LG-DimABSA: A Language-Grounded Approach to Dimensional Aspect-Based Sentiment Analysis

Anushka Sharma 2023113006
Disha Pant 2023101085
Sama Harshika 2023101004
Vignesh Vembar 2023102019

Abstract

Traditional sentiment analysis relies on discrete polarity labels, which fail to capture the continuous nature of affect. Dimensional sentiment modeling using the valence–arousal (VA) framework provides a richer representation by encoding both polarity and intensity. While Aspect-Based Sentiment Analysis (ABSA) enables fine-grained sentiment assignment to aspect–opinion pairs, existing DimABSA approaches often rely on end-to-end models with limited interpretability.

We propose LG-DimABSA, a language-grounded staged pipeline that decomposes the task into candidate extraction, aspect–opinion pair matching, category classification, and VA regression. By explicitly modeling linguistic structure, our approach improves transparency and modularity while achieving competitive predictive performance with reduced computational cost and faster inference compared to end-to-end baselines. We incorporate dependency-based span extraction, biaffine pair modeling, constraint-aware categorization, and contrastive learning for VA prediction, demonstrating the effectiveness of explicit linguistic decomposition for efficient and interpretable dimensional sentiment analysis.

Codebase: https://github.com/wig-nesh/LG-DimABSA

1 Introduction

The growth of social media and online review platforms has led to large volumes of user-generated opinionated text. Analyzing this data is valuable for tasks such as understanding consumer feedback, public opinion, and detecting harmful content like hate speech. Sentiment analysis therefore plays an important role in extracting insights from large-scale textual data.

2 Literature Study

Aspect-Based Sentiment Analysis (ABSA) aims to identify opinions expressed toward specific targets within a text, enabling fine-grained sentiment understanding beyond broad, sentence-level polarity. Classic ABSA tasks, such as Aspect Sentiment Triplet Extraction (ASTE) and Aspect Sentiment Quadruple Prediction (ASQP), have historically treated sentiment as a coarse, discrete category (e.g., positive, neutral, negative). However, discrete labels often fail to capture subtle variations in emotional intensity. To address this, Dimensional ABSA (DimABSA) extends the paradigm by mapping sentiment to a continuous, two-dimensional space based on Russell's circumplex model (Russell, 1980) of emotions, assigning real-valued valence (pleasantness) and arousal (activation) scores to extracted aspect–opinion pairs. Recent shared tasks and benchmarks, including the SIGHAN-2024 Dimensional ABSA task (Tong and Wei, 2024) and SemEval-2026 Task 3, have formalized this formulation and provided datasets for evaluating dimensional sentiment prediction.

2.1 Neural Approaches to DimABSA and Categorization

Most recent systems designed for DimABSA rely heavily on large pretrained transformers, implemented either as structured pipelines or end-to-end generative models. For example, (Zhang, 2024) proposed a hybrid approach combining a BERT-based extraction pipeline with a Large Language Model (LLM) for sentiment intensity prediction. Recent shared-task systems further explore alternative LLM-driven formulations; (Jiang et al., 2024) reformulated the task as a paraphrase generation problem, using a Seq2Seq (T5-based) model to generate natural language sentences that linearize sentiment intensity quadru-

ples in an end-to-end manner. (Kang et al., 2024) proposes a multi-agent collaboration framework where multiple LLM-based agents jointly perform extraction, classification, and intensity estimation through structured interaction.

In addition to sentiment intensity prediction, DimABSA systems also include aspect category classification, where each aspect–opinion pair is assigned a semantic category. For the specific sub-task of aspect categorization, recent approaches frequently rely on pretrained models like BERT to obtain contextual representations of text, subsequently applying neural classifiers for direct category prediction (Zhang, 2024). Increasingly, these studies integrate LLMs or span-based tagging to jointly perform extraction and classification tasks.

2.2 Pipelined and Multi-Step Approaches to ABSA

As an alternative to monolithic architectures, several studies decompose sentiment analysis into multi-stage pipelines. In the context of Aspect Sentiment Triplet Extraction (ASTE), early frameworks adopt a two-stage strategy that first predicts sentiment elements such as aspects, opinions, and sentiment labels using sequence tagging models, and subsequently pairs these elements to construct valid triplets. Later works, such as (Peng et al., 2020), further explored specialized mechanisms like grid tagging to explicitly model the relationship between aspect and opinion spans in a unified fashion. Related approaches in dependency parsing have demonstrated the effectiveness of biaffine scoring functions for modeling pairwise interactions between textual elements (Dozat and Manning, 2017), which motivates our choice of explicit span-level interaction modeling.

