{4}$, where $\hat{\mathcal{Y}}^{(m)}$ denotes the prediction from model m.

Step 3: Normalization, alignment, and voting. Because different models may use slightly different surface forms, we normalize targets and opinions by case-folding and whitespace/punctuation normalization, while preserving span grounding. We then align candidates primarily by the review-level key (t, a).[1] A candidate is marked as *high-consensus* if it reaches a strict majority agreement on inclusion and sentiment across the four models; otherwise it is marked as *low-consensus*.

Step 4: Judger verification for low-consensus cases. Approximately 28% of reviews contain at least one low-consensus candidate. We send all low-consensus candidates to an independent judger (Gemini-2.5-Pro), which verifies whether each quad is supported by explicit textual evidence and satisfies the schema constraints, outputting ACCEPT/REJECT. Accepted candidates are merged with the high-consensus set to form the final annotation.

3.4 Data Quality

We assess label quality by measuring agreement between the aggregated annotations and a human-labeled gold set $\mathcal{D}_{\text{human}}$ (760 reviews). On this gold set, the aggregated labels achieve quad-level micro Precision/Recall/F1 of 0.930/0.810/0.866 under strict span grounding and aspect-inventory matching. We additionally conduct expert verification on a random sample for sanity checking; this verification is used only for quality assessment and does not affect training or evaluation labels. We discuss remaining noise sources (e.g., boundary ambiguity between product aspects and service-related mentions) in §7.

3.5 Dataset Characteristics

Table 1 presents the key dataset features and compares them with commonly used datasets such as Restaurant-ACOS and Laptop-ACOS.

Our E-ABSA20K dataset possesses the following three characteristics:

1. Extremely long texts and scattered evidence: On average, each review contains 63 words and 6 aspects, which is much more complex than reviews for Laptop-ACOS and Restaurant-ACOS.

2. Coexistence of multiple aspects and susceptibility to aspect mismatch: A single review typically involves multiple aspects simultaneously, and the sentiments may be opposite (mixed sentiments across aspects within a single review).

3. Strong domain specificity and difficult label boundaries: The four categories differ significantly in aspect distribution and expression habits, containing a large number of domain terms and colloquial weak sentiment expressions, leading to more blurred boundaries between neutral and weak positive/weak negative.

4 Method: Two-stage Propose-and-Verify ABSA

4.1 Overview

We study review-level quad-based ABSA (Fig. 2), where the goal is to extract a set of sentiment quads $\mathcal{Y} = \{(t_i, a_i, o_i, s_i)\}_{i=1}^{n}$ from a review x. Here t denotes the opinion target (PRODUCT or an explicit product type/part mentioned in x), a is an aspect category chosen from a fixed inventory, o is an opinion phrase, and $s_i \in \{\texttt{positive}, \texttt{negative}, \texttt{neutral}\}$. Our aspect inventory list can be found in Appendix 4.

We propose a two-stage **propose-and-verify** framework that decouples (i) high-recall proposal from (ii) strict verification and (iii) review-level consolidation. Importantly, while Stage 2 may optionally produce intermediate *judgments* for analysis and distillation, **all task metrics are computed solely on the final output quads** $\mathcal{Y}$.

4.2 Stage 1: High-recall Proposal under Schema Constraints

Stage 1 generates a set of candidates $\tilde{\mathcal{Y}} = \{(t, a, o, s, e)\}$, where e is an evidence span copied from the review. Stage 1 is optimized for recall and intentionally avoids scope filtering (e.g., it may include candidates from comparisons, conditionals, or uncertain statements, which are later verified).

To reduce ungrounded extractions, we enforce hard schema constraints: (i) a must be an exact match from the aspect inventory list; (ii) $t \in$

[1] When multiple candidates share the same (t, a), we keep the one with the strongest cross-model support and prefer spans that exactly match the source text; ties are sent to the judger.

Source	Domain/Dataset	#Reviews	Avg. words	P90 words	Quads/Rev
Our E-ABSA20K (Long Reviews)	WomenBags	5,000	57.9	85	5.3
	Dresses	5,000	64.5	92	6.4
	Cosmetics	5,000	66.3	97	6.0
	Furniture	5,000	66.8	98	6.3
	All (E-ABSA20K)	**20,000**	**63.9**	**93**	**6.0**
Prior Benchmarks (Short Reviews)	Restaurant-ACOS	2,221	15.1	28	1.6
	Laptop-ACOS	4,010	15.7	28	1.4

Table 1: Statistics of our E-ABSA20K dataset compared with prior short-text ABSA benchmarks. E-ABSA20K reviews are substantially longer and denser in aspects.

$\{\text{PRODUCT}\} \cup \mathcal{P}(x)$, where $\mathcal{P}(x)$ denotes product types/parts explicitly mentioned in x (no invented targets; if uncertain, fall back to PRODUCT); and (iii) e must be a verbatim substring of x supporting the candidate. We cap the number of candidates per review to K=20 to bound inference cost and the verification workload.

4.3 Stage 2: Verification and Review-level Consolidation

Stage 2 takes the review and the Stage 1 candidates $(x, \tilde{\mathcal{Y}})$ and outputs the final set of quads $\mathcal{Y}$.

