Takoyaki at SemEval-2026 Task 3: Ensembling LLM Predictions using Demonstration Retrieval for Dimensional Aspect-based Sentiment Analysis

Kosuke Yamada **Sho Takase** **Ryosuke Kohita**
CyberAgent
{yamada_kosuke, takase_sho, kohita_ryosuke}@cyberagent.co.jp

Abstract

This paper describes our system for SemEval-2026 Task 3 (DimABSA). We participate in Subtask 2 (DimASTE), which requires extracting triplets of aspect term, opinion term, and valence-arousal scores from review sentences, and Subtask 3 (DimASQP), which additionally requires aspect category classification to form quadruplets. Our proposed system consists of a multi-step pipeline: (1) **retrieval-based in-context learning** using BM25 to select relevant demonstrations for LLM inference, (2) **agreement-based ensemble** combining LLM predictions from multiple retrieval variants, and, for a subset of datasets, (3) **error-pattern correction** refining uncertain predictions using correction rule sets based on training data. Retrieval-based ICL and the agreement-based ensemble show consistent improvements across languages and domains. Error-pattern correction yields further improvement for the Japanese dataset. To further investigate output quality beyond automated evaluation metrics, we conducted human evaluation. The results suggest that LLM-based labeling achieves higher agreement with gold labels than human annotators, and additionally indicate a discrepancy between automated scores and practical output quality.

1 Introduction

Aspect-based sentiment analysis (ABSA) identifies specific aspects of an entity and classifies the associated sentiment, typically with coarse-grained categorical labels (Pontiki et al., 2014; Zhang et al., 2021; Zhou et al., 2024). Dimensional ABSA (DimABSA) (Lee et al., 2024), in contrast, represents sentiment as continuous valence-arousal (VA) scores based on Russell's circumplex model (Russell, 1980, 2003), enabling fine-grained sentiment analysis. However, it remains underexplored across diverse languages and domains due to the scarcity of annotated resources. SemEval-2026 Task 3 (DimABSA) (Yu et al., 2026; Lee et al., 2026) addresses this gap by providing a benchmark spanning multiple languages and domains.

In this task, we participate in Subtask 2 (DimASTE), which requires extracting triplets of aspect term, opinion term, and valence-arousal scores from review sentences, and Subtask 3 (DimASQP), which additionally requires aspect category classification to form quadruplets.[1] To bypass the high overhead of language- and domain-specific fine-tuning, we explore the efficacy of in-context learning (ICL) (Brown et al., 2020; Dong et al., 2024) and analyze the key factors behind its performance. Our system consists of a two-to-three-step pipeline: (1) retrieval-based ICL, which uses BM25 (Robertson and Zaragoza, 2009) to select training sentences similar to the target sentence as demonstrations for inference by LLMs such as Gemini (Gemini Team, 2023) or OpenAI GPT (OpenAI, 2025), (2) an agreement-based ensemble that integrates LLM predictions obtained from multiple retrieval strategies, and (3) error-pattern correction that refines uncertain predictions using correction rule sets based on training data, applied optionally on a subset of datasets.

Our experimental results show that retrieval-based demonstration selection yields a substantial gain, while advanced prompting strategies such as contrastive ICL (Mo et al., 2024) and cheat-sheet ICL (Honda et al., 2025) do not provide consistent benefit. The agreement-based ensemble effectively manages the trade-off between precision and recall inherent in combining multiple retrieval variants. For the Japanese dataset, error-pattern correction yields further improvement beyond the ensemble. To further investigate output quality beyond auto-

[1] We skip Subtask 1 (DimASR), which predicts VA scores for aspect and opinion terms provided as input, because we are interested in predicting sentiment for aspect and opinion terms that are themselves extracted from review sentences rather than given in advance.

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 1707–1723

mated evaluation metrics, we additionally conduct human evaluation on the Japanese dataset. The results suggest that LLM-based labeling achieves higher agreement with gold labels than human annotators, and indicate a discrepancy between automated scores and practical output quality, as the majority of predictions receive high quality ratings.

Among the eight Subtask 3 datasets, our system ranked first on English Restaurant, English Laptop, and Tatar Restaurant, second on Japanese Hotel, third on Russian Restaurant, fourth on Ukrainian Restaurant, fifth on Chinese Restaurant, and seventh on Chinese Laptop. On Subtask 2, it ranked first on English Restaurant and English Laptop, second on Tatar Restaurant, fifth on Russian Restaurant, Ukrainian Restaurant, and Chinese Restaurant, seventh on Japanese Hotel, and eighth on Chinese Laptop.

2 Background

2.1 Task Definition

The shared task (Yu et al., 2026) defines two structured extraction subtasks that share the same review sentences and continuous sentiment annotations. Subtask 2 (Dimensional Aspect Sentiment Triplet Extraction, DimASTE) extracts triplets of aspect term, opinion term, and valence-arousal (VA) scores, while Subtask 3 (Dimensional Aspect Sentiment Quad Prediction, DimASQP) further predicts an aspect category following the `Entity#Attribute` schema and thus yields quadruplets. Both subtasks combine text span extraction for aspect and opinion terms and continuous regression for VA scores on a 1–9 scale (5 = neutral), grounded in Russell's circumplex model (Russell, 1980, 2003), and Subtask 3 additionally requires closed-set classification for aspect categories. For example, given the input *"The Thai food was average to good, but the delivery was terrible."*, Subtask 2 expects the triplets (*Thai food*, *average to good*, 6.75#6.38) and (*delivery*, *terrible*, 2.88#6.62), while Subtask 3 expects the quadruplets (*Thai food*, `FOOD#QUALITY`, *average to good*, 6.75#6.38) and (*delivery*, `SERVICE#GENERAL`, *terrible*, 2.88#6.62). Both subtasks cover six languages including Chinese, English, Japanese, Russian, Tatar, and Ukrainian across restaurant, laptop, and hotel domains.

Systems are evaluated with continuous F1 (cF1), which extends standard F1 to handle continuous VA scores (Yu et al., 2026). A prediction is counted as a match only if its aspect term and opinion term exactly match those of a gold tuple, and Subtask 3 additionally requires the aspect category to match. Matched pairs then receive a similarity score inversely proportional to the VA distance. cF1 is the harmonic mean of continuous precision (cPre) and recall (cRec).[2]

2.2 Related Work

Categorical ABSA has progressed from aspect-level classification tasks (Pontiki et al., 2014, 2015, 2016) to triplet extraction modeled as pipeline and position-aware tagging (Peng et al., 2020; Xu et al., 2020), and further to quadruplet prediction reformulated as paraphrase generation with seq2seq models (Zhang et al., 2021, 2023). More recently, LLMs have been applied to ABSA through ICL and prompting (Zhang et al., 2024; Zhou et al., 2024) or fine-tuning (Šmíd et al., 2024), but best practices for leveraging LLMs in structured extraction, including demonstration selection and inference strategies, remain largely underexplored.

DimABSA was first explored as a shared task at SIGHAN-2024 (Lee et al., 2024), where participants tackled Chinese restaurant reviews with VA scores across three subtasks including intensity prediction, triplet extraction, and quadruplet extraction. Top systems combined BERT-based extraction with LLM classification (Xu et al., 2024) or applied ICL with retrieval-based demonstration selection (Zhu et al., 2024; Meng et al., 2024). The task at SemEval-2026 (Yu et al., 2026) extends this to six languages and three domains.

