# AILS-NTUA at SemEval-2026 Task 3: Efficient Dimensional Aspect-Based Sentiment Analysis

**Stavros Gazetas** **Giorgos Filandrianos** **Maria Lymperaiou**
**Paraskevi Tzouveli** **Athanasios Voulodimos** **Giorgos Stamou**
School of Electrical and Computer Engineering, AILS Laboratory
National Technical University of Athens
stavrosgaz9@gmail.com, {geofila, marialymp}@ails.ece.ntua.gr
tpar@mail.ntua.gr, thanosv@mail.ntua.gr, gstam@cs.ntua.gr

## Abstract

In this paper, we present the AILS-NTUA system for Track-A of SemEval-2026 Task 3 on Dimensional Aspect-Based Sentiment Analysis (DimABSA), which encompasses three complementary problems: Dimensional Aspect Sentiment Regression (DimASR), Dimensional Aspect Sentiment Triplet Extraction (DimASTE), and Dimensional Aspect Sentiment Quad Prediction (DimASQP) within a multilingual and multi-domain framework. Our methodology combines fine-tuning of language-appropriate encoder backbones for continuous aspect-level sentiment prediction with language-specific instruction tuning of large language models using LoRA for structured triplet and quadruplet extraction. This unified yet task-adaptive design emphasizes parameter-efficient specialization across languages and domains, enabling reduced training and inference requirements while maintaining strong effectiveness. Empirical results demonstrate that the proposed models achieve competitive performance and surpass the provided baselines in most evaluation settings[1].

## 1 Introduction

Aspect-Based Sentiment Analysis (ABSA) seeks to discern the sentiment directed at a specific aspect of an entity, facilitating a fine-grained analysis that transcends sentiment polarity. In recent years, research has increasingly moved from predicting categorical sentiment for isolated elements toward structured ABSA formulations that extract richer opinion structures, such as triplets in Aspect Sentiment Triplet Extraction (ASTE) (Peng et al., 2020) and quadruplets in Aspect Sentiment Quad Prediction (ASQP) (Zhang et al., 2021a), built around aspect terms/categories and opinion expressions with sentiment polarity (Zhang et al., 2022). By replacing the sentiment polarity with continuous valence–arousal (VA) scores, where valence reflects positivity/negativity and arousal reflects activation/intensity (Russell, 1980), more nuanced emotional information can be provided, as explored in a shared task (Lee et al., 2024).

SemEval-2026 Task 3: Dimensional Aspect-Based Sentiment Analysis (DimABSA) extends ABSA to a multilingual and multi-domain setting by utilizing VA annotations to facilitate analysis at both the aspect and sentiment levels (Yu et al., 2026). Our work revolves around Track-A (Lee et al., 2026), which includes three subtasks: Subtask 1 - Dimensional Aspect Sentiment Regression (DimASR), Subtask 2 - Dimensional Aspect Sentiment Triplet Extraction (DimASTE), and Subtask 3 - Dimensional Aspect Sentiment Quad Prediction (DimASQP), which we explore in this work.

To address the three subtasks, we propose a unified yet task-specialized framework that combines aspect-conditioned regression with instruction-tuned structured generation: DimASR is modeled as continuous VA regression, while DimASTE and DimASQP are cast as constrained JSON generation. Our contributions are: 1 A parameter-efficient multilingual regression framework for DimASR, fine-tuning language-appropriate encoders per language–domain pair to predict valence–arousal scores, outperforming the provided baselines in most settings. 2 A unified LoRA-based instruction-tuning pipeline for DimASTE and DimASQP on Llama and Qwen models ($\leq$14B), evaluated under zero-, few-, and supervised setups, achieving competitive or superior cF1 compared to larger fully fine-tuned LLMs. 3 An empirical study of translation-based cross-lingual transfer, analyzing the trade-offs between multilingual adaptation and translation-induced noise in lower-resource settings.

[1] Code: stavgaz/Semeval2026-Efficient-DimABSA



*Proceedings of the The 20th International Workshop on Semantic Evaluation (2026)*, pages 1212–1227

## 2 Related Work

**Tagging and span-based structured ABSA.** Earlier work on structured ABSA commonly revolved around using sequence labeling, tagging, and span-based methods to jointly extract triplets (aspect, opinion, sentiment) and quadruplets (aspect, category, opinion, sentiment). Regarding ASTE, approaches such as position-aware tagging (Xu et al., 2020), grid-based token-pair labeling (Wu et al., 2020), and span-based formulations (Xu et al., 2021) were implemented. As for ASQP, non-generative approaches frequently employ table-filling or grid-tagging to predict relations among the elements in a unified manner (Zhou et al., 2023), building on earlier quadruple settings such as ACOS (Cai et al., 2021). These strategies motivate further research that substitutes discrete labeling methods with serialized tuple generation.

**Generative structured ABSA.** Structured ABSA has increasingly been formulated as text generation, where models decode serialized sentiment structures directly from the input text. Early work used encoder-decoder models such as T5/BART to generate the targets in unified text-to-text frameworks (Zhang et al., 2021b; Yan et al., 2021). More recent work has utilized large language models (LLMs) for ABSA under zero-shot and few-shot prompting, as well as instruction-style fine-tuning, showing promise across different domains and languages (Wu et al., 2025; Wang et al., 2024; Zhou et al., 2024; Simmering and Huoviala, 2023). This groundwork motivated our use of instruction-tuned LLMs for structured ABSA in the dimensional (VA) setting of DimABSA.

**Dimensional ABSA.** The VA annotations have recently been adopted in ABSA as a way to extract fine-grained information. Most notably, the SIGHAN-2024 dimABSA shared task studied dimensional ABSA with subtasks covering intensity prediction and structured triplet/quadruplet extraction in VA space (Lee et al., 2024). In this shared task, many approaches were proposed for the Chinese language, including span-based extraction with contrastive learning (Tong and Wei, 2024), paraphrase/generation-style prediction (Jiang et al., 2024), and LLM-based approaches, including few-shot in-context learning and fine-tuning (Meng et al., 2024; Xu et al., 2024).

## 3 Task and Dataset Description

### 3.1 Task Definition

The task is defined over the following elements: Aspect Term ($A$), Aspect Category ($C$), Opinion Term ($O$), and Valence–Arousal (VA). The *Aspect Category* follows a hierarchical schema of the form `ENTITY#ATTRIBUTE`, where *ENTITY* denotes the general aspect entity (e.g., *FOOD*, *SERVICE*) and *ATTRIBUTE* specifies the evaluated property of that entity (e.g., *QUALITY*, *PRICE*), with the symbol # acting as a delimiter between the two semantic levels. The Valence–Arousal pair is represented as `V#A`, where $V, A \in [1.00, 9.00]$ and values are reported with two-decimal precision. Task format and examples are provided in App. A.

Track-A[2] includes the three previously mentioned subtasks: DimASR: given a text and one or more target aspects, predict a VA score for each aspect. DimASTE: given a text, extract all triplets $(A, O, VA)$. DimASQP: given a text, extract all quadruplets $(A, C, O, VA)$. All textual outputs are case-sensitive.

### 3.2 Dataset Description

The DimABSA benchmark comprises six languages—*Chinese (ZHO), English (ENG), Japanese (JPN), Russian (RUS), Tatar (TAT), and Ukrainian (UKR)*—and spans four application domains: *Restaurant, Laptop, Hotel, and Finance*. For each available language–domain configuration, the dataset is organized into dedicated *training*, *validation*, and *test* partitions. Individual instances correspond to domain-specific textual reviews annotated with aspect-level dimensional sentiment information. Notably, the Finance domain is available exclusively for the DimASR subtask.

Certain training partitions—namely the English Restaurant and Laptop datasets, the Chinese Restaurant dataset, and the Japanese Hotel dataset—contain *NULL* annotations corresponding to implicitly expressed sentiment targets, resulting in increased difficulty during model training. Furthermore, we observe distributional variability across dataset partitions, including differences in review length, sentiment structure density (i.e., quadruplets per review), and category frequency. Detailed dataset statistics and exploratory data analysis are provided in App. B.

[2]Official Task repository: DimABSA/DimABSA2026

### 3.3 Evaluation Metrics

**DimASR** is evaluated using Root Mean Squared Error (RMSE) between the predicted and gold valence-arousal (VA) scores, computed over both dimensions.

**DimASTE and DimASQP** are evaluated using continuous F1 (cF1), which unifies the exact-match evaluation of categorical elements with the accuracy of VA prediction. A prediction contributes only if its categorical elements exactly match a gold annotation in the same sentence (i.e., $(A, O)$ for triplets and $(A, C, O)$ for quadruplets).