Verification. For each candidate, Stage 2 checks (1) **grounding** (whether $t/o/s$ are supported by the evidence span and the review), and (2) **scope** (whether the candidate refers to the purchased product rather than alternatives/comparisons, generic statements, or unrelated entities). It may also tag the **modality** (asserted vs. conditional/suggestion/uncertain) to avoid over-committing to non-asserted claims. For interpretability and distillation, Stage 2 can additionally output per-candidate *judgments* (e.g., scope/modality/include decisions). These judgments are *not* used in evaluation; they are auxiliary metadata.

Consolidation. Stage 2 then consolidates the retained candidates into $\mathcal{Y}$ by enforcing a unique key per (t, a). We define a normalization function $\text{norm}(t)$ to reduce superficial target variants: $\text{norm}(t) = \text{PRODUCT}$ if t is PRODUCT or a product type (e.g., "bag"); otherwise $\text{norm}(t)$ is the lowercase head of the explicit part name (e.g., "zipper", "strap"). We then use the consolidated key $k = (\text{norm}(t), a)$. For candidates sharing the same key, we merge opinion phrases (concatenated with "; ") and resolve polarity conflicts by setting $s = \texttt{negative}$ when both positive and negative evidence are present.

4.4 Single-stage Baseline (Output Constraints)

To ensure a fair comparison, our single-stage baseline uses the **same task schema** as the two-stage system: it must output a set of quads (t, a, o, s) with a restricted to the same aspect inventory list and t restricted to PRODUCT or explicit product types/parts mentioned in the review. We also enforce structured, machine-parseable outputs (JSON-only in our prompts) and apply the same post-processing for parsing failures across methods. This isolates the effect of the two-stage decomposition from differences in output formatting or label space.

4.5 Prompt Design (Summary)

Stage 1 is prompted to output *JSON-only* candidates under the above constraints and to keep comparison/conditional/hedged mentions for recall. Stage 2 is prompted to follow a fixed decision order (grounding $\rightarrow$ scope $\rightarrow$ modality $\rightarrow$ include) and then apply global consolidation with the key $k = (\text{norm}(t), a)$; full prompts are provided in Appendix 5.

5 Experimental Setup

5.1 Task, Data, and Metrics

Given a review, models output sentiment quads (t, a, o, s). We evaluate quad-level micro Precision/Recall/F1. Experiments use four categories (*WomenBags, Dresses, Cosmetics, Furniture*), each with 5k reviews split 7:1:2, and we report category-wise results and Avg-4cate (macro-average of micro-F1 scores across four categories).

5.2 Models and Inference Settings

To evaluate the performance gains of the two-stage approach over the single-stage approach across different model sizes, we choose the Qwen3

family, spanning model size from 4B to 235B (4B/8B/14B/32B/235B). We use zero-shot and few-shot (4 demonstrations) prompting, sharing the same prompt family between single-stage and two-stage. All experiments use temperature 0.6.

Why Qwen3? We chose the Qwen3 family for three reasons: its consistent size spectrum (4B to 235B) enables controlled scaling analysis; its strong instruction-following capabilities reduce formatting failures; and its open-source ecosystem ensures reproducibility. Our method is model-agnostic, but Qwen3 serves as a strong, representative backbone for our study.

5.3 Hard-case and Efficiency Evaluation

Hard-case subsets are constructed via surface triggers (conditional/comparison/uncertain/other-entity). Efficiency is measured by average total tokens per review (summed across calls for two-stage) and reported as Pareto frontiers on a sampled *WomenBags* subset. We use these trigger-based subsets only for post-hoc evaluation; they are not used for training, prompt selection, or hyperparameter tuning.

5.4 Specialized Inference and Training Setups

CoT Prompting. To assess the impact of reasoning, we compare direct JSON output (*no-think*) with a variant that encourages internal reasoning (*think*). For our two-stage method, reasoning is enabled only in Stage 2.

SFT and Distillation. We conduct supervised fine-tuning (SFT) on Qwen3-8B to evaluate different training strategies, including single-stage SFT, fine-tuning only Stage 2, and fine-tuning both stages. We also explore distilling verification behavior from a strong teacher (Qwen3-235B) into the 8B model.

6 Results

6.1 Overall Performance

Table 2 reports the main results. Across all model sizes, categories, and prompting regimes, the two-stage pipeline consistently improves micro-F1 over single-stage, with larger gains for smaller backbones. The performance gains of the two-stage approach are particularly pronounced on our proposed E-ABSA20K dataset, which is expected given its longer texts and more scattered evidence compared to the other benchmarks. The improvements are typically driven by higher recall while maintaining comparable precision, consistent with our design: Stage 1 prioritizes coverage and Stage 2 filters out-of-scope or unsupported candidates and consolidates duplicates. We also conducted experiments under few-shot conditions. The results are detailed in Appendix A.4

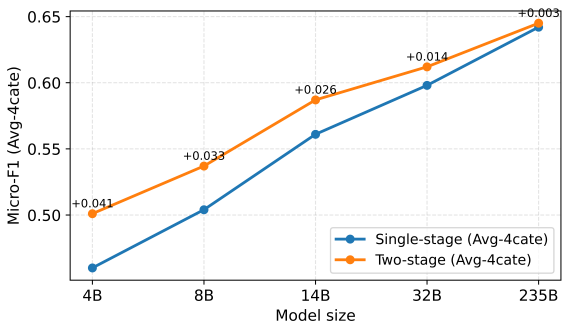

Figure 3: **Scaling behavior and gain convergence of two-stage inference (Avg-4cate, zero-shot):** Micro-F1 of single-stage vs. two-stage across Qwen3 model sizes.