3 System Overview

Figure 1 illustrates our system, which combines two to three steps: retrieval-based ICL, agreement-based ensemble, and error-pattern correction. Since Subtask 2 (DimASTE) differs from Subtask 3 (DimASQP) only in the absence of aspect categories, we use the same pipeline for both subtasks and obtain Subtask 2 triplets by dropping the aspect category from the Subtask 3 quadruplet outputs. The remainder of this section therefore describes the pipeline for Subtask 3.

3.1 Step 1: Retrieval-Based ICL

A simple approach to ICL for LLMs is to sample demonstrations at random. However, demonstrations similar to the target input tend to share

[2] Formal definition in Appendix A.

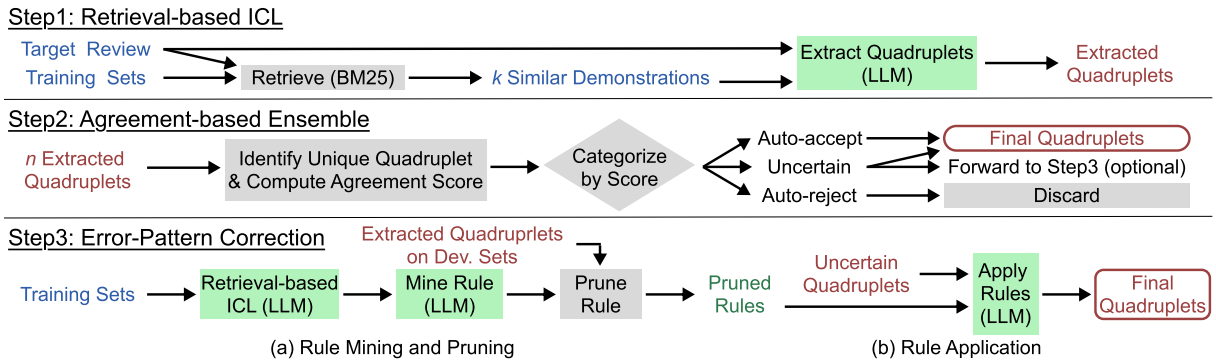

Figure 1: Our system overview. Steps 1 and 2 are applied to all datasets; Step 3 is optional for a subset of datasets.

structural patterns with it, leading to more consistent predictions (Liu et al., 2022; Luo et al., 2023), so for each target input we instead retrieve similar sentences with gold annotations using BM25 (Robertson and Zaragoza, 2009) and construct an ICL prompt with these as demonstrations.

We further run multiple parallel retrieval variants with varying tokenization schemes: character bigram, character trigram, and word segmentation. For word segmentation, we use jieba (Sun, 2012) for Chinese and UniDic-lite (McCann, 2020) for Japanese, and whitespace tokenization for all other languages. We use English prompts for all languages and additionally Japanese prompts for Japanese, yielding n=6 variants (3 tokenization schemes $\times$ 2 prompt languages); other languages use n=3.[3]

3.2 Step 2: Agreement-Based Ensemble

The union of n variants maximizes recall but degrades precision. We compute an *agreement score* for each unique quadruplet, defined as the number of variants, out of n, that produced the same quadruplet under exact string match on aspect term, aspect category, and opinion term. LLM outputs are categorized into three groups:

- **Auto-accept** ($= n$)**.** All variants agree. Output directly with high confidence.
- **Auto-reject** ($\leq t$)**.** Agreement at or below threshold t. Treated as noise and discarded.
- **Uncertain** ($t < \cdot < n$)**.** Intermediate agreement. Forwarded to Step 3.

For accepted outputs, VA scores are averaged across agreeing variants. When Step 3 is not applied, uncertain predictions are retained in the output without modification. The threshold t controls the balance between precision and recall, and we set $t = \lfloor n/2 \rfloor$ based on development data.

3.3 Step 3: Error-Pattern Correction

Comparing model predictions with gold annotations on development data reveals recurring error patterns such as category confusion and aspect granularity mismatch. We therefore mine correction rules from these patterns and apply them to uncertain predictions from Step 2.[4]

Rule mining and pruning We apply retrieval-based ICL (as in Step 1) to each training instance. An LLM simultaneously performs error analysis and generates correction rules by identifying similar gold–prediction pairs whose aspect or opinion spans share partial character overlap or whose category shares either the entity or attribute component, and describing the corresponding correction actions. Each rule is expressed as a natural language instruction that specifies a recurring error type and the transformation to apply, such as stripping a morphological suffix from an opinion span or removing an honorific prefix from an aspect term. Rules are then selected based on performance on the development set.

Rule application This performs correction and deletion on uncertain predictions from Step 2. The LLM receives uncertain predictions and correction rules, then independently decides whether to correct each element of each prediction. If correction results in duplicate quadruplets, their VA scores are averaged.

4 Experiments

Since our Subtask 2 (DimASTE) system drops aspect categories from the Subtask 3 (DimASQP)

[3]Prompt template in Appendix I.

[4]Prompt templates in Appendix I.

Dataset	Pipeline	Baseline cF1	Our cF1	Rank
eng-rest	Step 1–2	37.46	**65.14**	**1/16**
eng-lap	Step 1–2	27.95	**42.27**	**1/17**
jpn-hot	Step 1–3	19.43	40.86	2/10
rus-rest	Step 1–2	29.63	51.30	3/10
tat-rest	Step 1–2	23.80	**47.36**	**1/11**
ukr-rest	Step 1–2	29.71	50.19	4/10
zho-rest	Step 1–3	28.59	49.66	5/11
zho-lap	Step 1–3	19.00	37.45	7/11

Table 1: Submitted results on Subtask 3 (DimASQP). Baseline cF1 is the best score among the organizer-provided baseline systems in Lee et al. (2026). Step 1–2 = retrieval-based ICL with agreement-based ensemble. Step 1–3 = Step 1–2 with error-pattern correction. Rank shows unofficial competition standing.

Dataset	Pipeline	Baseline cF1	Our cF1	Rank
eng-rest	Step 1–2	49.20	**70.21**	**1/20**
eng-lap	Step 1–2	44.24	**63.66**	**1/19**
jpn-hot	Step 1–3	34.64	53.40	7/13
rus-rest	Step 1–2	42.42	55.64	5/13
tat-rest	Step 1–2	35.77	50.92	2/13
ukr-rest	Step 1–2	42.20	54.38	5/12
zho-rest	Step 1–3	35.29	53.82	5/12
zho-lap	Step 1–3	24.94	47.58	8/12

Table 2: Submitted results on Subtask 2 (DimASTE). Column definitions follow Table 1.

output, we report main results for both subtasks but conduct all subsequent analyses on Subtask 3 only.

