# QuadAI at SemEval-2026 task 3: ensemble learning of hybrid RoBERTa and LLMs for dimensional aspect-based sentiment analysis

Vink, A.J.W. de; Ventirozos, F.K.; Amat Lefort, N.; Han, L.; Kochmar, E.; Ghosh, D.; ... ; Zampieri, M.

# QuadAI at SemEval-2026 Task 3: Ensemble Learning of Hybrid RoBERTa and LLMs for Dimensional Aspect-Based Sentiment Analysis

**A.J.W. de Vink**[1], **Filippos Karolos Ventirozos**[2], **Natalia Amat-Lefort**[1], **Lifeng Han**[*,1,3]

[1] LIACS, Leiden University, NL
[2] Manchester Metropolitan University, UK
[3] Leiden University Medical Center, NL
[*] *Corresponding Author: l.han@liacs.leidenuniv.nl*

## Abstract

We present our system for SemEval-2026 Task 3 on dimensional aspect-based sentiment regression. Our approach combines a hybrid RoBERTa encoder, which jointly predicts sentiment using regression and discretized classification heads, with large language models (LLMs) via prediction-level ensemble learning. The hybrid encoder improves prediction stability by combining continuous and discretized sentiment representations. We further explore in-context learning with LLMs and ridge-regression stacking to combine encoder and LLM predictions. Experimental results on the development set show that ensemble learning significantly improves performance over individual models, achieving substantial reductions in RMSE and improvements in correlation scores. Our findings demonstrate the complementary strengths of encoder-based and LLM-based approaches for dimensional sentiment analysis. Our development code and resources will be shared at https://github.com/aaronlifenghan/ABSentiment

## 1 Introduction

Aspect-based sentiment analysis (ABSA) is a natural language processing (NLP) task that involves a few sub-tasks that include aspect term extraction, aspect category detection, opinion extraction, and aspect sentiment classification (Zhang et al., 2022). It has witnessed the traditional ML, deep learning, and pretrained language model (PLM) approaches. The Transformer-based models (both encoders (Liao et al., 2021; Chauhan et al., 2025) and decoders (Mughal et al., 2024; Ventirozos et al., 2025b,a)) have been the dominant methods nowadays for such tasks, while challenges remain, such as data scarcity, domain application, and modeling complex aspect-opinion relationships (Nazir et al., 2020; Zhang et al., 2022).

The shared task we attended this year has two tracks: Track-A "Dimensional Aspect-Based Sentiment Analysis (DimABSA)" and Track-B "Dimensional Stance Analysis (DimStance)". We attended Task-A.1 "DimASR - Valence-Arousal (VA) Prediction" and Track-B, leaving "A.2: DimASTE - Triplet Extraction" and "A.3: DimASQP - Quadruplet Extraction" into our future work. The shared task data is described at (Lee et al., 2026; Becker et al., 2026) As examples of Task-A1 and Task-B, we list the text format of input and output in Figure 1 and 2. Valence indicates positivity and negativity with more dimensions. Arousal indicates emotional intensity from high to low. For instance, (Happy, Delight, Excited) can be located in the corner of "positive and high" emotion, while (depressed, bored, tired) can be in the "negative and low" emotion (Yu et al., 2026).[1]

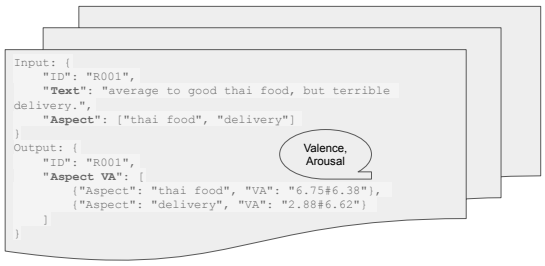


Figure 1: Example of TaskA1 data: Dimensional Aspect Sentiment Regression (DimASR)

In this work, we introduce related work to our proposed method, the methodology design and development (Hybrid RoBERTa, LLMs, ensemble), the evaluation results from development sets, and our submissions to the shared task. Due to unforeseen circumstances, we did not manage to submit all the methods we developed; however, we will continue our testing for offline development, and our codes and resources will be shared publicly for open science.

[1] https://github.com/DimABSA/DimABSA2026

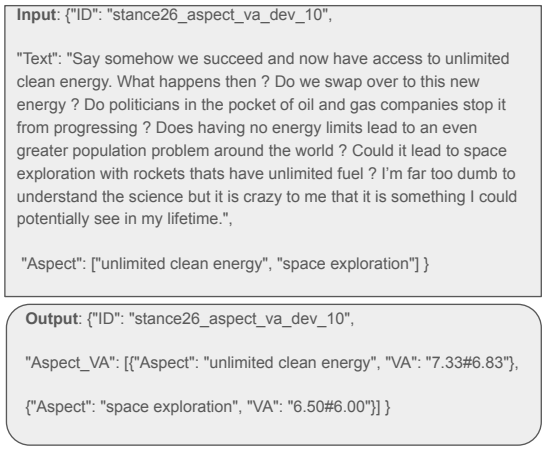


Figure 2: Example of TaskB data: Dimensional Aspect Sentiment Regression (DimASR)

## 2 Related Work

### 2.1 Beyond VA Scores

In addition to the Valance and Arousal (VA) score, Shi et al. (2025) tried to use another psychological emotion dimension "Dominance", i.e., the degree of control or influence. To address the issues of standard ABSA models that rely on word embeddings and attention but do not use structured emotion knowledge, they proposed the Graph Attention Network (GAT) method to model relationships between words and capture the syntactic dependencies. Similarly, there are other recent works using graph knowledge to address sentiment analysis, such as ReviewGraph (de Vink et al., 2025).