6.2 Scaling Behavior: Diminishing Two-stage Gains with Larger Backbones

Figure 3 studies how the benefit of our two-stage *propose-and-verify* decomposition changes with backbone scale. Both single-stage and two-stage extraction improve steadily as model size increases. However, the absolute improvement brought by two-stage inference shrinks with larger backbones. This trend suggests a convergence-like behavior: as the backbone becomes stronger, it internally learns part of the grounding and scope control that Stage 2 explicitly enforces, leaving less room for additional gains. Conversely, for smaller and medium-sized models, explicit verification and review-level consolidation provide a substantial robustness boost, which explains why the two-stage pipeline can narrow the performance gap to much larger single-stage models under practical deployment constraints.

6.3 Efficiency: Performance vs. Latency

Figure 4 reports the performance–latency Pareto frontier on *WomenBags*, with μF on the y-axis and average end-to-end request time (RT) on the x-axis. For single-stage, RT corresponds to one model call per review; for two-stage, RT is the sum of two serial calls, $\mathrm{RT} = \mathrm{RT}_{\mathrm{S1}} + \mathrm{RT}_{\mathrm{S2}}$. Overall, two-stage improves μF with a moderate latency increase due to the additional verification and con-

Model	SFT	Method	WomenBags	Dresses	Cosmetics	Furniture	Restaurant	Laptop
Qwen3-4B	w/o	single-stage	0.482	0.484	0.425	0.451	0.271	0.139
		two-stage	**0.521**↑	**0.524**↑	**0.460**↑	**0.500**↑	**0.285**↑	**0.151**↑
	w/	single-stage	0.512	0.518	0.458	0.497	0.609	0.410
		two-stage	**0.552**↑	**0.559**↑	**0.487**↑	**0.532**↑	**0.618**↑	**0.419**↑
Qwen3-8B	w/o	single-stage	0.530	0.526	0.464	0.495	0.389	0.199
		two-stage	**0.554**↑	**0.559**↑	**0.497**↑	**0.540**↑	**0.392**↑	**0.213**↑
	w/	single-stage	0.567	0.564	0.509	0.528	0.611	0.417
		two-stage	**0.595**↑	**0.593**↑	**0.527**↑	**0.567**↑	**0.615**↑	**0.421**↑
Qwen3-14B	w/o	single-stage	0.582	0.597	0.516	0.550	0.439	0.274
		two-stage	**0.599**↑	**0.628**↑	**0.543**↑	**0.578**↑	**0.450**↑	**0.278**↑
	w/	single-stage	0.601	0.625	0.538	0.579	0.612	0.414
		two-stage	**0.621**↑	**0.658**↑	**0.565**↑	**0.587**↑	**0.615**↑	0.412
Qwen3-32B	w/o	single-stage	0.629	0.646	0.554	0.565	0.443	0.267
		two-stage	**0.640**↑	**0.670**↑	**0.572**↑	**0.586**↑	**0.446**↑	**0.271**↑
Qwen3-235B	w/o	single-stage	0.641	0.672	0.597	0.629	0.471	0.278
		two-stage	**0.653**↑	**0.689**↑	**0.603**↑	**0.637**↑	**0.482**↑	0.256

Table 2: Micro-F1 for single-stage vs two-stage (zero-shot) on E-ABSA20K and open-source datasets.

solidation call, and it *reduces the need for very large backbones* under practical latency budgets (e.g., Qwen3-32B two-stage is more accurate than Qwen3-32B single-stage while remaining faster than Qwen3-235B single-stage). Token statistics are reported in Appendix A.

6.4 Hard-case Stratified Evaluation

Table 3 shows that two-stage yields the largest gains on conditional and comparison subsets, mainly through improved precision, indicating more accurate scope attribution and grounding-based filtering in discourse-hard contexts.

6.5 Chain-of-Thought (CoT) Prompting

Our results show that CoT primarily boosts the recall of single-stage prompting, especially for smaller models. However, it yields only marginal gains for our two-stage framework, suggesting that our explicit decomposition already externalizes the key reasoning steps. Full results are in Appendix Table 7.

6.6 Supervised Fine-Tuning (SFT)

SFT consistently improves performance over zero-shot prompting. Notably, our two-stage SFT still outperforms single-stage SFT, especially on hard cases, by directly optimizing the verification and consolidation logic. This indicates that the architectural benefits of our framework persist even with supervised training. Detailed comparisons are available in Appendix Table 11.

SFT vs. two-stage. Even after SFT, two-stage remains consistently better than single-stage, especially on hard cases; detailed comparisons are reported in Appendix A.6.

