

Large Language Models as Automatic Annotators and Annotation Adjudicators for Fine-Grained Opinion Analysis

Gaurav Negi Margaret Alessandra Waskow John P. McCrae
Omnia Zayed Paul Buitelaar

Data Science Institute, University of Galway

{g.negil, margaret.waskow, john.mccrae,
omnia.zayed, paul.buitelaar}@universityofgalway.ie

Abstract

Fine-grained opinion analysis enables a detailed understanding of expressed sentiments and their corresponding targets. However, annotating domain-specific datasets requires substantial human effort and financial resources. To address this limitation, we investigate the feasibility of using large language models (LLMs) as automatic annotators for fine-grained opinion analysis. We use a declarative annotation pipeline to reduce prompt engineering when identifying fine-grained opinion spans, and compare rule-based voting against an LLM-based adjudicator for reconciling conflicting predictions. We trial the pipeline across different model sizes on the Aspect Sentiment Triplet Extraction (ASTE) and Aspect-Category-Opinion-Sentiment (ACOS) tasks. Our findings indicate a critical performance bifurcation: LLMs are reliable at the span level yet struggle to faithfully reproduce the relational structures connecting those spans. The better aggregation strategy depends on this complexity: voting suffices for triplets, while quadruples require an adjudicator that re-reads the source. Treating annotator disagreement as an informative signal rather than noise, we further show that agreement between LLM annotators is a calibrated predictor of correctness, and can be used to delegate cases with high annotator disagreement to humans.

1 Introduction

Opinion¹ analysis has evolved beyond simple sentiment polarity (positive, negative, neutral) to fine-grained formulations that also capture the contextual elements of an opinion. Consider the following example:

¹Unless stated otherwise, we use the term opinion as a broad concept that covers sentiment and its associated information, such as opinion target and the person who holds the opinion, and use the term sentiment to mean only the underlying positive, negative, or neutral polarity implied by opinion.

Example: “*I had hoped for better battery life, as it had only about 2–1/2 hours doing heavy computations (8 threads using 100% of the CPU).*”

Here a *negative* sentiment is expressed according to sentence- or document-level annotations. But with a fine-grained formulation, the following additional annotations are desired: (i) an **opinion span** expressing the polarity, in this example that is “*hoped for better*”, (ii) an explicit instantiation of the target in the text called an **aspect term** (in the example: “*battery life*”), They together form an aspect sentiment triplet as specified according to the Aspect Sentiment Triplet Extraction (ASTE) task (Wu et al., 2020). Adding coarse-grained aspect categories (e.g., “*Battery#Operational_performance*” in the example above) to identify the target entity and attribute enables the capture of implicit targets that lack explicit aspect terms, adhering to ACOS quadruple guidelines (Cai et al., 2021).

Derived mostly from SemEval review data, current fine-grained opinion datasets support system benchmarking, but their scale and specificity hinder real-world applicability (Zhang et al., 2023). Furthermore, human opinion annotation remains a challenging task requiring strict guidelines (Toprak et al., 2010). While human annotation remains the gold standard for opinion annotation, its lack of scalability motivates the assessment of automatic annotation approaches. Because fine-grained opinion annotation is inherently subjective, disagreement among annotators is systematic rather than random. In this work, we treat it as a signal of uncertainty used to decide where disagreement should be resolved through automated aggregation policies and when it should be reserved for human adjudication.

In this work, we investigate the potential of LLMs to automatically annotate text with fine-grained opinions. The LLMs have been shown to be few-shot learners (Kojima et al., 2022) and to exhibit instruction-following and reasoning ca-

pabilities (Hao et al., 2024). Through our study, we enquire into the extent to which they can successfully annotate fine-grained opinions and their failure modes. Understanding the efficacy of LLM-driven annotations can help us make an informed decision on the extent to which these models can fulfil the opinion annotator role or assist human annotators.

We also propose a dedicated LLM-based Adjudication approach that combines multiple opinion annotations for a given text input to resolve inter-annotator disagreement and obtain final opinion annotations. Voting thresholds and LLM-based adjudication represent different decision policies under disagreement, and we characterise the conditions under which each is appropriate. This is inspired by the process of annotation adjudication (Hovy and Lavid, 2010), which includes making a final judgement after considering the conflicting annotations from multiple annotators. Commonly used adjudication methods, such as traditional majority voting, assume a discrete label set. However, ASTE and ACOS opinions form open-ended triplets and quadruples, preventing direct application of traditional majority voting (Uma et al., 2021). We therefore adapt traditional voting into two rule-based tuple-level and element-wise aggregation baselines to compare against our LLM-based adjudication. Our contributions, evaluated across three model scales on six SemEval datasets, are:

- **Declarative LLM annotation pipeline:** A schema-constrained pipeline that uses LLMs declaratively, without hand-crafted prompts, to extract opinions under the ASTE and ACOS specifications.
- **Annotation adjudication:** An LLM-as-adjudicator method for merging annotations from multiple LLM annotators, benchmarked against exact, flexible and element-wise voting aggregators adapted to the combinatorial output space of both tasks.
- **Uncertainty estimation and utilisation:** An element- and pairwise decomposition reveals strong span-level but weak relational alignment with human annotations. Moreover, annotator disagreement serves as a calibrated confidence signal (transferring from a dev sample to test) for selective human routing.

2 Related Work

Opinion Formulations Opinion mining has progressed from feature-based summarisation (Hu and Liu, 2004; Liu and Zhang, 2012) to fine-grained Aspect-Based Sentiment Analysis (ABSA). This work investigates two extractive ABSA tasks: Aspect Sentiment Triplet Extraction (ASTE) (Xu et al., 2020; Wu et al., 2020) and Aspect-Category-Opinion-Sentiment (ACOS) quadruple extraction (Cai et al., 2021).

LLMs as Annotators Dataset creation encompasses processes from selection to adjudication (Klie et al., 2024). Large Language Models (LLMs) have started being employed as annotators for subjective tasks such as span (Kasner et al., 2026), argument-quality (Mirzakhmedova et al., 2024), and propaganda annotation (Hasanain et al., 2024). This study extends their application to fine-grained opinion annotation through inference-time prompt optimisation, integrating the training-free approach of rule-based methods (Qiu et al., 2011) with contemporary LLM reasoning.

Prompt Engineering Given that LLMs demonstrate prompt sensitivity (Zhuo et al., 2024) and task variability (Wang and Wang, 2025), this work employs **DSPy** (Khattab et al., 2024) to automatically synthesise prompts from few-shot examples, rather than relying on manual prompt engineering. For output aggregation, we prefer multiple outputs from diverse LLM-annotators over single-model redundancy such as self-consistency (Wang et al., 2023). Fixed-rule voting baselines are evaluated alongside an LLM-as-adjudicator approach inspired by stacked generalisation (Wolpert, 1992).

3 Dataset

Task	Dataset	Train	Dev	Test
ASTE	lap14	906	219	328
	res14	1126	310	492
	res15	607	148	322
	res16	857	210	326
ACOS	laptop	2934	326	816
	restaurant	1530	171	583

Table 1: Dataset Breakdown

The datasets used in our experiments are described in Table 1. The data schema relating the ASTE and ACOS specifications, task definitions, and annotated examples is provided in Appendix A.

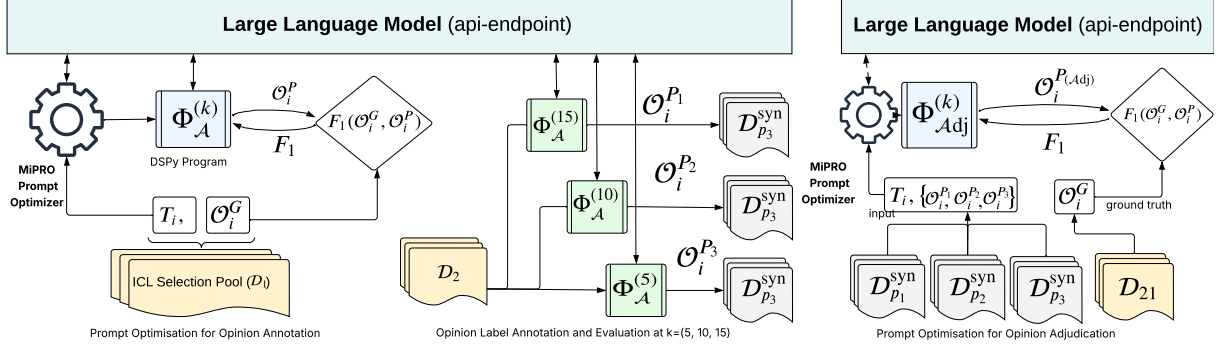


Figure 1: **(Left) Annotation:** in the synthesis stage the DSPy program Φ_A constructs optimised prompts from k human-annotated examples $\mathcal{O}_i^G$ drawn from the ICL selection pool $\mathcal{D}_1$; in the annotation stage it generates predicted opinions $\mathcal{O}_i^P$ for input texts T_i . The exemplar count $k \in \{5, 10, 15\}$ is selected on $\mathcal{D}_2$. **(Right) Adjudication:** the redundant annotation sets produced at each k are combined with the source texts of $\mathcal{D}_{21}$ to optimise the adjudication program $\Phi_{Adj}^{(k)}$, and k for adjudication is selected on $\mathcal{D}_{22}$.

4 Methodology

The fine-grained opinion annotation pipeline has multiple stages: (i) Preparing and evaluating LLM-specific prompts, (ii) Redundant Opinion annotation using multiple LLMs, and (iii) Annotation adjudication using an LLM to create collaborative opinion annotations. In addition, we implement rule-based voting aggregators (Section 4.2.1) that serve as baselines against which the LLM-based adjudication is compared.

4.1 Problem Definition

On a conceptual level, the process of annotating opinions using the fine-grained opinion formulations involves identifying a set of opinions expressed in the input text T_i . As with human annotators, the process generates multiple annotations for the same input T_i , followed by adjudication that resolves disagreements between annotators to produce final annotations. These steps can be formalised as:

LLM as opinion annotators. Given the input text T_i from corpora $\mathcal{T}$, our objective is to extract all expressed opinions $\mathcal{O}_i = \{o_1, o_2, \dots, o_n\}$. The elements of single opinion (o_j) is task dependant:

- ASTE opinions are formulated as the triple $o_j = (at, s, op)$, where at is the aspect term, s is the sentiment and op is the opinion term.
- ACOS opinions are quadruples $o_j = (at, s, op, ac)$, where at , s and op encode similar concepts as ASTE and ac is aspect category made up of entity E and attribute A (expressed as $E\#A$).

The LLM-driven annotation pipeline maps input

corpora $\mathcal{T}$ to $\mathcal{O}^k$ using k LLM-based annotators.

Opinion Adjudication. We evaluate two main strategies for opinion adjudication:

- **LLM-Based Adjudication (Adj_{LLM}):** LLM-based adjudication to merge opinion sets $(\mathcal{O}^1, \dots, \mathcal{O}^k)$ produced by k annotators given the source text $\mathcal{T}$: $\text{Adj}_{\text{LLM}} : (\mathcal{T}, \mathcal{O}^1, \dots, \mathcal{O}^k) \rightarrow \mathcal{O}$. This formulation is analogous to stacking (Wolpert, 1992) in ensemble learning.
- **Voting Adjudication Baselines (Vot):** Rule-based counterparts Adj_{LLM} that operate strictly on the annotation sets: $\text{Vot} : (\mathcal{O}^1, \dots, \mathcal{O}^k) \rightarrow \mathcal{O}$. Retain opinions supported by a minimum annotator threshold (Section 4.2.1) and are used for benchmarking purposes.

4.2 Declarative Annotation and Adjudication Pipeline

We structure the generation and adjudication steps into a multi-stage declarative pipeline powered by a DSPy program. Both the annotation and adjudication stages rely on the MIPROv2 prompt optimiser to systematically refine the instructions and in-context examples (Opsahl-Ong et al., 2024). The pipeline is trained and evaluated across two data splits: $\mathcal{D}_1$ (the first half of the development set) and $\mathcal{D}_2$ (the second half of the development set).

Pipeline stages The pipeline is optimised in three stages, illustrated in Figure 1: **(i)** MIPROv2 optimises the annotation program Φ_A on $\mathcal{D}_1$ at three few-shot levels, $k \in \{5, 10, 15\}$, yielding one optimised program per In-Context exemplar. **(ii)** These

programs are applied to the unseen texts of $\mathcal{D}_2$, generating redundant candidate sets $\{\mathcal{O}_i^{P_1}, \mathcal{O}_i^{P_2}, \mathcal{O}_i^{P_3}\}$ for each input. The annotation exemplar count (k) and the optimised program to be retained are selected here, by F_1 against gold. (iii) $\mathcal{D}_2$ is then divided in two: the adjudicator $\Phi_{Adj}^{(k)}$ is optimised with MIPROv2 on the candidate sets of $\mathcal{D}_{21}$, while its own exemplar count is selected on $\mathcal{D}_{22}$, which later supplies the confidence mapping of Section 8. The adjudicator performs a joint extraction conditioned on the source text and the candidate sets, emitting the full annotation set in a single pass rather than selecting among candidates element by element. Both selection steps are analogous to hyperparameter search on a development set; all results reported in Sections 6 and 7 are computed exclusively on held-out test splits.

4.2.1 Voting Adjudication

We propose two rule-based voting aggregators as adjudication performance baselines (pseudocode in Algorithm 1 and 2, Appendix C):

In brief, **tuple-level voting** votes over whole annotator tuples, grouped either by **exact matching** of normalised keys or by **flexible matching** over span overlap, retaining clusters supported by at least θ annotators, making the required degree of consensus an explicit, tunable policy choice; **element-wise voting** aligns tuples across annotators and sets each field independently by majority vote, allowing it to **recombine fields** from different annotators without introducing unproposed elements. Full descriptions and pseudocode are provided in Appendix C.