### 2.2 Language/Domain Specific ABSA

For language-specific ABSA, Lee et al. (2024) introduced a shared task for the Chinese language, which attracted 11 teams, focusing on intensity prediction, triplet extraction and quadruple extraction.

For domain specific work on ABSA, Chakraborty et al. (2020) used active learning on scientific reviews, using 8,000 peer reviews from ICLR conference, including (review text, score, paper decision). The work focused on the relationship between aspect sentiment (positive/negative) and paper outcome (accept/reject).

### 2.3 Hybrid Models for ABSA

Zhang et al. (2024) used hybrid setting of BERT based encoder models and LLMs. They first use BERT pipeline to extract aspects, categories, and opinions, then use LLM with QLoRA fine-tuning to predict sentiment intensity based on the BERT outputs.

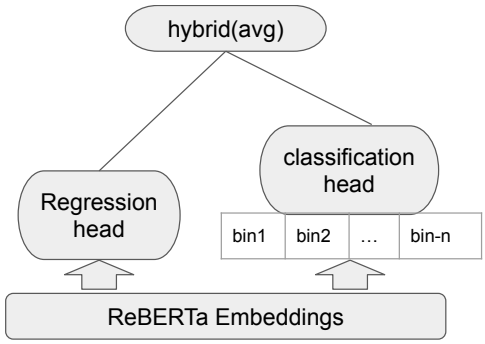


Figure 3: Hybrid RoBERTa

Another hybrid model conducted by Liang et al. (2022) combines neural networks and contextual feature representations. Their model integrates word embeddings with attention mechanisms to capture aspect-specific contextual information. Experimental results demonstrated that the hybrid approach outperforms traditional neural models on benchmark datasets. This work highlights the effectiveness of hybrid architectures for fine-grained sentiment prediction.

More challenges, tasks, and methodologies on ABSA can be found in earlier surveys (Nazir et al., 2020; Zhang et al., 2022).

## 3 Methodology

### 3.1 Hybrid RoBERTa

We designed a hybrid encoder-based model using averaged scores from regression and discretized bin integrated classification, as showin in Figure 3. We firstly used RoBERTa embedding as the encoder. Then, in parallel, we trained a regression head and a discretized classification head. The core idea of discretizing the target space is that we take the continuous embedding variable and split it into n bins. The final layer outputs an n-dimensional logit vector then applied with softmax and trained with cross-entropy loss. The advantage of discretized classification is that it is expected to be a more stable training than regression and expresses confidence over bins.

Lastly, the final prediction of Hybrid RoBERTa is obtained by averaging both outputs ($w = 0.5$). We list the equations below. Regression output:

$$\hat{y}_{reg} \tag{1}$$

Classification expected value:

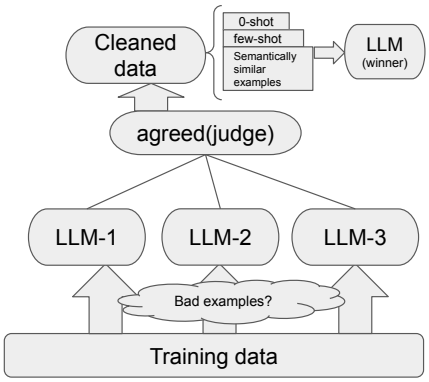


Figure 4: Triple-LLMs workflow

$$\hat{y}_{cls} = \sum_{i=1}^{B} p_i c_i \tag{2}$$

Final prediction:

$$\hat{y} = w\hat{y}_{reg} + (1-w)\hat{y}_{cls} \tag{3}$$

Training objective:

$$L = L_{reg} + \alpha L_{cls} \tag{4}$$

### 3.2 LLMs

As a starting point, we explore the difference between:

- zero-shot prompting (no examples)
- random examples (40)
- picked semantically similar examples (40, 90, 200, 600)

The semantical similarity is according to the embedding similarity scores at sentence level. We utilized the OpenAI's model for sentence embeddings[2].

Looking into some examples, we also decided to filter out low quality ones, such as cases where the labels do not look right. We call this step as Data Cleaning. The step involved firstly using HDBScan[3] to cluster all the ones used for in-context learning (i.e. training-set) instances according to the two dimensions of valence and arousal. HDBScan uses the DBScan approach but converts it into hierarchical clustering, for which we used the given, default, hyper-parameters for the clustering.