6.7 Why Two-stage Works

Taken together, the main results (Table 2), hard-case analysis (Table 3), and ablations (Table 8) provide converging evidence that two-stage inference improves review-level ABSA by decomposing the task into (i) high-recall proposal, (ii) explicit scope/grounding/modality verification, and (iii) review-level consolidation with a unique (t, a) key.

First, Stage 1 prioritizes recall under strict schema constraints, which reduces omission errors for fine-grained targets/aspects that are easily missed in single-pass generation. Second, Stage 2 explicitly verifies grounding and scope, which primarily reduces false positives in discourse-hard settings: in Table 3, two-stage achieves large precision gains on conditional and comparison subsets (e.g., for Qwen3-8B, P improves from $0.567 \rightarrow 0.659$ on conditionals and $0.549 \rightarrow 0.651$ on comparisons), indicating more accurate scope attribution and evidence-based filtering. Third, consolidation enforces review-level consistency by merging duplicates and resolving conflicting opinions under the same (t, a); removing this step leads to a clear drop in μF in the ablation study (Table 8).

Finally, the efficiency results in §6.3 show that these gains translate into a better performance–

Hard-case subset (quad-level)	Qwen3-8B (Avg over 4 categories)						Qwen3-14B (Avg over 4 categories)					
	single-stage			two-stage			single-stage			two-stage		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Conditional (if/choose/would)	0.567	0.463	0.510	0.659	0.463	0.544	0.584	0.549	0.566	0.671	0.550	**0.605**
Comparison (than/instead/vs)	0.549	0.457	0.499	0.651	0.464	**0.542**	0.570	0.539	0.554	0.662	0.548	**0.599**
Other-entity (shipping/service)	0.572	0.477	0.520	0.659	0.489	**0.561**	0.596	0.566	0.581	0.673	0.572	**0.619**
Uncertain (maybe/not sure)	0.545	0.438	0.486	0.555	0.445	0.494	0.562	0.505	0.532	0.571	0.513	0.541

Table 3: Hard-case stratified evaluation: results are macro-averaged over four categories (WomenBags, Dresses, Cosmetics, Furniture).

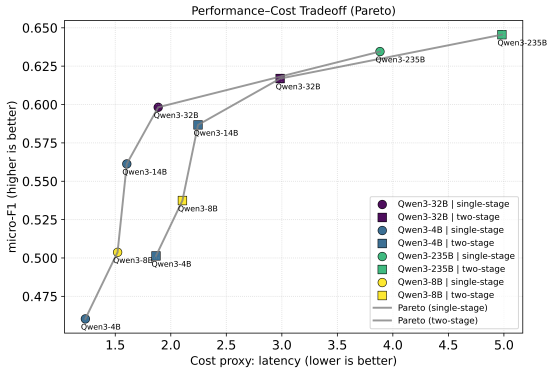

Figure 4: Performance–latency Pareto frontier on the WomenBags category (micro-F1 vs. average end-to-end RT) for single-stage and two-stage ABSA.

latency trade-off in practice: two-stage improves accuracy with a moderate latency overhead and can reduce the need for very large backbones under realistic budgets (see Figure 4).

7 Limitations

E-ABSA20K is partially annotated via multi-LLM voting and judger verification, which may leave residual noise, especially for borderline service-related mentions. Our experiments focus on the Qwen3 family and trigger-based hard-case subsets, so absolute numbers may vary across backbones and more implicit discourse phenomena. Two-stage inference incurs extra latency and should be weighed against deployment constraints.

8 Ethics and Privacy

We collect reviews from public e-commerce platforms and anonymize potentially identifying information (e.g., phone numbers, addresses, order IDs). We will release the dataset for research use only and follow platform terms and applicable data protection policies.

9 Conclusion

We introduced **E-ABSA20K**, a multi-domain dataset for review-level, quad-based ABSA in real-world e-commerce scenarios. Compared with prior public resources, E-ABSA20K contains substantially longer and noisier reviews with more fine-grained aspect coverage, making it well-suited for studying long-context extraction, cross-sentence evidence, and scope confusion (e.g., comparisons and conditionals).

To address these challenges, we proposed a **two-stage propose-and-verify framework** that decomposes ABSA into (i) high-recall candidate proposal under strict schema constraints, (ii) explicit grounding/scope/modality verification, and (iii) review-level consolidation that enforces a unique key per (t, a). Experiments across four product categories and five Qwen3 model scales show consistent improvements over single-stage prompting, with the largest gains appearing on discourse-hard subsets (comparison/conditional/other-entity) where single-pass generation is prone to scope leakage and ungrounded outputs. Notably, two-stage inference substantially narrows the gap to much larger backbones and yields a more favorable accuracy–latency trade-off for deployment. Our latency-based Pareto analysis further indicates that the two-stage decomposition can improve accuracy while reducing reliance on very large backbones under practical deployment budgets. Finally, we demonstrated that Stage 2 decisions can be distilled from a strong teacher into a smaller verifier, providing an additional path toward robust and cost-effective deployment.

In future work, we will strengthen the annotation protocol and expand evaluation to more backbones and multilingual settings, as well as investigate grounding constraints and learning-based consolidation to improve faithfulness in long reviews.