3 Dataset

We evaluate our aspect extraction, opinion pairing, and categorization models using the datasets from the DimABSA Shared Task (Yu and Mohammad, 2025). The data consists of review text from domains such as laptops and restaurants, which commonly contain multiple aspects with different sentiments. In our experiments, we use the English domain and follow the official train, development, and test splits for fair comparison with prior work.

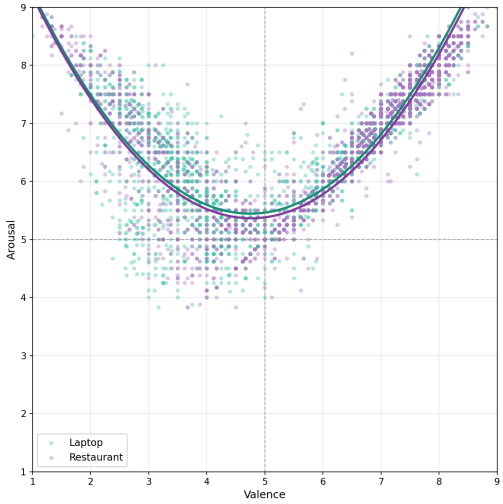

Figure 1: Scatterplot of Valence Arousal from the Train Set

3.1 Data Source and Features

Each instance in the dataset consists of a raw text sequence and an annotated list of target quadruplets. Each quadruplet is structured in the format of (aspect term, aspect category, opinion expression, valence-arousal score).

The aspect category is a pair of an entity and an attribute. The valence and arousal are represented as a pair of real valued scores. This captures both the polarity and the intensity of the sentiment expressed towards each aspect.

```
{
  "text": "The laptop has a great battery
life but the screen is dull.",
  "quadruplets": [
    ("battery life", "BATTERY#LIFE", "great",
(8.2, 7.8)),
    ("screen", "DISPLAY#QUALITY", "dull",
(2.6, 6.6))
  ]
}
```

3.2 Data Preprocessing

The raw text was first cleaned to normalize punctuation and resolve tokenizer artifacts. This included removing extra whitespace, standardizing quotation marks, fixing separated contractions etc.

Since our method focuses on explicit aspect–opinion relationships, we retained only valid pairs by removing samples with `NULL` opinions. Sentences without at least one valid pair after filtering were also discarded, ensuring that training and evaluation used only complete aspect–opinion relationships.

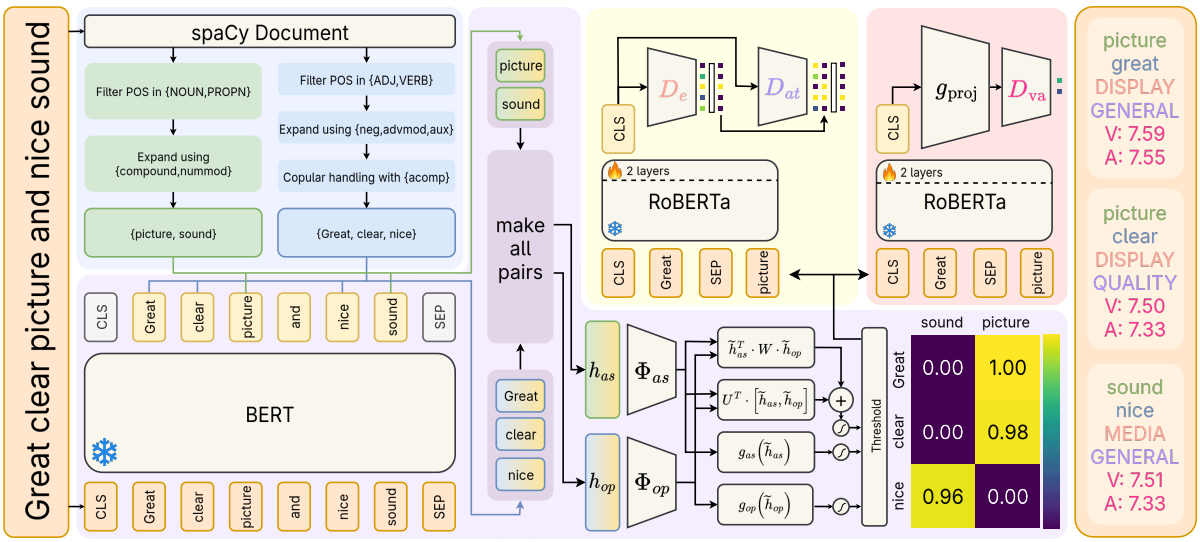

Figure 2: Overview of the proposed staged pipeline for DimABSA.