## 4 System Overview

### 4.1 Aspect-Based Regression for DimASR

For DimASR, we fine-tune pretrained transformer encoders for aspect-conditioned regression. Given an input sentence $x$ and a target aspect $a$, we concatenate them into a single sequence (e.g., `Aspect:` $a$`. Sentence:` $x$`.`) and feed it to the encoder backbone. We obtain a pooled representation and predict valence and arousal using two scalar regression heads. To train the models, we optimize a weighted combination of Mean Squared Error (MSE) and Concordance Correlation Coefficient (CCC) (Lin, 1989), and we add a VA-guided triplet regularizer using the standard hinge triplet objective (Schroff et al., 2015). Since VA labels form a continuous affective space, the triplet term encourages aspect-conditioned representations with similar gold VA scores to be closer than those with distant VA scores, providing relational supervision complementary to MSE and CCC losses. We use one backbone per language and fine-tune separate models for each domain within that language. All datasets are flattened to contain entries in the form of Review/Aspect. At inference, each review's VA scores are grouped back. A more thorough explanation is provided in App. D.

**Language backbones.** We use language-appropriate transformer encoders from the BERT, RoBERTa, and DeBERTa families (Devlin et al., 2019; Liu et al., 2019; He et al., 2020) and employ XLM-R (Conneau et al., 2020) for languages without a dedicated backbone. Specifically, we use DeBERTa as the backbone for English and Japanese, RoBERTa for Chinese, a BERT-style model for Russian, and XLM-R for Ukrainian and Tatar. Details about the selected backbones are provided in App. D.

### 4.2 Instruction-Tuned LLM Generation for DimASTE and DimASQP

In order to extract the sentiment elements from the review, we approach DimASTE and DimASQP as constrained text generation. Given an input review, the model generates a JSON-formatted list of sentiment structures. This enables joint prediction of all required elements in a single decoding pass and avoids multi-stage extraction pipelines.

**Unified prompting with and without categories.** Both subtasks share the same instruction-tuning pipeline and differ only in whether aspect categories are provided and predicted. For DimASQP, the prompt includes the domain-specific list of valid categories, whereas for DimASTE the category list is omitted. For partitions containing *NULL* labels, we design language-specific instructions that discourage the model from predicting *NULL* as an easy solution when the aspect and/or opinion is implicit in the review. All instructions are written in the same language as the input data. The exact templates used for training and inference are reported in App. F.

**Structured outputs and VA handling.** To reduce formatting errors and simplify post-processing, we split VA into explicit numeric keys (Valence, Arousal) in the generated JSON during training. Gold labels are converted from the original $V\#A$ strings into floats and serialized with two-decimal formatting. At inference time, we parse the generated JSON, constrain VA values to the valid $[1.00, 9.00]$ range, and then map predictions back to the required submission format.

**Backbones and parameter-efficient fine-tuning.** We instruction fine-tune open LLMs from the Llama (Grattafiori et al., 2024) and Qwen (Qwen et al., 2025) families using LoRA (Hu et al., 2021), training separate adapters for each language/domain dataset while keeping the task formulation fixed. We use the instruct versions of Llama 3.1 8B for English, Qwen 2.5 7B for Chinese, and Qwen 2.5 14B for Japanese, Russian, Ukrainian, and Tatar. The models were chosen based on their support in each respective language. For the lowest resource languages (JPN, RUS, UKR, TAT), a larger model of 14B parameters was chosen to bridge the gap in multilingual support of the LLM.

## 5 Results

**Experimental setup.** All experiments were conducted using NVIDIA A100 GPU. Models were trained on the training set, with results reported on validation and test. Hyperparameters appear in App. G.

**Baselines.** For all subtasks, two baseline models were provided: Kimi K2 Thinking (32B) and Qwen 3 14B. The dataset paper (Lee et al., 2026) reports Zero-shot and One-shot results with GPT-5 mini and Kimi K2 Thinking, as well as supervised fine-tuning of Qwen 3 14B, Mistral 3 14B, Llama 3.3 70B, and GPT-OSS 120B, which we include for comparison. Test set results are given in App. C.

### 5.1 DimASR Results

Tables 1 and 2 present the results for the DimASR subtask in Dev and Test sets. We observe that language-appropriate encoder backbones outperform the provided baselines, except on the Tatar Restaurant dataset, where Kimi K2 Thinking achieves superior performance. Our models outperformed larger LLM-based approaches discussed in the benchmark paper in all English and Chinese domains, along with Japanese Finance, despite having fewer parameters. In Japanese Hotel and Russian/Ukrainian Restaurant, our approach is only outperformed by GPT-OSS, while in Tatar Restaurant, it is outperformed by both GPT-OSS and Kimi K2 (One-shot setting). (Table 7).

Performance drops are most pronounced in lower-resource languages (notably Tatar), which we attribute to smaller training sets and weaker language-specific pretraining. We report the Pearson Correlation Coefficient (PCC) for deeper analysis. PCC is consistently higher for valence across languages and domains, suggesting that arousal is less directly lexicalized and harder to infer from text. The performance difference between Dev and Test sets can be attributed to split size variations, as shown in App B.

| **Lang.** | **Domain** | **RMSE**$_{VA}$ ↓ | **PCC**$_V$ ↑ | **PCC**$_A$ ↑ |
|---|---|---|---|---|
| ENG | Laptop | 1.0166 | 0.9174 | 0.5768 |
| | Restaurant | 0.9208 | 0.9295 | 0.6882 |
| ZHO | Finance | 0.5438 | 0.8241 | 0.5981 |
| | Laptop | 0.7443 | 0.8863 | 0.7042 |
| | Restaurant | 0.8007 | 0.8419 | 0.6748 |
| RUS | Restaurant | 1.3808 | 0.8810 | 0.5130 |
| TAT | Restaurant | 1.9234 | 0.5231 | 0.3859 |
| UKR | Restaurant | 1.3692 | 0.9435 | 0.5622 |
| JPN | Finance | 0.9881 | 0.8088 | 0.4398 |
| | Hotel | 0.9844 | 0.9076 | 0.5825 |

Table 1: DimASR Dev set results per language and domain.

| **Lang.** | **Domain** | **RMSE**$_{VA}$ ↓ | **PCC**$_V$ ↑ | **PCC**$_A$ ↑ |
|---|---|---|---|---|
| ENG | Laptop | 1.4401 | 0.8520 | 0.5073 |
| | Restaurant | 1.3933 | 0.8757 | 0.6105 |
| ZHO | Finance | 0.5425 | 0.8327 | 0.6436 |
| | Laptop | 0.7457 | 0.8738 | 0.7133 |
| | Restaurant | 1.0023 | 0.8258 | 0.5852 |
| RUS | Restaurant | 1.7236 | 0.7865 | 0.5075 |
| TAT | Restaurant | 2.1144 | 0.5831 | 0.3104 |
| UKR | Restaurant | 1.6724 | 0.8189 | 0.5358 |
| JPN | Finance | 0.9635 | 0.7920 | 0.3497 |
| | Hotel | 0.7484 | 0.9105 | 0.7074 |

Table 2: DimASR Test set results per language and domain.

### 5.2 DimASTE Results

Table 3 presents the results for the DimASTE subtask. All models achieve competitive performance on both sets, outperforming the competition baselines. Compared to the results reported in the benchmark paper (which includes much larger fine-tuned LLMs), our $\leq$14B models display comparable and, in most cases, better results on cF1, except in Tatar Restaurant (outperformed by Llama 3.3 70B) and Japanese Hotel (outperformed by GPT-OSS), as shown in Table 7. The instruct variants of our models seem to help by improving instruction-following and structured output adherence. In addition to fine-tuning, we also report the performance of Qwen 2.5 14B on the Dev set in a Zero-shot and Few-shot setting. Few-shot worked better than Zero-shot, but the overall performance was underwhelming. The results and insights of this study can be found in App H.

The Dev–Test cF1 gaps correspond to variations in review length and structure density (tuples per review), as shown by the Population Stability Index (PSI) heatmap and split statistics in App. B. The effect is significant for Chinese Restaurant, where length and tuples per review vary considerably, and the Test split is denser, increasing the likelihood of missed structures and reducing cF1. Similar

| Lang. | Model | Domain | cF1 ↑ | |
|---|---|---|---|---|
| | | | Dev | Test |
| ENG | Llama 3.1 8B | Laptop | 0.5962 | 0.5311 |
| | | Restaurant | 0.7668 | 0.6518 |
| ZHO | Qwen 2.5 7B | Laptop | 0.4193 | 0.4646 |
| | | Restaurant | 0.5916 | 0.5042 |
| JPN | Qwen 2.5 14B | Hotel | 0.4879 | 0.5021 |
| RUS | Qwen 2.5 14B | Restaurant | 0.4766 | 0.4988 |
| UKR | Qwen 2.5 14B | Restaurant | 0.4958 | 0.4725 |
| TAT | Qwen 2.5 14B | Restaurant | 0.4272 | 0.3874 |

Table 3: DimASTE results per language and domain.

patterns appear for English Restaurant and Laptop, where Test reviews are slightly longer or denser than Dev. In lower-resource settings (e.g., Tatar Restaurant), smaller split sizes amplify variance, so moderate length or density differences yield visible Dev–Test fluctuations.