References

Tom B. 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 M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Siva Uday Sampreeth Chebolu, Franck Dernoncourt, Nedim Lipka, and Thamar Solorio. 2024a. Oats: Opinion aspect target sentiment quadruple extraction dataset for aspect-based sentiment analysis. *Preprint*, arXiv:2309.13297.

Siva Uday Sampreeth Chebolu, Franck Dernoncourt, Nedim Lipka, and Thamar Solorio. 2024b. Roast: Review-level opinion aspect sentiment target joint detection for absa. *Preprint*, arXiv:2405.20274.

Xuanwen Ding, Jie Zhou, Liang Dou, Qin Chen, Yuanbin Wu, Arlene Chen, and Liang He. 2024. Boosting large language models with continual learning for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4367–4377, Miami, Florida, USA. Association for Computational Linguistics.

Joseph J Peper, Wenzhao Qiu, Ryan Bruggeman, Yi Han, Estefania Ciliotta Chehade, and Lu Wang. 2024. Shoes-ACOSI: A dataset for aspect-based sentiment analysis with implicit opinion extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15477–15490, Miami, Florida, USA. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Paul F. Simmering and Paavo Huoviala. 2023. Large language models for aspect-based sentiment analysis. *Preprint*, arXiv:2310.18025.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Cristian Danescu-Niculescu-Mizil, Jim Devlin, Christopher Donahue, William Fedus, Alexey Fodor, Hubert Fort, Naman Goyal, Ankit Gupta, Abhishek Hazra, and 65 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Qianlong Wang, Keyang Ding, Hengxin Gao, Hui Wang, and Ruifeng Xu. 2025. Error comparison optimization for large language models on aspect-based sentiment analysis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18630–18646, Vienna, Austria. Association for Computational Linguistics.

Chengyan Wu, Bolei Ma, Zheyu Zhang, Ningyuan Deng, Yanqing He, and Yun Xue. 2025. Evaluating zero-shot multilingual aspect-based sentiment analysis with large language models. *International Journal of Machine Learning and Cybernetics*, 16(10):8079–8101.

Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. 2021. Aspect sentiment quad prediction as paraphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9209–9219, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *Preprint*, arXiv:2203.01054.

A Appendix

A.1 Aspect Inventory

Table 4 summarizes the complete aspect inventories for the four domains in E-ABSA20K: WomenBags (34 aspects), Dresses (29 aspects), Cosmetics (42 aspects), and Furniture (33 aspects). These inventories constrain both annotation and model outputs, requiring each aspect to be an exact match from the predefined list of the corresponding domain.

A.2 Stage 2 Distillation (Teacher–Student) for a Deployable Verifier

Table 9 shows that **Stage 2 distillation** (Qwen3-235B → Qwen3-8B) improves Avg-4cate micro-F1, largely via precision gains, consistent with reduced hallucinations and scope leakage. We use teacher-proposed candidates during distillation to avoid candidate-regime mismatch; hard-case results are in Table 10.

A.3 Ablation

Table 8 confirms that both Stage 2 verification and global consolidation contribute: Stage 1 alone increases recall but substantially hurts precision, verification restores precision while retaining recall, and consolidation further improves consistency by reducing duplicates and conflicts.

A.4 Few-shot result

Few-shot result is shown in Figures 5 and 6

A.5 CoT Prompting and Stage-2 Reasoning

We evaluate whether chain-of-thought (CoT) style reasoning improves structured ABSA extraction and how it interacts with our two-stage decomposition. All CoT experiments use the same decoding configuration as the main setup (temperature 0.6) and report quad-level micro-F1 macro-averaged over the four categories (Avg-4cate).

Single-stage CoT. We compare a *no-think* prompt, which instructs the model to provide a direct JSON output, with a *think* variant. The "think" prompt encourages the model to perform internal reasoning steps before generating the final output, while still requiring the final output to be JSON-only. The task schema, aspect inventory, and output constraints are kept identical across both settings to ensure a fair comparison.

Two-stage CoT (Stage2-think). For our two-stage framework, Stage 1 is always kept as a *no-think* high-recall candidate proposal step. We then optionally enable *think* mode **only in Stage 2** (verification and consolidation). This setup allows us to isolate whether explicit reasoning is more beneficial for the initial proposal or for the subsequent verification, grounding, and consolidation steps.

A.6 Supervised Fine-tuning (SFT) and Distillation Protocols

We conduct a controlled comparison of supervised training strategies on the Qwen3-8B backbone, under the same aspect inventory and JSON output schema. We denote the Stage 1 proposer as S_1 and the Stage 2 verifier+consolidator as S_2. Superscripts indicate the model instance: T for the teacher model, S for the student model, and dep for the deployed proposer used at inference time. We distill only Stage 2, hence S_2^{S} denotes the student verifier. For a review x, $S_1(x)$ outputs up to K=20 candidates $\tilde{\mathcal{Y}}$, and $S_2(x, \tilde{\mathcal{Y}})$ outputs the final quads $\mathcal{Y}$.

Single-stage SFT. We fine-tune a one-pass extractor to directly map an input review x to its final quad set $\mathcal{Y}^*$, where $\mathcal{Y}^* = \{(t, a, o, s)\}$ is the ground-truth annotation from our dataset. The training objective is to learn the mapping $x \rightarrow \mathcal{Y}^*$.