[2]text-embedding-3-large

[3]https://github.com/scikit-learn-contrib/hdbscan

```
CLUSTER_CRITIQUE_PROMPT = (
    """You are an expert in sentiment analysis.
Below is a cluster of aspect-sentiment examples grouped by
      similar
Valence-Arousal (VA) values.

DEFINITIONS:
- Valence: 1=very negative to 9=very positive
- Arousal: 1=very calm to 9=very intense

CLUSTER EXAMPLES:
{examples_text}

TASK: Identify which examples (if any) have
      spurious/incorrect VA labels
that don't match the text sentiment. An example is spurious
      if:
1. The VA values don't match the sentiment expressed in the
      text
2. The aspect sentiment is clearly different from the labeled
      values
3. The label seems inconsistent with similar examples in this
      cluster

Return ONLY valid JSON with NO extra text:
{{
  "spurious_indices": [0, 5, 12],
  "reasoning": "Example 0: Text is very negative but valence
      is too high.
  Example 5: ..."
}}

If no examples are spurious, return:
{{"spurious_indices": [], "reasoning": "All labels appear
      correct"}}"""
)
```

Figure 5: Cluster critique prompt for VA-based sentiment analysis

Following for each cluster we had separate three LLMs be presented each cluster, similar VA scores, and ask in the prompt to pinpoint which one of these is an outlier or not. The data cleaning prompt includes: Persona, Valence/Arousal spectrum definitions, Cluster inputs, Task and Output format example, which are detailed in Figure 5.

Finally, if all three LLMs agree that a specific instance(s) in a cluster is an outlier, we would remove them from the available pool of in-context learning candidates.

This is shown in Figure 4. Exact LLMs we used for this task are Gemini-3 Pro, Opus 4.5, and GPT 5.2 for data cleaning with cross validation. Table 8 in the Appendix, lists the specific versions used and in Appendix A.1 we provide evidence in our preliminary experiments that using only the "clean" instances with our Data Cleaning pipeline yields better scores with LLM prediction.

To reduce variance and improve model robustness, we explore the ensemble learning strategy, which will be described below.

### 3.3 Ensemble Learning

For Ensemble Learning, we design the prediction-level fusion (*aka* late fusion or model stacking) of Hybrid RoBERTa and LLMs with optional other features, as in Figure 6. For this work, we in-

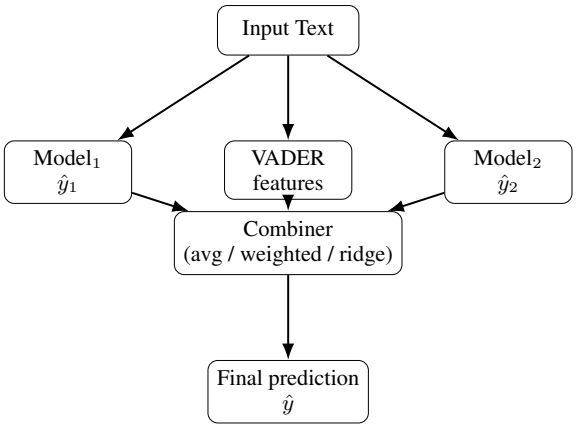


Figure 6: Prediction-level ensemble architecture combining base models and optional VADER features.

corporate lexical sentiment features derived from VADER (Hutto and Gilbert, 2014), including compound, positive, negative, and neutral polarity scores, as auxiliary inputs to the ensemble combiner.

We detail the ensemble prediction into mathematical formulas below: Given $K$ base models, each producing a prediction $\hat{y}_k$ for an input $x$, the final prediction is obtained via a combiner $g(\cdot)$:

$$\hat{y} = g(\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_K, \mathbf{f}), \tag{5}$$

where $\mathbf{f}$ denotes optional external features (e.g., VADER scores).

**Simple averaging:**

$$\hat{y} = \frac{1}{K} \sum_{k=1}^{K} \hat{y}_k \tag{6}$$

**Weighted averaging:**

$$\hat{y} = \sum_{k=1}^{K} w_k \hat{y}_k, \quad \sum_{k=1}^{K} w_k = 1 \tag{7}$$

**Ridge stacking:**

$$\hat{y} = \mathbf{w}^\top [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_K, \mathbf{f}] + b \tag{8}$$

The VADER features include: compound (continuous [-1,1]), pos, neu, neg. So the stacking vector becomes:

$$\mathbf{x} = [\hat{y}_1,\ \hat{y}_2,\ \text{compound},\ \text{pos},\ \text{neu},\ \text{neg}]$$

The rationale for including VADER as a feature is that the ensemble combines a lexicon and rule-based component, which is domain-robust and fast, with neural-based encoder models and LLMs.