4.1 Setup

We use the eight Subtask 3 (DimASQP) datasets, covering six languages and three domains (`eng-rest`, `eng-lap`, `jpn-hot`, `rus-rest`, `tat-rest`, `ukr-rest`, `zho-rest`, `zho-lap`).[5] We use Gemini 3.0 Pro (Gemini Team, 2023) as our primary model for all submitted runs.[6] For Step 3 error-pattern mining, we additionally run Gemini 3.0 Flash on training data using the same 50-shot retrieval setup, excluding the target instance from demonstrations. Rule mining and rule application themselves use Gemini 3.0 Pro. We enforce quadruplet output format using Gemini's structured output feature, which constrains decoding to a predefined JSON schema and guarantees syntactically valid responses. We set k=50 demonstrations per variant as a practical trade-off between cF1, cost, and latency.[7]

4.2 Main Results

Tables 1 and 2 report our submitted results on Subtask 3 (DimASQP) and Subtask 2 (DimASTE) alongside the best cF1 among the organizer-provided baseline systems reported in Lee et al. (2026). The two organizer-provided baselines reported in Lee et al. (2026) are Kimi K2 Thinking (Kimi Team, 2025) prompted in a one-shot setting and Qwen3-14B (Yang et al., 2025) fine-tuned with QLoRA (Dettmers et al., 2023), and Kimi K2 Thinking achieves the higher cF1 on every dataset in both subtasks, so we report its score as the Baseline cF1 in Tables 1 and 2. The scores across datasets vary substantially, from 37.45 (`zho-lap`) to 65.14 (`eng-rest`) on Subtask 3 and from 47.58 (`zho-lap`) to 70.21 (`eng-rest`) on Subtask 2, which suggests that the difficulty of the task varies across datasets. The ensemble pipeline was used for five datasets (`eng-rest`, `eng-lap`, `rus-rest`, `tat-rest`, `ukr-rest`), while the optional error-pattern correction was additionally applied to three datasets (`jpn-hot`, `zho-rest`, `zho-lap`) where it improved development performance. Our system outperforms the Baseline cF1 on all eight datasets in both subtasks by a large margin, with absolute cF1 gains ranging from 14.32 to 27.68 on Subtask 3 and from 12.18 to 22.64 on Subtask 2.

Our system ranks first on three of the eight Subtask 3 datasets (`eng-rest`, `eng-lap`, `tat-rest`) and on two Subtask 2 datasets (`eng-rest`, `eng-lap`), with per-dataset ranks shown in Tables 1 and 2. Although the systems of other teams have not yet been disclosed, these rankings suggest that our approach is particularly effective for very high-resource (English) and very low-resource (Tatar) languages, while fine-tuned or language-specific methods may hold an advantage for mid-resource languages. On Subtask 2, our advantage is largely limited to the two English datasets, and our relative standing on the non-English datasets is lower than on Subtask 3. For example, `jpn-hot` drops from second to seventh and `tat-rest` from first to second. This likely reflects that pipeline choices such as whether to apply error-pattern correction were tuned on Subtask 3 development performance, and that dropping aspect categories post hoc does not separately re-optimize the pipeline for triplet extraction.

[5] Dataset statistics in Appendix B.
[6] Model comparison and API details in Appendix E.
[7] Shot count comparison in Appendix F.

Method	cPre	cRec	cF1
Random ICL	30.53	35.44	32.80
Retrieval-based ICL (1)	35.62	41.33	38.26
+ Agreement-Based Ensemble (1–2)	35.47	43.16	38.94
+ Error-Pattern Correction (1–3)	38.64	43.35	40.86

Table 3: Component ablation on `jpn-hot`. Each subsequent row adds one method to the row above. Numbers in parentheses indicate the steps included in the pipeline.

4.3 Ablation Study

Table 3 decomposes each pipeline step's contribution on `jpn-hot`.[8] BM25 retrieval over random selection yields a substantial gain (+5.46 cF1), confirming that the choice of demonstrations strongly influences ICL performance on structured prediction tasks. The agreement-based ensemble adds a further +0.68 cF1. This gain comes primarily from recall (+1.83), as the union of six variants covers more gold quadruplets, while precision is largely maintained (−0.15).[9] The error-pattern correction further improves cF1 by +1.92, mainly through improved precision (+3.17).[10]

4.4 Error Analysis

Table 4 shows correction examples on `jpn-hot`. Cases 1 and 2 are successful: a *te*-form suffix is trimmed from the opinion, and an honorific prefix *o-* is stripped from the aspect. Case 3 is a failure: the same prefix rule incorrectly reduces *oyu* (hot water) to *yu*, missing that *oyu* is a lexicalized compound. This shows that pattern-based rules can over-generalize to fixed expressions. As in Case 2 and Case 3, we also find inherently difficult annotation cases in the gold data, where near-identical expressions receive different spans because annotators face genuine ambiguity, making it hard to distinguish system errors from legitimate annotation variability.

4.5 Discussion on Alternative Strategies

We explored two promising strategies to further improve retrieval-based ICL. Contrastive ICL (C-ICL) (Mo et al., 2024) enriches demonstrations with explicit negative contrast, and cheat-sheet ICL (Honda et al., 2025) distills training examples into a concise annotation guide supplied as additional context. All three methods use bigram-based BM25 with 50-shot demonstrations on `jpn-hot`.

[8] Full per-dataset pipeline results are in Appendix G.
[9] Agreement bucket distribution in Appendix C.
[10] Rule pruning statistics in Appendix D.

1	*Text*	駅から近くてとても良かった
	Before	駅 / 近くて (chikaku-te)
	After	駅 / 近く (chikaku) ✓
2	*Text*	ツインのお部屋は広くて清潔、...
	Before	お部屋 (o-heya) / 清潔
	After	部屋 (heya) / 清潔 ✓
3	*Text*	お湯...がとても良かったです
	Before	お湯 (o-yu) / とても良かった
	After	湯 (yu) / とても良かった ×

Table 4: Correction examples (*Aspect* / *Opinion*) on `jpn-hot`.

Method	cF1
Standard ICL	37.51
Contrastive ICL (C-ICL)	39.12
Cheat-sheet ICL	36.49

Table 5: Advanced prompting strategies on `jpn-hot`. Standard ICL denotes the plain retrieval-based baseline without additional prompting strategies.

Because DimASQP lacks explicit negatives for C-ICL, we constructed them from training-time prediction errors.

As Table 5 shows, C-ICL yielded a modest improvement over the baseline, but the gain did not justify the additional cost of negative construction. The ambiguous boundary between correct and incorrect labels makes reliable negatives difficult to obtain. Cheat-sheet ICL slightly degraded performance because annotation inconsistencies in the training data led to contradictory rules.

4.6 Human Evaluation

Automatic metrics such as cF1 rely on exact match against gold labels, but the DimASQP task involves subjective judgments where multiple valid annotations may exist for a single sentence. To investigate this, we conducted two human evaluations on `jpn-hot` with 80 test instances and three Japanese-speaking annotators, focusing on triplets only.[11] Throughout this section, *triplet* refers to the categorical elements (aspect term, opinion term, aspect category) of a DimASQP quadruplet and does not denote the VA-inclusive DimASTE triplet defined in Section 2.1. VA scores are not assessed in the human evaluation and are evaluated only through cF1 in the automatic results.

[11] Annotation guidelines in Appendix J.

Comparison Pair	Exact F1	Similar F1
Gold–Gemini	**0.333**	**0.673**
Gold–Human (avg.)	0.240	0.581
Human–Human (avg.)	0.213	0.587

Table 6: Annotation agreement on `jpn-hot`.

4.6.1 Annotation Agreement

In the first evaluation, annotators independently labeled all 80 instances and we computed pairwise F1 to quantify labeling consistency. We report two matching criteria: **Exact F1** requires all three elements of a predicted triplet to exactly match a reference triplet, while **Similar F1** relaxes aspect and opinion matching to token-level partial overlap, capturing near-miss predictions that differ only in span boundaries.