Two-stage SFT (Stage2-only). In this setup, we keep Stage 1 *fixed* as a zero-shot prompter and only fine-tune Stage 2. Crucially, the candidates $\tilde{\mathcal{Y}}$ used for training Stage 2 are generated by the *same* Stage 1 configuration used at inference time (the "deployed" proposer, S_1^{dep}). Formally, for each training review x, we first generate candidates $\tilde{\mathcal{Y}}^{\mathrm{dep}} = S_1^{\mathrm{dep}}(x)$ and then train Stage 2 on pairs of $\big((x, \tilde{\mathcal{Y}}^{\mathrm{dep}}), \mathcal{Y}^*\big)$. This design aligns the candidate distribution between training and inference, mitigating distribution shift for the verifier.

Two-stage SFT (Stage1+Stage2). Here, we fine-tune both stages. Stage 1 is trained to produce higher-quality, schema-valid candidates (still capped at $K = 20$), and Stage 2 is trained as described above, using the candidates from the fine-tuned Stage 1.

Stage 2 Distillation (Teacher–Student). We distill the teacher verifier's behavior into a smaller student verifier (Qwen3-8B) using a strong teacher (Qwen3-235B, under *think*). To avoid proposal-regime mismatch, we train the student Stage 2 on candidates proposed by the teacher Stage 1. Concretely:

1. **Teacher proposal:** generate candidates $\tilde{\mathcal{Y}}^{\mathrm{T}} = S_1^{\mathrm{T}}(x)$.
2. **Teacher verification:** generate pseudo labels $\mathcal{Y}^{\mathrm{T}} = S_2^{\mathrm{T}}(x, \tilde{\mathcal{Y}}^{\mathrm{T}})$.
3. **Student training:** train the student verifier S_2^{S} on pairs $\big((x, \tilde{\mathcal{Y}}^{\mathrm{T}}), \mathcal{Y}^{\mathrm{T}}\big)$.

This trains the student to verify the same candidate distribution that the teacher verifier sees.

Supervision Sources. For all SFT experiments, the ground truth $\mathcal{Y}^*$ refers to the aggregated/voted quads from the E-ABSA20K dataset. For distillation, the supervision signal is the teacher's output $\mathcal{Y}^{\mathrm{T}}$.

Training Protocol. Unless otherwise noted, all SFT and distillation variants use LoRA fine-tuning for 10 epochs with a learning rate of 5×10^{-5}, a batch size of 4, and a maximum sequence length of 4096. All models are trained on the same training split.

Domain	Aspect inventory
WomenBags (34)	Overall Satisfaction; Material Type; Material Accuracy; Color Accuracy; Color Preference; Appearance; Size Accuracy; Price; Value for Money; Customer Service; Shipping Speed; Delivery Experience; Packaging; Return & Refund Policy; Construction Quality; Durability; Occasion Suitability; Sizing Guidance Clarity; Design Aesthetic; Weight; Capacity; Shape Retention; Odor; Brand Authenticity; Accessories Included; Security Features; Cleaning & Maintenance; Hardware Quality; Hardware Design; Strap Adjustability; Material Quality; Water Resistance; Comfort; Interior Organization
Dresses (29)	Overall Satisfaction; Material Type; Material Quality; Material Accuracy; Product-Image Consistency; Color Accuracy; Color Preference; Size Accuracy; Fit; Length; Comfort; Design Aesthetic; Construction Quality; Durability; Stretch; Transparency; Lining; Occasion Suitability; Ease of Use; Price; Value for Money; Care & Maintenance; Customer Service; Shipping Speed; Delivery Experience (not speed); Packaging; Return/Refund Policy; Brand Authenticity; Accessories Included
Cosmetics (42)	Adhesion; Aesthetic Appeal; After-Sales Service; Authenticity; Blendability; Brand; Color; Color Payoff; Color Variety; Coverage; Design; Drying Time; Durability; Ease of Makeup Removal; Ease of Use; Efficacy; Longevity; Moisturizing; Oil Control; Packaging; Pigmentation; Portability; Price Value; Product Quality; Product Size; Safety; Shade Matching; Finish; Smudge Resistance; Sun Protection; Texture / Feel; Transfer Resistance; User Experience; Versatility; Water Resistance; Logistics Service; Order Accuracy; Product Availability; Suitable Skin Type; Expectation vs Reality; Overall; Others
Furniture (33)	After-Sales Service; Appearance & Design; Assembly & Installation; Authenticity; Breathability & Ventilation; Cleaning & Maintenance; Comfort; Damage & Defects; Durability; Ease of Use; Energy Efficiency; Environmental Friendliness; Ergonomics; Expectation vs Reality; Fit & Space Compatibility; Floor Protection; Functionality; Load Capacity; Logistics & Delivery; Material; Mobility & Portability; Moisture & Water Resistance; Noise Performance; Odor; Price Value; Product Quality; Safety; Storage; Surface Properties; Return Policy; Seller Trustworthiness; Overall; Others

Table 4: Aspect inventories for the four product categories in E-ABSA20K.