**Training objective (ridge stacking).** Given training targets $\mathbf{y} \in \mathbb{R}^n$ and the design matrix $\mathbf{X} \in \mathbb{R}^{n \times (K+F)}$ whose $i$-th row is $\mathbf{x}^{(i)} = [\hat{y}_1^{(i)}, \ldots, \hat{y}_K^{(i)}, \mathbf{f}^{(i)}]$, we learn $\mathbf{w}$ and $b$ by:

$$\min_{\mathbf{w}, b} \ \frac{1}{n} \|\mathbf{y} - (\mathbf{X}\mathbf{w} + b\mathbf{1})\|_2^2 \ + \ \lambda \|\mathbf{w}\|_2^2, \tag{9}$$

We construct $\mathbf{X}$ using out-of-fold predictions of the base models to avoid label leakage.

## 4 Model Training and Development

### 4.1 Hybrid ReBERTa for Track-A1

The system performance on the development set of *laptop* category from encoder-based models is shown in Table 1 and 2. Table 1 presents the overall comparisons among regression-only, discretization-bin, and hybrid (averaging two). From the Error score metrics, we can see that the Hybrid model produced much better output on two metrics, MSE and RMSE, which have a bigger margin decrease in the error scores. However, it produced similar (or comparable) scores on RMSE(v) and RMSE(a), to regression-only and Bin models, respectively. For Pearson correlation scores, the Regression-only model produced the highest scores, though not much difference from the Hybrid model. Table 2 displays the Top-10 hybrid configurations ranked by averaging RMSE on the Laptop Dev set. We tried different sets of triple values for the parameters exhaustively; however, for future development, it would be more suitable to carry out automated hyperparameter tuning, e.g., OPTUNA (Akiba et al., 2019) to explore.

In addition, Table 3 shows the Hybrid RoBERTa performance on Task-A1 *restaurant* dev data. We can see from the scores that the hybrid model achieved the best performance on this data across all tested metrics, including both error scores and correlations. Importantly, the MSE score of the hybrid model is almost down to half that of the regression model (0.4919 vs 0.8176). Large improvement margins can also be observed from RMSE scores.

### 4.2 LLMs on Track-A Laptop Dev

The LLM output evaluation on the *laptop* Dev set is shown in Table 6, where we can see that, in comparison to Hybrid RoBERT in Table 2, the LLMs produced an even lower RMSE score of 0.695, vs 0.7361. In addition, it increased the correlation

Table 1: Performance of the Hybrid RoBERTa model on SemEval Task 1 (**Laptop**, Dev set). The model combines a regression head and a discretized classification head (31 bins), with the final prediction obtained by averaging both outputs ($w = 0.5$). Error categories: lower is better; in 3 categories hybrid is much better, except for RMSE that is comparable to Bin. Pearson correation $\rho$ higher is better rho( Regress>Hybrid>Bin)

| Model Variant | MSE ↓ | RMSE ↓ | $\text{RMSE}_v$ ↓ | $\text{RMSE}_a$ ↓ | $\rho_v$ ↑ | $\rho_a$ ↑ | $\rho_{mean}$ ↑ |
|---|---|---|---|---|---|---|---|
| Regression only | 0.6140 | 0.7836 | 0.7201 | 0.8423 | 0.9102 | 0.5505 | 0.7304 |
| Bin (expected value) | 0.6238 | 0.7898 | 0.8340 | **0.7430** | 0.8974 | 0.5126 | 0.7050 |
| Hybrid (average) | **0.5419** | **0.7361** | **0.7214** | 0.7506 | 0.9074 | 0.5388 | 0.7231 |

Table 2: Top-10 Hybrid configurations ranked by average RMSE on the **Laptop** Dev set. $num\_bins$ denotes the number of discretization bins, $\alpha$ the classification loss weight, and $w$ the regression–classification averaging weight.

| $num\_bins$ | $\alpha$ | $w$ | $\text{RMSE}_{avg}$ ↓ | $\rho_{mean}$ ↑ | $\rho_a$ ↑ |
|---|---|---|---|---|---|
| 31 | 0.5 | 0.5 | **0.7361** | 0.7231 | 0.5388 |
| 11 | 1.0 | 0.5 | 0.7368 | 0.7182 | 0.5277 |
| 31 | 0.5 | 0.4 | 0.7376 | 0.7252 | 0.5421 |
| 7 | 0.2 | 0.5 | 0.7405 | 0.7287 | 0.5473 |
| 31 | 0.5 | 0.3 | 0.7431 | 0.7270 | 0.5448 |
| 11 | 1.0 | 0.4 | 0.7448 | 0.7185 | 0.5278 |
| 7 | 0.2 | 0.4 | 0.7473 | 0.7293 | 0.5485 |
| 11 | 0.5 | 0.5 | 0.7481 | 0.7251 | 0.5523 |
| 31 | 1.0 | 0.5 | 0.7488 | 0.7138 | 0.5278 |
| 31 | 0.2 | 0.4 | 0.7516 | 0.7151 | 0.5234 |

mean score from 0.7231 to 0.757. More information about the LLM implementation and comparisons can be found in Appendix 8.