Table 6 shows the results.[12] The Exact F1 of Gold–Human is only 0.240 and the Exact F1 of Human–Human is 0.213, far below levels typically considered reliable, reflecting genuine ambiguities in the task. In contrast, Similar F1 scores are substantially higher across all pairs, above 0.50 in every comparison, indicating that a large share of predictions are near-misses that differ only in span boundaries rather than being fundamentally wrong. The Exact F1 of Gold–Gemini is 0.333, exceeding that of human annotators under both exact and similar matching, highlighting both the inherent difficulty of DimABSA annotation and the strong capability of Gemini as a base model for this task.

4.6.2 Prediction Quality Scoring

In the second evaluation, annotators scored each of the 169 Gemini-predicted triplets from the same 80 instances on a 1–5 quality scale. Table 7 reports the distribution: about 80.8% of predicted triplets scored 4–5, while scores 1–2 account for only 10.7%. Krippendorff's α (ordinal) (Krippendorff, 2011) is 0.373, confirming that fine-grained quality judgments remain subjective even among native speakers. The gap between cF1 and human satisfaction suggests that cF1 may penalize near-miss predictions that humans consider acceptable. These results indicate that incorporating at least partial-match credit into the evaluation metric would better align scoring with human judgment.

[12]Full pairwise matrix in Appendix H.

Score	Mean	Std
5	126.0	14.73
4	10.7	6.03
3	14.3	9.24
2	15.0	10.44
1	3.0	3.00

Table 7: Distribution of human quality scores for 169 Gemini-predicted triplets on 80 sampled `jpn-hot` instances. Each row shows the mean and standard deviation of the number of triplets receiving that score, averaged over three annotators.

5 Conclusion

We proposed a two-to-three-step ICL pipeline for both Subtask 2 (DimASTE) and Subtask 3 (DimASQP), ranking first on three of the eight Subtask 3 datasets and on two Subtask 2 datasets. Retrieval-based ICL yielded substantial gains through highly relevant demonstrations, and agreement-based ensemble further improved recall via broader coverage. Error-pattern correction partially improved precision on the Japanese dataset. Human evaluation confirms a gap between automatic scores and practical quality, with LLM-based labeling achieving higher agreement with gold labels than human annotators. Annotation inconsistencies fundamentally limit both evaluation and data-driven approaches. Future work should explore soft evaluation metrics capturing semantic closeness beyond exact match, alongside more consistent annotation guidelines.

Limitations

Our system relies entirely on ICL without fine-tuning, so the potential gains from combining ICL with parameter updates remain unexplored. The pipeline runs multiple retrieval variants and multiple LLM calls per instance, which incurs substantial API costs. Ablation and alternative strategy experiments were conducted primarily on Japanese data. We observed similar tendencies during development on other languages, but these were not systematically evaluated, so cross-lingual generalizability should be interpreted with caution. The human evaluation was conducted by three annotators on a single domain (`jpn-hot`), and broader evaluation across languages and domains would strengthen the findings. Our manual evaluation focuses on triplets rather than full quadruplets, so VA score quality is assessed only indirectly through cF1.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 (NeurIPS)*, volume 33, pages 1877–1901.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128.

Gemini Team. 2023. Gemini: A family of highly capable multimodal models. ArXiv preprint arXiv:2312.11805.

Ukyo Honda, Soichiro Murakami, and Peinan Zhang. 2025. Distilling many-shot in-context learning into a cheat sheet. In *Findings of the Association for Computational Linguistics: EMNLP 2025*.

Kimi Team. 2025. Kimi K2: Open agentic intelligence. ArXiv preprint arXiv:2507.20534.

Klaus Krippendorff. 2011. Computing Krippendorff's alpha-reliability. Departmental Papers (ASC).

Lung-Hao Lee, Liang-Chih Yu, Natalia Loukashevich, Ilseyar Alimova, Alexander Panchenko, Tzu-Mi Lin, Zhe-Yu Xu, Jian-Yu Zhou, Guangmin Zheng, Jin Wang, Sharanya Awasthi, Jonas Becker, Jan Philip Wahle, Terry Ruas, Shamsuddeen Hassan Muhammad, and Saif M. Mohammad. 2026. DimABSA: Building multilingual and multidomain datasets for dimensional aspect-based sentiment analysis. ArXiv preprint arXiv:2601.23022.

Lung-Hao Lee, Liang-Chih Yu, Suge Wang, and Jian Liao. 2024. Overview of the SIGHAN 2024 shared task for Chinese dimensional aspect-based sentiment analysis. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing*, pages 165–174.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out*, pages 100–114.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Zhao. 2023. Dr.ICL: Demonstration-retrieved in-context learning. In *Workshop on Robustness of Zero/Few-Shot Learning in Foundation Models*.

Paul McCann. 2020. fugashi, a tool for tokenizing Japanese in python. In *Proceedings of Second Workshop for NLP Open Source Software*, pages 44–51, Online. Association for Computational Linguistics.

Ling-ang Meng, Tianyu Zhao, and Dawei Song. 2024. DS-group at SIGHAN-2024 dimABSA task: Constructing in-context learning structure for dimensional aspect-based sentiment analysis. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing*, pages 127–132.

Ying Mo, Jiahao Liu, Jian Yang, Qifan Wang, Shun Zhang, Jingang Wang, and Zhoujun Li. 2024. C-ICL: Contrastive in-context learning for information extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10099–10114.

OpenAI. 2025. OpenAI GPT-5 system card. ArXiv preprint arXiv:2601.03267.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8600–8607.

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*, pages 19–30.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, pages 486–495.

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*, pages 27–35.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

James A. Russell. 1980. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178.

James A. Russell. 2003. Core affect and the psychological construction of emotion. *Psychological Review*, 110(1):145–172.

Jakub Šmíd, Pavel Priban, and Pavel Kral. 2024. LLaMA-based models for aspect-based sentiment analysis. In *Proceedings of the 14th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 63–70.

Junyi Sun. 2012. jieba: Chinese text segmentation tool. `https://github.com/fxsjy/jieba`.

Hongling Xu, Delong Zhang, Yice Zhang, and Ruifeng Xu. 2024. HITSZ-HLT at SIGHAN-2024 dimABSA task: Integrating BERT and LLM for Chinese dimensional aspect-based sentiment analysis. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing*, pages 175–185.

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*, pages 2339–2349.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. ArXiv preprint arXiv:2505.09388.

Liang-Chih Yu, Jonas Becker, Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Lung-Hao Lee, Ying-Lung Lin, Jin Wang, Jan Philip Wahle, Terry Ruas, Alexander Panchenko, Ilseyar Alimova, Kai-Wei Chang, Lilian Wanzare, Nelson Odhiambo, Bela Gipp, and Saif M. Mohammad. 2026. SemEval-2026 task 3: Dimensional aspect-based sentiment analysis (DimABSA). In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*. Association for Computational Linguistics.

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.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Pan, and Lidong Bing. 2024. Sentiment analysis in the era of large language models: A reality check. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3881–3906, Mexico City, Mexico. 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.

Changzhi Zhou, Dandan Song, Yuhang Tian, Zhijing Wu, Hao Wang, Xinyu Zhang, Jun Yang, Ziyi Yang, and Shuhao Zhang. 2024. A comprehensive evaluation of large language models on aspect-based sentiment analysis. ArXiv preprint arXiv:2412.02279.

Senbin Zhu, Hanjie Zhao, Xingren Wang, Shanhong Liu, Yuxiang Jia, and Hongying Zan. 2024. ZZU-NLP at SIGHAN-2024 dimABSA task: Aspect-based sentiment analysis with coarse-to-fine in-context learning. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing*, pages 112–120.