5 Experimental Setup

Model Configuration In our experiments, we used models of three different sizes: mini (4B parameters), small (14B parameters), and medium (32B parameters), with three distinct LLMs per scale (Table 2).

Ann.	Mini (~4B)	Small (~14B)	Med (~30B)
$\mathcal{A}_1$	Qwen3-4B-Thinking	Qwen3-14B	Qwen3-30B-A3B-Thinking
$\mathcal{A}_2$	MiniCPM3-4B	gpt-oss-20b [†]	DeepSeek-R1-Distill-Qwen-32B
$\mathcal{A}_3$	Phi-4-mini-reasoning	DeepSeek-R1-Distill-Qwen-14B	gpt-oss-20b [†]

Table 2: LLMs used as annotators. [†]Used with both Small and Med.

For each model, we identify its optimal configuration by validating performance at $k \in \{5, 10, 15\}$ on the $\mathcal{D}_2$ split; models are then ranked

as $\mathcal{A}_1, \mathcal{A}_2$, and $\mathcal{A}_3$ based on these F_1 validation scores.² The LLM assigned to $\mathcal{A}_1$ is also selected for the adjudication stage because of its demonstrated efficacy on the annotation task. The adjudication process is analogous to appointing a high-performing human annotator as adjudicator (Hovy and Lavid, 2010). To achieve this, we train a separate DSPy program and then optimise via MIPROv2, obtaining Adj . This two-stage approach ensures that, while the redundant annotations are generated by a diverse ensemble, the final synthesis is performed by the most capable model in the tier, using a prompt specifically optimised to resolve inter-model disagreements. However, while distinct pipelines with different DSPy programs are utilised, this selection criterion for adjudication might introduce cases of self-selection bias, mirroring the inherent challenges of the traditional annotation-adjudication design that motivated our architecture.

Hardware and cost All experiments ran on a single NVIDIA A40, with vLLM (Kwon et al., 2023) serving each model; hyperparameters are constant across models, with generated output length 16384 and temperature 0. Individual runs are therefore near-deterministic, and we rely on consistency across model scales, tasks and datasets together with the paired bootstrap tests of Appendix H. Annotation costs \$0.017–\$0.036 per sentence, 15–30 $\times$ below the £0.41 ($\approx$ \$0.55) reported for a single crowd-sourced ABSA pass (Donhauser et al., 2026), while providing three-fold redundancy. (More latency and throughput figures are in Appendix E.4.)

Evaluation Because exact-match metrics are notoriously stringent for combinatorial tasks requiring joint span, category, and sentiment alignment (Barnes et al., 2021), we supplement standard evaluation with granular diagnostics. Specifically, we measure reference-based performance using exact-match, element-wise, and pairwise F_1 scores against ground-truth labels to isolate boundary versus relational errors. In tandem, we measure annotator consensus via Inter-Annotator Agreement (IAA) on span co-extractions to evaluate agreement among LLMs before adjudication.

Baselines We benchmark against published results for three standard SFT paradigms (sequence

²dspy artefacts: <https://github.com/ANON-1331/anon-repo-annotation-opinion>

tagging (Xu et al., 2020), grid-based span labeling (Wu et al., 2020), and generation (Yan et al., 2021)), (rule-based) Double-Propagation (Qiu et al., 2011), and UnifiedABSA (Wang et al., 2024) (a multi-task supervised baseline trained on annotations across all ABSA tasks, contrasting our training-free approach). Additionally, the rule-based voting aggregators from Section 4.2.1 ($\mathcal{A}_{dj}(Vot_{=})$, $\mathcal{A}_{dj}(Vot_{\approx})$, and $\mathcal{A}_{dj}(Vot_{el})$ with $\theta = 2/3$) serve as rule-based baselines against our LLM adjudicator $\mathcal{A}_{dj}(\mathcal{A}_1)$.

6 Results

Triplet (ASTE) and quadruplet (ACOS) F_1 scores are separately reported against human gold references on held-out SemEval test splits. To evaluate aggregation gains, paired bootstrap tests are conducted (Appendix H).

6.1 Comparison with Supervised Baselines

Inference-time adaptation trails supervised SFT on both tasks, but the margin diverges sharply between them. On **ASTE** the gap is domain-dependent: on Lap14 supervised models lead decisively (UnifiedABSA at 62.45% and BART_{ABSA} at 58.69% F_1 , vs. 47.77% for Med $\mathcal{A}_{dj}(Vot_{=})$), whereas across the restaurant datasets it closes considerably: on Res15, Medium $\mathcal{A}_{dj}(Vot_{=})$ (64.04%) outperforms even the fine-tuned UnifiedABSA (63.25%) without parameter updates, while Res16 exhibits a small margin (70.91% Mini $\mathcal{A}_{dj}(Vot_{=})$ vs. 72.10%) and Res14 a moderate one (69.27% Med $\mathcal{A}_{dj}(\mathcal{A}_1)$ vs. 73.22%). Each domain-best result yields statistically significant improvements over individual annotators ($p < 0.05$, marked */+), highlighting strong inference-time lead generalisation.

On **ACOS** no such convergence occurs: UnifiedABSA (42.58% on Laptop and 60.60% on Restaurant) and BART-CRN (38.32% and 48.90%) retain a clear lead. Unlike SFT baselines optimised directly on task-specific training data, inference-time methods cannot rely on the learnt dataset specific likelihoods, and two properties specific to quadruples widen the gap further: (i) **implicit opinions and aspects**, since the model must first decide whether an aspect or opinion is explicit or implicit before mapping it to a category; and (ii) **search-space complexity**, as the Laptop taxonomy spans 112 categories and yields a best F_1 of only 18.66%, whereas Restaurant, with 12 categories, presents a much simpler classification task and reaches a competitive 39.98%.

	Model	ASTE (F_1)				ACOS (F_1)	
		Lap14	Res14	Res15	Res16	Laptop	Rest.
SFT	JET	51.04	62.40	57.53	63.83	–	–
	GTS	54.58	70.20	58.67	67.58	–	–
	BART _{ABSA}	58.69	65.25	59.26	67.62	–	–
	JET-ACOS	–	–	–	–	23.81	39.01
	Extract-classify	–	–	–	–	35.80	44.61
	BART-CRN	–	–	–	–	38.32	48.90
	UnifiedABSA	62.45	73.22	63.25	72.10	42.58	60.60
Mini	Double-Prop.	–	–	–	–	08.00	21.04
	$\mathcal{A}_1$	43.15	63.86	55.11	66.36	13.07	30.31
	$\mathcal{A}_2$	35.95	57.31	47.61	59.01	12.12	21.39
	$\mathcal{A}_3$	23.59	38.53	34.68	45.34	01.58	05.18
	$\mathcal{A}_{dj}(Vot_{\approx})$	39.91 ⁺	61.29 ⁺	52.53 ⁺	66.85 ⁺	9.21	26.19 ⁺
	$\mathcal{A}_{dj}(Vot_{=})$	43.54 ⁺	64.73 ⁺	56.21 ⁺	70.91 [*]	10.33	25.54 ⁺
	$\mathcal{A}_{dj}(Vot_{el})$	41.10 ⁺	63.81 ⁺	52.88 ⁺	66.97 ⁺	13.18	28.79 ⁺
	$\mathcal{A}_{dj}(\mathcal{A}_1)$	45.69 [*]	67.64 [*]	58.18 [*]	67.89 ⁺	16.18 [*]	31.63 ⁺
	$\mathcal{A}_1$	44.93	63.42	60.28	64.99	09.69	21.36
	$\mathcal{A}_2$	41.08	60.06	53.96	58.98	07.66	22.92
Small	$\mathcal{A}_3$	32.95	57.42	50.56	62.44	06.70	15.73
	$\mathcal{A}_{dj}(Vot_{\approx})$	42.82	63.84 ⁺	57.03 ⁺	65.51	9.00	24.13 ⁺
	$\mathcal{A}_{dj}(Vot_{=})$	45.03 ⁺	66.02 [*]	59.53 ⁺	64.63 ⁺	8.15	25.12 ⁺
	$\mathcal{A}_{dj}(Vot_{el})$	42.06	63.79 ⁺	57.46 ⁺	64.61	08.78	24.13 ⁺
	$\mathcal{A}_{dj}(\mathcal{A}_1)$	40.24	61.65	56.38	62.69	10.92 [*]	29.65 [*]
	$\mathcal{A}_1$	45.64	68.67	61.98	65.37	15.75	28.41
	$\mathcal{A}_2$	43.49	62.46	60.26	63.04	09.78	17.47
Med	$\mathcal{A}_3$	41.08	60.06	53.96	58.98	07.66	22.92
	$\mathcal{A}_{dj}(Vot_{\approx})$	44.83	66.69 ⁺	61.29	63.83	11.93 ⁺	23.81
	$\mathcal{A}_{dj}(Vot_{=})$	47.77 ⁺	68.99 ⁺	64.04⁺	64.63	12.06 ⁺	27.28 ⁺
	$\mathcal{A}_{dj}(Vot_{el})$	46.36	66.57 ⁺	61.41	62.43	12.90 ⁺	27.31 ⁺
	$\mathcal{A}_{dj}(\mathcal{A}_1)$	46.81	69.27 ⁺	62.51	67.14 ⁺	18.66 [*]	39.98 [*]
	$\mathcal{A}_1$	45.64	68.67	61.98	65.37	15.75	28.41
	$\mathcal{A}_2$	43.49	62.46	60.26	63.04	09.78	17.47

Table 3: Overall F_1 scores for LLM-based annotators and adjudication on ASTE (triplets) and ACOS (quadruples). */+ mark significant improvements over all three / at least two individual annotators ($p < 0.05$, Appendix H). Bold marks the best overall score per dataset; italics the best among inference-time methods. Complete results with precision and recall, and baseline references, are given in Tables 18 and 17 (Appendix F).

6.2 Scaling and Adjudication Gains

Both tasks peak at the **Med** tier, where LLM adjudication $\mathcal{A}_{dj}(\mathcal{A}_1)$ generally improves upon the best individual annotator ($\mathcal{A}_1$). Reported absolute gains $\mathcal{A}_{dj}(\mathcal{A}_1)$ and F_1 gains over $\mathcal{A}_1$ across the four datasets of +2.54/3.78/3.07/1.53% for Mini models, and +1.17/0.60/0.53/1.77% for Medium models.

The observed non-monotonic decrease in performance for the ASTE Small tier (Qwen3-14B) is attributable to the adjudicator model rather than annotation quality, since exact voting ($Vot_{=}$) continues to outperform $\mathcal{A}_1$ (66.02% vs. 63.42% on Res14). This outcome may be explained by the lack of long-chain reasoning ("thinking") states in the Small model, which are present in the Mini

and Med models (Guo et al., 2025). Supporting this hypothesis, substituting the 14B model with a reasoning-distilled Qwen3-4B-Thinking adjudicator partially restores performance (see Table 8, Appendix B).

LLM vs. Voting Adjudication. Both adjudication families show clear precision-recall trade-offs. Exact matching ($Vot_{=}$) prioritizes precision by discarding non-verbatim tuples (ASTE: 75.00% on Mini Res16, 57.09% vs. 42.34% for $\mathcal{A}_1$ on Lap14; ACOS: 59.36% Res, 32.57% Lap at Mini), while $Vot_{\approx}$ and Vot_{el} recover recall (ASTE: 48.24% on Med Lap14; ACOS: 27.18% vs. 16.27% on Mini Res).

Comparing $\mathcal{A}_{dj}(\mathcal{A}_1)$ against the top voting baseline separates the two tasks cleanly. On **ASTE**, $Vot_{=}$ and $\mathcal{A}_{dj}(\mathcal{A}_1)$ are closely matched, split 7 to 5 across 12 cells ($Vot_{=}$ is best performing all four Small, Lap14/Res15 Med, Res16 Mini; $\mathcal{A}_{dj}(\mathcal{A}_1)$ stays ahead on Lap14/Res14/Res15 Mini, Res14/Res16 Med). Margins remain under 4.80 points (67.49 vs. 62.69 on Small Res16) and under 1 point on Med Lap14/Res14. On **ACOS**, $\mathcal{A}_{dj}(\mathcal{A}_1)$ dominates all 6 cells by +1.92 (Small Lap: 10.92 vs. 9.00) to +12.67 points (Med Res: 39.98 vs. 27.31). Exact voting is best performing only on 1 cell vs. 4 for Vot_{el} due to quadruple recall collapse (8.13% Med Lap; 19.43% Med Res), showing adjudication gains scale directly with relational complexity.

7 Analysis

The element-wise and pairwise decomposition of triplet and quadruple evaluations, detailed in Tables 19 and 20 (Appendices G), reveals a consistent pattern across both tasks. Alignment with human annotations measured is strong at the individual span level but degrades systematically as relational complexity increases.

Performance Hierarchy and Label Alignment:

Across both tasks, sentiment polarity (s) consistently exhibits the highest alignment with human annotations, approaching, and in some settings exceeding, 90% F1. Aspect term (at) and opinion term (op) spans show lower, yet promising F1 scores reaching up to $\sim 80\%$ post-adjudication at medium scale. Performance declines predictably as models are required to identify interdependent structures. In the **ASTE** task (Table 19), while assigning the correct polarity to an aspect ($s\&at$) is more challenging than individual labelling, the

most degrading bottleneck appears to be $at\&op$ identification, i.e. the process of correctly linking an opinion span to its specific target.

Domain Disparity and Complexity We observe a pronounced domain disparity in the ACOS task, particularly on laptops where untuned performance lags due to aspect-category (ac) prediction. Specifically, ac alignment reaches only 36.30% on Med Laptop versus 54.91% on Restaurant, maintaining an 18-25 point gap across scales. Across both domains (Table 20), degradation is highest for pairings involving ac : at Med Laptop, the lowest pairs are $op\&ac$ (22.61%), $at\&ac$ (25.95%), and $s\&ac$ (33.35%), all trailing $at\&op$ (39.57%). Additionally, opinion-span scores fall well below ASTE counterparts (55.29% on Med Laptop vs. 72.73% on Med Lap14) as implicit elements require explicit **None** labelling during quadruple annotation.