Additionally, we observe a substantial performance gap between English Restaurant and Laptop, despite English being a high-resource language. This can be attributed to the number of NULL entries in the Laptop Train set, which cover nearly half of it (App. B). This introduces a skewed supervision signal that may bias the model toward predicting NULL, even when aspects and/or opinions are explicit.

### 5.3 DimASQP Results

Consistent with the previous subtask, our models show the same behavior when moving to a more complex task. The drop in performance is attributed to the introduction of categories in the extracted fields. We achieve competitive performance, outperforming the provided baselines except in English Laptop (outperformed by Kimi K2 Thinking, as well as GPT-5 mini) and showcasing comparable results against much larger LLMs proposed in the benchmark paper, surpassing their cF1 in most language/domain settings except in Tatar Restaurant (outperformed by Llama 3.3 70B) and Japanese Hotel (outperformed by GPT-OSS) following DimASTE, as shown in Table 7. Zero-shot and Few-shot were also tested using Qwen 2.5 14B on the Dev set, yielding results similar to those of the previous subtask (App H).

For DimASQP, we additionally fine-tuned larger LLMs than those proposed, namely Llama 3.1 70B, Qwen 2.5 32B, and Qwen 2.5 72B, using the same strategy. They displayed competitive performance on the Dev set across the higher-resource languages and fell off in the lower ones. Our proposed efficient models outperformed them consistently. More on this study can be found in App H.

| Lang. | Model | Domain | cF1 ↑ | |
|---|---|---|---|---|
| | | | Dev | Test |
| ENG | Llama 3.1 8B | Laptop | 0.3556 | 0.2694 |
| | | Restaurant | 0.7560 | 0.5988 |
| ZHO | Qwen 2.5 7B | Laptop | 0.3662 | 0.3703 |
| | | Restaurant | 0.5665 | 0.4544 |
| JPN | Qwen 2.5 14B | Hotel | 0.4218 | 0.3747 |
| RUS | Qwen 2.5 14B | Restaurant | 0.4774 | 0.4369 |
| UKR | Qwen 2.5 14B | Restaurant | 0.4941 | 0.4154 |
| TAT | Qwen 2.5 14B | Restaurant | 0.4033 | 0.3306 |

Table 4: DimASQP results per language and domain.

The gap between the Restaurant and Laptop domains becomes larger with the addition of categories. In English and Chinese, models perform much better in the restaurant domain due to higher category/label diversity in the laptop domain compared to other domains. We observe the shift in categories in English Laptop from Train to Dev and Test sets in the PSI heatmap in App. B, which explains the performance drop in this setting. Notably, the Chinese Laptop model performed better without category mentions in the prompt.

### 5.4 DimASQP Results via Translation

Motivated by the performance of our model in the English Restaurant dataset, we translated the Train and Dev sets into English using Qwen 2.5 14B, fine-tuned Llama 3.1 8B on the translated training data, and evaluated it on the translated Dev set. We then mapped the predicted English structures back to the original language using Qwen 2.5 14B. Additionally, we evaluated an English-trained model (trained on English Restaurant) on the translated Dev sets and applied the same back-mapping.

Compared to training on original-language data, translation-based training leads to an overall performance drop, suggesting that translation introduces noise (idiom shift, span drift, or label misalignment). The English-trained model performed worse only on Chinese Restaurant, consistent with the Chinese–English gap. The Chinese translation gains suggest weaker direct transfer from an English-trained model, so translation-based adapta-

| Lang. | Translate+Train | ENG-Model |
|---|---|---|
| ZHO | 0.36 | 0.26 |
| RUS | 0.31 | 0.31 |
| UKR | 0.24 | 0.25 |
| TAT | 0.20 | 0.22 |

Table 5: Restaurant DimASQP (cF1), Dev set, Llama 3.1 8B: translation training vs. English-trained model.

tion can partially compensate despite added noise. In other languages, the English-trained model mostly achieved better performance (except Russian where performance is tied), reflecting the difficulty of translating while preserving language-specific idioms. In practice, back-mapping often yields paraphrases rather than original spans, increasing mismatch under exact-match evaluation.

## 6 Conclusion

In our work, we address DimABSA, introduced in SemEval-2026 Task 3. We propose a parameter-efficient methodology that utilizes lightweight transformer encoders for DimASR and LLM backbones with up to 14B parameters for DimASTE and DimASQP. Across languages and domains, our system achieves competitive results on the official evaluation, showing that efficient, smaller-scale models can perform well for DimABSA. In total, we aspire for our methodology to assist future research in the DimABSA setting.

## Acknowledgements

This work was carried out within the framework of the Pharos AI Factory project, funded by the European High-Performance Computing Joint Undertaking (EuroHPC JU) under Grant Agreement No. 101234269 as part of the Horizon Europe and by the Greek Public Investments Program.

## Limitations

Our system has several limitations. First, we train separate models (or LoRA adapters) for each language/domain dataset, which improves specialization but increases the number of checkpoints to manage and does not directly exploit cross-lingual or cross-domain transfer. Second, structured prediction for DimASTE/DimASQP is evaluated under an exact-match criterion for categorical elements; despite prompt design and greedy decoding, generation can still produce format errors and paraphrased spans, which are heavily penalized by cF1. Third, performance is less stable in low-resource settings where the development sets are small and distribution shifts between splits can be substantial, leading to higher variance and weaker generalization. Finally, our experiments were limited by compute resources (single GPU), restricting broader hyperparameter sweeps and more extensive exploration of larger backbones.

## References

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.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8440–8451. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. DeBERTa: Decoding-enhanced BERT with disentangled attention. *arXiv preprint arXiv:2006.03654*.

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

Yunfan Jiang, Tianci Liu, and Hengyang Lu. 2024. JN-NLP at SIGHAN-2024 dimABSA task: Extraction of sentiment intensity quadruples based on paraphrase generation. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 121–126, Bangkok, Thailand. Association for Computational Linguistics.

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. Mohammed. 2026. Dimabsa: Building multilingual and multidomain datasets for dimensional aspect-based sentiment analysis. *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 (SIGHAN-10)*, pages 165–174, Bangkok, Thailand. Association for Computational Linguistics.

Lawrence I.-Kuei Lin. 1989. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, 45(1):255–268.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

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 (SIGHAN-10)*, pages 127–132, Bangkok, Thailand. 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.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

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

Zeliang Tong and Wei Wei. 2024. CCIIPLab at SIGHAN-2024 dimABSA task: Contrastive learning-enhanced span-based framework for Chinese dimensional aspect-based sentiment analysis. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 102–111, Bangkok, Thailand. Association for Computational Linguistics.

Qianlong Wang, Hongling Xu, Keyang Ding, Bin Liang, and Ruifeng Xu. 2024. In-context example retrieval from multi-perspectives for few-shot aspect-based sentiment analysis. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8975–8985, Torino, Italia. ELRA and ICCL.

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

Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020. Grid tagging scheme for aspect-oriented fine-grained opinion extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2576–2585, Online. Association for Computational Linguistics.

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 (SIGHAN-10)*, pages 175–185, Bangkok, Thailand. Association for Computational Linguistics.

Lu Xu, Yew Ken Chia, and Lidong Bing. 2021. Learning span-level interactions for aspect sentiment triplet extraction. 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 4755–4766, Online. Association for Computational Linguistics.

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)*, pages 2339–2349, Online. 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.

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. 2021a. Aspect sentiment quad prediction as paraphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9209–9219, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021b. Towards generative 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 2: Short Papers)*, pages 504–510, Online. Association for Computational Linguistics.

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

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. *Preprint*, arXiv:2412.02279.

Junxian Zhou, Haiqin Yang, Yuxuan He, Hao Mou, and Junbo Yang. 2023. A unified one-step solution for aspect sentiment quad prediction. *Preprint*, arXiv:2306.04152.

## A Data Format

The data format for each subtask is provided in Table 6.