A Evaluation Metric: Continuous F1

We describe the continuous F1 (cF1) metric used for both Subtask 2 (DimASTE) and Subtask 3 (DimASQP) (Yu et al., 2026). For each sentence, predicted and gold tuples are matched by *exact string match* on categorical elements (after lowercasing). The matching key consists of aspect term and opinion term for Subtask 2, and Subtask 3 additionally requires the aspect category to match, so its key is (aspect, category, opinion). If multiple predictions match the same gold tuple, the gold tuple is treated as a false negative ($\mathrm{cTP} = 0$).

For a matched pair with predicted VA scores (V_p, A_p) and gold scores (V_g, A_g), the continuous true positive is:

$$\mathrm{cTP} = \max\left(0,\ 1 - \frac{\sqrt{(V_p - V_g)^2 + (A_p - A_g)^2}}{D_{\max}}\right) \tag{1}$$

where $D_{\max} = \sqrt{8^2 + 8^2} = \sqrt{128}$ is the maximum Euclidean distance in the VA space (scores range from 1 to 9). A perfect VA match yields $\mathrm{cTP} = 1$, and maximum distance yields $\mathrm{cTP} = 0$. Predictions with VA values outside $[1, 9]$ receive $\mathrm{cTP} = 0$.

Let $\mathrm{TP}_{\mathrm{cat}}$, $\mathrm{FP}_{\mathrm{cat}}$, and $\mathrm{FN}_{\mathrm{cat}}$ denote the counts of categorically matched, unmatched predictions, and unmatched gold tuples, respectively.

$$\mathrm{cPre} = \frac{\sum_t \mathrm{cTP}(t)}{\mathrm{TP}_{\mathrm{cat}} + \mathrm{FP}_{\mathrm{cat}}} \tag{2}$$

$$\mathrm{cRec} = \frac{\sum_t \mathrm{cTP}(t)}{\mathrm{TP}_{\mathrm{cat}} + \mathrm{FN}_{\mathrm{cat}}} \tag{3}$$

The denominator of cPre equals the total number of predicted tuples, and that of cRec equals the total number of gold tuples. The cF1 score is then:

$$\mathrm{cF1} = \frac{2 \cdot \mathrm{cPre} \cdot \mathrm{cRec}}{\mathrm{cPre} + \mathrm{cRec}} \tag{4}$$

Dataset	Language / Domain	Train	Dev	Test
eng-rest	English / Restaurant	2,284	200	1,000
eng-lap	English / Laptop	4,076	200	1,000
jpn-hot	Japanese / Hotel	1,600	200	800
rus-rest	Russian / Restaurant	1,240	48	630
tat-rest	Tatar / Restaurant	1,240	48	630
ukr-rest	Ukrainian / Restaurant	1,240	48	630
zho-rest	Chinese / Restaurant	6,050	300	1,000
zho-lap	Chinese / Laptop	3,490	300	1,000

Table 8: Dataset statistics shared by Subtask 2 (DimASTE) and Subtask 3 (DimASQP). The two subtasks share identical sentences and splits and differ only in the annotated element set.

Bucket	Count	Share
Auto-accept (agreement $= 6$)	768	42.5%
Auto-reject (agreement ≤ 3)	148	8.2%
Uncertain ($3 <$ agreement < 6)	893	49.4%
Total	1,809	100.0%

Table 9: Agreement bucket distribution on the jpn-hot test set with n=6 variants and threshold t=3. Counts refer to candidate quadruplet groups formed by the union of all six variants' predictions over 800 test sentences.

B Dataset Statistics

Table 8 summarises the eight datasets shared by Subtask 2 (DimASTE) and Subtask 3 (DimASQP), where the two subtasks use identical sentences and splits and differ only in whether aspect categories are annotated. The task covers six languages and three domains: restaurant (English, Chinese, Russian, Tatar, Ukrainian), laptop (English, Chinese), and hotel (Japanese). Training set sizes range from 1,240 sentences (rus-rest, tat-rest, ukr-rest) to 6,050 sentences (zho-rest).

C Agreement Bucket Distribution

Table 9 reports the distribution of candidate quadruplet groups across the three agreement buckets defined in Section 3.2 on the jpn-hot test set with n=6 variants and threshold t=3. The union of all six variants' predictions over the 800 test sentences yields 1,809 candidate groups.

The distribution shows that the ensemble separates candidates into three qualitatively different regions. Auto-accept captures 42.5% of all candidates under the strictest criterion (unanimous agreement across six variants), which produces a sizeable high-confidence core that exceeds the auto-reject mass by more than five times. Auto-reject accounts for only 8.2%, confirming that setting t=$\lfloor n/2 \rfloor$=3 is conservative and does not over-prune the recall surface. The largest single region is the uncertain bucket at 49.4%, where Step 3 error-pattern correction is applied when it helps on the development set. This concentration of mass in the middle region motivates the optional Step 3, since nearly half of the candidate groups are neither unanimously supported nor reliably rejectable and thus benefit most from targeted rule-based correction.

Dataset	Original	Pruned	Reduction
jpn-hot	43	14	67%
zho-rest	44	20	55%
zho-lap	41	5	88%

Table 10: Number of correction rules before and after pruning. Reduction = percentage of rules removed.

D Rule Pruning Statistics

Table 10 reports the number of correction rules before and after pruning for the three datasets where Step 3 is applied. Each rule is mined from training-set errors and retained only if it improves development-set performance when applied; the remaining rules are discarded. The pruning reduces the rule set by 55–88%.

Table 11 lists the full set of 43 rules mined for jpn-hot together with whether each rule was retained after development-set pruning and which extraction element it targets. Among 19 aspect rules 8 survive, while only 2 of 14 opinion rules and 4 of 10 category rules remain, so opinion-level normalisation has the lowest retention rate.

E LLM Comparison and API Details

Table 12 compares Gemini and OpenAI GPT models on jpn-hot under identical single-variant bigram BM25 ICL conditions with k=50 demonstrations. Gemini 3.0 Pro outperforms all evaluated alternatives by a wide margin. Gemini 3.0 Pro (gemini-3-pro-preview) and Gemini 3.0 Flash (gemini-3-flash-preview) are accessed via the Google AI API (https://ai.google.dev/). GPT-5 nano (gpt-5-nano), GPT-5 mini (gpt-5-mini), and GPT-5.2 (gpt-5.2) are accessed via the OpenAI API (https://platform.openai.com/). All API calls were made between January 12 and February 28, 2026.