Adjudication Effects and Mechanism The LLM adjudicator ($\mathcal{A}_{dj}(\mathcal{A}_1)$) mitigates annotator noise and improves multi-element alignment across most settings for e.g., ASTE Lap14 Mini $at\&op$ alignment rises from 48.67% ($\mathcal{A}_1$) to 52.62%, and ACOS Res Med $op\&ac$ jumps from 37.66% to 48.85% (Table 19, 20). Conversely, exact voting ($Vot_{=}$) struggles with complex interdependencies, dropping Laptop Med sentiment alignment from 89.86% ($\mathcal{A}_1$) to 54.41%. To explain these dynamics, we categorise emitted tuples into *copy* (verbatim candidate tuple), *recombination* (novel mix of candidate elements), and *generation* (contains novel elements from the source text).

Task	Scale	Vot_{el}	$\mathcal{A}_{dj}(\mathcal{A}_1)$
ASTE	Mini	2.4 (42)	18.4 (158)
	Small	0.0 (23)	0.0 (48)
	Med	0.0 (19)	9.6 (115)
ACOS	Mini	6.9 (159)	15.8 (481)
	Small	2.4 (169)	6.3 (495)
	Med	3.5 (198)	13.4 (1002)

Table 4: Precision (%) of non-copy output, i.e., the tuples matching no annotator tuple verbatim with bucket size in parentheses.

While exact/flexible copies verbatim and element-wise voting (Vot_{el}) offer limited recombination ($\leq 10\%$), the LLM adjudicator $\mathcal{A}_{dj}(\mathcal{A}_1)$ uniquely enables generation by re-reading the source text. Consequently, ASTE outputs remain overwhelmingly copies (94–98%), whereas higher-dimensional Med ACOS shifts heavily to recombination (14%)

and generation (33%, leaving 53% copies, Appendix D). Crucially, when departing from verbatim copying, the LLM adjudicator emits far more non-copy tuples than Vot_{el} (1002 vs 198 on Med ACOS) with substantially higher accuracy, outperforming Vot_{el} by $2.3\text{--}7.7\times$ where Vot_{el} ever succeeds, and 9.6% vs 0% on Med ASTE (Table 4).

7.1 Inter-Annotator Agreement of LLM-Based Annotators

Data	Scale	at	op	ac	at&s	op&s
ASTE (triplets)						
lap14	Mini	0.6125	0.49	–	0.59	0.46
	Small	0.73	0.66	–	0.72	0.66
	Med	0.75	0.69	–	0.75	0.69
res14	Mini	0.70	0.61	–	0.68	0.59
	Small	0.79	0.74	–	0.78	0.74
	Med	0.82	0.77	–	0.80	0.77
res15	Mini	0.70	0.57	–	0.67	0.54
	Small	0.81	0.72	–	0.80	0.71
	Med	0.84	0.75	–	0.82	0.74
res16	Mini	0.72	0.63	–	0.69	0.60
	Small	0.79	0.77	–	0.78	0.75
	Med	0.82	0.79	–	0.81	0.78
ACOS (quadruples)						
laptop	Mini	0.60	0.37	0.22	0.58	0.37
	Small	0.70	0.65	0.31	0.69	0.65
	Med	0.70	0.64	0.33	0.69	0.64
restaurant	Mini	0.61	0.41	0.41	0.59	0.40
	Small	0.73	0.64	0.49	0.71	0.64
	Med	0.71	0.65	0.50	0.69	0.64

Table 5: Krippendorff- α inter-annotator agreement among the three LLM annotators, before adjudication (*at*: aspect term, *op*: opinion span, *ac*: aspect category, *at&s/op&s*: span with sentiment). *ac* applies to ACOS only. Shading marks: **low** ($\alpha < 0.667$) and **high** ($\alpha \geq 0.80$) reliability.

While joint evaluation shows promising ASTE results (Table 3), ACOS performance (Table 3) indicates that LLMs cannot yet perform complex annotation independently. However, element-wise analysis (Tables 19 and 20) reveals strong span-level potential. To assess this further for annotation reliability, we report pre-adjudication Inter-Annotator Agreement (IAA) using Krippendorff’s α (Table 5). These agreement metrics reveal that reliability for ASTE increases with model parameter size, with triplet span elements **approaching highly reliable** levels. Conversely, this trend is notably weaker for ACOS, where quadruple elements plateau at **moderately reliable** levels at best. A detailed error analysis is provided in Appendix 9.

8 Uncertainty Estimation and Utilisation

The agreement scores in Section 7.1 are reported for the entire test-split annotations produced by the LLM-annotation pipeline. In this section, we investigate whether disagreement is informative *per item*. Let a tuple’s *support* $s \in \{0, \dots, 3\}$ be the number of annotators who independently propose it, and let $agree_i$ be the mean pairwise F_1 among the three annotator sets for sentence i . Importantly, this measure quantifies uncertainty and is reference-free, i.e., it does not require human-annotated references.

Support orders correctness, but is scarce for quadruples Precision rises monotonically with support for both tasks at every scale (Figure 2a): on ASTE Small from 0.00 at $s=0$ to 0.79 at $s=3$, and on ACOS Med from 0.13 to 0.68. How much output actually reaches high support, however, differs sharply. Unanimity accounts for 33.7-56.7% of ASTE output but only 1.7-7.5% of ACOS output: 33, 153, and 115 tuples at Mini, Small, and Med. The high-confidence region that makes agreement actionable for triplets barely exists for quadruples: the same scarcity that leaves a fixed voting threshold little to select, and hence why re-reading the source outperforms counting votes there.

Recalibrated support transfers across splits Expected calibration error (Guo et al., 2017) weights each support level by its share of the output, but it is only as meaningful as the confidence estimate $\hat{c}$ it is measured against. Consider the formula:

$$ECE = \sum \frac{n_s}{N} |P(\text{correct} | s) - \hat{c}(s)|, \quad (1)$$

We use the *vote* share for confidence estimation, defining $\hat{c}(s) = s/3$ for our $M=3$ annotators, a cross-model counterpart of the sample frequency underlying self-consistency. However, this misfits by construction: it asserts absolute certainty at $s=3$ (though true accuracy is 50-85%) and impossibility at $s=0$ (though accuracy reaches 18%). We therefore estimate $\hat{c}$ from data, setting $\hat{c}(s) = P_{\mathcal{D}_{22}}(\text{correct} | s)$ on $\mathcal{D}_{22}$ (disjoint from the $\mathcal{D}_{21}$ split on which the adjudicator was optimised), freezing it, and evaluating on the held-out test splits, since fitting and evaluating on one split is circular. The mapping transfers: ECE falls from 0.162 to 0.073 pooled by task and scale, improving in 15 of the 18 domain and scale settings (Appendix E).

A calibrated support value converts a count into an estimated probability of correctness, so once the

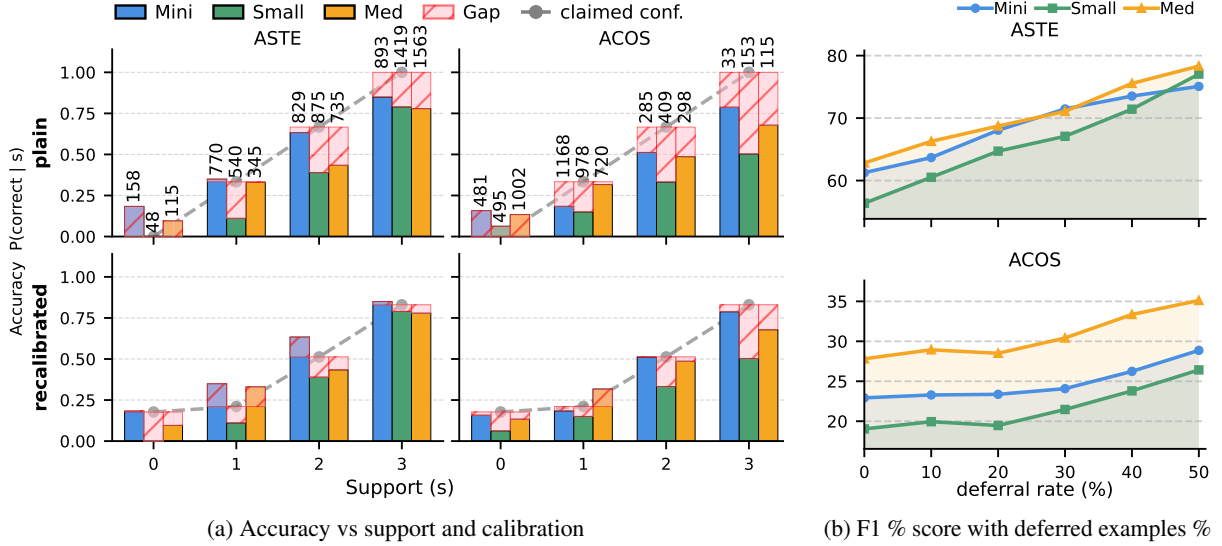


Figure 2: Disagreement as an uncertainty signal at two granularities. **(a)** Bars indicate $P(\text{correct} | s)$ per support level s (bucket sizes annotated). Dashed and solid lines denote nominal ($s/3$) and calibrated (from $\mathcal{D}_{22}$) confidences. Precision increases monotonically with support across all six task $\times$ scale cells. Unanimity is frequent for ASTE ($n=893/1419/1563$ at Mini/Small/Med) but rare for ACOS ($n=33/153/115$). The fitted mapping corrects the nominal mapping’s forced boundary constraints (0 at $s=0$; 1 at $s=3$), halving average expected calibration error (ECE) from 16.2 to 7.3. Bucket-wise ECE (nominal $\rightarrow$ fitted for Mini/Small/Med) is: ASTE (7.7 $\rightarrow$ 8.5, 22.9 $\rightarrow$ 7.9, 19.1 $\rightarrow$ 6.8); ACOS (15.3 $\rightarrow$ 2.1, 20.8 $\rightarrow$ 11.8, 11.1 $\rightarrow$ 6.8). ASTE Mini’s slight regression stems from a coincidentally strong nominal baseline. **(b)** Sentences are ranked by mean pairwise agreement. Withholding the least-agreed 50% boosts micro- F_1 on the remainder by 13.8–20.7 points for ASTE vs. 5.9–7.4 for ACOS.

mapping is fitted, a precision target can be set on new, unlabelled data. Since $\mathcal{D}_{22}$ also selected k , the fitted values may be slightly optimistic, though the adjudicator itself never saw this data during optimisation.

Disagreement as a Deferral Signal At the sentence level, agreement (agree_i) correlates with adjudicated output quality across all scales. However, the two tasks occupy starkly different regimes: agreement is a moderately strong triage signal for ASTE triplets ($\rho = 0.43\text{--}0.58$) but a weak one for ACOS quadruples ($\rho = 0.26\text{--}0.34$, all $p < 10^{-22}$; details in Appendix E.2). This divergence is clear during triage. Withholding the lowest-agreement half of the data raises ASTE micro- F_1 by 13.8 to 20.7 points (Small: 56.4 $\rightarrow$ 77.0), whereas ACOS gains only 5.9 to 7.4 points and plateaus below 30% deferral (Figure 2b). Ultimately, disagreement can effectively drive a human-LLM division of labour by routing difficult sentences to humans while retaining the rest for automated adjudication. Paralleling Khurana et al. (2024), who use worker disagreement for model abstention, we replace abstention with human delegation. Notably, this strategy is far more effective for opinion triplets (ASTE) than opinion quadruples (ACOS).

9 Error Analysis

Failure modes fall into four categories: (i) opinion-target mislinking (*at&op*), (ii) category misassignment (*ac*), (iii) implicit-element errors, and (iv) span-boundary divergence. Unlike tuple-level prescribed metrics that obscure these granular errors, element-reorganising methods expose them (Tables 19, 20).

Table 6 highlights how inherent task subjectivity and framing drive these failure modes. In ACOS, plausible category shifts (e.g., Gold’s *lap-top#op_perf* vs A_1 ’s *lap#general*) are partial misalignments that Exact Match F1 conflates with total failure. Similarly in ASTE (“*It arrived so fast...*”), annotators extract implicit targets like “*delivery*” or “*arrival*” that are absent from the text; while semantically accurate, these violate strict explicit-span conventions and are penalised as hallucinations. This reveals both the promise and the limitations of LLM-based adjudication. On one hand, its ability to recombine and generate elements allows it to capture human labelling dynamics better than rigid voting mechanisms. On the other hand, this flexibility becomes a drawback when task guidelines strictly forbid extracting terms anchored outside the text.