4 Methodology

4.1 Task Formulation

Given an input sentence $s = \{t_1, t_2, ..., t_n\}$, the goal of LG-DimABSA is to extract a set of quadruplets of the form:

$$(as, op, ca = (en, at), se = (va, ar))$$

where as is an aspect span, op is an opinion span, $ca \in \mathcal{E} \times \mathcal{A}$ denotes a structured category composed of an entity $en \in \mathcal{E}$ and attribute $at \in \mathcal{A}$, and a sentiment pair $se \in [1, 9]^2$ of $va \in [1, 9]$ and $ar \in [1, 9]$ representing the continuous valence and arousal scores in the affective circumplex space.

Our modular pipeline decomposes this structured prediction problem into a sequence of progressively constrained transformations. Specifically, we factorize the mapping as:

$$s \to (A, O) \to P \to (C, S)$$

where:

- $A = \{as_1, as_2, ..., as_l\}$ denotes candidate aspect spans
- $O = \{op_1, op_2, ..., op_m\}$ denotes candidate opinion spans
- $P \subseteq A \times O$ represents valid aspect-opinion pairs
- $C \subseteq \mathcal{E} \times \mathcal{A}$ assigns a structured category to each pair
- $S \subseteq [1, 9]^2$ maps each pair to a point in the valence-arousal space

The pipeline consists of four stages:

1. Candidate Extraction: Propose aspect and opinion span candidates.
2. Pair Matching: Identify valid aspect–opinion pairs from span representations.
3. Category Classification: Assign an entity–attribute category to each valid pair.
4. VA Regression: Map each pair to a point in the valence–arousal space.

This decomposition introduces strong inductive biases at each stage, enabling efficient learning under limited supervision while maintaining interpretability and modularity. Our model avoids the combinatorial complexity of jointly modeling all components in a single end-to-end architecture. We use classical methods for high recall candidate generations, and neural methods to filter and classify candidates, allowing us to leverage the strengths of both paradigms.

4.2 Candidate Extraction

A central challenge in DimABSA is the combinatorial explosion of possible aspect–opinion interactions. Given a sentence of length n, naively considering all token-level pairs results in $\mathcal{O}(n^2)$ candidate interactions, the majority of which are invalid. This not only introduces significant noise but also leads to poor sample efficiency.

To address this, we introduce a high-recall proposal generation stage that restricts the search space to linguistically plausible spans. Concretely, we extract candidate aspect and opinion spans:

$$A = \{as_1, as_2, ..., as_l\}$$
$$O = \{op_1, op_2, ..., op_m\}$$

where each span corresponds to a contiguous subsequence of tokens in s.

Our approach leverages dependency parsing using (Honnibal et al., 2020) to construct these candidates, prioritizing recall over precision. While this stage may introduce false positives, downstream components are designed to filter invalid pairs, making high recall the more critical objective.

4.2.1 Aspects

Aspect expressions correspond to entities or attributes that are being evaluated. Linguistically, these expressions are most commonly realized as nominal phrases, often centered around nouns and proper nouns with attached modifiers.

We therefore construct candidate aspect spans by first identifying all tokens whose part-of-speech tags belong to `{"NOUN","PROPN"}`, and treating them as potential heads of aspect phrases.

When then expand each head using its local dependency structure. Specifically, we include dependents connected via `{"compound","nummod"}`.

This allows the model to capture multi-token expressions where modifiers are integral to meaning (e.g., "battery life", "USB ports"). The resulting span is defined as the minimal contiguous segment covering the head and its selected dependents.

4.2.2 Opinions

Opinion expressions encode sentiment toward aspects and are typically realized as predicate-like structures, most commonly adjectives or verbs, often modified by intensifiers or negations.

We identify candidate opinion spans by selecting tokens with part-of-speech tags in `{"ADJ", "VERB"}`, which serve as anchors for sentiment-bearing expressions.

We then expand each anchor using its dependency neighborhood, incorporating modifiers connected via `{"advmod", "neg", "aux"}`.

In addition, we explicitly handle copular constructions, where sentiment is expressed via adjectival complements linked through a copula (e.g., "the screen is great"). In such cases, `{"acomp"}` is included in the expansion to ensure that the opinion span captures the complete sentiment expression.

The resulting spans correspond to minimal units that preserve the compositional structure of sentiment expressions (e.g., "not very good", "highly recommend"), rather than treating sentiment as isolated tokens.

4.3 Biaffine Pair Matching

Even after restricting the search space to candidate spans A and O, the cartesian product $A \times O$ still contains many invalid combinations. Naively modeling all token-level interactions using self-attention introduces substantial noise and requiring large amounts of supervision to learn meaningful associations.

To address this, we formulate pair identification as a span-level relation modeling problem using two classifiers and a biaffine scoring mechanism. This allows us to explicitly model the asymmetric relationship between aspect and opinion spans while operating over a much smaller set of candidates.