Status	Rule Name	Target
✓	Honorific Prefix Removal	aspect
×	Staff/Person Suffix Removal	aspect
✓	Scope Limiter Removal	aspect
×	Attribute to Entity Mapping (Abstract)	aspect
✓	Head Noun Extraction (General Modifiers)	aspect
✓	Demonstrative Removal	aspect
×	Station Name Generalization	aspect
✓	Generalizer Suffix Removal	aspect
×	Generalize Room Attributes to Entity	aspect
×	Map Specific Dishes/Menu to Meal Category	aspect
×	Attribute Service Actions to Agents	aspect
×	Map Facility Components to Whole	aspect
✓	Standardize Location Terminology	aspect
×	Group Amenities under Category	aspect
×	Convert User Consequence to Object Quality	aspect
✓	Filter Objective State Descriptions	aspect
×	Normalize Location Facts to Evaluation	aspect
×	Simplify Situational Aspects	aspect
✓	Filter Functional Descriptions	aspect
×	Remove Polite Auxiliaries	opinion
×	Normalize Connective Adjective Forms	opinion
×	Strip Existence/State Verbs	opinion
×	Remove Aspect Context (Subject/Object)	opinion
×	Simplify Nominalizers and Vague Endings	opinion
×	Strip Polite Copulas	opinion
×	Remove Redundant Aspect Subject	opinion
×	Isolate Sentiment from Reason	opinion
×	Trim "Totemo" Inconsistency	opinion
×	Convert Spatial/Physical Facts to Evaluation	opinion
×	Normalize User Reactions to Facility Quality	opinion
✓	Convert Sensory Observations to Judgments	opinion
×	Remove Descriptive Mechanisms	opinion
✓	Filter Mere Existence	opinion
✓	Normalize Opinion Politeness	category
✓	Simplify Te-form Adjectives	category
✓	Extract Attribute from Possessive Target	category
×	Generalize Hotel Targets to NULL	category
✓	Trim Emotional Verb Phrases	category
×	Location Specificity	category
×	Hygiene to Cleanliness	category
×	Physical Dimensions to Design	category
×	Equipment to Facilities	category
×	Meal Plans to Food/Drinks	category

Table 11: Full list of 43 rules mined for `jpn-hot` and whether each rule was retained by development-set pruning (✓) or discarded (×). The Target column indicates which extraction element each rule rewrites.

F Number of Demonstrations

Table 13 shows the effect of demonstration count k on cF1 for `jpn-hot` under bigram BM25 retrieval with a single variant. Performance improves steadily from k=5 (34.66) to k=200 (38.29), but decreases at k=500 (36.72), likely because a very long context dilutes the relevance signal of retrieved demonstrations. Input tokens (and thus cost) grow roughly linearly with k while output tokens stay nearly constant around 100K, so the USD cost is dominated by the prompt and scales from $3.17 at k=5 to $111.45 at k=500. We select k=50 for all submitted runs as it provides a practical balance between performance, inference cost, and latency.

Model	cF1
Gemini 3.0 Pro	**37.59**
Gemini 3.0 Flash	33.85
GPT-5.2	31.45
GPT-5 nano	29.72
GPT-5 mini	29.66

Table 12: Model comparison on `jpn-hot` using bigram BM25 retrieval with 50-shot demonstrations.

Shots	cF1	Input (M)	Output (K)	Cost (USD)
5	34.66	1.00	98.1	3.17
10	36.53	1.52	96.7	4.21
20	37.30	2.60	97.5	6.38
50	37.51	5.87	98.0	12.92
100	37.92	11.38	98.4	23.94
200	38.29	22.37	99.9	45.94
500	36.72	55.12	100.5	111.45

Table 13: Effect of demonstration count on `jpn-hot` under bigram BM25 retrieval with a single variant. Input and output columns report cumulative token counts in millions (M) and thousands (K) respectively, and cost is computed with Gemini 3 Pro rates ($2.00 per 1M input tokens and $12.00 per 1M output tokens).

G Detailed Results by Pipeline Variant

Table 14 extends the component ablation in Table 3 to all eight test sets, reporting the cumulative contribution of each pipeline step using released test labels for post-hoc rescoring. Each dataset block follows the same method progression as Table 3: random ICL as the baseline, retrieval-based ICL (Step 1), agreement-based ensemble (Step 1–2), and error-pattern correction (Step 1–3). Bold indicates the pipeline variant submitted at competition time, chosen by development-set performance as reported in Section 4.2.

The agreement-based ensemble (step 1–2) improves over single-variant retrieval-based ICL in all eight datasets, confirming that ensemble consistency is a reliable source of gain regardless of language or domain. Error-pattern correction (step 1–3) further improves cF1 only for `jpn-hot`, while it regresses relative to step 1–2 on the test set for `zho-rest` and `zho-lap` despite development-set gains, suggesting that correction rules do not fully generalise to the test distribution for these datasets. For the remaining five datasets where step 1–2 was submitted, step 1–3 consistently underperforms, indicating that error-pattern correction is sensitive to the availability of reliable development-set signals for rule pruning.

Method	cPre	cRec	cF1
eng-rest			
Random ICL	62.84	63.58	63.21
Retrieval-based ICL (1)	64.43	64.80	64.62
+ Agreement-Based Ensemble (1–2)	66.45	63.89	**65.14**
+ Error-Pattern Correction (1–3)	63.64	60.83	62.20
eng-lap			
Random ICL	39.77	40.18	39.98
Retrieval-based ICL (1)	40.31	40.46	40.39
+ Agreement-Based Ensemble (1–2)	43.36	41.15	**42.23**
+ Error-Pattern Correction (1–3)	40.86	39.68	40.26
jpn-hot			
Random ICL	30.53	35.44	32.80
Retrieval-based ICL (1)	35.62	41.33	38.26
+ Agreement-Based Ensemble (1–2)	35.47	43.16	38.94
+ Error-Pattern Correction (1–3)	38.64	43.35	**40.86**
rus-rest			
Random ICL	43.23	52.00	47.21
Retrieval-based ICL (1)	44.07	54.63	48.78
+ Agreement-Based Ensemble (1–2)	48.34	54.65	**51.30**
+ Error-Pattern Correction (1–3)	45.84	55.92	50.38
tat-rest			
Random ICL	40.88	48.22	44.25
Retrieval-based ICL (1)	41.85	50.38	45.71
+ Agreement-Based Ensemble (1–2)	44.94	50.05	**47.36**
+ Error-Pattern Correction (1–3)	39.11	45.38	42.01
ukr-rest			
Random ICL	41.92	50.17	45.67
Retrieval-based ICL (1)	44.23	54.03	48.64
+ Agreement-Based Ensemble (1–2)	47.47	53.23	**50.19**
+ Error-Pattern Correction (1–3)	44.87	50.55	47.54
zho-rest			
Random ICL	42.51	51.03	46.38
Retrieval-based ICL (1)	45.02	53.70	48.98
+ Agreement-Based Ensemble (1–2)	47.96	52.66	50.20
+ Error-Pattern Correction (1–3)	46.09	53.83	**49.66**
zho-lap			
Random ICL	29.55	36.46	32.64
Retrieval-based ICL (1)	33.94	41.53	37.35
+ Agreement-Based Ensemble (1–2)	36.83	41.21	38.90
+ Error-Pattern Correction (1–3)	34.00	41.69	**37.45**

Table 14: Per-dataset pipeline comparison across all eight test sets using released test labels for post-hoc rescoring. Method names follow Table 3. Bold indicates the pipeline variant submitted, selected at submission time using development-set performance.

H Detailed Human Evaluation

Table 15 shows the full pairwise agreement matrix on the 80 sampled jpn-hot instances. Among annotators, Annotator A aligns most closely with the gold standard (Exact F1 0.329), while B and C score lower (0.221 and 0.171), yielding the 0.240 average in Section 4.6.

Pair (ref → hyp)	Exact F1	Similar F1
Gold → Gemini	0.333	0.673
Gold → Annotator A	0.329	0.605
Gold → Annotator B	0.221	0.572
Gold → Annotator C	0.171	0.567
Gemini → Annotator A	0.321	0.700
Gemini → Annotator B	0.159	0.527
Gemini → Annotator C	0.130	0.564
Annotator A → B	0.227	0.546
Annotator A → C	0.216	0.627
Annotator B → C	0.196	0.589

Table 15: Full pairwise annotation agreement on jpn-hot. Gold (original task annotations), Gemini (model outputs), A/B/C (human annotators).