## B Exploratory Data Analysis

**Split Statistics.** Regarding DimASR, the VA distributions for all the datasets are provided in the dataset paper (Lee et al., 2026). Figure 1 summarizes the size of each split, along with the aspects per review for DimASR. We mainly focus our EDA on the other subtasks where the data have more dimensions. The plots presented all contain information about DimASQP, but the datasets are essentially the same as in DimASTE, aside from the keyword change (Triplet/Quadruplet) and the existence of categories. We provide the number of reviews per split and tuples per review in each split in Figure 2. We can see the substantial difference in size between Dev and Test, which can contribute to performance differences.

**NULL Aspect/Opinion instances.** Following that, we examine the NULL labels in each Train set, as they can bias the model during training (Figure 3). English datasets contain a large number of NULL labels, with more than half of the NULL instances representing missing opinions. Even though the actual target was missing, sentiment was expressed but not directed at a specific aspect, making these examples valuable as well. The main problem with those NULL entries is the bias they can create for the model to easily output NULL when the aspect/opinion is hard to find in the text. The prompt design was oriented to diminish this bias.

**Splits' distributions shift.** Lastly, we present a PSI heatmap to identify the differences between each split's distributions (PSI < 0.1: No significant shift, 0.1 - 0.2: Moderate shift, > 0.2: Significant shift). We compare the review lengths, the number of quadruplets per review, and the category distributions between splits for each language and domain combination. This heatmap is useful for explaining specific behaviors in our results.

## C Benchmark Models

**Competition baselines.** We provide the Hugging Face identifiers for the competition baseline models, which are Kimi K2 Thinking (32B) [3] and Qwen 3 14B [4].

**Benchmark paper.** Additionally, we provide the model identifiers for the systems used in the benchmark paper, including Llama 3.3 70B[5], Mistral 3 14B[6], and GPT-OSS 120B[7]. We also report results for GPT-5 mini, accessed via the OpenAI API.

**Benchmark Paper Results** Table 7 summarizes the results of the Benchmark paper (including the baseline models).

[3] moonshotai/Kimi-K2-Thinking
[4] Qwen/Qwen3-14B
[5] meta-llama/Llama-3.3-70B-Instruct
[6] mistralai/Ministral-3-14B-Base-2512
[7] openai/gpt-oss-120b

| Subtask | Example (JSON) |
|---|---|
| **DimASR** | `{"ID": "rest26_aspect_va_dev_1", "Text": "Great diner food and breakfast is served all day", "Aspect_VA": [{"Aspect": "diner food", "VA": "7.25#6.75"}, {"Aspect": "breakfast", "VA": "7.25#6.75"}]}` |
| **DimASTE** | `{"ID": "rest26_aste_dev_2", "Text": "Customer service was fantastic and food was awesome", "Triplet": [{"Aspect": "Customer service", "Opinion": "fantastic", "VA": "7.33#7.33"}, {"Aspect": "food", "Opinion": "awesome", "VA": "7.67#7.67"}]}` |
| **DimASQP** | `{"ID": "rest26_asqp_dev_1", "Text": "Food and coffee are great", "Quadruplet": [{"Aspect": "Food", "Category": "FOOD#QUALITY", "Opinion": "great", "VA": "7.67#7.83"}, {"Aspect": "coffee", "Category": "DRINKS#QUALITY", "Opinion": "great", "VA": "7.67#8.00"}]}` |

Table 6: Example JSON format for each Subtask.

| Subtask | Lang. | Domain | Zero-shot | | One-shot | | Supervised Fine-Tuning | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | GPT-5 mini | Kimi K2 Thinking | GPT-5 mini | Kimi K2 Thinking | Qwen 3 14B | Mistral 3 14B | Llama 3.3 70B | GPT-OSS 120B |
| DimASR | ENG | Restaurant | 2.9490 | 2.3432 | 2.3926 | 2.1461 | 2.6427 | 2.6316 | 2.5244 | 1.4605 |
| | ENG | Laptop | 3.2115 | 2.6546 | 2.5637 | 2.1893 | 2.8089 | 2.6258 | 2.7354 | 1.5269 |
| | JPN | Hotel | 3.1406 | 2.3294 | 2.1607 | 1.7553 | 2.2906 | 2.2999 | 2.6255 | 0.7188 |
| | JPN | Finance | 2.6760 | 2.3379 | 1.9243 | 1.6396 | 1.8964 | 2.0700 | 2.4191 | 1.0188 |
| | RUS | Restaurant | 2.5447 | 2.0630 | 2.0390 | 1.7768 | 2.1528 | 2.3617 | 2.5089 | 1.4775 |
| | TAT | Restaurant | 2.6645 | 2.3636 | 2.2308 | 1.9380 | 2.6367 | 3.0463 | 2.9165 | 1.7153 |
| | UKR | Restaurant | 2.5628 | 2.0782 | 2.0438 | 1.7805 | 2.2121 | 2.4592 | 2.5709 | 1.5166 |
| | ZHO | Restaurant | 2.7125 | 2.2623 | 2.2467 | 1.8959 | 2.0073 | 1.9373 | 2.4463 | 1.0349 |
| | ZHO | Laptop | 2.4790 | 2.0426 | 1.9380 | 1.6440 | 1.7706 | 1.8267 | 2.3633 | 0.8032 |
| | ZHO | Finance | 2.6547 | 2.9662 | 2.0094 | 1.9652 | 1.4707 | 1.8900 | 2.5632 | 0.6511 |
| DimASTE | ENG | Restaurant | 0.4993 | 0.5101 | 0.5034 | 0.4920 | 0.4483 | 0.2930 | 0.5418 | 0.5442 |
| | ENG | Laptop | 0.4491 | 0.4519 | 0.4874 | 0.4424 | 0.3827 | 0.2736 | 0.4664 | 0.4515 |
| | JPN | Hotel | 0.1727 | 0.3148 | 0.2487 | 0.3464 | 0.1622 | 0.1458 | 0.4694 | 0.5397 |
| | RUS | Restaurant | 0.4016 | 0.4031 | 0.3730 | 0.4242 | 0.3341 | 0.1774 | 0.4590 | 0.4262 |
| | TAT | Restaurant | 0.3419 | 0.3503 | 0.3159 | 0.3577 | 0.2020 | 0.1154 | 0.4101 | 0.3578 |
| | UKR | Restaurant | 0.4014 | 0.4082 | 0.3326 | 0.4220 | 0.3099 | 0.1595 | 0.4517 | 0.4250 |
| | ZHO | Restaurant | 0.3190 | 0.3728 | 0.2425 | 0.3529 | 0.2509 | 0.1446 | 0.4789 | 0.4759 |
| | ZHO | Laptop | 0.2372 | 0.2227 | 0.2815 | 0.2494 | 0.2099 | 0.1182 | 0.4344 | 0.4366 |
| DimASQP | ENG | Restaurant | 0.4036 | 0.3740 | 0.3816 | 0.3746 | 0.2673 | 0.2058 | 0.5048 | 0.5013 |
| | ENG | Laptop | 0.2304 | 0.2805 | 0.2842 | 0.2795 | 0.1529 | 0.1293 | 0.2483 | 0.2411 |
| | JPN | Hotel | 0.0907 | 0.1309 | 0.1542 | 0.1943 | 0.0400 | 0.0311 | 0.3577 | 0.4151 |
| | RUS | Restaurant | 0.2508 | 0.2656 | 0.2505 | 0.2963 | 0.1682 | 0.1000 | 0.4118 | 0.3683 |
| | TAT | Restaurant | 0.1974 | 0.2310 | 0.2025 | 0.2380 | 0.0954 | 0.0611 | 0.3702 | 0.3094 |
| | UKR | Restaurant | 0.2465 | 0.2974 | 0.2180 | 0.2971 | 0.1641 | 0.0975 | 0.4070 | 0.3663 |
| | ZHO | Restaurant | 0.2481 | 0.2975 | 0.1891 | 0.2859 | 0.1605 | 0.0934 | 0.4391 | 0.4249 |
| | ZHO | Laptop | 0.1356 | 0.1569 | 0.1921 | 0.1900 | 0.1124 | 0.0728 | 0.3506 | 0.3551 |

Table 7: Combined Benchmark paper's results across subtasks. DimASR values are $\mathrm{RMSE}_{VA}$; DimASTE and DimASQP values are cF1.