4.3.1 Aspect and Opinion Representation

We first encode the input sentence using BERT, producing contextualized token representations:

$$H = \{h_1, h_2, ..., h_n\}, h_i \in \mathbb{R}^d$$

For each candidate span $[\text{start}, \text{end}]$, we compute fixed-length representation via mean pooling:

$$h_{[\text{start},\text{end}]} = \frac{1}{\text{end} - \text{start} + 1} \sum_{i=\text{start}}^{\text{end}} h_i$$

This aggregation captures the semantic content of the span while preserving contextual information from the encoder.

Aspect and opinion spans play inherently asymmetric roles in sentiment relations. To reflect this, we project span representations into distinct latent spaces using separate multi-layer perceptrons:

$$\tilde{h}_{as} = \Phi_{as}(h_{as})$$
$$\tilde{h}_{op} = \Phi_{op}(h_{op})$$

This separation enables the model to encode role-specific features relevant to aspects and opinions, respectively.

4.3.2 Validity Classifiers

Due to the recall orientated nature of the candidate extraction, we are left with a large number of invalid aspect and opinion spans. To mitigate this, we introduce binary classifiers that predict the validity of each candidate span:

$$p(as) = \sigma\big(g_{as}(\tilde{h}_{as})\big)$$
$$p(op) = \sigma\big(g_{op}(\tilde{h}_{op})\big)$$

These classifiers predict the probability of each aspect and opinion candidate actually being a relevant aspect or opinion. During training, we use binary cross-entropy loss against the ground-truth labels for valid spans. At inference time, we can filter out candidates with probability greater than 0.5.

$$A' = \{as \in A \mid p(as) > 0.5\}$$
$$O' = \{op \in O \mid p(op) > 0.5\}$$

4.3.3 Biaffine Scoring Mechanism

We model the interaction between an aspect $ap \in A'$ and an opinion $op \in O'$ using a biaffine scoring function consisting of two main components.

A Linear term capturing independent contributions of aspect and opinion representations:

$$U^\top \big[\tilde{h}_{as}; \tilde{h}_{op}\big] = U_{as}^\top \tilde{h}_{as} + U_{op}^\top \tilde{h}_{op}$$

A Bilinear term modeling multiplicative interactions between aspect and opinion representations:

$$\tilde{h}_{as}^\top W \tilde{h}_{op} = \sum_{i,j} W_{i,j} \tilde{h}_{as_i} \tilde{h}_{op_j}$$

These two are put together along with a bias term to compute the final score for the pair:

$$\text{score}(as, op) = \tilde{h}_{as}^\top W \tilde{h}_{op} + U^\top \big[\tilde{h}_{as}; \tilde{h}_{op}\big] + b$$

where W captures bilinear interactions, U models linear dependencies over concatenated representations, and b is the bias term.

The bilinear term allows the model to capture multiplicative interactions between aspect and opinion representations, while the linear term provides additional flexibility for modeling independent contributions. This formulation has been shown to be effective for asymmetric relation modeling, such as dependency parsing, and is well-suited for capturing directed sentiment associations.

The probability of a pair in $A' \times O'$ being valid is then computed using a sigmoid activation for all remaining candidate pairs:

$$p(as, op) = \sigma(\text{score}(as, op))$$

During training, we supervise this using binary cross-entropy loss against the ground-truth labels for valid aspect-opinion pairs. At inference time, we can filter pairs based on a threshold (e.g., 0.5) to obtain the final set of valid pairs:

$$P = \{(as, op) \in A' \times O' \mid p(as, op) > 0.5\}$$

4.4 Category Classification with Constraint-Aware Decoding

Given a valid aspect–opinion pair $(as, op) \in P$, the next step is to assign a structured category $c \in \mathcal{E} \times \mathcal{A}$, where $\mathcal{E}$ and $\mathcal{A}$ denote the sets of entity and attribute labels, respectively. This requires modeling both the semantic content of the pair and the compatibility between entity–attribute combinations.

4.4.1 Shared Representation

Since we already have the corresponding aspect-opinion pairs, at this stage we create inputs:

$$\grave{s} = [as, \text{SEP}, op]$$

and pass them into RoBERTa to obtain a contextualized representation that captures the interaction between the aspect and opinion spans using the [CLS] token. During training the last 2 layers of the encoder are unfrozen to allow for task-specific fine-tuning, while the rest of the parameters remain frozen to preserve general language understanding.