### 4.3 Ensemble on TrackA Laptop Dev

Table 4 shows the results from ensemble learning of two models, with/without the VADER feature on the *laptop* data, which show much improvement in comparison to individual models (hybrid RoBERTa and LLMs), especially on RMSE scores. For the weighted ensemble, weights were selected via grid search over the interval [0,1] with step size 0.1, optimizing RMSE on the development set. The goal is that 1) better models get more influence; 2) it reduces bias compared to equal averaging. The experimental results show that average weighting produced RMSE score 0.6399 vs Weighted 0.6344 (lower and better). The results also show that the VADER feature did not make an improvement in the evaluation scores, even a slight degradation, which indicates that VADER probably adds noise or a redundant signal. In addition, the weighted avg and stacking produced the same RMSE scores (0.6344), although other scores are different. This might suggest that the linear ridge basically learned weights approximately [0.3, 0.7], so stacking approximately a weighted average. This often happens with only 2 models. To explore this, future work shall explore more models for the ensemble.

### 4.4 Hybrid RoBERTa on Track-B

Table 5 shows the performance of Hybrid RoBERTa on Track-B English data - Environmental Protection. The best hybrid configuration and the evaluation metrics are listed. Once again, the hybrid model performed the best over individual regression and bin-expected models.

## 5 Model Submissions

Due to unforeseen situations, we were only available to submit the system output for TaskA.1 using the Hybrid RoBERTa model, i.e., without the LLMs and ensemble-learning variations. In addition, we did not submit for Track-B. The initial/unofficial ranking from the organisers on our Hybrid ReBERTa (lightweight) shows that it achieved 16/30 and 22/33 on laptop and restaurant data, respectively. On laptop data, the best performing team has a score of 1,2408, while Hybrid RoBERTa has **1,4062**, which is closer to the best team and much better than the bottom-ranking team 1,8486 and baseline 2,8053. Similarly, on restaurant data, the best performing team has 1,1035 error score, while Hybrid RoBERTa produced **1,3632**, much better than the last team 1,9115 and baseline 2,791. Considering the very **low cost** from encoder-based Hybrid RoBERTa with **constrained** training, this performance is very promising, as shown in Table 9 and 10 for full ranking, and Table 7 for brief summary.

## 6 Conclusions and Future Work

In this system paper, we introduced a hybrid encoder model RoBERTa averaging the weighted performance of regression and discretized classification heads. In addition, we explored the prediction-level fusion (late fusion or model stacking) for ensembling the output scores of the hybrid encoder and LLM; however, for future work, we would like to explore different kinds of ensemble methods, e.g., stacking ensemble from (Romero et al., 2025), automatic hyper-parameter finetuning (Akiba et al.,

Table 3: **Hybrid** RoBERTa results on the **Restaurant** development set for SemEval Task 1 (Valence/Arousal). The model combines a regression head and a hard-bin classification head; the *average* prediction is a weighted combination with `pred_weight` $= 0.5$. Best configuration and detailed dev metrics are shown.

| Domain | Variant | MSE↓ | RMSE↓ | $\text{RMSE}_v$↓ | $\text{RMSE}_a$↓ | $\rho_v$↑ | $\rho_a$↑ | $\rho_{mean}$↑ |
|---|---|---|---|---|---|---|---|---|
| Restaurant | Regression | 0.8176 | 0.9042 | 0.8212 | 0.9802 | 0.9205 | 0.6458 | 0.7832 |
| | Bin-expected | 0.5369 | 0.7327 | 0.8154 | 0.6395 | 0.9130 | 0.6408 | 0.7769 |
| | Average ($w = 0.5$) | **0.4919** | **0.7013** | **0.6692** | **0.7320** | **0.9217** | **0.6679** | **0.7948** |

Best config (Restaurant): `num_bins` $= 11$, `alpha_cls` $= 0.2$, `pred_weight` $= 0.5$.

Table 4: **Ensemble** results on the Laptop Dev set for SemEval Task 1 (Valence/Arousal). We compare simple averaging, weighted averaging, and ridge-regression stacking (out-of-fold, OOF) with and without additional VADER-based features. Best (lowest) RMSE among valid (non-leaking) settings is highlighted.

| Setting | Method | MSE↓ | RMSE↓ | $\text{RMSE}_v$↓ | $\text{RMSE}_a$↓ | $\rho_v$↑ | $\rho_a$↑ | $\rho_{mean}$↑ |
|---|---|---|---|---|---|---|---|---|
| Without VADER | Avg | 0.4095 | 0.6399 | 0.5835 | 0.6918 | 0.9391 | 0.6105 | 0.7748 |
| | Weighted (w=[0.3,0.7]) | 0.4025 | **0.6344** | 0.5629 | 0.6987 | 0.9421 | 0.6124 | 0.7773 |
| | Stacking (Ridge, OOF) | 0.4025 | **0.6344** | 0.5713 | 0.6918 | 0.9397 | 0.5893 | 0.7645 |
| With VADER | Avg | 0.4095 | 0.6399 | 0.5835 | 0.6918 | 0.9391 | 0.6105 | 0.7748 |
| | Weighted (w=[0.3,0.7]) | 0.4025 | **0.6344** | 0.5629 | 0.6987 | 0.9421 | 0.6124 | 0.7773 |
| | Stacking (Ridge, OOF) | 0.4079 | 0.6387 | 0.5718 | 0.6992 | 0.9396 | 0.5793 | 0.7594 |