## D DimASR Models

**Transformer backbones.** Table 8 summarizes the encoder backbones used for each language, along with their corresponding Hugging Face model identifiers.

| Language | Model family | Model ID |
|---|---|---|
| English | DeBERTa | `yangheng/deberta-v3-base-absa-v1.1`[8] |
| Japanese | DeBERTa | `ku-nlp/deberta-v3-base-japanese`[9] |
| Chinese | RoBERTa | `hfl/chinese-roberta-wwm-ext`[10] |
| Russian | BERT-style | `DeepPavlov/rubert-base-cased`[11] |
| Ukrainian | XLM-R | `FacebookAI/xlm-roberta-base`[12] |
| Tatar | XLM-R | `FacebookAI/xlm-roberta-base`[13] |

Table 8: Language-specific encoder backbones used for DimASR.

**Regression head.** Given an aspect-conditioned input, the encoder produces token representations $\{\mathbf{h}_t\}_{t=1}^{T}$. We compute an attention-pooled representation $\mathbf{z}$ as

$$s_t = \mathbf{w}^\top \mathbf{h}_t, \quad \alpha_t = \frac{\exp(s_t)}{\sum_{j=1}^{T} \exp(s_j)}, \quad \mathbf{z} = \sum_{t=1}^{T} \alpha_t \mathbf{h}_t, \tag{1}$$

with padding tokens masked before normalization. Valence and arousal are predicted using two linear heads:

$$\tilde{v} = \mathbf{W}_v \mathbf{z} + b_v, \qquad \tilde{a} = \mathbf{W}_a \mathbf{z} + b_a. \tag{2}$$

[8] yangheng/deberta-v3-base-absa-v1.1
[9] ku-nlp/deberta-v3-base-japanese
[10] hfl/chinese-roberta-wwm-ext
[11] DeepPavlov/rubert-base-cased
[12] FacebookAI/xlm-roberta-base
[13] FacebookAI/xlm-roberta-base

**Training objective.** Let $\mathbf{y}_i = (v_i, a_i)$ and $\tilde{\mathbf{y}}_i = (\tilde{v}_i, \tilde{a}_i)$ denote the gold and predicted VA pairs for instance $i$ in a minibatch of size $B$. Labels are normalized to $[0, 1]$ during training and rescaled/clipped at inference. The MSE term is

$$\mathcal{L}_{\text{MSE}} = \frac{1}{B} \sum_{i=1}^{B} \|\tilde{\mathbf{y}}_i - \mathbf{y}_i\|_2^2 . \tag{3}$$

For each dimension $k \in \{v, a\}$, CCC is computed over the minibatch as

$$\text{CCC}_k = \frac{2\sigma_{\tilde{y}^{(k)} y^{(k)}}}{\sigma^2_{\tilde{y}^{(k)}} + \sigma^2_{y^{(k)}} + \left(\mu_{\tilde{y}^{(k)}} - \mu_{y^{(k)}}\right)^2}, \tag{4}$$

and aggregated as $\text{CCC} = \lambda_v \text{CCC}_v + \lambda_a \text{CCC}_a$.

We additionally sample triplets $(i, p, n)$ according to gold VA distances, with positives close to the anchor and negatives farther away. The triplet loss is computed on pooled representations $\mathbf{z}$:

$$\ell_{i,p,n} = \max\Big(0,\ \|\mathbf{z}_i - \mathbf{z}_p\|_2 - \|\mathbf{z}_i - \mathbf{z}_n\|_2 + m\Big), \tag{5}$$

$$\mathcal{L}_{\text{tri}} = \frac{1}{|\mathcal{T}|} \sum_{(i,p,n)\in\mathcal{T}} \ell_{i,p,n}. \tag{6}$$

The final loss is

$$\mathcal{L} = (1-\beta)\mathcal{L}_{\text{base}} + \beta\mathcal{L}_{\text{tri}}, \tag{7}$$

$$\mathcal{L}_{\text{base}} = \gamma\mathcal{L}_{\text{MSE}} + (1-\gamma)(1-\text{CCC}), \tag{8}$$

where $\gamma$ balances MSE and CCC, and $\beta$ controls the contribution of the VA-guided triplet regularizer.

# E DimASTE and DimASQP Models

We use Llama 3.1 8B Instruct [14] for English, Qwen 2.5 7B Instruct [15] for Chinese, and Qwen 2.5 14B Instruct [16] for Japanese, Russian, Ukrainian, and Tatar.

# F Language Specific Prompts

We construct prompts using the native chat template of each backbone to improve instruction-following and JSON adherence. The full training prompts for DimASQP can be found in Table 14. The DimASTE training prompts are the same, without the category mentions. For inference, we use exactly the same prompt without the answer.

[14] meta-llama/Llama-3.1-8B-Instruct
[15] Qwen/Qwen2.5-7B-Instruct
[16] Qwen/Qwen2.5-14B-Instruct

# G Experimental Settings

## G.1 DimASR Settings.

In all experiments, we set $\gamma = 0.3$ in $\mathcal{L}_{\text{base}}$, and weight the CCC aggregation as $\lambda_v = 0.3$ and $\lambda_a = 0.7$. For the final objective, we set $\beta = 0.05$, yielding a $0.95/0.05$ weighting between $\mathcal{L}_{\text{base}}$ and $\mathcal{L}_{\text{tri}}$. The triplet term is computed on the pooled encoder representations $\mathbf{z}$, allowing gradients from $\mathcal{L}_{\text{tri}}$ to update the encoder.

The hyperparameters used for training can be found in Table 9. We use a two-stage learning-rate schedule: a step-wise linear warmup for the first 10% of training steps (updated every mini-batch), followed by an epoch-wise ReduceLROnPlateau schedule applied after each validation epoch, monitoring $\text{RMSE}_{\text{VA}}$ (factor 0.5, patience 2).

| **Hyperparameter** | **Value** |
|---|---|
| Learning rate ($\eta$) | $2e-5$ |
| Batch size | 16 |
| Dropout | 0.3 |
| Warmup ratio | 0.1 |
| Early stopping patience | 5 |
| Max sequence length | 128 |
| Optimizer | AdamW |
| Scheduler | Warmup + ReduceLROnPlateau |
| Epochs | 30 |

Table 9: Training hyperparameters for the DimASR models.

## G.2 DimASTE & DimASQP Settings

We apply LoRA adapters to the main attention and feed-forward projection layers, keeping the number of trainable parameters small while retaining a strong adaptation capacity. For efficiency, we load base models in 4-bit precision and train only the LoRA adapters. The hyperparameters used for instruction fine-tuning our LLMs can be found in Table 10.

# H Additional Ablation Studies

## H.1 Zero-shot & Few-shot

We also tested our methodology in a zero-shot/few-shot scenario for DimASTE and DimASQP using Qwen 2.5 14B Instruct on the Dev set (Tables 11 & 12). For few-shot prompting, we randomly sample three demonstrations from the training set of the corresponding language-domain pair and prepend them under a single instruction, following the same output schema as the task. We include each demonstration as an input-output pair (review plus its gold JSON) and apply the same post-processing as in our

| Hyperparameter | Value |
|---|---|
| Train epochs ($E$) | 1 |
| Per-device batch size | 2 |
| Gradient accumulation steps | 4 |
| Effective batch (per device) | $2 \times 4 = 8$ |
| Learning rate ($\eta$) | $2e-4$ |
| Weight decay | $1 \times 10^{-4}$ |
| Warmup ratio | 0.03 |
| LR scheduler | linear |
| Optimizer | paged_adamw_32bit |
| Max sequence length | 2048 |
| Quantization | 4-bit |
| Precision | bf16 |
| Max grad norm | 0.3 |
| LoRA rank ($r$) | 16 |
| LoRA $\alpha$ | 32 |
| LoRA dropout | 0.2 |
| LoRA target modules | `[q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj]` |

Table 10: LLM fine-tuning configuration (4-bit loading + LoRA adapters) for DimASTE & DimASQP.

main pipeline before scoring. We decode greedily to improve format stability and JSON adherence.