Model Size	WomenBags						Dresses					
	single-stage			two-stage			single-stage			two-stage		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Qwen3-4B	0.575	0.511	0.541	0.595	0.539	0.566	0.565	0.563	0.564	0.576	0.593	0.584
Qwen3-8B	0.593	0.531	0.560	0.603	0.556	0.579	0.638	0.595	0.616	0.648	0.612	0.629
Qwen3-14B	0.638	0.589	0.612	0.647	0.609	0.627	0.674	0.643	0.658	0.674	0.650	0.662
Qwen3-32B	0.692	0.595	0.640	0.692	0.614	0.650	0.753	0.649	0.697	0.755	0.655	0.701
Qwen3-235B	0.715	0.611	0.659	0.717	0.621	0.666	0.763	0.649	0.702	0.766	0.664	0.711

Table 5: Micro Precision/Recall/F1 for single-stage vs two-stage (few-shot) on WomenBags and Dresses.

A.6.1 Does SFT eliminate the need for two-stage inference?

To investigate whether supervised fine-tuning (SFT) can close the performance gap between single-stage and two-stage methods, we conduct a series of experiments on the Qwen3-8B backbone. Our findings indicate that while SFT improves performance across the board, the architectural benefits of our two-stage framework persist.

Single-stage SFT learns domain-specific expressions but can still suffer from scope leakage and inconsistency in long reviews. In contrast, **two-stage SFT**, by directly optimizing the verification and consolidation logic, demonstrates superior robustness. Full results comparing single-stage and two-stage SFT variants are presented in Appendix A.6 (Table 11).

This advantage is particularly evident on our hard-case subsets. As detailed in the Appendix (Table 12), two-stage SFT variants achieve the most consistent gains on conditional and comparison subsets, where single-stage models often struggle with scope or modality confusion. The challenge of distinguishing product aspects from service mentions in the "other-entity" subset persists across all methods.

Model Size	Cosmetics						Furniture					
	single-stage			two-stage			single-stage			two-stage		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Qwen3-4B	0.532	0.485	0.508	0.548	0.518	0.533	0.569	0.482	0.522	0.588	0.524	0.554
Qwen3-8B	0.563	0.512	0.536	0.576	0.535	0.555	0.646	0.524	0.579	0.659	0.544	0.596
Qwen3-14B	0.570	0.528	0.548	0.571	0.540	0.555	0.647	0.543	0.591	0.660	0.556	0.603
Qwen3-32B	0.653	0.550	0.597	0.654	0.568	0.608	0.674	0.557	0.610	0.697	0.560	0.621
Qwen3-235B	0.690	0.610	0.648	0.695	0.619	0.655	0.729	0.610	0.664	0.733	0.613	0.668

Table 6: Micro Precision/Recall/F1 for single-stage vs two-stage (few-shot) on Cosmetics and Furniture.

Model Size	**Single no-think**			**Single think**			**Two-stage (none)**			**Two-stage (Stage2-think)**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Qwen3-4B	0.533	0.406	0.460	0.543	0.469	**0.503**	0.564	0.452	0.501	0.566	0.449	0.501
Qwen3-14B	0.590	0.537	0.561	0.601	0.567	0.584	0.610	0.567	0.587	0.620	0.567	**0.592**
Qwen3-235B	0.694	0.585	0.635	0.702	0.602	0.648	0.701	0.598	0.645	0.702	0.602	**0.648**

Table 7: **Effect of CoT prompting (Avg-4cate).** "think" asks the model to reason internally but output JSON only. "Two-stage (Stage2-think)" enables *think* only for verification and consolidation. **Bold** denotes the best F1 within each row.

Setting	Description	Qwen3-8B (Avg-4cate)			Qwen3-14B (Avg-4cate)		
		P	R	F1	P	R	F1
Single-stage (baseline)	One-pass extraction and normalization.	0.562	0.457	0.504	0.590	0.537	0.561
Stage1 only	Candidate generation; directly used as final output.	0.420	0.688	0.521	0.444	0.702	0.544
Stage1 + Stage2 verify (w/o consolidation)	Verify candidates (scope/modality/include), no global merge/dedup.	0.572	0.503	0.535	0.585	0.569	0.577
Full two-stage	Verify + global consolidation (merge/dedup/conflict resolution).	**0.584**	0.498	**0.537**	**0.610**	0.567	**0.587**

Table 8: Ablation study of the two-stage framework. Metrics are micro Precision/Recall/F1, macro-averaged over four categories (WomenBags, Dresses, Cosmetics, Furniture).

Method (Stage 2: Qwen3-8B, Avg-4cate)	**P**	**R**	**F1**
Two-stage (prompt-only)	0.584	0.498	0.537
Two-stage + Stage2-distill (teacher $\rightarrow$ student)	0.667	0.518	0.583

Table 9: **Stage 2 distillation (Avg-4cate).** Pseudo labels are generated by a Qwen3-235B teacher under *think*. We use the teacher's own Stage 1 candidates to match the teacher's verification regime and avoid candidate-regime mismatch during distillation.