[ACOS] Sentence: <i>this computer is great for playing videos in a classroom setting of 10 people , connecting to an hdmi tv to show videos and powerpoints , and has a fantastic battery life .</i>			
Gold: <i>s:</i> Positive <i>at:</i> computer <i>ac:</i> laptop#op_perf <i>op:</i> great <i>s:</i> Positive <i>at:</i> battery life <i>ac:</i> battery#quality <i>op:</i> fantastic	$\mathcal{A}_1$: <i>s:</i> Positive <i>at:</i> computer <i>ac:</i> lap#general <i>op:</i> great <i>s:</i> Positive <i>at:</i> battery life <i>ac:</i> lap#portability <i>op:</i> fantastic	$\mathcal{A}_2$: <i>s:</i> Positive <i>at:</i> computer <i>ac:</i> mul_device#op_perf <i>op:</i> great <i>s:</i> Positive <i>at:</i> battery life <i>ac:</i> lap#portability <i>op:</i> fantastic <i>s:</i> Positive <i>at:</i> computer <i>ac:</i> lap#connectivity <i>op:</i> great	$\mathcal{A}_3$: <i>s:</i> Positive <i>at:</i> computer <i>ac:</i> cpu#general <i>op:</i> great <i>s:</i> Positive <i>at:</i> hdmi tv <i>ac:</i> port#connectivity <i>op:</i> NULL <i>s:</i> Positive <i>at:</i> battery life <i>ac:</i> battery#quality <i>op:</i> fantastic
<i>Adj(V_{otel}):</i> <i>s:</i> Positive <i>at:</i> computer <i>ac:</i> lap#general <i>op:</i> great <i>s:</i> Positive <i>at:</i> battery life <i>ac:</i> lap#portability <i>op:</i> fantastic	<i>Adj(V_≈):</i> <i>s:</i> Positive <i>at:</i> computer <i>ac:</i> lap#general <i>op:</i> great <i>s:</i> Positive <i>at:</i> battery life <i>ac:</i> lap#portability <i>op:</i> fantastic	<i>Adj(V₌):</i>	<i>Adj($\mathcal{A}_1$):</i> <i>s:</i> Positive <i>at:</i> computer <i>ac:</i> lap#op_perf <i>op:</i> great <i>s:</i> Positive <i>at:</i> battery life <i>ac:</i> battery#quality <i>op:</i> fantastic
[ASTE] Sentence: <i>It arrived so fast and customer service was great .</i>			
Gold: <i>s:</i> Positive <i>at:</i> customer service <i>op:</i> great	$\mathcal{A}_1$: <i>s:</i> Positive <i>at:</i> <u>delivery</u> <i>op:</i> fast <i>s:</i> Positive <i>at:</i> customer service <i>op:</i> great	$\mathcal{A}_2$: <i>s:</i> Positive <i>at:</i> <u>arrival</u> <i>op:</i> fast <i>s:</i> Positive <i>at:</i> customer service <i>op:</i> great	$\mathcal{A}_3$: <i>s:</i> Positive <i>at:</i> customer service <i>op:</i> great <i>s:</i> Negative <i>at:</i> <u>price</u> <i>op:</i> arrived so fast
<i>Adj(V_{otel}):</i> <i>s:</i> Positive <i>at:</i> <u>delivery</u> <i>op:</i> fast <i>s:</i> Positive <i>at:</i> customer service <i>op:</i> great	<i>Adj(V_≈):</i> <i>s:</i> Positive <i>at:</i> customer service <i>op:</i> great	<i>Adj(V₌):</i> <i>s:</i> Positive <i>at:</i> customer service <i>op:</i> great	<i>Adj($\mathcal{A}_1$):</i> <i>s:</i> Positive <i>at:</i> <u>delivery</u> <i>op:</i> fast <i>s:</i> Positive <i>at:</i> customer service <i>op:</i> great

Table 6: Qualitative error analysis of ACOS cells display the tuple (*s*: sentiment, *at*: aspect term, *ac*: aspect category, *op*: opinion span). ASTE rows omit the category field. NULL indicates an implicit element. **Incorrect** and **hallucinated** entries are highlighted relative to the Gold annotation.

10 Conclusion

This work introduces a declarative LLM-based opinion-annotation pipeline and an adjudication method for resolving disagreement between annotators. Evaluation across three model scales against human-annotated ground truth reveals a bifurcation in performance: LLMs accurately detect individual spans reliably. However, their structural fidelity declines notably as relational complexity increases. When benchmarked against exact, flexible, and element-wise voting methods, the LLM adjudicator matches strict consensus on ASTE triplets yet is the only method that consistently improves ACOS quadruples, since it is the only approach that re-examines the source text. Therefore, the optimal aggregation strategy depends on relational

complexity. Furthermore, disagreement is predictive: tuple support orders correctness and, once recalibrated, can be transferred as a confidence estimate. Sentence-level disagreement identifies likely failures, enabling contested sentences to be routed to human annotators rather than accepting or rejecting the pipeline in its entirety. At current capability levels, the pipeline therefore provides reliable span and span-sentiment annotations, supporting its use as a high-fidelity pre-annotation assistant and data augmentation tool, rather than as a stand-alone replacement for human annotators.

Limitations

Several limitations warrant consideration, starting with our use of model quantisation to deploy large-scale models on consumer-grade hardware,

specifically the 32B variants. While this approach ensures computational efficiency, it may introduce subtle reasoning trade-offs that increase friction in faithfully reproducing relational structures. Secondly, the pipeline’s declarative method with inference-time adaptation is theoretically domain-agnostic. However, our current assessment is limited to SemEval-derived review datasets from the laptop and restaurant domains. Consequently, the pipeline’s robustness on more complex or novel unannotated corpora remains a subject for future research. Finally, the process is not strictly annotation-free, as it relies on a small number of in-context learning examples ($k \in \{5, 10, 15\}$). While our method is resource-efficient as it does not require fine-tuning or parametrised optimisation, it still requires a modest foundation of high-quality human labels (at least 5-15 examples) to automatically synthesise prompts.

More fundamentally, an emerging line of work questions whether subjective tasks should be aggregated to a consensus at all, arguing instead that full annotation distributions should be retained and modelled directly (Plank, 2022; Leonardelli et al., 2023). Extending our pipeline to preserve rather than resolve disagreement remains for future work.

Acknowledgement

This work was supported by Research Ireland under Grant Number SFI/12/RC/2289_P2 (Insight_2).

References

- Jeremy Barnes, Lilja Øvrelid, and Erik Velldal. 2021. [If you’ve got it, flaunt it: Making the most of fine-grained sentiment annotations](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 49–62, Online. Association for Computational Linguistics.
- Hongjie Cai, Rui Xia, and Jianfei Yu. 2021. [Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions](#). 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 340–350, Online. Association for Computational Linguistics.
- Niklas Donhauser, Jakob Fehle, Nils Constantin Hellwig, Markus Weinberger, Udo Kruschwitz, and Christian Wolff. 2026. [Annotation quality in aspect-based sentiment analysis: a case study comparing experts, students, crowdworkers, and large language models](#). In *The fourth workshop on the role of resources in the age of large language models (RESOURCEFUL 2026)*, pages 73–88.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. [On calibration of modern neural networks](#). In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML’17*, pages 1321–1330, Sydney, NSW, Australia. JMLR.org.
- Daya Guo, Dejian Yang, Haowei Zhang, et al. 2025. [Deepseek-r1: Incentivizing reasoning in llms via reinforcement learning](#). *Nature*, 645(8081):633–638.
- Shibo Hao, Yi Gu, Haotian Luo, Tianyang Liu, Xiyao Shao, Xinyuan Wang, Shuhua Xie, Haodi Ma, Adithya Samavedhi, Qiyue Gao, Zhen Wang, and Zhiting Hu. 2024. [LLM reasoners: New evaluation, library, and analysis of step-by-step reasoning with large language models](#). In *First Conference on Language Modeling*.
- Maram Hasanain, Fatema Ahmad, and Firoj Alam. 2024. [Large language models for propaganda span annotation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14522–14532, Miami, Florida, USA. Association for Computational Linguistics.
- Eduard Hovy and Julia Lavid. 2010. Towards a ‘science’ of corpus annotation: a new methodological challenge for corpus linguistics. *International journal of translation*, 22(1):13–36.
- Minqing Hu and Bing Liu. 2004. [Mining and summarizing customer reviews](#). In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 168–177. ACM.
- Zdeněk Kasner, Vilém Zouhar, Patrícia Schmidtová, Ivan Kartáč, Kristýna Onderková, Ondrej Platek, Dimitra Gkatzia, Saad Mahamood, Ondrej Dusek, and Simone Balloccu. 2026. [LLMs as span annotators: A comparative study of LLMs and humans](#). In *Proceedings of the First Workshop on Multilingual Multicultural Evaluation*, pages 1–22, Rabat, Morocco. Association for Computational Linguistics.
- Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. [Dspy: Compiling declarative language model calls into self-improving pipelines](#). In *ICLR*.
- Urja Khurana, Eric Nalisnick, Antske Fokkens, and Swabha Swayamdipta. 2024. [Crowd-Calibrator: Can Annotator Disagreement Inform Calibration in Subjective Tasks?](#) In *First Conference on Language Modeling*.
- Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych. 2024. [Analyzing dataset annotation quality management in the wild](#). *Computational Linguistics*, 50(3):817–866.

- Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with pagedattention](#). In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP '23*, page 611–626, New York, NY, USA. Association for Computing Machinery.
- Elisa Leonardelli, Gavin Abercrombie, Dina Almanea, Valerio Basile, Tommaso Fornaciari, Barbara Plank, Verena Rieser, Alexandra Uma, and Massimo Poesio. 2023. [SemEval-2023 Task 11: Learning with Disagreements \(LeWiDi\)](#). In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2304–2318, Toronto, Canada. Association for Computational Linguistics.
- Bing Liu and Lei Zhang. 2012. [A Survey of Opinion Mining and Sentiment Analysis](#). In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 415–463. Springer US, Boston, MA.
- Nailia Mirzakhmedova, Marcel Gohsen, Chia-Hao Chang, and Benno Stein. 2024. [Are large language models reliable argument quality annotators?](#) In *Robust Argumentation Machines - First International Conference, RATIO 2024, Bielefeld, Germany, June 5-7, 2024, Proceedings*, volume 14638 of *Lecture Notes in Computer Science*, pages 129–146. Springer.
- Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. 2024. [Optimizing instructions and demonstrations for multi-stage language model programs](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9340–9366, Miami, Florida, USA. Association for Computational Linguistics.
- Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. [Knowing what, how and why: A near complete solution for aspect-based sentiment analysis](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8600–8607.
- Barbara Plank. 2022. [The “Problem” of Human Label Variation: On Ground Truth in Data, Modeling and Evaluation](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. [Opinion word expansion and target extraction through double propagation](#). *Comput. Linguistics*, 37(1):9–27.
- Cigdem Toprak, Niklas Jakob, and Iryna Gurevych. 2010. Sentence and expression level annotation of opinions in user-generated discourse. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10*, page 575–584, USA. Association for Computational Linguistics.
- Alexandra Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. [Learning from disagreement: A survey](#). *J. Artif. Intell. Res.*, 72:1385–1470.
- Julian Junyan Wang and Victor Xiaoqi Wang. 2025. [Assessing consistency and reproducibility in the outputs of large language models: Evidence across diverse finance and accounting tasks](#). *CoRR*, abs/2503.16974.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations*.
- Zengzhi Wang, Rui Xia, and Jianfei Yu. 2024. [Unified ABSA via Annotation-Decoupled Multi-Task Instruction Tuning](#). *IEEE Transactions on Knowledge and Data Engineering*, 36(11):7242–7254.
- David H. Wolpert. 1992. [Original contribution: Stacked generalization](#). *Neural Netw.*, 5(2):241–259.
- Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020. [Grid tagging scheme for aspect-oriented fine-grained opinion extraction](#). *CoRR*, abs/2010.04640.
- Haoliang Xiong, Zehao Yan, Chuhan Wu, Guojun Lu, Shiguan Pang, Yun Xue, and Qianhua Cai. 2023. [BART-based contrastive and retrospective network for aspect-category-opinion-sentiment quadruple extraction](#). *International Journal of Machine Learning and Cybernetics*, 14(9):3243–3255.
- Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. [Position-aware tagging for aspect sentiment triplet extraction](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2339–2349. Association for Computational Linguistics.
- Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021. [A Unified Generative Framework for Aspect-based Sentiment Analysis](#). 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 2416–2429, Online. Association for Computational Linguistics.
- Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2023. [A survey on aspect-based sentiment analysis: Tasks, methods, and challenges](#). *IEEE Transactions on Knowledge and Data Engineering*, 35(11):11019–11038.

Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. 2024. [ProSA: Assessing and understanding the prompt sensitivity of LLMs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1950–1976, Miami, Florida, USA. Association for Computational Linguistics.

A Dataset Details

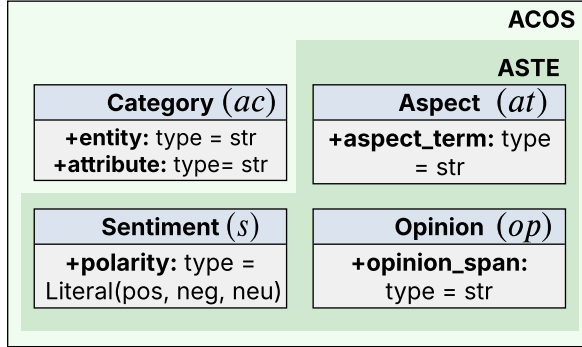


Figure 3: ACOS & ASTE Specifications

Because fine-tuning LLMs is not an objective of our annotation pipeline, we do not utilise training splits. Instead, we use the development split to provide In-context Learning (ICL) examples, which are included in the optimised prompt. The test splits remain unchanged, allowing us to evaluate performance across multiple LLMs in our experiments. The schema in Figure 3 illustrates the relationship between the two formulations we explore in this work.

ACOS	
Input	<i>"I have eaten here three times and have found the quality and variety of the fish to be excellent"</i>
Output	(quality, excellent, Pos, food#quality) (variety, excellent, Pos, food#style)
ASTE	
Input	<i>"moules were excellent, lobster ravioli was very salty."</i>
Output	(moules, excellent, Pos) (lobster ravioli, salty, Neg)

Table 7: Fine-Grained Opinion Examples

- **ASTE**: Aspect Sentiment Triple Extraction (ASTE) falls under the umbrella of ABSA tasks. Proposed by (Peng et al., 2020), it is a task for extracting opinion triplets, thus providing one-shot answer to: **What** target is being discussed, **How** is the sentiment (e.g. positive) and **Why** is this sentiment (e.g. "excellent").
- **ACOS**: This specification Aspect-Category-Opinion-Sentiment(ACOS) Quadruple Extraction, to extract aspect, aspect category, opinion and sentiment as quadruples in text and provide full support for aspect-based sentiment analysis with implicit aspects and opinions

³. The primary focus of this formulation is the fine-grained analysis of the opinion target. This task is more challenging than ASTE because it not only adds a category on top of the existing opinion triplet, but also takes into account implicit cases (i.e. missing explicit aspect terms or opinion spans) (Cai et al., 2021).