Few-shot performed much better than zero-shot across all languages and domains, but it still couldn't reach the performance of fine-tuning. These techniques yield better results in high-resource languages such as English and Chinese. Notably, for English Laptop, few-shot prompting approaches the fine-tuned Dev performance, suggesting that in some cases, in-context demonstrations can leverage the model's prior domain knowledge and reduce the need for full fine-tuning; however, this trend is less consistent in lower-resource languages.

| Lang. | Domain | Dev cF1 ↑ | |
|---|---|---|---|
| | | Zero-shot | Few-shot |
| ENG | Laptop | 0.48 | 0.59 |
| | Restaurant | 0.53 | 0.59 |
| ZHO | Laptop | 0.23 | 0.27 |
| | Restaurant | 0.37 | 0.43 |
| JPN | Hotel | 0.17 | 0.19 |
| RUS | Restaurant | 0.33 | 0.37 |
| UKR | Restaurant | 0.33 | 0.35 |
| TAT | Restaurant | 0.15 | 0.16 |

Table 11: DimASTE Dev set results in zero-shot and few-shot prompting (cF1) using Qwen 2.5 14B Instruct.

| Lang. | Domain | Dev cF1 ↑ | |
|---|---|---|---|
| | | Zero-shot | Few-shot |
| ENG | Laptop | 0.22 | 0.32 |
| | Restaurant | 0.42 | 0.51 |
| ZHO | Laptop | 0.14 | 0.18 |
| | Restaurant | 0.23 | 0.31 |
| JPN | Hotel | 0.06 | 0.11 |
| RUS | Restaurant | 0.25 | 0.30 |
| UKR | Restaurant | 0.24 | 0.25 |
| TAT | Restaurant | 0.07 | 0.15 |

Table 12: DimASQP Dev set results in zero-shot and few-shot prompting (cF1) using Qwen 2.5 14B Instruct.

### H.2 Fine-tuning larger LLMs

Using the same strategy and hyperparameters, we tried to utilize larger LLMs for DimASQP, such as Llama 3.1 70B Instruct [17], Qwen 2.5 32B Instruct[18], and Qwen 2.5 72B Instruct[19]. Due to resource constraints, we could not further explore hyperparameter sensitivity or conduct multiple runs.

| Lang. | Model | Domain | Dev cF1 ↑ |
|---|---|---|---|
| ENG | Llama 3.1 70B | Laptop | 0.3373 |
| | | Restaurant | 0.7441 |
| ZHO | Qwen 2.5 32B | Laptop | 0.3119 |
| | | Restaurant | 0.5220 |
| | Qwen 2.5 72B | Laptop | 0.3675 |
| | | Restaurant | 0.5508 |
| JPN | Qwen 2.5 32B | Hotel | 0.3044 |
| | Qwen 2.5 72B | | 0.3414 |
| RUS | Qwen 2.5 32B | Restaurant | 0.3714 |
| | Qwen 2.5 72B | | 0.4332 |
| UKR | Qwen 2.5 32B | Restaurant | 0.3610 |
| | Qwen 2.5 72B | | 0.4328 |
| TAT | Qwen 2.5 32B | Restaurant | 0.2812 |
| | Qwen 2.5 72B | | 0.3467 |

Table 13: DimASQP development set results (cF1) after LoRa-tuning larger LLMs.

Under the same fine-tuning recipe, larger backbones did not consistently improve cF1 over our $\leq$14B models, and in several lower-resource settings, they performed worse. This suggests that scaling benefits may require more careful hyperparameter tuning.

[17] meta-llama/Llama-3.1-70B-Instruct
[18] Qwen/Qwen2.5-32B-Instruct
[19] Qwen/Qwen2.5-72B-Instruct

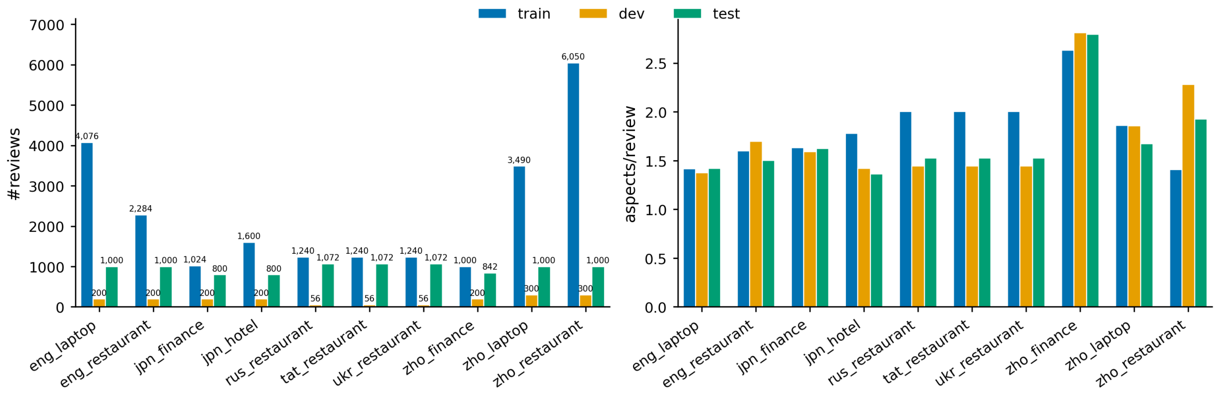


Figure 1: DimASR dataset statistics across splits: number of reviews (left) and aspects per review (right).

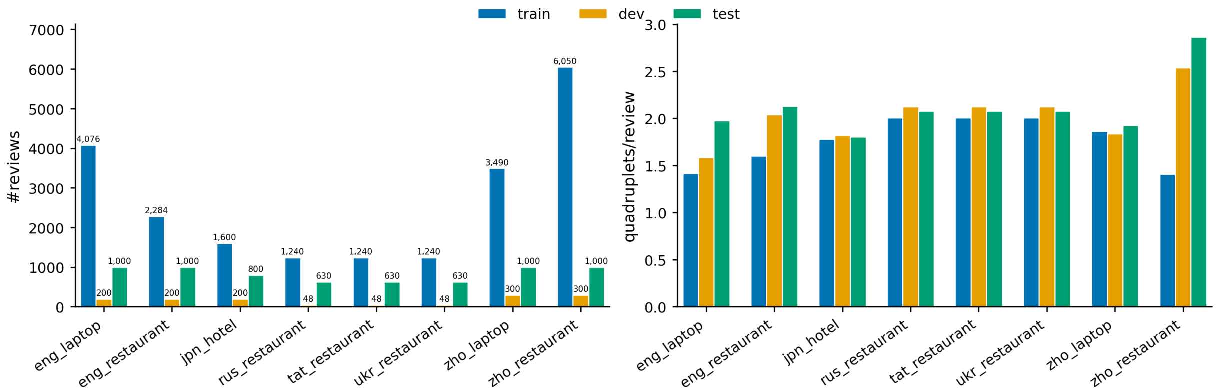


Figure 2: DimASTE and DimASQP dataset statistics across splits: number of reviews (left) and quadruplets per review (right).

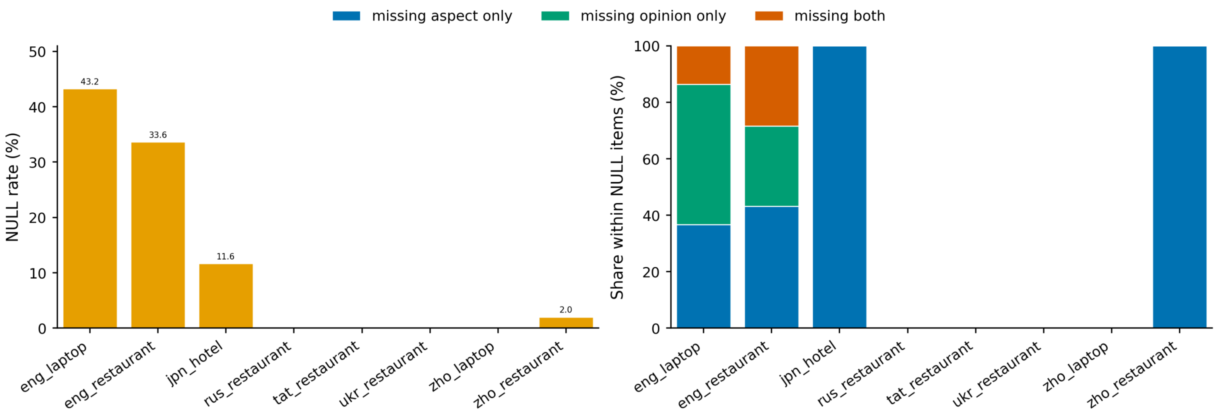


Figure 3: DimASTE and DimASQP NULL analysis on Train set: NULL rate (left) and composition of NULL cases (right).