Table 5: Track B (English) Environmental Protection — best hybrid configuration and DEV results.

| **Best config** | **num_bins** | $\alpha_{\text{cls}}$ | **pred_weight** ($w$) |
|---|---|---|---|
| | 21 | 1.0 | 0.3 |

| **Variant** | **MSE** | **RMSE** | **$\text{RMSE}_v$** | **$\text{RMSE}_a$** | **$\text{Pearson}_v$** | **$\text{Pearson}_a$** |
|---|---|---|---|---|---|---|
| Regression | 2.0287 | 1.4243 | 1.7162 | 1.0546 | 0.5311 | 0.2221 |
| Bin-Expected | 2.2221 | 1.4907 | 1.8301 | 1.0464 | 0.5272 | 0.0664 |
| Average ($w$=0.3) | 1.9661 | 1.4022 | 1.6932 | 1.0322 | 0.5312 | 0.2234 |

*Note.* Pearson mean: Regression = 0.3766, Bin-Expected = 0.2968, Average = 0.3773. Also: $\text{RMSE}_{\text{avg}} = 1.4022$, $\text{RMSE}_{\text{reg}} = 1.4243$, $\text{RMSE}_{\text{cls}} = 1.4907$.

†Full-fit stacking is trained on the full dev set and evaluated on the same dev set; it is therefore optimistic and should not be used for fair model selection. OOF stacking is the appropriate estimate.

| Model | MSE | RMSE | $\text{RMSE}_v$ | $\text{RMSE}_a$ | $\rho$ |
|---|---|---|---|---|---|
| LLM (ICL) | 0.484 | 0.695 | 0.633 | 0.752 | 0.757 |

Table 6: LLM dev results (Laptop).

| Dataset | Rank | RMSE | Baseline |
|---|---|---|---|
| Laptop | 16 / 29 | 1.4062 | 2.8053 |
| Restaurant | 22 / 32 | 1.3632 | 2.7910 |

Table 7: Summary of our system performance on the SemEval-2026 Track A.1 datasets. Lower RMSE is better.

2019), as well as multi-modal sentiment tasks (Lindevelt et al., 2026).

## Limitations

There are some limitations from our work. 1) Due to time limitations, we did not apply LLMs and ensembles on the test set, but on the Dev set. We will explore our system performance on test sets offline. 2)To test model generalisability, we plan to explore the performance on languages other than English, such as Chinese. 3) For LLM Model selection, we applied larger models Gemini, Claude, and GPT. It would be more inclusive to include smaller LLMs for comparisons.

## Acknowledgments

We thank the reviewers for valuable comments on our work.

## References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. *Preprint*, arXiv:1907.10902.

Jonas Becker, Liang-Chih Yu, Shamsuddeen Hassan Muhammad, Jan Philip Wahle, Terry Ruas, Idris Abdulmumin, Lung-Hao Lee, Nelson Odhiambo, Lilian Wanzare, Wen-Ni Liu, Tzu-Mi Lin, Zhe-Yu Xu, Ying-Lung Lin, Jin Wang, Maryam Ibrahim Mukhtar, Bela Gipp, and Saif M. Mohammad. 2026. Dimstance:

Multilingual datasets for dimensional stance analysis. *Preprint*, arXiv:2601.21483.

Souvic Chakraborty, Pawan Goyal, and Animesh Mukherjee. 2020. Aspect-based sentiment analysis of scientific reviews. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, pages 207–216.

Amit Chauhan, Aman Sharma, and Rajni Mohana. 2025. An enhanced aspect-based sentiment analysis model based on roberta for text sentiment analysis. *Informatica*, 49(14).

AJW de Vink, Natalia Amat-Lefort, and Lifeng Han. 2025. Reviewgraph: A knowledge graph embedding based framework for review rating prediction with sentiment features. *arXiv preprint arXiv:2508.13953*.

C. J. Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the 8th International Conference on Weblogs and Social Media (ICWSM)*.

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. *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.

Bin Liang, Hang Su, Lin Gui, Erik Cambria, and Ruifeng Xu. 2022. Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. *Knowledge-based systems*, 235:107643.

Wenxiong Liao, Bi Zeng, Xiuwen Yin, and Pengfei Wei. 2021. An improved aspect-category sentiment analysis model for text sentiment analysis based on roberta. *Applied Intelligence*, 51(6):3522–3533.

David Lindevelt, Suzan Verberne, and Joost Broekens. 2026. The correlation between emotion in text and speech segments is limited: A cross-modal study. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 2611–2621, Rabat, Morocco. Association for Computational Linguistics.

Nimra Mughal, Ghulam Mujtaba, Sarang Shaikh, Aveenash Kumar, and Sher Muhammad Daudpota. 2024. Comparative analysis of deep natural networks and large language models for aspect-based sentiment analysis. *Ieee Access*, 12:60943–60959.

Ambreen Nazir, Yuan Rao, Lianwei Wu, and Ling Sun. 2020. Issues and challenges of aspect-based sentiment analysis: A comprehensive survey. *IEEE Transactions on Affective Computing*, 13(2):845–863.