4.4.2 Entity Prediction

The entity prediction task is modeled as a multi-class classification problem over the set of entity labels $\mathcal{E}$. We use a feed-forward neural network that takes the [CLS] representation as input and produces logits over the entity label space.

$$\widehat{en}_{\text{logits}} = D_{en}(h_{\text{CLS}})$$
$$p(en \mid \grave{s}) = \sigma\big(\widehat{en}_{\text{logits}}\big)$$

During training, we use cross-entropy loss against the ground-truth entity labels. At inference time, we select the entity with the highest predicted probability:

$$\widehat{en} = \arg\max_{en} p(en \mid \grave{s})$$

4.4.3 Attribute Prediction

The attribute prediction task is similarly modeled as a multi-class classification problem over the set of attribute labels $\mathcal{A}$. We use a separate feed-forward neural network that also takes the [CLS] representation as input to produce logits over the attribute label space.

$$at_{\text{logits}} = D_{at}(h_{\text{CLS}})$$

Not all entity-attribute combinations are valid. For example, certain attributes may only apply to specific entities. To model this, we query a lookup table with the predicted entity to retrieve a mask that is applied to the logits.

Figure 3: Illustration of the contrastive learning objective structuring the embedding space according to VA values.

$$at_{\text{masked}} = \begin{cases} at_{\text{logits}} & \text{if } M_{\widehat{en},at} = 1 \\ -\infty & \text{else} \end{cases}$$

$$p(at \mid \grave{s}, \widehat{en}) = \sigma(at_{\text{masked}})$$

During training, we use cross-entropy loss against the ground-truth attribute labels, using teacher forcing by passing in gold entities. At inference time, the prediction is made by selecting the attribute with the highest probability:

$$\hat{at} = \arg\max_{at} p(at \mid \grave{s}, \widehat{en})$$

This gives us the predicted category for the aspect-opinion pair as $\hat{ca} = (\widehat{en}, \hat{at}) \in C$.

4.5 VA Regression in Continuous Sentiment Space

Given a valid aspect–opinion pair (as, op) along with its marked input representation $\grave{s}$ from the categorization, our goal is to learn a mapping to a sentiment pair $se = (va, ar)$

4.5.1 Representation Learning

We once again use RoBERTa and pass in $\grave{s}$ to obtain a contextualized representation using the [CLS] token. Here too, the last 2 layers are unfrozen during training to allow for task-specific fine-tuning.

For regression, just this representation may not be sufficient to capture the nuanced information. To enhance the model's ability to capture affective nuances, we introduce an additional feed-forward layer that projects the [CLS] representation into a lower-dimensional embedding space:

$$z = g_{\text{proj}}(h_{\text{CLS}})$$

To encourage the embedding space to reflect the structure of the VA space, we train this head using a constrastive loss that encourages pairs with similar VA values to have nearby embeddings, while dissimilar pairs are pushed apart. This structuring of the embedding space allows the model to capture affective relationships more effectively, improving regression performance. Given a pair of samples (i, j), the contrastive loss is defined as:

$$\mathcal{L}_{\text{contrast}} = -\log \frac{\exp\big(\text{sim}(z_i, z_j)/\tau\big)}{\sum \exp(\text{sim}(z_i, z_k)/\tau)}$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity, and τ is a temperature hyperparameter. This objective induces a geometry in which distances between embeddings correspond to differences in affective states, enabling the model to learn a more structured representation of sentiment.

Upon visualizing the learned embedding space using dimensionality reduction techniques like t-SNE, we observe that samples with similar VA values cluster together. Pretrained RoBERTa embeddings seemed to have clusters of similar aspects, but finetuning with the contrastive objective reorganized the space to prioritize clustering by similar opinions.

4.5.2 Regression Objective

From the learned embedding z, we predict valence and arousal using a simple feed-forward regression head:

$$\hat{se} = (\hat{va}, \hat{ar}) = D_{va}(z)$$

During training we use mean squared error loss against the ground-truth VA scores.

5 Baselines

We evaluate the proposed pipeline against two representative baselines that capture both structured and end-to-end modeling paradigms in DimABSA.

5.1 Contrastive Learning-Enhanced Span-based Framework (CLESF)

We implement a span-based, contrastive learning framework inspired by (Tong and Wei, 2024), which represents a strong structured baseline. Since their original model is designed for Chi-

```
system_prompt = (
"You are an assistant for Aspect-Based Sentiment Analysis (ABSA). "
"Your task is to extract sentiment quadruplets from laptop reviews. "
"Each quadruplet consists of: Aspect, Category, Opinion, Valence, and Arousal.
"
"Category MUST be in the exact format 'ENTITY#ATTRIBUTE'. Choose ENTITY from:
<ENTITY LIST>"
"Choose ATTRIBUTE from: <ATTRIBUTE LIST>"
"Valence and Arousal are floating-point numbers between 1.00 and 9.00. "
"CRITICAL RULES: \n"
"1. You MUST extract 'Aspect' and 'Opinion' EXACTLY as they appear in the
original text, character-for-character.\n"
"2. INCLUDE ALL MODIFIERS. Do not truncate terms. For example, if the text says
'battery life', the Aspect must be 'battery life', NOT 'battery'. If the text
says 'very bright', the Opinion must be 'very bright', NOT 'bright'.\n"
"3. DO NOT correct spelling mistakes or typos (e.g., if the text says
'performence', output 'performence', NOT 'performance').\n"
"4. DO NOT hallucinate Categories. It MUST uniquely match one of the explicit
combos listed.\n"
"Format the output strictly as a JSON list of dictionaries like: "
'[{"Aspect": "...", "Category": "LAPTOP#GENERAL", "Opinion": "...", "Valence":
5.0, "Arousal": 5.0}]'
)
```