I Prompt Templates

We use three categories of prompt templates across the pipeline. Figure 2 shows the retrieval-based ICL template for Step 1, which instructs the model to extract DimASQP quadruplets from review text using BM25-retrieved demonstrations. Figure 3 shows the correction template for Step 3, which supplies a pruned rulebook and asks the model to judge and correct each extracted element independently. Figures 4 and 5 show the error mining templates used to induce the rulebook from training-set errors before rule pruning.

J Annotation Guidelines

The following guidelines were originally written in Japanese and administered to three native Japanese-speaking annotators; we provide an English translation below, with examples in the original Japanese. Guideline 1, shown in Figure 6, was constructed based on the annotation guidelines described in Lee et al. (2026) and covers triplet annotation, where annotators independently identify Aspect Term, Opinion Term, and Aspect Category triplets. Guideline 2, shown in Figure 7, covers quality scoring of Gemini-predicted triplets on a 1–5 scale.

System Prompt

You are an expert in aspect-based sentiment analysis. Please extract the following Quadruplets from the given Japanese hotel review text:

1. **Aspect**: The specific aspect being discussed (e.g., "staff", "breakfast", "room")
2. **Opinion**: The opinion expression about the aspect (e.g., "kind", "delicious")
3. **Category**: The category of the aspect. Choose from the following categories: `{categories}`
4. **VA** (Valence and Arousal): Expressed in "Valence#Arousal" format (e.g., "6.33#6.17")
    - Valence: 1–9 scale for pleasantness (9=most pleasant, 1=most unpleasant)
    - Arousal: 1–9 scale for emotional intensity (9=strongest, 1=weakest)

If there are multiple aspects, include multiple elements in the array.

Few-Shot Examples (appended to system prompt when $k > 0$)

Here are examples of input text and expected output:

`Input: {input`$_1$`}`
`Output: {output`$_1$`}`

⋮ (k examples total)

`Input: {input`$_k$`}`
`Output: {output`$_k$`}`

User Prompt

Please analyze the following text.

`Input: {input}`

Figure 2: Step 1 ICL prompt, illustrated with `jpn-hot`. The system prompt (green) defines the quadruplet extraction task and category constraints. The few-shot block (orange) is appended when $k > 0$, with demonstrations retrieved by BM25. Other languages use the same structure with language- and domain-specific text. Placeholders in braces are filled at inference time.

System Prompt

You are an expert in output correction for Aspect-Based Sentiment Analysis (ABSA).
Task Overview: Correct ABSA prediction results based on the rulebook.
Absolute Constraints:
- Aspect and Opinion must be strings extracted from the input text.
- It is forbidden to output words that do not exist in the text as Aspect or Opinion.
- Category is a selection from the defined category list, not text extraction.
- VA is not subject to correction (keep as is).

Correction Policy (Aggressive Mode):
- If patterns match the rulebook, *actively correct* them.
- Judge each element (Aspect, Opinion, Category) independently.

Output in JSON format.

User Prompt

ABSA Task: Output Correction (Aggressive Mode)

Task: Correct the following prediction based on the rulebook.
Correction Policy: If patterns match the rulebook, actively make corrections.
Absolute Constraints:
- Aspect and Opinion must be strings extracted from the text.
- It is forbidden to output words that do not exist in the text as Aspect or Opinion.

Rulebook: `{rules}`

Target Data:
```
Text: {text}
Aspect: {pred_aspect}  Opinion: {pred_opinion}
Category: {pred_category}  VA: {pred_va}
```

Figure 3: Step 3 correction prompt. The system prompt (green) enforces absolute constraints, requiring that Aspect and Opinion be verbatim text spans while VA scores remain uncorrected. The user prompt (blue) supplies the pruned rulebook and requests structured JSON output with per-element reasoning and applied rule IDs. Each rule in the rulebook has a *Condition* describing the error pattern and an *Action* specifying the correction. Placeholders in braces are filled at inference time.

System Prompt

You are an expert in error analysis for Aspect-Based Sentiment Analysis (ABSA). Discover patterns inductively from the data and create a rulebook with reasoned transformation rules. Output your response in JSON format.

User Prompt (Element Analysis: similar / miss)

 # {target_element} Element: Comparative Analysis of match vs {error_type}

Task Create a **rulebook for converting {error_type} to match**.
{element_description}
{error_type_description}

Match Cases ({num_match} cases) The following are cases where Gold and Pred match exactly: {match_examples}

{error_type} Cases ({num_error} cases) The following are cases where Gold and Pred are {error_type_label}:
{error_examples}

Output Format (JSON) Please output in the following JSON format:

```
{
  "trends": [
    "Trend 1: Describe trends found by comparing match and {error_type}",
    "Trend 2: ...", "..."
  ],
  "rules": [
    {
      "rule_name": "Name of the rule (concise)",
      "condition": "When to apply (characteristics of Pred)",
      "action": "replace: replace with ~ / delete: delete",
      "reason": "Why this rule is effective (linguistic/domain reasons)",
      "examples": ["Case: 'XXX' -> 'YYY'", "Case: 'AAA' -> 'BBB'"]
    }
  ],
  "summary": "Brief summary of overall trends and recommendations"
}
```

Guidelines for Creating Rules

1. Create rules with reasons. Do not just say "if A then B"; explain **why**. Good example:

```
{
  "rule_name": "Remove polite suffixes",
  "condition": "Pred ends with polite forms (e.g., 'です', 'ます', 'ました')",
  "action": "replace: remove the polite suffix and convert to basic form",
  "reason": "In ABSA annotation standards, opinion expressions are recorded
             in basic form. Polite forms are common in speech but gold data
             is standardized in basic form.",
  "examples": ["'良かったです' -> '良かった'", "'美味しかったです' -> '美味しかった'"]
}
```

2. Avoid simplistic rules. Do not create rules that depend only on specific words, delete without reason, or have only a single example.

3. Include examples but keep it general. Write rules to apply to similar patterns, not just the given examples.

4. Must be determinable from Pred only. At runtime, Gold is unknown; conditions must be determinable from Pred alone.

Notes: `trends`: all discovered trends. `rules`: list with reasons. `summary`: 2–3 sentences.

Figure 4: Step 3 error mining prompt for element-level analysis. Applied to each combination of element type and error type to induce correction rules from training-set prediction errors. Element types are aspect, opinion, and category. Error types are similar and miss. The prompt supplies matched cases alongside error cases and asks the model to produce generalizable correction rules in JSON format. Placeholders in braces are filled at inference time.

System Prompt

You are an expert in output correction for Aspect-Based Sentiment Analysis (ABSA).
Task Overview: Correct ABSA prediction results based on the rulebook.
Absolute Constraints:
- Aspect and Opinion must be strings extracted from the input text.
- It is forbidden to output words that do not exist in the text as Aspect or Opinion.
- Category is a selection from defined category list, not text extraction.
- VA is not subject to correction (keep as is).

Correction Policy (Aggressive Mode):
- If patterns match the rulebook, *actively correct* them.
- Judge each element (Aspect, Opinion, Category) independently.

Output in JSON format.