Pablo Romero, Lifeng Han, and Goran Nenadic. 2025. Medication extraction and entity linking using stacked and voted ensembles on LLMs. In *Proceedings of the Second Workshop on Patient-Oriented Language Processing (CL4Health)*, pages 303–315, Albuquerque, New Mexico. Association for Computational Linguistics.

Xuefeng Shi, Weiping Ding, Min Hu, Xin Kang, and Fuji Ren. 2025. Triple dimensional psychology knowledge encouraging graph attention networks to exploit aspect-based sentiment analysis. *Scientific Reports*, 15(1):27109.

Filippos Ventirozos, Peter A. Appleby, and Matthew Shardlow. 2025a. Are you sure you're positive? consolidating chain-of-thought agents with uncertainty quantification for aspect-category sentiment analysis. In *Proceedings of the 1st Workshop for Research on Agent Language Models (REALM 2025)*, pages 309–326, Vienna, Austria. Association for Computational Linguistics.

Filippos Karolos Ventirozos, Peter Appleby, and Matthew Shardlow. 2025b. Aspect–sentiment quad prediction with distilled large language models. In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing - Natural Language Processing in the Generative AI Era*, pages 1309–1319, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Liang-Chih Yu, Jonas Becker, Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Lung-Hao Lee, Ying-Lung Lin, Jin Wang, Jan Philip Wahle, Terry Ruas, Natalia Loukachevitch, Alexander Panchenko, Ilseyar Alimova, Lilian Wanzare, Nelson Odhiambo, Bela Gipp, Kai-Wei Chang, and Saif M. Mohammad. 2026. Semeval-2026 task 3: Dimensional aspect-based sentiment analysis (dimabsa). *Preprint*, arXiv:2604.07066.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 35(11):11019–11038.

Yice Zhang, Hongling Xu, Delong Zhang, and Ruifeng Xu. 2024. A hybrid approach to dimensional aspect-based sentiment analysis using bert and large language models. *Electronics*, 13(18):3724.

# A LLMs

In Table 8 we provide the exact LLM version models used for our experiments. Then we discuss our preliminary experiments, the significance of each configuration, and the best model and configuration. For our final result, we were unable to run the best model (i.e. Gemini-3-pro) with its configurations. However, we report the ensemble of three vendor models using ICL with 40 semantically retrieved examples on the aforementioned dev set (§4.2). For the comparisons below, we used subsets of the test set.

| Provider | Model | Config |
|---|---|---|
| OpenAI | `gpt-5.2` | effort=high verbosity=low responses API |
| Anthropic | `claude-opus-4-5` | ext. thinking budget=16k |
| Google | `gemini-3-pro` | thinking=HIGH safety=OFF |

Table 8: LLMs used in the multi-vendor ensemble. Model IDs abbreviated; full versions: `claude-opus-4-5-20251101`, `gemini-3-pro-preview`.

## A.1 Effect of Training-Data Cleaning

To experiment the effect of whether the Data Cleaning process is useful, we compared using our three LLMs voted average with each having a distinct persona and 40 semantically similar retrieved examples in the prompt, a configuration that was deemed robust. We compared using the configuration with the 40 examples retrieved from all the training set (3,795 instances) versus (2,576 instances) from the cleaned training set. The results demonstrated that the RMSE favourably dropped from $0.091$ to $0.026$, a $71.4\%$ relative reduction. In detail, gpt-5.2 alone improves by $62.5\%$ ($0.096 \rightarrow 0.036$), Claude-opus-4-5 by $69.8\%$ ($0.091 \rightarrow 0.028$), and Gemini-3-pro by $85.8\%$ ($0.122 \rightarrow 0.017$).

As a control for background model drift, we compare gpt-5.2's zero-shot RMSE across four independent runs and find it stable within $\pm 0.006$ of $0.180$. Because zero-shot prompting does not consult the training pool, this stability isolates the observed gain to the pool rather than to provider-side model updates.

## A.2 Effect of Model Choice

Regarding which model performed best, we observed Gemini-3-pro ($0.017$) $<$ Claude-opus-4-5 ($0.028$) $<$ gpt-5.2 ($0.036$). Even when using the three LLMs averaged scores ($0.026$) the Gemini alone would outperform this ensemble by $33\%$ on tested subset. The vendor-diversity hypothesis—that averaging across heterogeneous model families produces a stronger signal—therefore does not hold in this regime, as the strongest model alone exceeds any blend that includes weaker constituents.

## A.3 Effect of In-Context Examples and Retrieval Method

At the outset, few-shot prompting consistently outperforms zero-shot prompting across retrieval methods. Evidently, in four GPT-5.2 configurations, $k = 40$ reduces RMSE from $\sim 0.18$ to $\sim 0.10$ (a $\sim 45\%$ relative reduction; paired-bootstrap CI excludes $0$).