Figure 4: System Prompt for LLM Baseline. Huge lists have been shortened for brevity.

nese, we adapt it to the English setting by replacing MacBERT with RoBERTa-base as the underlying encoder. Candidate spans are exhaustively generated up to a maximum length of 10 tokens. Each span is represented by concatenating its start and end token embeddings, providing a compact yet expressive representation. To enhance positional awareness, Rotary Positional Embeddings (RoPE) are incorporated within the attention mechanism.

Following span generation, a pruning strategy is applied to retain the top-k most probable aspect and opinion spans. Pair representations are then constructed by combining the corresponding span embeddings along with a max-pooled representation of the intermediate context. The pipeline learns multiple objectives, including span identification, aspect–opinion pairing, sentiment regression, and category classification, using dedicated neural classifiers.

5.2 End to End LLM

As a contrasting paradigm, we employ an end-to-end large language model baseline using Qwen2-1.5B-Instruct. This model is evaluated in a zero-shot setting, without task-specific fine-tuning. We design a system prompt that explicitly defines the DimABSA task, including extraction requirements, category constraints, and the expected valence-arousal output format. The model is instructed to generate predictions as a structured JSON list of quadruplets, which are subsequently parsed for evaluation.

6 Metrics

Evaluating a staged DimABSA pipeline requires measuring performance at each intermediate step as well as in the fully composed system, since errors propagate across stages and can significantly affect downstream predictions. Accordingly, we report metrics tailored to each subtask, and distinguish between strict and relaxed matching criteria to account for span-level variability and provide a more nuanced assessment of extraction and pairing quality. In addition to stage-wise evaluation, we report end-to-end pipeline performance to quantify cumulative error and compare it against gold-pair evaluation to isolate the impact of upstream components.

6.1 Candidate Extraction

We evaluate candidate extraction using a strict lemma-head matching criterion, where spans are reduced to their head lemmas to account for surface variation.

We report precision, recall, and F1-score for both aspect and opinion candidates. To assess completeness, we use coverage, defined as:

$$\text{Coverage}_i = \begin{cases} 1 \text{ if } \text{overlap}(p_i, g_i) = 1 \\ 0 \text{ otherwise} \end{cases}$$

$$\text{Coverage} = \frac{1}{N} \sum_{i=1}^{N} \text{Coverage}_i$$

6.2 Pair Matching

We evaluate pair matching under both relaxed and strict criteria to differentiate between failures due to semantic understanding and noise in the spans.

$$\text{Precision} = \frac{|p \in P : \exists g \in G, \text{overlap}(p, g)|}{|P|}$$

$$\text{Recall} = \frac{|g \in G : \exists p \in P, \text{overlap}(p, g)|}{|G|}$$

The overlap function changes for strict and relaxed matching:

$$\text{overlap\_s}(p, g) = \begin{cases} 1 \text{ if } p = g \\ 0 \text{ otherwise} \end{cases}$$

$$\text{overlap\_r}(p, g) = (\text{span}(p_{as}) \cap \text{span}(g_{as}) \neq \emptyset) \wedge (\text{span}(p_{op}) \cap \text{span}(g_{op}) \neq \emptyset)$$

Further to test the performance of the validity heads, we report precision, recall and F1 score on the binary classification of aspects and opinions separately.

6.3 Category Classification

For category classification we report Accuracy, Precision, Recall, Macro F1 and Micro F1 scores on the entity and attribute classification tasks separately. We also report the combined scores for all theme categories together, where a prediction is only considered correct if both the entity and attribute are correctly predicted.

6.4 VA Regression

For testing valence and arousal regression performance, we report MAE and Spearman correlation for both dimensions separately. To get a combined metric, we use cF1.

$$c_{TP} = 1 - \frac{\text{dist}(p_{va}, g_{va})}{\text{max distance}}$$

From this we get cPrec, cRec, and finally cF1.