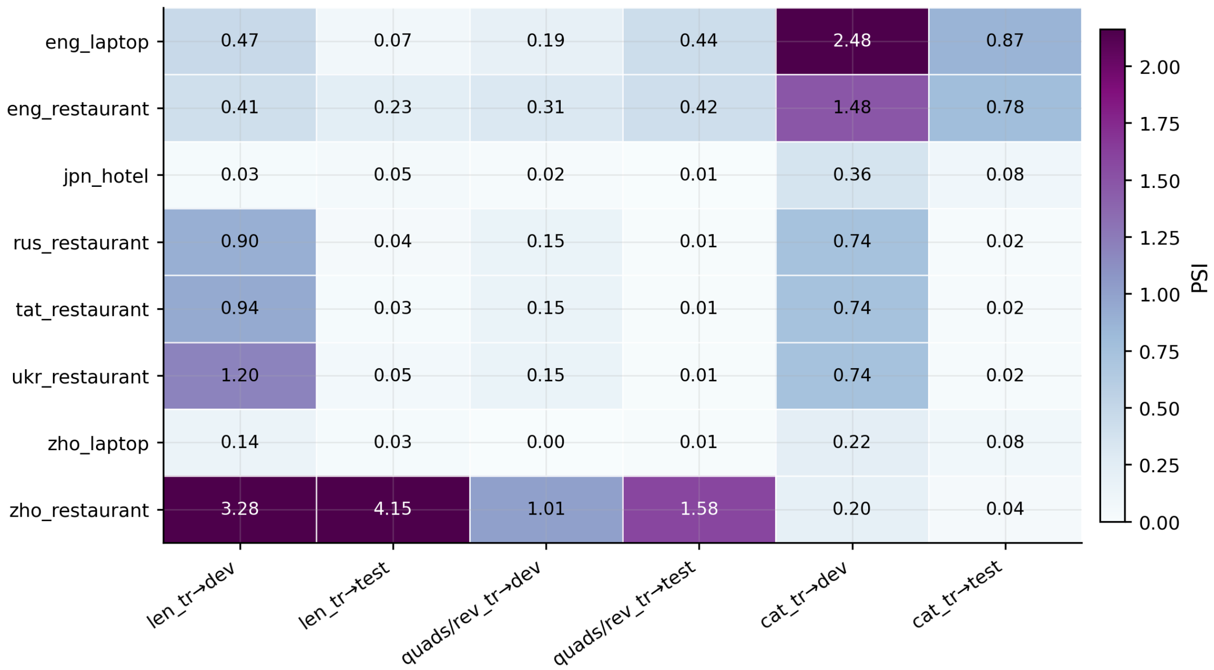


Figure 4: DimASTE and DimASQP distribution shift measured by PSI between Train and Dev/Test across features (review length, quadruplets/review, and category).

## Prompt Templates Used for Training (DimASQP)

<table>
<tr><th>Lang_Domain</th><th>Instruction</th><th>Training chat template (user + assistant)</th></tr>
<tr><td>eng_restaurant</td><td>You are an expert sentiment analysis tool. Your task is to extract all<br>↪ quadruplets from the given review.<br>Here are the rules:<br>1. You MUST format your answer as a single, valid JSON list of objects.<br>2. Each object MUST have the keys: 'Aspect', 'Category', 'Opinion',<br>↪ 'Valence', and 'Arousal'.<br>3. 'Aspect' and 'Opinion' MUST be exact substrings from the input text.<br>- Copy the text EXACTLY as it appears, including spacing,<br>↪ punctuation errors, or typos.<br>- Do NOT correct or normalize the text.<br>3b. Always try your best to find explicit 'Aspect' and 'Opinion'<br>↪ mentions.<br>- The 'Opinion' may appear before or after the 'Aspect' in the<br>↪ text; always extract the full 'Aspect' and 'Opinion',<br>↪ regardless of order in the sentence.<br>- Use 'NULL' only if a real 'Aspect' or 'Opinion' span truly does<br>↪ not exist.<br>3c. 'Opinion' MUST be complete and include all adjectives, negations,<br>↪ and modifiers.<br>4. 'Category' MUST be one of the following values: [{CATEGORIES}].<br>5. 'Valence' and 'Arousal' MUST be numbers from 1.00 to 9.00, with up<br>↪ to two decimal places.<br>6. Your entire response MUST be *only* the JSON list.<br>Now, perform the task for the following review:</td><td><|begin_of_text|><|start_header_id|>user<|end_header_id|><br><br>{instruction}<br><br>Review: "{text}"<|eot_id|><br><br><|start_header_id|>assistant<|end_header_id|><br>{answer_json}<|eot_id|></td></tr>
<tr><td>eng_laptop</td><td>You are an expert sentiment analysis tool. Your task is to extract all<br>↪ quadruplets from the given review.<br>Here are the rules:<br>1. You MUST format your answer as a single, valid JSON list of objects.<br>2. Each object MUST have the keys: 'Aspect', 'Category', 'Opinion',<br>↪ 'Valence', and 'Arousal'.<br>3. 'Aspect' and 'Opinion' MUST be exact substrings from the input text.<br>- Copy the text EXACTLY as it appears, including spacing,<br>↪ punctuation errors, or typos.<br>- Do NOT correct or normalize the text.<br>3b. Always try your best to find explicit 'Aspect' and 'Opinion'<br>↪ mentions.<br>- The 'Opinion' may appear before or after the 'Aspect' in the<br>↪ text; always extract the full 'Aspect' and 'Opinion',<br>↪ regardless of order in the sentence.<br>- Use 'NULL' only if a real 'Aspect' or 'Opinion' span truly does<br>↪ not exist.<br>3c. 'Opinion' MUST be complete and include all adjectives, negations,<br>↪ and modifiers.<br>3d. 'Aspect' MUST be complete and include descriptive adjectives.<br>4. 'Category' MUST be one of the following values: [{CATEGORIES}].<br>5. 'Valence' and 'Arousal' MUST be numbers from 1.00 to 9.00, with up<br>↪ to two decimal places.<br>6. Your entire response MUST be *only* the JSON list.<br>Now, perform the task for the following review:</td><td><|begin_of_text|><|start_header_id|>user<|end_header_id|><br><br>{instruction}<br><br>Review: "{text}"<|eot_id|><br><br><|start_header_id|>assistant<|end_header_id|><br>{answer_json}<|eot_id|></td></tr>
<tr><td>zho_restaurant</td><td>你是一個精準且嚴格的中文情感四元組抽取模型。<br>你的任務是從評論中抽取所有可能且明確存在的四元組，<br>↪ 並且不要遺漏任何明確的 Aspect-Opinion 組合，同時嚴格避免假陽性。<br><br>每個四元組包含以下欄位（欄位順序必須固定）：<br>"Aspect"（評論中出現的詞，必須完全照抄原文字串）<br>"Category"（必須與 {CATEGORIES} 中的某一項完全一致，不可創造、<br>↪ 改寫或合併新分類）<br>"Opinion"（完整且連續描述該 Aspect 的情緒片段，必須完全照抄原文字串，<br>↪ 包括所有否定、程度、副詞或連續修飾語）<br>"Valence"（情緒正負強度，1.00–9.00，小數兩位，<br>↪ 基於語義強度的主觀連續估計）<br>"Arousal"（情緒喚起強度，1.00–9.00，小數兩位，<br>↪ 基於語義強度的主觀連續估計）<br><br>請遵守以下規則：<br>1. 輸出必須是一個 JSON 陣列，不得包含任何非 JSON 文字<br>2. Aspect 與 Opinion 必須完全照抄原文，不可修改、標準化、裁切或推測<br>3. 若評論中存在多個不同的 Aspect 或 Opinion，需分別生成多個 JSON 物件；<br>↪ 若同一 Aspect 對應多個不同 Opinion，也需分別生成多個四元組<br>4. 僅當評論中存在明確情緒表達，但無法在原文中找到任何可對應的 Aspect 或<br>↪ Opinion 文字片段時，才允許填寫 "NULL"<br>5. 若評論中不存在任何明確可抽取的情感四元組，輸出空陣列 []<br>6. 嚴格避免假陽性：僅在 Aspect 與 Opinion<br>↪ 均在原文中明確存在時才生成四元組<br>7. 請檢查評論中所有可能的 Aspect-Opinion 組合，<br>↪ 務必完整覆蓋所有明確表達的情感<br>8. 僅輸出 JSON，不要附加任何說明、分析或解釋<br><br>請根據以下評論抽取情感四元組：</td><td><|im_start|>user<br>{instruction}<br><br>評論："{text}"<br><|im_end|><br><|im_start|>assistant<br>{answer_json}<br><|im_end|></td></tr>
</table>

Continued on next page.