Intuitively, semantic retrieval consistently outperformed random retrieval when using $k = 40$ ICL instances. We conducted two GPT-5.2 runs per condition (same model, same prompt, same test slice). Both semantic runs (RMSE $0.104$, $0.116$; mean $0.110$) outperformed both random runs ($0.126$, $0.121$; mean $0.124$); a mean relative reduction of $\sim 11\%$.

When we picked our best model candidate Gemini-3-pro on the cleaned pool, increasing the number of semantically retrieved ICL examples from $k = 90$ to $k = 200$ reduced RMSE more than four-fold ($0.11 \rightarrow 0.025$); a further increase to $k = 600$ yields an additional but smaller gain to $0.016$.

Overall, performance improves with both the use and the quality of in-context examples: few-shot provides the largest gain, semantic retrieval offers consistent improvements over random, and increasing $k$ yields diminishing but meaningful returns.

## A.4 Effect of Persona Assignment

We test persona sensitivity with three replicates of a 3-Gemini ensemble at $k = 40$, varying only the assignment of three expert personas intended to induce response diversity. RMSE varies by at most $0.009$ across permutations, far smaller than the training-pool effect (§A.1) and the single-model gap (§A.2). This suggests limited impact at this scale. No no-persona baseline was tested, so results reflect interchangeability rather than absolute

| Score ↓ | Team | Rank |
|---|---|---|
| 1,2408 | LogSigma | 1 |
| 1,2425 | TeleAI | 2 |
| 1,2769 | Bert Kittens | 3 |
| 1,2833 | RPI Team | 4 |
| 1,2942 | SokraTUM | 5 |
| 1,3048 | NLANGPROC | 6 |
| 1,3098 | ICT-NLP | 7 |
| 1,3261 | PICT | 8 |
| 1,3283 | The Classics | 9 |
| 1,3455 | YangS_team | 10 |
| 1,3501 | NTNU-SMIL | 11 |
| 1,3612 | PALI | 12 |
| 1,3654 | Habib university | 13 |
| 1,3841 | Cortexa | 14 |
| 1,3946 | SCU_Mesclab | 15 |
| **1,4062** | **QuadAI** | **16** |
| 1,4109 | HUS@NLP-VNU | 17 |
| 1,4190 | Pixel Phantoms | 18 |
| 1,4242 | SCUZANE | 19 |
| 1,4336 | UNF-BMI | 20 |
| 1,4394 | PAI | 21 |
| 1,4401 | AILS-NTUA | 22 |
| 1,4557 | LexiCore | 23 |
| 1,4562 | NCL-BU | 24 |
| 1,5073 | EmberAI | 25 |
| 1,5080 | REGLAT | 26 |
| 1,5412 | hdharpure | 27 |
| 1,5924 | DUTH | 28 |
| 1,8486 | surface3 | 29 |
| 2,8053 | Baseline | – |

Table 9: Initial SemEval-2026 Track A.1 ranking on the English Laptop dataset. Lower scores are better. Our system, **QuadAI**, ranked 16th among participating teams and outperformed the official baseline.

benefit.

## B QuadAI Ranking Among Teams

We list the initial/un-official ranking we received for reference in Table 9 and 10 for full ranking on Track-A1 laptop and restaurant data respectively (Yu et al., 2026).

| Score ↓ | Team | Rank |
|---|---|---|
| 1,1035 | LogSigma | 1 |
| 1,1812 | Bert Kittens | 2 |
| 1,1958 | PICT | 3 |
| 1,2006 | RPI Team | 4 |
| 1,2116 | NLANGPROC | 5 |
| 1,2139 | TeleAI | 6 |
| 1,2141 | PAI | 7 |
| 1,2270 | SRCB | 8 |
| 1,2277 | SCU_Mesclab | 9 |
| 1,2324 | The Classics | 10 |
| 1,2676 | ICT-NLP | 11 |
| 1,2745 | HUS@NLP-VNU | 12 |
| 1,2772 | YangS_team | 13 |
| 1,2846 | NTNU-SMIL | 14 |
| 1,2866 | PALI | 15 |
| 1,2990 | NTNU-SMIL | 16 |
| 1,3011 | SokraTUM | 17 |
| 1,3049 | Habib university | 18 |
| 1,3168 | Scmhl5 | 19 |
| 1,3270 | Cortexa | 20 |
| 1,3483 | SCUZANE | 21 |
| **1,3632** | **QuadAI** | **22** |
| 1,3656 | Pixel Phantoms | 23 |
| 1,3742 | LexiCore | 24 |
| 1,3920 | UNF-BMI | 25 |
| 1,3933 | AILS-NTUA | 26 |
| 1,4265 | TeamLasse | 27 |
| 1,4268 | satyam_1909 | 28 |
| 1,4678 | EmberAI | 29 |
| 1,4861 | NCL-BU | 30 |
| 1,5003 | hdharpure | 31 |
| 1,9115 | surface3 | 32 |
| 2,7910 | Baseline | 33 |

Table 10: Initial SemEval-2026 Track A.1 ranking on the English Restaurant dataset. Lower scores are better. Our system, **QuadAI**, ranked 22nd and outperformed the official baseline.