7 Results and Analysis

7.1 Candidate Extraction

As shown in Table 1, the candidate extraction stage achieves high recall across both datasets, with aspect recall exceeding 98% and opinion recall reaching up to 91.7% on the restaurant domain. This indicates that the extraction module is highly effective at capturing most relevant spans, ensuring minimal information loss for downstream stages.

However, precision is comparatively lower (60–73%), suggesting that the model generates a substantial number of spurious candidates which reflects an intentional design trade-off: prioritizing recall to maximize coverage, while delegating filtering to later stages.

Coverage remains high (94–96% for aspects), confirming that the majority of gold spans are included in the candidate set. Notably, opinion coverage is lower, particularly on the laptop dataset (76%), indicating that opinion expressions are harder to fully enumerate, likely due to their greater lexical and syntactic variability.

Data	Type	Prec	Rec	F1	Cov
L	A	60.1	96.9	70.0	96.0
	O	61.6	82.6	67.4	76.0
R	A	69.8	97.1	77.6	94.0
	O	72.9	91.7	78.1	83.5

Table 1: Candidate extraction performance. L: Laptop, R: Restaurant, A: Aspect, O: Opinion

7.2 Pair Matching

The validity classifiers achieve strong performance across both datasets, with F1 scores exceeding 85% for aspects and reaching up to 91.8% for opinions Table 2. This validates the staged design, where an over-generating extractor is followed by a discriminative filtering step.

Table 3 highlights a significant gap between strict and relaxed evaluation of matching. Under relaxed matching, F1 scores improve substantially (e.g., 58.2 -> 71.7 on laptops), demonstrating that many predicted pairs are semantically correct but misaligned at span boundaries.

Interestingly, the restaurant domain shows higher recall and coverage, suggesting that domain-specific linguistic regularities (e.g., shorter, more explicit opinion phrases) make pairing easier compared to the more diverse laptop domain.

Data	Type	Prec	Rec	F1
L	A	85.0	86.7	85.8
	O	80.2	90.2	84.9
R	A	80.1	92.6	85.9
	O	87.7	96.2	91.8

Table 2: Validity classifier performance. L: Laptop, R: Restaurant; A: Aspect, O: Opinion.

Data	Prec		Rec		F1		Cov	
	S	R	S	R	S	R	S	R
L	68.2	84.8	50.8	62.1	58.2	71.7	41.0	52.0
R	58.1	72.4	65.0	78.6	61.3	75.3	55.0	69.5

Table 3: Biaffine matcher performance. L: Laptop, R: Restaurant, S: Strict, R: Relaxed

7.3 Category Classification

Table 4 reveals a clear disparity between entity and attribute prediction. Entity classification achieves strong performance (up to 92.2 macro F1), while attribute classification is significantly weaker (as low as 21.7 macro F1).

This gap indicates that attributes are substantially more ambiguous and fine-grained, making them harder to classify reliably.

The discrepancy between macro and micro F1 suggests class imbalance, where frequent categories dominate micro scores while rare categories degrade macro performance.

Data	Type	Acc	Macro F1	Micro F1
L	E	86.4	61.7	86.4
	A	56.5	38.9	56.5
	C	49.8	25.0	49.8
R	E	96.5	92.2	96.5
	A	45.4	21.7	45.4
	C	44.4	41.5	44.4

Table 4: Categorization performance. L: Laptop, R: Restaurant; E: Entity, A: Attribute, C: Category.

7.4 VA Regression

From Table 5, the model achieves moderate regression performance, with MAE values between 0.55 and 0.73 and Spearman correlations ranging from 57.7 to 71.7.

Valence prediction is consistently more accurate than arousal, suggesting that polarity-related signals are easier to infer than intensity-related ones. This observation aligns with prior findings indicating that valence contributes more strongly and reliably to affective processing, whereas arousal is comparatively more variable and less consistently encoded

Data	Type	MAE	Spearman	cF1
L	V	0.62	71.7	—
	A	0.56	57.7	—
	C	0.59	—	58.4
R	V	0.74	69.4	—
	A	0.63	64.4	—
	C	0.68	—	59.1

Table 5: VA regression performance on dev set. L: Laptop, R: Restaurant; V: Valence, A: Arousal, C: Combined.

7.5 End-to-End Pipeline and Comparison with Baselines

Table 6 compares our staged pipeline against a structured span-based baseline (CLESF) and a zero-shot large language model (LLM) on the laptop domain.

Our approach achieves substantially higher recall for both aspects and opinions (96.9 / 82.6) compared to CLESF (69.9 / 64.3) and the LLM baseline (68.4 / 56.0). This highlights the effectiveness of our language guided candidate generation which ensures high upstream coverage and reduces the risk of missing relevant entities.

CLESF achieves higher pair F1 (74.9 vs 64.9), by adopting a precision-oriented strategy, generating cleaner but fewer pairs, while missing a significant portion of valid aspect–opinion relations. In contrast, our approach captures a broader set of candidate pairs.