<table>
<thead>
<tr><th>Lang/Domain</th><th>Instruction (original prompt)</th><th>Training chat template (user + assistant)</th></tr>
</thead>
<tbody>
<tr><td>zho_laptop</td><td>你是一個精準且嚴格的中文情感四元組抽取模型。<br>你的任務是從評論中抽取所有可能且明確存在的四元組，<br>↪ 並且不要遺漏任何明確的 Aspect-Opinion 組合，同時嚴格避免假陽性。<br><br>每個四元組包含以下欄位（欄位順序必須固定）：<br>"Aspect"（評論中出現的詞，必須完全照抄原文字串）<br>"Category"<br>"Opinion"（完整且連續描述該 Aspect 的情緒片段，必須完全照抄原文字串，<br>↪ 包括所有否定、程度、副詞或連續修飾語）<br>"Valence"（情緒正負強度，1.00–9.00，小數兩位，<br>↪ 基於語義強度的主觀連續估計）<br>"Arousal"（情緒喚起強度，1.00–9.00，小數兩位，<br>↪ 基於語義強度的主觀連續估計）<br><br>請遵守以下規則：<br>1. 輸出必須是一個 JSON 陣列，不得包含任何非 JSON 文字<br>2. Aspect 與 Opinion 必須完全照抄原文，不可修改、標準化、裁切或推測<br>3. 若評論中存在多個不同的 Aspect 或 Opinion，需分別生成多個 JSON 物件；<br>↪ 若同一 Aspect 對應多個不同 Opinion，也需分別生成多個四元組<br>4. 僅當評論中存在明確情緒表達，但無法在原文中找到任何可對應的 Aspect 或<br>↪ Opinion 文字片段時，才允許填寫 "NULL"<br>5. 若評論中不存在任何明確可抽取的情感四元組，輸出空陣列 []<br>6. 嚴格避免假陽性：僅在 Aspect 與 Opinion<br>↪ 均在原文中明確存在時才生成四元組<br>7. 請檢查評論中所有可能的 Aspect-Opinion 組合，<br>↪ 務必完整覆蓋所有明確表達的情感<br>8. 僅輸出 JSON，不要附加任何說明、分析或解釋<br><br>請根據以下評論抽取情感四元組：</td><td><|im_start|>user<br>{instruction}<br><br>評論："{text}"<br><|im_end|><br><|im_start|>assistant<br>{answer_json}<br><|im_end|></td></tr>
<tr><td>jpn_hotel</td><td>あなたは日本語レビューから感情四つ組を抽出するモデルです。<br>↪ 以下の規則に厳密に従い、<br>↪ 入力テキストから四つ組をすべて抽出してください。<br>↪ 【使用可能なカテゴリ一覧】{CATEGORIES}【抽出項目】1. "Aspect"：<br>↪ テキスト内に実際に出現する対象語（原文を一字一句そのまま抽出）2.<br>↪ "Category"：上記カテゴリ一覧から最も適切なカテゴリ名を選択 3.<br>↪ "Opinion"：Aspect に対する評価表現（原文をそのまま抽出し、否定・<br>↪ 程度語も含める）4. "Valence"：感情の正負強度（1.00〜9.00）5.<br>↪ "Arousal"：感情の覚醒度（1.00〜9.00）【出力規則】- JSON<br>↪ 配列のみを出力し、余計な文字を一切含めないこと。-<br>↪ 原文に存在しない語句を作らないこと。- 明確な Aspect または Opinion<br>↪ がない場合は "NULL" を使用する。-<br>↪ 抽出すべき四つ組が存在しない場合は [] を返す。-<br>↪ 複数ある場合は要素を分けてすべて出力する。</td><td><|im_start|>user<br>{instruction}<br><br>レビュー本文："{text}"<br><|im_end|><br><|im_start|>assistant<br>{answer_json}<br><|im_end|></td></tr>
<tr><td>rus_restaurant</td><td>Вы — эксперт по анализу настроений. Ваша задача — извлечь все<br>↪ эмоциональные четверки из данного пользовательского отзыва.<br>Вот правила:<br>1. Вы ДОЛЖНЫ форматировать ответ как один корректный JSON-список<br>↪ объектов.<br>2. Каждый объект ДОЛЖЕН содержать ключи: 'Aspect', 'Category',<br>↪ 'Opinion', 'Valence' и 'Arousal'.<br>3. 'Aspect' и 'Opinion' ДОЛЖНЫ быть точными подстроками из исходного<br>↪ текста.<br>- Копируйте текст ТОЧНО как он есть, включая пробелы, пунктуацию и<br>↪ ошибки.<br>- НЕ исправляйте и не нормализуйте текст.<br>- 'Opinion' может идти перед или после 'Aspect'; извлекайте полные<br>↪ 'Aspect' и 'Opinion', независимо от порядка в предложении.<br>- 'Opinion' ДОЛЖНО быть полным, включая все прилагательные,<br>↪ отрицания и модификаторы.<br>4. 'Category' ДОЛЖЕН быть одним из следующих значений: [{CATEGORIES}].<br>5. 'Valence' и 'Arousal' ДОЛЖНЫ быть числами от 1.00 до 9.00 с<br>↪ точностью до двух знаков после запятой.<br>6. Ваш ответ должен содержать *только* JSON-список.<br>Теперь выполните задачу для следующего отзыва:</td><td><|im_start|>user<br>{instruction}<br><br>Отзыв: "{text}"<br><|im_end|><br><|im_start|>assistant<br>{answer_json}<br><|im_end|></td></tr>
</tbody>
</table>

Continued on next page.

| Lang/Domain | Instruction (original prompt) | Training chat template (user + assistant) |
|---|---|---|
| **ukr_restaurant** | Ти є високоточним інструментом аспектно-орієнтованого аналізу<br>↪ тональності.<br>Твоє завдання — витягнути всі релевантні чотирики (Aspect–Category–<br>↪ Opinion–Valence–Arousal) з поданого тексту відгуку.<br><br>Правила:<br>1. Відповідь має бути єдиним валідним JSON-масивом об' єктів.<br>2. Кожен об' єкт відповідає одній парі Aspect–Opinion, що виражає одну<br>↪ окрему оцінку.<br>3. Кожен об' єкт повинен містити рівно такі ключі: "Aspect",<br>↪ "Category", "Opinion", "Valence", "Arousal".<br>4. "Aspect" і "Opinion" повинні бути точними підрядками з тексту<br>↪ відгуку — копіюй символ у символ, без змін, нормалізації чи<br>↪ перефразування.<br>5. "Opinion" має включати всі прикметники, заперечення та підсилювачі,<br>↪ що формують емоційну оцінку аспекту.<br>6. "Category" повинна бути рівно одним значенням зі списку дозволених<br>↪ категорій: [{CATEGORIES}]. Жодні інші значення заборонені.<br>7. "Valence" і "Arousal" — дійсні числа від 1.00 до 9.00 включно,<br>↪ обов' язково з двома десятковими знаками. Valence: 1.00 — вкрай<br>↪ негативна, 9.00 — вкрай позитивна оцінка. Arousal: 1.00 — низька<br>↪ емоційна інтенсивність, 9.00 — дуже висока інтенсивність.<br>8. Якщо у відгуку немає жодної релевантної пари Aspect–Opinion,<br>↪ поверни порожній JSON-масив [].<br>9. Відповідь повинна містити лише JSON-масив — без пояснень,<br>↪ коментарів або будь-якого додаткового тексту.<br><br>Ось текст відгуку: | <\|im_start\|>user<br>{instruction}<br><br>Текст відгуку: "{text}"<br><\|im_end\|><br><\|im_start\|>assistant<br>{answer_json}<br><\|im_end\|> |
| **tat_restaurant** | Син — татар телендәге күзәтү текстларыннан аспект-дүртлекләрне чыгару<br>↪ өчен махсуслашкан система.<br>Түбәндәге кагыйдәләрне төгәл үтә:<br>1) Җавап БАРЫ тик дөрес JSON-массив булырга тиеш, башка бернинди текст<br>↪ язма.<br>2) Һәр элементта мәҗбүри кырлар:<br>↪ "Aspect","Category","Opinion","Valence","Arousal".<br>3) "Aspect" һәм "Opinion" тексттан үзгәртүсез күчерелә.<br>4) "Aspect" һәм "Opinion" теләсә нинди тәртиптә очрый ала, ләкин<br>↪ текстта мөмкин булган ҺӘР аспект–фикер парын аерым дүртлек итеп<br>↪ чыгару мәҗбүри.<br>5) "Opinion" барлык сыйфатларны, көчәйткечләрне һәм кире формаларны<br>↪ тулысынча эченә ала.<br>6) "Category" түбәндәгеләрнең берсе генә булырга тиеш: [{CATEGORIES}].<br>7) "Valence" һәм "Arousal" 1.00–9.00 диапазонында, ике дөңгеләк санга<br>↪ кадәр күрсәтелә.<br><br>Күзәтү тексты: | <\|im_start\|>user<br>{instruction}<br><br>Текст бәяләмәсе "{text}"<br><\|im_end\|><br><\|im_start\|>assistant<br>{answer_json}<br><\|im_end\|> |