B Small-Tier Adjudicator Ablation

Table 8: Small-tier adjudicator ablation on ASTE (P / R / F_1).

Dataset	Qwen3-14B	Qwen3-4B-Thinking
Lap14	38.89 / 41.77 / 40.24	43.36 / 43.20 / 43.28
Res14	59.13 / 64.38 / 61.65	65.66 / 67.28 / 66.46
Res15	50.66 / 63.56 / 56.38	55.55 / 65.08 / 59.94
Res16	57.01 / 69.64 / 62.69	61.48 / 72.37 / 66.48

C Voting Adjudication Algorithms

Algorithm 1 presents the tuple-level and element-wise voting aggregators described in Section 4.2.1. **tuple-level Voting**: Illustrated in Algorithm 1, this method votes over the global set of annotator-tuple pairs $X = \bigcup_{i=1}^M \{(i, e)\}$ across M annotators using two matching variants: (i) **exact matching**, which groups tuples with identical normalised keys (including the aspect category for ACOS quadruples), and (ii) **flexible matching**, which handles span discrepancies via an undirected graph $G = (X, E)$. An edge connects (i, e_a) and (j, e_b) iff $s_a = s_b$ and both target and opinion spans overlap ($W(t_a) \cap W(t_b) \neq \emptyset \wedge W(o_a) \cap W(o_b) \neq \emptyset$). Implicit (null) spans map to a unique token to prevent matching explicit spans. Using Union-Find, we extract connected components as clusters $\mathcal{C}$ and discard those supported by fewer than θ unique voters ($|\{i \mid (i, e) \in C\}| < \theta$). For each surviving cluster C , the majority surface form e^* is selected as representative, breaking frequency ties by the longest explicit opinion span. **Element-Wise Voting**: Illustrated in Algorithm 2, this method operates on individual tuple fields rather than whole tuples. Tuples are aligned greedy-style (strongest alignment first, max one tuple per annotator) if $\geq 50\%$ of fields agree and either an explicit span overlaps or the tuples are identical. Each aligned cluster emits a single tuple whose fields are independently set by majority vote, breaking ties by

³Opinion in ACOS task refers to the span that expresses sentiment.

In: $\{A_1, \dots, A_M\}, \sim \in \{\text{exact}, \text{flex}\}, \theta$
Out: consensus set Y

```

1.  $X \leftarrow \bigcup_i \{(i, e) : e \in A_i\}$ 
2. if  $\sim \text{exact}$  then
3.    $C \leftarrow \text{partition } X \text{ by norm}(e)$ 
4. else
5.    $C \leftarrow \text{Components on span overlap}$ 
6.    $Y \leftarrow \emptyset$ 
7. for each  $C \in C$  do
8.   if  $|\{(i, e) \in C\}| \geq \theta$  then
9.      $e^* \leftarrow \text{argmax}_u(\text{freq}(u), \text{len})$ 
10.     $Y \leftarrow Y \cup \{e^*\}$ 
11. return  $Y$ 

```

Alg 1: tuple-level voting groups full tuples via exact or flexible matching.

highest annotator rank then longest surface form. Unlike tuple-level voting, this variant can **recombine fields** from different annotators, though it never introduces unproposed elements.

In: $\{A_1, \dots, A_M\}$ over fields F ; θ
Out: consensus set Y

```

1.  $X \leftarrow \bigcup_i \{(i, e) : e \in A_i\}$ 
2.  $E \leftarrow \{(a, b) : \text{ann}(a) \neq \text{ann}(b)\}$ 
3.  $\text{AGREE}(e_a, e_b)$ 
4.  $C \leftarrow \text{Components}(X, E), \text{greedy},$ 
5.    $\leq 1 \text{ tuple/annotator}$ 
6.    $Y \leftarrow \emptyset$ 
7. for each  $C \in C$  do
8.   if  $|\{(i, e) \in C\}| \geq \theta$  then
9.      $e^*[f] \leftarrow \text{MAJ}_f(C), \forall f \in F$ 
10.     $Y \leftarrow Y \cup \{e^*\}$ 
11. return  $Y$ 

```

Alg 2: Element-wise voting aligns tuples across shared fields and constructs representative tuples field-by-field.

D Adjudication Mechanism: Full Breakdown

Per-cell decomposition of each emitted tuple into copy (*match*), recombination, and generation (*unique*), with the correct count and the derived generation share and synthesis-correct rate.

Full mechanism breakdown Per-cell copy/recomb/generate counts with correct totals and derived %gen and %correct-from-synthesis, for the adjudicator and element-wise voting.

E Calibration of support

Table 10 gives the per-bucket precision underlying Figure 2(a) under exact matching. Precision is monotone in support for both tasks at every scale; the counts show how unevenly the high-support region is populated, with ACOS unanimity reaching only 33–153 tuples.

Scale	Method	copy	recomb	gen	Total
ASTE (3-element)					
Med	Exact	2527 (1590) <i>100.0%</i>	–	–	2527
	Flex	2713 (1588) <i>100.0%</i>	–	–	2713
	Elem	2868 (1639) <i>99.3%</i>	19 (0) <i>0.7%</i>	–	2887
	LLM	2643 (1651) <i>95.8%</i>	17 (2) <i>0.6%</i>	98 (9) <i>3.6%</i>	2758
Small	Exact	2392 (1494) <i>100.0%</i>	–	–	2392
	Flex	2620 (1505) <i>100.0%</i>	–	–	2620
	Elem	2805 (1555) <i>99.2%</i>	23 (0) <i>0.8%</i>	–	2828
	LLM	2834 (1522) <i>98.3%</i>	36 (0) <i>1.2%</i>	12 (0) <i>0.4%</i>	2882
Mini	Exact	1976 (1354) <i>100.0%</i>	–	–	1976
	Flex	2323 (1366) <i>100.0%</i>	–	–	2323
	Elem	2711 (1515) <i>98.5%</i>	42 (1) <i>1.5%</i>	–	2753
	LLM	2492 (1553) <i>94.0%</i>	60 (12) <i>2.3%</i>	98 (17) <i>3.7%</i>	2650
ACOS (4-element)					
Med	Exact	792 (272) <i>100.0%</i>	–	–	792
	Flex	1728 (325) <i>100.0%</i>	–	–	1728
	Elem	1843 (387) <i>90.3%</i>	198 (7) <i>9.7%</i>	–	2041
	LLM	1133 (451) <i>53.1%</i>	291 (50) <i>13.6%</i>	711 (84) <i>33.3%</i>	2135
Small	Exact	748 (225) <i>100.0%</i>	–	–	748
	Flex	1613 (285) <i>100.0%</i>	–	–	1613
	Elem	1761 (304) <i>91.2%</i>	169 (4) <i>8.8%</i>	–	1930
	LLM	1540 (360) <i>75.7%</i>	226 (11) <i>11.1%</i>	269 (20) <i>13.2%</i>	2035
Mini	Exact	469 (220) <i>100.0%</i>	–	–	469
	Flex	1437 (290) <i>100.0%</i>	–	–	1437
	Elem	1806 (390) <i>91.9%</i>	159 (11) <i>8.1%</i>	–	1965
	LLM	1487 (387) <i>75.6%</i>	116 (23) <i>5.9%</i>	365 (53) <i>18.5%</i>	1968

Table 9: Tuple output composition and correctness for ASTE (3-element) and ACOS (4-element). For each method, the upper row gives Total (Correct) and the lower italic row gives that category’s share of the method’s overall output. Exact and Flex cannot produce recombined or unique tuples by construction; these cells are marked “–”.

Task	Scale	s=0	s=1	s=2	s=3
ASTE	Mini	.184 (158)	.349 (770)	.633 (829)	.850 (893)
	Small	.000 (48)	.111 (540)	.390 (875)	.790 (1419)
	Med	.096 (115)	.330 (345)	.434 (735)	.779 (1563)
ACOS	Mini	.158 (481)	.184 (1168)	.512 (285)	.788 (33)
	Small	.063 (495)	.150 (978)	.333 (409)	.503 (153)
	Med	.134 (1002)	.317 (720)	.487 (298)	.678 (115)

Table 10: $P(\text{correct} \mid s)$ with bucket size in parentheses, exact matching.

Under flexible matching (Table 11) the ordering survives for ASTE but inverts for ACOS Mini and Med between $s=0$ and $s=1$, since flexible support 0 isolates tuples no annotator matched even loosely. Flexible agreement is also the weaker signal: for ASTE Mini it moves 46% more tuples into the top bucket (1305 against 893) while precision there falls from .850 to .765, because it admits approximate matches while correctness is still scored exactly.

Task	Scale	$s=0$	$s=1$	$s=2$	$s=3$
ASTE	Mini	.146 (48)	.277 (473)	.541 (824)	.765 (1305)
	Small	.000 (7)	.083 (289)	.288 (652)	.677 (1934)
	Med	.000 (16)	.263 (99)	.312 (551)	.700 (2092)
ACOS	Mini	.243 (136)	.113 (551)	.240 (574)	.326 (706)
	Small	.026 (78)	.072 (415)	.119 (519)	.290 (1023)
	Med	.172 (203)	.127 (355)	.215 (512)	.371 (1065)

Table 11: $P(\text{correct} \mid s)$ under flexible matching, bucket size in parentheses.

Task	Domain / Scale	nominal	fitted
ASTE	lap14 / Mini	.105	.072
	lap14 / Small	.289	.147
	lap14 / Med	.262	.125
	res14 / Mini	.091	.142
	res14 / Small	.184	.040
	res14 / Med	.141	.035
	res15 / Mini	.122	.051
	res15 / Small	.261	.108
	res15 / Med	.220	.075
	res16 / Mini	.078	.086
	res16 / Small	.220	.066
	res16 / Med	.235	.099
ACOS	laptop / Mini	.239	.108
	laptop / Small	.249	.178
	laptop / Med	.163	.067
	restaurant / Mini	.097	.115
	restaurant / Small	.152	.060
	restaurant / Med	.129	.122

Table 12: ECE per domain and scale; bold marks the lower of the two.

E.1 Selective prediction over support

Thresholding on support turns the signal into an emission rule: retain only tuples with $s \geq \tau$. The existing $Vot_{=}$ aggregator at $\theta = 2/3$ is one point on this curve, at $\tau=2$. Because abstention here discards a tuple rather than deferring a decision, raising τ costs recall as well as coverage—from .63 to .30 on ASTE Mini, and from .22 to .01 on ACOS Mini, where the high-precision corner covers almost nothing.

Task	Scale	$\tau=3$		$\tau=2$		$\tau=1$		$\tau=0$	
		cov	P	cov	P	cov	P	cov	P
ASTE	Mini	.34	.85	.65	.75	.94	.62	1.00	.60
	Small	.49	.79	.80	.64	.98	.54	1.00	.53
	Med	.57	.78	.83	.67	.96	.62	1.00	.60
ACOS	Mini	.02	.79	.16	.54	.76	.26	1.00	.23
	Small	.07	.50	.28	.38	.76	.23	1.00	.19
	Med	.05	.68	.19	.54	.53	.40	1.00	.27

Table 13: Coverage and precision at each support threshold over $\mathcal{Adj}(\mathcal{A}_1)$ output; $\tau=0$ is unfiltered.

E.2 Sentence-level agreement

Spearman rank correlation between agree_i and $F_1(\mathcal{O}_i^{\text{adj}}, \mathcal{O}_i^G)$, with 95% intervals from a sentence-level bootstrap. The two tasks occupy different regimes, the weakest ASTE value exceeding the strongest ACOS one.

Task	Scale	ρ	p	95% CI
ASTE	Mini	0.429	1.9×10^{-66}	[0.386, 0.474]
	Small	0.577	9.4×10^{-131}	[0.534, 0.617]
	Med	0.469	3.7×10^{-81}	[0.421, 0.511]
ACOS	Mini	0.307	6.5×10^{-32}	[0.254, 0.361]
	Small	0.341	2.5×10^{-39}	[0.287, 0.393]
	Med	0.260	5.3×10^{-23}	[0.206, 0.312]

Table 14: Sentence-level agreement versus adjudicated output quality.

E.3 Agreement-based triage

Sentences are ranked by agree_i and the lowest-scoring fraction withheld; Table 15 reports micro- F_1 over the retained remainder. ASTE improves steadily, while ACOS is near-flat below 30% and non-monotone between 10% and 20% at Small and Med, so deferral there pays only under substantial human effort.

Task	Scale	0%	10%	20%	30%	40%	50%
ASTE	Mini	61.3	63.7	68.1	71.5	73.5	75.1
	Small	56.4	60.5	64.7	67.1	71.4	77.0
	Med	62.8	66.3	68.7	71.1	75.6	78.3
ACOS	Mini	22.9	23.3	23.4	24.1	26.2	28.9
	Small	19.0	19.9	19.5	21.5	23.8	26.4
	Med	27.8	28.9	28.5	30.4	33.4	35.1

Table 15: Retained micro- F_1 against deferral rate.

E.4 Latency and Throughput

Per-sentence latency was derived from modification-timestamp gaps across 79 in-

tact runs on a single NVIDIA A40 (48 GB). Outliers ($> Q_3 + 1.5 \times \text{IQR}$; 1.5–9% of gaps) were discarded as restart stalls. Timings are pooled strictly within (task, model, step, shots) groups, excluding small folders and sub-2 s write-buffered runs, yielding $\sim 22.3\text{k}$ timed annotations (Table 16), of which 12.7k cover the models used in this work. Paired same-model rows show ACOS consistently slower than ASTE at equal shot settings (e.g. Qwen3-8B: 14.7–17.1 s vs. 17.6–27.4 s), which is why the ACOS-only annotator timings yield a conservative cost bound. Few-shot prompting increases latency up to $2\times$ for large adjudicators (Qwen3-30B on ASTE: 28.9 s zero-shot vs. 58.7 s 10-shot), with modest impact on smaller models. Costs use July 2026 A40 rates (\$0.29–\$0.44/h; £1 $\approx$ \$1.34).