User Prompt (Excess Analysis)

 # Analysis of Over-detection (Aspect-Opinion Pairs)

Task Analyze "over-detections" where there is no corresponding Gold for model predictions, and create a **rulebook**.
Goals:
- Identify **patterns to delete** (delete)
- Identify **patterns that become correct with modification** (replace)
- Avoid incorrectly processing correct Preds (match)

Reference of Correct Patterns (extracted from {num_match} match cases) The following are Aspect-Opinion pairs correctly predicted (Gold and Pred matched): {gold_examples}

Over-detection Cases ({num_excess} cases) The following are predictions with no corresponding Gold: {excess_examples}

Output Format (JSON)

```
{
  "trends": [
    "Trend 1: Describe patterns common to over-detections",
    "Trend 2: ...", "..."
  ],
  "rules": [
    {
      "rule_name": "Name of the rule (concise)",
      "condition": "When to apply (characteristics of Pred)",
      "action": "delete: delete / replace: replace with ~",
      "reason": "Why this rule is effective (linguistic/domain reasons)",
      "examples": ["Case description 1", "Case description 2"]
    }
  ],
  "summary": "Brief summary of overall trends and recommendations"
}
```

Guidelines for Creating Rules
1. Consider both delete and replace. Over-detections are either completely unnecessary (→ delete) or become correct with modification (→ replace to different Aspect/Opinion).

2. Create rules with reasons. Do not just say "delete"; explain **why** to delete/replace.

3. Avoid simplistic rules. Do not delete based only on specific words or without reason.

4. Include examples but keep it general. Write rules to apply to similar patterns.

Notes: `trends`: all discovered trends. `rules`: both delete and replace with reasons. `summary`: 2–3 sentences.

Figure 5: Step 3 error mining prompt for over-detection analysis. The system prompt (green) is identical to the correction prompt in Figure 3. Applied once per dataset to identify patterns in over-detected predictions and produce delete or replace rules. Placeholders in braces are filled at inference time.

Guideline 1: Triplet Annotation

The task is to manually annotate (Aspect Term, Opinion Term, Aspect Category) triplets from the review texts in `test_sampled.xlsx`. Each annotator received a file named `test_sampled_{name}.xlsx`.

Task overview. Aspect-Based Sentiment Analysis (ABSA) identifies fine-grained opinion elements—what is being evaluated (Aspect) and how it is evaluated (Opinion)—from review text. Unlike standard sentiment classification, which assigns a single sentiment label to an entire text, ABSA extracts structured sentiment at the aspect level. Each triplet consists of three elements:

Element	Description	Example
Aspect	A word or phrase indicating an opinion target	スタッフ, 朝食, 部屋
Opinion	A word or phrase expressing sentiment toward the aspect	親切, 美味しかった, 狭かった
Category	A predefined label in `entity#attribute` format	`service#quality`, `food_drinks#quality`

Spreadsheet structure. Each annotator received a spreadsheet with the following columns:

Column	Description
`ID`	Unique data identifier (read-only)
`Text`	Review text (read-only)
`Triplet-1` ... `Triplet-10`	Annotation fields (annotator fills in)

Input format. Each triplet column is filled in the format: `Aspect, Opinion, Category`. If a text contains multiple sentiments, annotators fill `Triplet-1`, `Triplet-2`, ... in order. Unused columns are left blank. If the aspect is not explicitly mentioned in the text, `NULL` is used as the Aspect Term. If no extractable triplet exists in the text, all columns are left blank.

Category taxonomy. Entity: `hotel`, `rooms`, `facilities`, `room_amenities`, `service`, `location`, `food_drinks`. Attribute: `general`, `price`, `comfort`, `cleanliness`, `quality`, `design_features`, `style_options`, `miscellaneous`. Do not use any categories outside the predefined list.

Annotation rules:

Rule 1: Extract all Aspect-Opinion pairs.
If a sentence expresses multiple sentiments, annotate each as a separate triplet.
Example: 施設は少し古いですが部屋は綺麗に清掃されており、朝食が何より素晴らしく美味しいです

`Triplet-1`	`Triplet-2`	`Triplet-3`	`Triplet-4`
施設, 少し古い, `facilities#design_features`	部屋, 綺麗, `rooms#cleanliness`	朝食, 素晴らしく, `food_drinks#quality`	朝食, 美味しい, `food_drinks#quality`

Rule 2: Include adverbs and negations in the Opinion span. Adverbs and negations affect sentiment polarity or intensity and must be included in the Opinion term.
Example: 鮎も季節的に冷凍物なのかあまり美味しいと感じませんでした → (鮎, あまり美味しいと感じませんでした, `food_drinks#quality`). The adverb あまり and the negation 感じませんでした must both be included. Extracting only 美味しい would reverse the sentiment.

Rule 3: Extract only the relevant span. Avoid over- or under-extraction. Include only the sentiment-bearing portion.
Example: お風呂が大きくてよかったです → two triplets: (風呂, 大きくて, `facilities#design_features`) and (風呂, よかった, `facilities#quality`).

Rule 4: Copy verbatim from the source text. Aspect and Opinion terms must be copied verbatim from the source text. Do not paraphrase, normalize, or add words not present in the original.
Example: 朝食も地元の食材を用いた品々で、ジャージー牛乳やヨーグルトもとってもおいしかったです → (ジャージー牛乳, とってもおいしかった, `food_drinks#quality`) and (ヨーグルト, とってもおいしかった, `food_drinks#quality`). The Opinion must use the verbatim form とってもおいしかった from the text.

Rule 5: Use only predefined categories. Do not create, modify, or extend category labels beyond the taxonomy. Use only `entity#attribute` combinations from the predefined list.

Figure 6: Triplet annotation guideline administered to three annotators for the human evaluation on `jpn-hot`.

Guideline 2: Quality Scoring for Gemini Outputs

The task is to rate the quality of each AI-predicted (Gemini) triplet on a **1–5** scale for the same 80 sampled instances. Each annotator received a file named `test_sampled_gemini_{name}.xlsx`.

Spreadsheet structure.

Column	Description
`InstanceID`	Unique instance identifier (read-only)
`ID`	Original data ID (read-only)
`Text`	Review text (read-only)
`Triplet`	AI-predicted `Aspect, Opinion, Category` (read-only)
`Score`	**1–5 quality score (annotator fills in)**
`Note`	Optional free-text memo

Scoring rubric.

Score	Rating	Tolerance	Criteria
5	Perfectly correct	No correction needed	Aspect, Opinion, and Category are all correct.
4	Mostly correct	Not perfect, but no correction needed	Pair and Category correct, with slight span boundary deviation.
3	Partially correct	Usable with minor correction	Category error or large span boundary deviation.
2	Incorrect	Major correction needed	Hallucination or pairing error.
1	Completely wrong	Easier to redo than correct	Output unrelated to text, indicating fundamental task misunderstanding.

Scoring examples (from `jpn-hot`):

Score	Text	Triplet	Reason
5	スタッフも親切でまた近いうちに伺いますね	スタッフ, 親切, `service#quality`	All correct.
4	最近できたばかりでとても綺麗なホテルです	ホテル, 綺麗, `hotel#cleanliness`	Gold opinion is とても綺麗; pair and category are correct.
3	部屋が思ったより狭かった	部屋, 狭かった, `rooms#quality`	Correct category is `rooms#design_features`.
2	お風呂が大きくてよかったです	風呂, 大きい, `facilities#cleanliness`	Opinion should be 大きくて; category should be `design_features`. Multiple errors.
1	朝食もおいしかったです	ロビー, 広い, `location#general`	Entirely unrelated to the text.

Figure 7: Quality scoring rubric guideline for evaluating Gemini-predicted triplets on a 1–5 scale.