Hard-case subset	Two-stage (prompt-only)			Two-stage + Stage2-distill ($\tilde{\mathcal{Y}}^{\mathrm{T}}$)		
	P	**R**	**F1**	**P**	**R**	**F1**
Conditional (if/choose/would)	0.659	0.463	0.544	0.699	0.473	**0.564**
Comparison (than/instead/vs)	0.651	0.464	0.542	0.721	0.464	**0.565**
Uncertain (maybe/not sure)	0.555	0.445	0.494	0.575	0.450	**0.505**
Other-entity (shipping/service)	0.659	0.489	0.561	0.739	0.490	**0.589**

Table 10: **Hard-case evaluation of Stage 2 distillation (Qwen3-8B, Avg-4cate).** Distillation uses teacher-generated candidates $\tilde{\mathcal{Y}}^{\mathrm{T}}$ to better imitate the teacher's scope/grounding/modality decisions.

Method (Qwen3-8B, Avg-4cate)	P	R	F1
Single-stage (prompt-only)	0.562	0.457	0.504
Single-stage + SFT	0.594	0.463	0.520
Two-stage (prompt-only)	0.584	0.498	0.537
Two-stage + SFT (Stage2-only)	0.614	0.499	0.551
Two-stage + SFT (Stage1+Stage2)	0.606	**0.552**	**0.578**

Table 11: **SFT comparison (Avg-4cate).** For two-stage SFT, Stage 2 is trained on candidates produced by the *deployed* Stage 1, to match the inference-time candidate distribution and reduce verifier distribution shift.

Method (Qwen3-8B, Avg-4cate)	Conditional			Comparison			Uncertain			Other-entity		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Single-stage (prompt-only)	0.567	0.463	0.510	0.549	0.457	0.499	0.545	0.438	0.486	0.572	0.477	0.520
Single-stage + SFT ($x \to \mathcal{Y}^*$)	0.591	0.502	0.543	0.573	0.501	0.535	0.563	0.492	0.525	0.601	0.520	0.558
Two-stage (prompt-only)	0.659	0.463	0.544	0.651	0.464	0.542	0.555	0.445	0.494	0.659	0.489	0.561
Two-stage + SFT (Stage2-only)	0.699	0.473	0.564	0.721	0.464	0.565	0.575	0.450	0.505	0.739	0.490	0.589
Two-stage + SFT (Stage1+Stage2)	0.699	0.491	**0.577**	**0.722**	0.479	**0.576**	**0.579**	0.469	0.518	**0.732**	0.510	**0.601**

Table 12: **Hard-case SFT comparison (Avg-4cate).** For two-stage SFT, Stage 2 is trained on deployed-proposer candidates $\tilde{\mathcal{Y}}^{\mathrm{dep}}$ to avoid proposal-regime distribution shift.

Prompt: WomenBags Single-stage

```
You are an expert in fine-grained sentiment analysis for e-commerce fashion
    reviews.
Your task is to extract all relevant aspect sentiment tuples with expression
    type from a user review about WomenBags.
Each output tuple has the format: (target, aspect_category, opinion, sentiment)

-- target:
  -- If the product type/name and special parts are not mentioned, output: "
    PRODUCT" (default purchased item).
  -- If a specific type is mentioned (e.g., bag, tote, handbag), output: PRODUCT:
     XX, such as PRODUCT: bag, PRODUCT: tote.
  -- If a certain product part is mentioned (e.g., fabric, corners, lining,
    zipper, strap, hardware), output PRODUCT_PART: XX,
     such as PRODUCT_PART: strap, PRODUCT_PART: zipper, PRODUCT_PART: lining.
  Do NOT invent targets.

-- aspect_category: must be exactly one from the following predefined list:
["Overall Satisfaction","Material Type","Material Accuracy","Color Accuracy","
    Color Preference","Appearance","Size Accuracy","Price",
"Value for Money","Customer Service","Shipping Speed","Delivery Experience","
    Packaging","Return & Refund Policy",
"Construction Quality","Durability","Occasion Suitability","Sizing Guidance
    Clarity","Design Aesthetic","Weight","Capacity",
"Shape Retention","Odor","Brand Authenticity","Accessories Included","Security
    Features","Cleaning & Maintenance","Hardware Quality",
"Hardware Design","Strap Adjustability","Material Quality","Water Resistance","
    Comfort","Interior Organization"]

-- opinion: the minimal exact phrase from the review that conveys the user's
    view (do not paraphrase).
-- sentiment: one of ["positive","neutral","negative"].

Rules:
1. Extract every supported aspect from the candidate list -- multiple tuples are
     expected.
2. Each tuple must correspond to a distinct aspect category.
3. Keep "opinion" as a short, faithful span from the original text.
4. Output one tuple per line. Do not add explanations or markdown.
5. If no aspect from the list is mentioned (explicitly or implicitly), output
    nothing.
6. Pay attention to transitional/concessive/conditional sentences and mentions
    of irrelevant products; these do not belong to the review-level targets.
7. One review may mention multiple purchased items; distinguish targets and
    sentiments accordingly.
8. Handle sentiment conflicts under the same (target, aspect), or cases where
    multiple opinions should be combined.

Input review: "{review_text}"
Output:
```

Figure 5: Single-stage prompt for WomenBags.