Table 16: Per-sentence latency on an NVIDIA A40 across shot settings (per-group means and medians; post-outlier file counts). Upper blocks underpin the cost calculations in Section 5; the bottom block reports supplementary runs used solely to establish relative task speeds. DS=DeepSeek

Task	Model	Shots	Sent.	Mean (s)	Med. (s)
<i>Annotator passes (Table 2 models)</i>					
ACOS	DS-R1-Distill-Qwen-32B	0/5/10/15	2,395	63.5–69.4	61.6–67.7
ACOS	Qwen3-4B-Thinking	15	1,399	77.5	76.4
<i>Reference runs, scale-matched (not used in this work; task-ordering evidence only)</i>					
ASTE	Qwen3-8B	0/5/10/15	3,104	14.7–17.1	12.3–14.2
ACOS	Qwen3-8B	0/5/10/15	2,395	17.6–27.4	16.0–26.0
ASTE	DS-R1-0528-Qwen3-8B	0/5/10/15	3,077	16.7–34.7	16.1–20.3
ACOS	DS-R1-0528-Qwen3-8B	0/5/10/15	996	21.5–99.0	18.8–96.1
<i>Adjudication passes (Table 2 models)</i>					
ASTE	Qwen3-30B-A3B-Think	0/5/10/15	2,544	28.9–58.7	23.9–44.6
ASTE	Qwen3-14B	5	326	25.4	19.6
ASTE	Qwen3-4B-Thinking	5	1,461	45.9	39.2
ACOS	Qwen3-30B-A3B-Think	5/10/15	1,840	24.9–27.0	20.6–25.0
ACOS	Qwen3-14B	15	1,399	42.7	39.0
ACOS	Qwen3-4B-Thinking	5	1,370	50.7	48.2

F Complete Annotation and Adjudication Results

Tables 18 and 17 report the complete precision, recall, and F_1 scores summarised in Table 3.

G Element-Wise Assessment of ASTE and ACOS

Table 19 reports the element-wise and pairwise assessment of ASTE against the human annotations discussed in Section 7. Similarly, Table 20 reports the rigorous assessment of ACOS.

	Model	Laptop	Restaurant
		P / R / F1	P / R / F1
SFT	JET-ACOS	44.52 / 16.25 / 23.81	59.81 / 28.94 / 39.01
	Extract-classify	45.56 / 29.48 / 35.80	38.54 / 52.96 / 44.61
	BART-CRN	48.16 / 31.83 / 38.32	50.84 / 47.10 / 48.90
	UnifiedABSA	- / - / 42.58	- / - / 60.60
Inference-Time Adaptation	Double-Propagation	13.04 / 05.70 / 08.00	34.67 / 15.08 / 21.04
	$\mathcal{A}_1$	13.18 / 12.97 / 13.07	32.11 / 28.71 / 30.31
	$\mathcal{A}_2$	11.65 / 12.63 / 12.12	21.27 / 21.51 / 21.39
	$\mathcal{A}_3$	01.36 / 01.90 / 01.58	04.62 / 05.89 / 05.18
	Mini		
	$\mathcal{A}_{dj}(Vot_{\approx})$	10.94 / 7.96 / 9.21	33.22 / 21.62 / 26.19 ⁺
	$\mathcal{A}_{dj}(Vot_{=})$	32.57 / 6.14 / 10.33	59.36 / 16.27 / 25.54 ⁺
	$\mathcal{A}_{dj}(Vot_{el})$	13.21 / 13.15 / 13.18	30.59 / 27.18 / 28.79 ⁺
	$\mathcal{A}_{dj}(\mathcal{A}_1)$	16.44 / 15.92 / 16.18 [*]	32.90 / 30.46 / 31.63 ⁺
	$\mathcal{A}_1$	09.78 / 09.60 / 09.69	22.67 / 20.19 / 21.36
	$\mathcal{A}_2$	07.62 / 07.69 / 07.66	24.27 / 21.72 / 22.92
	$\mathcal{A}_3$	06.92 / 06.49 / 06.70	16.88 / 14.74 / 15.73
	Small		
	$\mathcal{A}_{dj}(Vot_{\approx})$	9.26 / 8.22 / 9.00	28.83 / 20.74 / 24.13 ⁺
	$\mathcal{A}_{dj}(Vot_{=})$	16.94 / 5.36 / 8.15	42.67 / 17.79 / 25.12 ⁺
	$\mathcal{A}_{dj}(Vot_{el})$	08.91 / 08.65 / 08.78	25.74 / 22.71 / 24.13 ⁺
	$\mathcal{A}_{dj}(\mathcal{A}_1)$	10.85 / 10.99 / 10.92 [*]	30.52 / 28.82 / 29.65 [*]
	$\mathcal{A}_1$	15.77 / 15.74 / 15.75	29.37 / 27.51 / 28.41
	$\mathcal{A}_2$	09.63 / 09.94 / 09.78	17.48 / 17.46 / 17.47
	$\mathcal{A}_3$	07.62 / 07.69 / 07.66	24.27 / 21.72 / 22.92
	Med		
	$\mathcal{A}_{dj}(Vot_{\approx})$	12.94 / 11.07 / 11.93 ⁺	26.66 / 21.51 / 23.81
	$\mathcal{A}_{dj}(Vot_{=})$	23.33 / 8.13 / 12.06 ⁺	45.76 / 19.43 / 27.28 ⁺
	$\mathcal{A}_{dj}(Vot_{el})$	12.82 / 12.98 / 12.90 ⁺	28.01 / 26.64 / 27.31 ⁺
	$\mathcal{A}_{dj}(\mathcal{A}_1)$	17.99 / 19.38 / 18.66 [*]	40.56 / 39.41 / 39.98 [*]

Table 17: LLM-based annotator performance on ACOS (P/R/F1) using an inference-time adaptation method. The baselines include a rule-based method (Qiu et al., 2011), sequence-labelling methods JET-ACOS and Extract-classify (Cai et al., 2021) and sequence-to-sequence generation method (BART-CRN by (Xiong et al., 2023)).

H Test For Statistical Significance

Tables 21 and 22 report paired-bootstrap comparisons of each aggregation method ($\mathcal{A}_{dj}(Vot_{\approx})$, $\mathcal{A}_{dj}(Vot_{=})$, $\mathcal{A}_{dj}(Vot_{el})$, and the LLM adjudicator $\mathcal{A}_{dj}(\mathcal{A}_1)$) against each individual annotator $\mathcal{A}_1$ – $\mathcal{A}_3$. Each cell gives the mean F1 difference (aggregated – individual, scaled to 0–100) with its 95% confidence interval and p -value; green shading marks a significant positive gain ($p < 0.05$). Green highlights that the adjudication performance is better than at least 2 of the individual annotators; yellow highlights mark that the adjudication performance is significantly better than all individual annotators.

	Model	Lap14	Res14	Res15	Res16	
		P / R / F1	P / R / F1	P / R / F1	P / R / F1	
SFT	JET	55.39 / 47.33 / 51.04	70.56 / 55.94 / 62.40	64.45 / 51.96 / 57.53	70.42 / 58.37 / 63.83	
	GTS	57.52 / 51.92 / 54.58	70.92 / 69.49 / 70.20	59.29 / 58.07 / 58.67	68.58 / 66.60 / 67.58	
	BART _{ABSA}	61.41 / 56.19 / 58.69	65.52 / 64.99 / 65.25	59.14 / 59.38 / 59.26	66.60 / 68.68 / 67.62	
	UnifiedABSA	- / - / 62.45	- / - / 73.22	- / - / 63.25	- / - / 72.10	
Inference-Time Adaptation	Mini	$\mathcal{A}_1$	42.34 / 43.39 / 43.15	62.27 / 64.98 / 63.86	50.96 / 60.60 / 55.11	62.43 / 70.81 / 66.36
		$\mathcal{A}_2$	34.05 / 38.08 / 35.95	54.71 / 60.16 / 57.31	41.70 / 55.46 / 47.61	52.42 / 67.51 / 59.01
		$\mathcal{A}_3$	21.63 / 25.92 / 23.59	36.18 / 41.22 / 38.53	30.12 / 40.87 / 34.68	40.65 / 51.27 / 45.34
		$\mathcal{A}_{dj}(Vot_{\approx})$	44.88 / 35.94 / 39.91 ⁺	64.99 / 57.98 / 61.29 ⁺	52.37 / 52.69 / 52.53 ⁺	65.72 / 68.03 / 66.85 ⁺
		$\mathcal{A}_{dj}(Vot_{=})$	57.09 / 35.19 / 43.54 ⁺	73.11 / 58.08 / 64.73 ⁺	61.63 / 51.65 / 56.21 ⁺	75.00 / 67.25 / 70.91 [*]
		$\mathcal{A}_{dj}(Vot_{el})$	40.32 / 41.90 / 41.10 ⁺	62.48 / 65.21 / 63.81 ⁺	48.53 / 58.09 / 52.88 ⁺	62.50 / 72.12 / 66.97 ⁺
		$\mathcal{A}_{dj}(\mathcal{A}_1)$	45.95 / 45.43 / 45.69 [*]	67.44 / 67.85 / 67.64 [*]	54.31 / 62.65 / 58.18 [*]	64.28 / 71.92 / 67.89 ⁺
	Small	$\mathcal{A}_1$	44.05 / 45.85 / 44.93	62.39 / 64.49 / 63.42	54.92 / 66.80 / 60.28	60.33 / 70.43 / 64.99
		$\mathcal{A}_2$	40.07 / 42.14 / 41.08	57.00 / 63.48 / 60.06	47.49 / 62.47 / 53.96	53.20 / 66.15 / 58.98
		$\mathcal{A}_3$	31.79 / 34.20 / 32.95	55.26 / 59.76 / 57.42	45.88 / 56.29 / 50.56	57.66 / 68.09 / 62.44
		$\mathcal{A}_{dj}(Vot_{\approx})$	45.01 / 40.85 / 42.82	63.90 / 63.77 / 63.84 ⁺	52.72 / 62.11 / 57.03 ⁺	61.68 / 69.84 / 65.51
		$\mathcal{A}_{dj}(Vot_{=})$	50.45 / 40.66 / 45.03 ⁺	68.80 / 63.46 / 66.02 [*]	58.07 / 61.07 / 59.53 ⁺	65.62 / 69.46 / 67.49 ⁺
		$\mathcal{A}_{dj}(Vot_{el})$	41.09 / 43.07 / 42.06	61.98 / 65.71 / 63.79 ⁺	51.74 / 64.60 / 57.46 ⁺	59.13 / 71.21 / 64.61
		$\mathcal{A}_{dj}(\mathcal{A}_1)$	38.89 / 41.77 / 40.24	59.13 / 64.38 / 61.65	50.66 / 63.56 / 56.38	57.01 / 69.64 / 62.69
	Med	$\mathcal{A}_1$	43.93 / 47.50 / 45.64	66.38 / 71.12 / 68.67	56.77 / 68.67 / 61.98	59.87 / 71.98 / 65.37
		$\mathcal{A}_2$	42.31 / 44.73 / 43.49	60.38 / 64.69 / 62.46	55.03 / 66.60 / 60.26	58.25 / 68.68 / 63.04
$\mathcal{A}_3$		40.07 / 42.14 / 41.08	57.00 / 63.48 / 60.06	47.49 / 62.47 / 53.96	53.20 / 66.15 / 58.98	
$\mathcal{A}_{dj}(Vot_{\approx})$		45.13 / 44.54 / 44.83	65.91 / 67.50 / 66.69 ⁺	56.76 / 66.59 / 61.29	59.62 / 68.67 / 63.83	
$\mathcal{A}_{dj}(Vot_{=})$		51.49 / 44.54 / 47.77 ⁺	70.32 / 67.70 / 68.99 ⁺	61.11 / 67.21 / 64.04⁺	61.51 / 68.09 / 64.63	
$\mathcal{A}_{dj}(Vot_{el})$		44.62 / 48.24 / 46.36	64.03 / 69.32 / 66.57 ⁺	55.28 / 69.07 / 61.41	57.10 / 68.87 / 62.43	
$\mathcal{A}_{dj}(\mathcal{A}_1)$		45.14 / 47.50 / 46.81	68.81 / 70.42 / 69.27 ⁺	57.66 / 68.24 / 62.51	62.33 / 72.76 / 67.14 ⁺	

Table 18: LLM-based annotator performance on ASTE (P/R/F1) using Inference-time adaptation methods. The baselines are SFT approaches using: sequence tagging method (BERT-JET by (Xu et al., 2020)), grid tagging method (BERT-GTS by (Wu et al., 2020) and sequence-to-sequence generative method by (Yan et al., 2021).