Our model outperforms both baselines on attribute classification (56.5 vs 36.2 vs 14.0). While CLESF achieves slightly higher entity F1 (88.4 vs 84.5), the gap is modest, and our model remains competitive.

In valence–arousal regression, our approach achieves the lowest errors (0.54 / 0.61), outperforming the LLM baseline (1.20 / 1.63) and being on par with CLESF (0.56 / 0.65). This demonstrates that using contrastive learning to shift the embedding space directly benefits continuous sentiment prediction.

From an efficiency standpoint, our model is significantly more lightweight, requiring 40% less VRAM than CLESF and nearly 3x faster inference, while vastly outperforming the LLM baseline, which has abysmal latency (1586 ms) and memory usage. This makes our approach more suitable for practical deployment scenarios where both performance and efficiency are critical.

7.6 Qualitative Analysis

To better understand model behavior, we examine specific cases from the dev set.

- **Success Case:** "The warranty is no good on this and you are going to need the warranty". The model correctly extracted the aspect "warranty" and paired it with the opinion "no good", predicted the correct entity WARRANTY, and estimated VA scores (V=2.79, A=5.82) very close to gold (V=2.80, A=5.80).
- **Failure Case 1 (Categorization):** "Great clear picture and nice sound", the pair (sound, nice) was incorrectly categorized as MEDIA#OPERATION instead of the gold MEDIA#GENERAL. This error reflects the fine-grained nature of attribute classification.
- **Failure Case 2 (VA Regression):** "This is a great laptop for media work, but the fans are super huge in this thing which makes the body very thick and bulky" shows a VA

Model	Asp R	Op R	Pair F1	Ent F1	Att F1	V MAE	A MAE	cF1	VRAM	Inf (ms)
Ours (L)	**96.9**	**82.6**	64.9	84.5	**56.5**	**0.54**	**0.61**	49.2	**1392**	**39.5**
CLESF (L)	69.9	64.3	**74.9**	**88.4**	36.2	0.56	0.65	**52.5**	2335	165
LLM (L)	68.4	56.0	54.3	41.4	14.0	1.20	1.63	38.8	3915	1586
Ours (R)	97.1	91.7	65.8	88.7	46.8	0.68	0.61	56.4	1392	45.9

Table 6: Pipeline evaluation results on dev set. L: Laptop, R: Restaurant; Asp: Aspect, Op: Opinion, Ent: Entity, Att: Attribute, V: Valence, A: Arousal. VRAM in MB, Inf: Inference time per sentence in ms.

regression failure. For the pair (fans, super huge), the model predicted V=7.1, A=7.8 while gold is V=3.8, A=7.2—a valence error of 3.4. The opinion "super huge" describes size (could be positive for a large trackpad, negative for bulky fans), and the model incorrectly interpreted it as positive sentiment.

More failure and success cases are included in our Github repo.

8 Conclusion

We presented LG-DimABSA, a language-grounded staged pipeline for dimensional aspect-based sentiment analysis that decomposes the task into interpretable components. By separating extraction, pairing, categorization, and regression, the approach offers improved transparency while remaining competitive with end-to-end models.

Our results show that high-recall candidate generation combined with structured filtering and biaffine modeling significantly improves coverage and robustness. Additionally, contrastive learning effectively structures the embedding space, leading to better valence–arousal prediction.

We identify attribute classification and arousal modeling as key challenges, and note that the relatively small dataset limits generalization and performance, highlighting the need for larger and more diverse benchmarks. Overall, our findings demonstrate the effectiveness of modular, linguistically grounded pipelines for structured sentiment analysis.

References

Timothy Dozat and Christopher D. Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. arXiv: 1611.01734 [cs.CL].

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. *Zenodo*.

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.

Xin Kang, Zhifei Zhang, Jiazheng Zhou, Yunong Wu, Xuefeng Shi, and Kazuyuki Matsumoto. 2024. TMAK-Plus at SIGHAN-2024 dimABSA Task: Multi-Agent Collaboration for Transparent and Rational Sentiment Analysis. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 88–95, Bangkok, Thailand.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing What, How and Why: A Near Complete Solution for Aspect-Based Sentiment Analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

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

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.

Lee, Wang, Wahle, Ruas, Chang Yu and Mohammad. 2025. Task:3 Dimensional Aspect-Based Sentiment Analysis (DimABSA). *Semeval 2026*.

Xu, Hongling, Zhang, Delong, Xu, Ruifeng Zhang. 2024. A Hybrid Approach to Dimensional Aspect-Based Sentiment Analysis Using BERT and Large Language Models. *Electronics* 13:3724.