Data	$\mathcal{A}$	Mini						Small						Med					
		<i>s</i>	<i>at</i>	<i>op</i>	<i>s&at</i>	<i>s&op</i>	<i>at&op</i>	<i>s</i>	<i>at</i>	<i>op</i>	<i>s&at</i>	<i>s&op</i>	<i>at&op</i>	<i>s</i>	<i>at</i>	<i>op</i>	<i>s&at</i>	<i>s&op</i>	<i>at&op</i>
lap14	$\mathcal{A}_1$	83.55	66.74	64.02	58.22	56.53	48.67	85.80	66.18	65.93	59.56	60.36	48.91	84.81	66.12	72.73	58.02	64.64	51.33
	$\mathcal{A}_2$	80.81	63.21	50.98	53.82	45.61	39.82	83.29	64.30	61.88	55.96	53.97	47.03	85.80	64.46	68.75	56.49	60.52	49.96
	$\mathcal{A}_3$	79.77	51.34	37.39	43.30	32.01	26.99	86.25	58.32	49.31	52.30	44.12	36.87	83.29	64.30	61.88	55.96	53.97	47.03
	$Adj(Vot_{el})$	83.50	66.46	61.18	58.25	53.88	46.03	85.59	66.18	61.68	59.00	55.36	47.47	84.94	67.28	72.46	58.22	63.98	52.93
	$Adj(Vot_{\approx})$	79.94	64.55	51.61	58.35	46.68	43.64	83.86	64.73	57.23	59.18	51.97	47.09	84.21	67.45	67.63	58.39	58.76	51.72
	$Adj(Vot_{=})$	72.22	60.57	55.37	54.57	50.58	47.47	79.94	64.47	59.59	59.15	54.42	49.33	79.88	65.31	70.37	56.63	61.96	54.51
	$Adj(\mathcal{A}_1)$	83.46	67.38	69.24	57.62	61.55	52.62	84.21	63.81	60.43	56.07	53.82	45.63	84.44	67.36	72.91	58.03	64.37	53.37
res14	$\mathcal{A}_1$	90.86	79.93	75.50	75.64	71.77	67.13	89.94	78.35	77.00	73.95	73.06	66.43	90.44	81.35	79.38	77.53	75.73	71.10
	$\mathcal{A}_2$	85.61	77.74	68.55	70.72	62.91	61.69	88.21	77.08	72.02	72.48	67.76	62.42	89.53	78.65	75.79	73.73	71.70	65.86
	$\mathcal{A}_3$	86.49	63.91	53.05	59.00	49.63	40.38	90.50	77.81	68.51	73.69	65.24	59.86	88.19	77.35	72.29	72.75	67.97	62.83
	$Adj(Vot_{el})$	90.91	80.50	74.82	76.63	71.10	66.70	90.46	79.09	75.56	75.56	71.78	66.17	89.80	80.74	77.76	76.44	73.92	69.47
	$Adj(Vot_{\approx})$	89.86	79.32	70.16	76.54	67.64	63.01	90.46	79.51	73.72	76.00	70.52	65.99	90.14	80.90	77.70	76.66	73.88	69.68
	$Adj(Vot_{=})$	86.84	77.50	73.20	75.13	70.77	66.33	89.61	79.67	75.10	76.87	72.00	68.15	89.06	82.05	79.04	77.80	75.31	72.07
	$Adj(\mathcal{A}_1)$	90.63	80.47	79.97	75.97	76.05	70.59	90.24	78.56	73.58	74.11	69.96	64.78	90.20	81.64	79.82	77.28	75.98	72.24
res15	$\mathcal{A}_1$	88.49	76.80	72.87	69.92	64.91	61.19	90.53	82.39	73.86	75.62	67.52	66.11	89.50	77.44	78.25	71.00	71.30	67.79
	$\mathcal{A}_2$	83.26	75.46	63.20	66.19	54.44	54.37	88.71	78.43	68.71	71.80	61.82	59.70	89.47	78.92	76.62	72.42	69.38	66.04
	$\mathcal{A}_3$	85.96	60.81	51.07	53.82	46.01	38.76	91.01	79.43	65.75	73.85	59.06	55.76	88.74	78.52	68.83	71.92	61.96	59.84
	$Adj(Vot_{el})$	88.52	76.97	71.20	69.11	63.14	59.68	90.15	82.66	71.95	76.15	64.91	63.72	89.10	78.89	77.93	72.71	70.31	67.46
	$Adj(Vot_{\approx})$	86.35	77.27	67.03	70.88	61.14	57.70	89.92	82.63	70.68	76.15	63.86	63.31	89.54	79.38	77.26	73.83	70.22	66.98
	$Adj(Vot_{=})$	82.17	74.52	70.48	69.30	64.52	61.63	85.63	81.23	72.51	74.36	65.88	65.79	87.32	79.41	78.70	73.71	71.98	69.55
	$Adj(\mathcal{A}_1)$	88.48	79.02	76.14	71.10	68.37	65.13	89.85	80.56	70.76	73.62	63.94	62.63	89.69	79.82	78.27	73.73	71.63	67.99
res16	$\mathcal{A}_1$	88.75	81.83	78.93	76.14	73.16	71.23	89.79	79.35	79.61	73.74	74.29	69.48	88.98	78.83	80.55	73.16	74.68	70.14
	$\mathcal{A}_2$	88.83	75.00	72.00	69.78	67.05	62.69	87.14	76.36	75.34	70.04	69.56	63.49	90.91	76.98	78.16	72.18	73.35	66.79
	$\mathcal{A}_3$	87.91	65.34	57.80	61.07	53.83	47.58	89.97	79.01	74.42	74.11	70.54	65.83	87.14	76.36	75.34	70.04	69.56	63.49
	$Adj(Vot_{el})$	90.60	83.09	77.79	78.08	72.85	71.31	89.26	79.88	78.40	74.28	73.31	68.67	88.95	77.56	79.33	71.54	73.46	67.20
	$Adj(Vot_{\approx})$	90.50	81.51	75.88	78.28	72.97	69.35	89.39	81.40	78.39	76.00	73.17	69.71	89.55	78.65	79.60	73.13	74.40	68.17
	$Adj(Vot_{=})$	88.30	79.81	78.92	77.46	76.46	72.97	88.63	80.77	79.96	75.77	74.92	71.64	88.67	78.46	79.47	73.12	74.13	69.07
	$Adj(\mathcal{A}_1)$	89.55	81.96	81.33	76.29	75.70	72.49	89.04	78.09	77.21	72.44	71.77	67.08	89.73	80.04	82.44	74.62	76.45	72.17

Table 19: Element-wise assessment of ASTE’s alignment with human annotations (F1 score).

Dataset	Scale & Annot	s	at	op	E	A	ac	s + at	s + op	at + op	s + ac	at + ac	op + ac	at + $s+op$
laptop	$\mathcal{A}_1$	89.48	68.97	57.05	53.05	54.05	30.75	63.47	54.96	42.50	28.26	21.68	18.06	40.65
	$\mathcal{A}_2$	88.52	62.30	45.09	66.95	54.12	34.99	56.61	42.59	30.39	31.43	23.54	18.19	28.55
	$\mathcal{A}_3$	73.39	37.60	17.16	46.63	39.58	19.53	32.25	15.98	10.42	16.43	7.24	3.40	9.91
	$Adj(Vot_{el})$	89.09	67.76	52.39	58.32	53.41	32.69	62.51	50.23	38.70	30.19	22.98	18.49	36.94
	$Adj(V_{\approx})$	82.05	61.69	32.91	52.45	44.75	26.65	58.15	32.10	26.62	25.04	19.41	11.42	25.90
	$Adj(V_{=})$	34.62	27.74	24.02	27.66	26.06	18.78	26.40	23.00	19.74	17.83	15.59	12.35	18.85
	$Adj(\mathcal{A}_1)$	89.70	68.45	55.21	67.81	54.83	39.87	63.12	52.30	40.37	36.29	27.76	22.75	38.50
Restaurant	$\mathcal{A}_1$	90.34	68.69	65.58	62.51	76.59	55.87	63.70	61.79	51.20	51.85	40.53	39.00	48.75
	$\mathcal{A}_2$	87.77	61.60	55.47	57.80	65.34	49.39	58.54	52.76	41.05	45.86	31.35	29.70	39.51
	$\mathcal{A}_3$	76.86	38.22	23.84	34.68	49.03	26.51	34.30	21.89	15.26	23.54	13.85	8.46	14.27
	$Adj(Vot_{el})$	88.83	67.82	62.41	63.37	72.93	55.27	63.53	59.85	49.01	52.04	39.68	37.80	47.37
	$Adj(V_{\approx})$	79.67	60.09	48.01	55.12	63.49	48.42	57.66	46.39	39.68	46.57	36.27	31.84	38.61
	$Adj(V_{=})$	47.22	37.26	38.32	40.88	41.27	36.48	35.81	37.33	31.91	35.16	29.26	30.95	31.39
	$Adj(\mathcal{A}_1)$	90.48	69.26	58.19	85.88	75.92	66.25	64.77	54.08	45.64	61.83	50.14	40.86	43.00
laptop	$\mathcal{A}_1$	90.88	59.26	36.86	65.58	59.28	40.56	55.24	35.22	22.85	37.48	23.66	16.83	21.88
	$\mathcal{A}_2$	89.30	57.66	44.10	49.61	49.31	21.97	53.06	41.23	27.21	19.92	15.54	10.62	25.65
	$\mathcal{A}_3$	90.62	63.35	38.24	48.28	50.50	21.86	58.98	35.99	25.16	20.13	13.67	10.19	23.89
	$Adj(Vot_{el})$	90.34	60.38	38.66	61.11	58.62	35.88	55.95	36.53	25.00	32.86	21.36	15.63	23.58
	$Adj(V_{\approx})$	85.08	63.30	32.27	56.82	52.31	31.56	58.85	30.50	22.91	29.13	22.48	12.78	21.66
	$Adj(V_{=})$	51.65	41.31	25.05	36.14	33.10	20.78	38.71	23.72	19.53	19.58	16.95	10.57	18.46
	$Adj(\mathcal{A}_1)$	90.85	60.04	34.75	76.28	60.71	46.74	55.77	32.77	22.59	43.32	27.08	18.10	21.37
restaurant	$\mathcal{A}_1$	90.45	57.85	46.60	67.11	78.10	58.95	54.22	44.18	34.11	54.98	37.52	27.80	32.59
	$\mathcal{A}_2$	87.93	60.00	49.27	54.07	68.22	49.11	56.03	46.46	37.32	44.55	36.23	27.91	35.69
	$\mathcal{A}_3$	90.26	64.57	50.73	51.98	67.26	43.78	60.42	46.77	36.58	39.51	28.45	22.93	34.34
	$Adj(Vot_{el})$	89.67	61.23	49.27	65.54	77.17	57.94	57.30	46.83	37.32	53.47	39.53	30.89	35.68
	$Adj(V_{\approx})$	83.36	63.62	43.92	62.64	70.02	55.62	59.93	41.81	35.88	52.36	41.07	29.49	34.21
	$Adj(V_{=})$	61.75	49.49	40.00	51.21	52.18	46.90	46.95	38.26	33.02	44.24	38.24	29.70	31.77
	$Adj(\mathcal{A}_1)$	90.08	64.18	47.57	79.63	84.32	71.48	59.87	44.72	37.71	66.00	49.19	37.46	35.78
laptop	$\mathcal{A}_1$	89.86	68.20	55.29	62.96	53.05	36.30	62.67	52.17	39.57	33.35	25.95	22.61	37.46
	$\mathcal{A}_2$	90.37	61.84	43.96	56.42	56.75	33.15	57.22	42.07	29.04	30.59	20.72	15.40	27.66
	$\mathcal{A}_3$	89.30	57.66	44.10	49.61	49.31	21.97	53.06	41.23	27.21	19.92	15.54	10.62	25.65
	$Adj(Vot_{el})$	90.17	66.60	50.05	61.46	53.97	34.87	61.89	47.46	35.71	32.28	24.09	18.71	33.79
	$Adj(V_{\approx})$	86.19	65.47	39.38	57.09	50.11	31.39	61.07	37.62	29.55	29.57	23.28	15.89	28.04
	$Adj(V_{=})$	54.41	44.31	30.08	38.64	35.85	24.22	41.90	28.90	24.34	22.88	20.60	14.42	23.42
	$Adj(\mathcal{A}_1)$	90.40	73.53	59.59	70.51	54.27	41.23	67.71	56.60	45.80	37.90	30.68	24.84	43.53
restaurant	$\mathcal{A}_1$	89.94	69.39	64.13	65.41	72.28	54.91	63.90	60.16	50.09	49.01	39.26	37.66	47.46
	$\mathcal{A}_2$	91.50	63.30	54.71	52.97	63.72	46.25	58.98	52.52	39.54	42.48	28.08	25.09	38.32
	$\mathcal{A}_3$	87.93	60.00	49.27	54.07	68.22	49.11	56.03	46.46	37.32	44.55	36.23	27.91	35.69
	$Adj(Vot_{el})$	91.08	69.29	60.44	65.23	73.31	56.22	64.37	57.55	47.90	51.58	39.25	36.29	45.87
	$Adj(V_{\approx})$	87.88	67.69	51.52	59.43	66.00	49.67	63.96	49.90	42.45	46.14	37.44	29.14	41.10
	$Adj(V_{=})$	63.03	52.60	46.53	47.01	50.73	42.66	49.71	45.48	39.41	39.74	36.29	31.48	38.47
	$Adj(\mathcal{A}_1)$	91.01	72.74	64.60	82.96	82.43	71.75	68.13	61.17	52.86	66.40	55.78	48.85	50.06

Table 20: FULL Element-wise assessment of ACOS alignment with human annotations (F1 score).

Model	Dataset	Method	A1	A2	A3
MINI	LAP14	$Adj(V_{\approx})$	-2.90 [-6.26, +0.32], p=0.0829	+3.97 [+0.86, +7.14], p=0.0135	+16.49 [+12.78, +20.21], p=0.0000
		$Adj(V_{=})$	+0.73 [-2.73, +3.99], p=0.6758	+7.60 [+4.33, +10.86], p=0.0000	+20.12 [+16.36, +23.80], p=0.0000
		$Adj(V_{el})$	-1.72 [-3.86, +0.39], p=0.1159	+5.15 [+1.48, +8.89], p=0.0075	+17.67 [+13.91, +21.53], p=0.0000
		$Adj(A_1)$	+2.87 [+0.34, +5.48], p=0.0297	+9.75 [+5.71, +14.02], p=0.0000	+22.26 [+17.79, +26.80], p=0.0000
	RES14	$Adj(V_{\approx})$	-2.84 [-5.35, -0.43], p=0.0229	+3.68 [+0.96, +6.30], p=0.0063	+22.81 [+19.69, +26.10], p=0.0000
		$Adj(V_{=})$	+0.61 [-1.73, +2.85], p=0.6109	+7.13 [+4.45, +9.69], p=0.0000	+26.26 [+23.08, +29.58], p=0.0000
		$Adj(V_{el})$	-0.32 [-2.18, +1.62], p=0.7340	+6.20 [+3.35, +9.11], p=0.0001	+25.34 [+21.98, +28.83], p=0.0000
		$Adj(A_1)$	+3.51 [+1.44, +5.68], p=0.0017	+10.03 [+6.40, +13.59], p=0.0001	+29.17 [+25.31, +33.06], p=0.0000
	RES15	$Adj(V_{\approx})$	-2.38 [-5.45, +0.57], p=0.1235	+4.94 [+1.98, +7.93], p=0.0014	+17.85 [+13.97, +21.89], p=0.0000
		$Adj(V_{=})$	+1.30 [-2.04, +4.51], p=0.4338	+8.61 [+5.56, +11.64], p=0.0000	+21.52 [+17.56, +25.47], p=0.0000
		$Adj(V_{el})$	-2.03 [-4.66, +0.46], p=0.1171	+5.29 [+2.14, +8.59], p=0.0010	+18.20 [+14.22, +22.36], p=0.0000
		$Adj(A_1)$	+3.28 [+0.32, +6.22], p=0.0315	+10.60 [+6.64, +14.58], p=0.0000	+23.51 [+19.22, +27.91], p=0.0000
	RES16	$Adj(V_{\approx})$	+0.37 [-2.03, +2.81], p=0.7596	+7.74 [+3.65, +11.80], p=0.0005	+21.51 [+17.75, +25.36], p=0.0000
		$Adj(V_{=})$	+4.43 [+2.03, +6.94], p=0.0005	+11.80 [+7.86, +15.84], p=0.0000	+25.57 [+21.93, +29.34], p=0.0000
		$Adj(V_{el})$	+0.48 [-1.28, +2.35], p=0.6007	+7.85 [+3.44, +12.40], p=0.0012	+21.62 [+17.78, +25.54], p=0.0000
		$Adj(A_1)$	+1.41 [-0.67, +3.48], p=0.1864	+8.78 [+4.19, +13.53], p=0.0006	+22.55 [+18.32, +26.76], p=0.0000
MSMALL	LAP14	$Adj(V_{\approx})$	-2.10 [-4.70, +0.38], p=0.1059	+1.75 [-0.71, +4.26], p=0.1667	+9.88 [+6.89, +12.90], p=0.0000
		$Adj(V_{=})$	+0.11 [-2.64, +2.71], p=0.9388	+3.95 [+1.52, +6.39], p=0.0020	+12.09 [+9.07, +15.18], p=0.0000
		$Adj(V_{el})$	-2.87 [-5.01, -0.87], p=0.0081	+0.98 [-1.65, +3.70], p=0.4734	+9.11 [+6.01, +12.21], p=0.0000
		$Adj(A_1)$	-4.64 [-7.11, -2.30], p=0.0001	-0.80 [-3.19, +1.67], p=0.5225	+7.34 [+4.30, +10.43], p=0.0000
	RES14	$Adj(V_{\approx})$	+0.42 [-1.83, +2.78], p=0.7175	+4.13 [+1.86, +6.33], p=0.0005	+6.24 [+3.91, +8.57], p=0.0000
		$Adj(V_{=})$	+2.61 [+0.35, +4.98], p=0.0302	+6.32 [+4.10, +8.51], p=0.0000	+8.42 [+6.05, +10.77], p=0.0000
		$Adj(V_{el})$	+0.37 [-1.76, +2.59], p=0.7342	+4.08 [+1.96, +6.21], p=0.0006	+6.19 [+3.80, +8.59], p=0.0000
		$Adj(A_1)$	-1.77 [-3.87, +0.41], p=0.1059	+1.94 [-0.53, +4.33], p=0.1196	+4.05 [+1.45, +6.64], p=0.0022
	RES15	$Adj(V_{\approx})$	-3.10 [-5.74, -0.48], p=0.0219	+3.24 [+0.24, +6.15], p=0.0337	+6.66 [+3.67, +9.69], p=0.0000
		$Adj(V_{=})$	-0.59 [-3.06, +1.80], p=0.6348	+5.74 [+2.83, +8.53], p=0.0001	+9.16 [+6.00, +12.30], p=0.0000
		$Adj(V_{el})$	-2.67 [-4.86, -0.61], p=0.0139	+3.66 [+0.79, +6.49], p=0.0138	+7.09 [+3.74, +10.47], p=0.0000
		$Adj(A_1)$	-3.75 [-6.33, -1.22], p=0.0045	+2.58 [-0.26, +5.35], p=0.0721	+6.01 [+2.73, +9.32], p=0.0003
	RES16	$Adj(V_{\approx})$	+0.52 [-2.71, +3.66], p=0.7543	+6.53 [+4.06, +9.21], p=0.0000	+3.07 [-0.59, +6.52], p=0.0925
		$Adj(V_{=})$	+2.49 [-0.76, +5.71], p=0.1310	+8.51 [+6.02, +11.20], p=0.0000	+5.04 [+0.96, +8.74], p=0.0117
		$Adj(V_{el})$	-0.38 [-3.38, +2.56], p=0.7949	+5.63 [+3.24, +8.25], p=0.0000	+2.16 [-1.78, +5.86], p=0.2600
		$Adj(A_1)$	-2.29 [-5.35, +0.67], p=0.1347	+3.72 [+1.16, +6.31], p=0.0061	+0.25 [-3.33, +3.66], p=0.8844
MED	LAP14	$Adj(V_{\approx})$	-0.81 [-3.36, +1.67], p=0.5269	+1.35 [-1.60, +4.32], p=0.3640	+3.76 [+1.17, +6.40], p=0.0040
		$Adj(V_{=})$	+2.12 [-0.33, +4.60], p=0.0935	+4.28 [+1.34, +7.31], p=0.0046	+6.69 [+4.11, +9.38], p=0.0000
		$Adj(V_{el})$	+0.71 [-1.33, +2.78], p=0.4969	+2.87 [-0.22, +6.13], p=0.0741	+5.28 [+2.45, +8.17], p=0.0000
		$Adj(A_1)$	+1.16 [-1.51, +3.83], p=0.3962	+3.33 [-0.00, +6.73], p=0.0530	+5.73 [+2.76, +8.80], p=0.0006
	RES14	$Adj(V_{\approx})$	-1.97 [-3.96, +0.02], p=0.0520	+4.24 [+1.38, +6.96], p=0.0036	+6.63 [+4.54, +8.71], p=0.0000
		$Adj(V_{=})$	+0.32 [-1.72, +2.33], p=0.7608	+6.53 [+3.67, +9.20], p=0.0001	+8.92 [+6.68, +11.17], p=0.0000
		$Adj(V_{el})$	-2.10 [-3.99, -0.21], p=0.0272	+4.11 [+1.27, +6.78], p=0.0043	+6.50 [+4.16, +8.86], p=0.0000
		$Adj(A_1)$	+0.60 [-1.17, +2.39], p=0.5121	+6.82 [+3.77, +9.78], p=0.0000	+9.21 [+6.53, +11.90], p=0.0000
	RES15	$Adj(V_{\approx})$	-0.69 [-3.19, +1.66], p=0.5679	+1.03 [-1.38, +3.44], p=0.4078	+7.33 [+4.38, +10.27], p=0.0000
		$Adj(V_{=})$	+2.06 [-0.49, +4.62], p=0.1134	+3.79 [+1.35, +6.19], p=0.0024	+10.08 [+7.17, +12.95], p=0.0000
		$Adj(V_{el})$	-0.57 [-2.88, +1.75], p=0.6209	+1.15 [-1.48, +3.82], p=0.3972	+7.45 [+4.58, +10.38], p=0.0000
		$Adj(A_1)$	+0.53 [-2.03, +3.17], p=0.6941	+2.25 [-0.65, +5.16], p=0.1233	+8.55 [+5.23, +11.94], p=0.0000
	RES16	$Adj(V_{\approx})$	-1.54 [-4.66, +1.53], p=0.3295	+0.80 [-1.60, +3.13], p=0.5066	+4.86 [+2.24, +7.64], p=0.0004
		$Adj(V_{=})$	-0.74 [-3.81, +2.26], p=0.6355	+1.60 [-0.75, +3.86], p=0.1677	+5.66 [+3.08, +8.39], p=0.0000
		$Adj(V_{el})$	-2.94 [-5.86, -0.14], p=0.0445	-0.60 [-2.93, +1.67], p=0.5951	+3.46 [+0.92, +6.11], p=0.0079
		$Adj(A_1)$	+1.77 [-0.91, +4.64], p=0.2118	+4.11 [+1.17, +7.17], p=0.0085	+8.17 [+5.02, +11.51], p=0.0000

Table 21: Statistical Significance Test Results - ASTE (scaled to 0–100)

Model	Domain	Method	A1	A2	A3
MINI	Laptop	$Adj(V_{\approx})$	-3.69 [-5.39, -2.06], p=0.0001	-2.91 [-4.65, -1.24], p=0.0009	+7.63 [+5.94, +9.36], p=0.0000
		$Adj(V_{=})$	-2.57 [-4.63, -0.57], p=0.0118	-1.79 [-3.61, -0.04], p=0.0518	+8.75 [+6.64, +10.98], p=0.0000
		$Adj(V_{el})$	+0.27 [-1.00, +1.53], p=0.6672	+1.06 [-0.96, +3.09], p=0.3142	+11.59 [+9.61, +13.64], p=0.0000
		$Adj(A_1)$	+3.27 [+1.89, +4.72], p=0.0000	+4.05 [+1.81, +6.33], p=0.0008	+14.59 [+12.23, +16.94], p=0.0000
	Restaurant	$Adj(V_{\approx})$	-4.13 [-6.22, -2.13], p=0.0002	+4.91 [+1.81, +7.94], p=0.0011	+21.11 [+17.89, +24.36], p=0.0000
		$Adj(V_{=})$	-4.78 [-7.65, -1.97], p=0.0008	+4.25 [+1.62, +6.80], p=0.0017	+20.45 [+16.82, +24.14], p=0.0000
		$Adj(V_{el})$	-1.53 [-3.46, +0.37], p=0.1200	+7.50 [+4.55, +10.41], p=0.0000	+23.70 [+20.57, +26.94], p=0.0000
		$Adj(A_1)$	+1.32 [-1.96, +4.62], p=0.4292	+10.35 [+6.42, +14.21], p=0.0000	+26.55 [+23.13, +29.92], p=0.0000
MSMALL	Laptop	$Adj(V_{\approx})$	-0.60 [-1.72, +0.50], p=0.2855	+1.34 [-0.18, +2.93], p=0.0891	+2.31 [+0.70, +3.96], p=0.0049
		$Adj(V_{=})$	-1.46 [-3.28, +0.29], p=0.1107	+0.48 [-0.90, +1.87], p=0.4882	+1.45 [+0.02, +2.94], p=0.0513
		$Adj(V_{el})$	-0.83 [-2.00, +0.32], p=0.1640	+1.12 [-0.28, +2.55], p=0.1208	+2.08 [+0.54, +3.64], p=0.0077
		$Adj(A_1)$	+1.31 [+0.12, +2.50], p=0.0321	+3.26 [+1.61, +5.01], p=0.0002	+4.22 [+2.45, +5.99], p=0.0000
	Restaurant	$Adj(V_{\approx})$	+3.00 [+0.94, +5.13], p=0.0054	+1.32 [-1.31, +4.07], p=0.3377	+8.39 [+5.17, +11.69], p=0.0000
		$Adj(V_{=})$	+3.98 [+1.47, +6.47], p=0.0017	+2.30 [-0.50, +5.00], p=0.1032	+9.38 [+6.19, +12.58], p=0.0000
		$Adj(V_{el})$	+3.00 [+1.01, +5.05], p=0.0030	+1.32 [-1.28, +4.01], p=0.3250	+8.40 [+5.14, +11.62], p=0.0000
		$Adj(A_1)$	+8.51 [+5.79, +11.39], p=0.0000	+6.84 [+4.16, +9.64], p=0.0000	+13.91 [+10.79, +17.09], p=0.0000
MED	Laptop	$Adj(V_{\approx})$	-3.74 [-5.36, -2.14], p=0.0000	+2.23 [+0.57, +3.92], p=0.0093	+4.27 [+2.47, +6.14], p=0.0000
		$Adj(V_{=})$	-3.61 [-5.66, -1.59], p=0.0009	+2.36 [+0.76, +4.03], p=0.0047	+4.40 [+2.60, +6.27], p=0.0000
		$Adj(V_{el})$	-2.77 [-4.30, -1.31], p=0.0001	+3.20 [+1.46, +5.02], p=0.0001	+5.24 [+3.47, +7.08], p=0.0000
		$Adj(A_1)$	+2.99 [+0.63, +5.34], p=0.0139	+8.96 [+6.53, +11.29], p=0.0000	+11.00 [+8.54, +13.42], p=0.0000
	Restaurant	$Adj(V_{\approx})$	-4.27 [-6.71, -1.87], p=0.0004	+6.55 [+4.00, +9.09], p=0.0000	+1.00 [-2.55, +4.50], p=0.5819
		$Adj(V_{=})$	-0.79 [-3.60, +1.98], p=0.5701	+10.02 [+7.46, +12.52], p=0.0000	+4.47 [+0.87, +7.98], p=0.0157
		$Adj(V_{el})$	-0.76 [-2.52, +1.05], p=0.3925	+10.05 [+7.04, +13.02], p=0.0000	+4.50 [+0.68, +8.32], p=0.0209
		$Adj(A_1)$	+11.91 [+8.53, +15.30], p=0.0000	+22.72 [+18.78, +26.48], p=0.0000	+17.17 [+13.97, +20.46], p=0.0000

Table 22: Statistical Significance Test Results - ACOS (scaled to 0–100)