Analytical Report: Instruct-DeBERTa for Aspect-Based Sentiment Analysis

Ali Sedaghi (Student ID: 40409844)
Based on the paper by Jayakody et al. (2024)

February 6, 2026

1 Main Problem and Importance

1.1 The Problem

Traditional sentiment analysis methods operate at the sentence or document level, assigning a single sentiment polarity (positive, negative, or neutral) to an entire text. This approach fails to capture the complexity of real-world feedback where a user may express conflicting opinions about different features within the same sentence (e.g., *"The food was excellent, but the service was slow"*).

1.2 Why It Is Important

Aspect-Based Sentiment Analysis (ABSA) is critical because it provides fine-grained insights. By extracting specific aspects (e.g., "food," "service") and determining the sentiment for each, businesses can pinpoint exact areas for improvement rather than relying on a vague overall score. This granularity is essential for actionable business intelligence, customer satisfaction enhancement, and targeted product development.

2 Inputs and Outputs

The system is designed as a pipeline that processes raw text and outputs structured sentiment data.

- **Input:** A raw textual review sentence (e.g., *"The battery life is amazing but the screen is too dim."*).
- **Output:** A set of aspect-sentiment pairs.
  - **Aspect Terms:** The specific features mentioned (e.g., *battery life*, *screen*).
  - **Sentiment Polarity:** The sentiment classification for each aspect (e.g., *Positive*, *Negative*).

3 Data Used

The paper evaluates the model using standard benchmarks in the ABSA domain to ensure comparability and reproducibility.

- **Source:** SemEval datasets from 2014, 2015, and 2016.
- **Domains:**
  - **Restaurant Reviews:** SemEval 2014 (Res-14), 2015 (Res-15), and 2016 (Res-16).
  - **Laptop Reviews:** SemEval 2014 (Lap-14).
- **Type:** Labeled sentences where aspect terms and their polarities are manually annotated.
- **Implementation Data:** For our reproduction experiments, we utilized a representative subset of the SemEval Restaurant dataset and a custom Laptop domain dataset to verify domain robustness.

4 Proposed Method

The paper proposes **Instruct-DeBERTa**, a hybrid pipeline that leverages the strengths of two distinct State-of-the-Art (SOTA) Transformer models. It decomposes the ABSA task into two pipelined sub-tasks: Aspect Term Extraction (ATE) and Aspect Sentiment Classification (ASC).

4.1 Methodology Schematic

The pipeline consists of two sequential steps utilizing distinct specialized models:

Input Review
↓
Step 1: Aspect Term Extraction (ATE)
Model: InstructABSA (Tk-Instruct)
Logic: Instruction Tuning (Prompt-based generation)
↓
Extracted Aspects (e.g., "food", "service")
↓
Step 2: Aspect Sentiment Classification (ASC)
Model: DeBERTa-V3-base-absa-v1
Logic: Contextual Classification (Sentence + Aspect Pair)
↓
Final Output: {("food", Positive), ("service", Negative)}

Figure 1: The Instruct-DeBERTa Hybrid Pipeline

4.2 Pipeline Steps

1. **Aspect Term Extraction (ATE):** Uses **InstructABSA**. This model is trained with instruction tuning. It takes the review sentence (wrapped in a prompt definition) and generates the aspect terms as a text sequence. InstructABSA showed superior performance in extraction tasks (F1 score > 92%).
2. **Aspect Sentiment Classification (ASC):** Uses **DeBERTa-V3-base-absa-v1**. The extracted aspects from Step 1 are paired with the original sentence. This pair is fed into the DeBERTa model, which classifies the sentiment of the aspect within that specific context.

4.3 Pseudocode

```python
# Algorithm: Instruct-DeBERTa Hybrid Pipeline

def predict_absa(review_text):
    # Step 1: Aspect Term Extraction (ATE) using InstructABSA
    # Input format requires specific instruction prompt
    prompt = f"Definition: Extract aspect terms... Input: {review_text}"
    extracted_terms = InstructABSA.generate(prompt)

    results = []

    # Step 2: Aspect Sentiment Classification (ASC) using DeBERTa
    for aspect in extracted_terms:
        # Classify sentiment for the specific aspect in the context of the review
        sentiment = DeBERTaV3.classify(text=review_text, text_pair=aspect)
        results.append((aspect, sentiment))

    return results
```

Listing 1: Instruct-DeBERTa Hybrid Pipeline Logic

5 Main Results, Limitations, and Future Work

5.1 Main Results

- **SOTA Performance:** The hybrid Instruct-DeBERTa model achieved the highest F1 scores for the joint task (extraction + classification) on both the SemEval Restaurant 2014 and Laptop 2014 datasets.
- **Robustness:** The model demonstrated consistent performance across different domains (Hospitality and Tech), outperforming other hybrid models and single-task baselines.
- **Experimental Verification:** In our code reproduction, the model successfully handled mixed-sentiment sentences (e.g., positive food, negative service) where the baseline sentence-level model failed.

5.2 Limitations

- **Pipeline Error Propagation:** Since the model is a pipeline, errors in the first step (Extraction) cannot be corrected in the second step (Classification). If an aspect is missed by InstructABSA, DeBERTa never gets a chance to classify it.
- **Prompt Sensitivity:** Experiments revealed that the ATE component (InstructABSA) is highly sensitive to the formatting of the input prompt. Without the correct "Definition/Input" structure, the model tends to hallucinate or repeat the input.
- **Computational Cost:** Running two distinct transformer models for every prediction is computationally more expensive than a single end-to-end model.

5.3 Ideas for Future Work

- **End-to-End Fine-Tuning:** Investigating a unified model that performs both tasks simultaneously to reduce inference latency.
- **Multilingual Adaptation:** Expanding the model to support low-resource languages by replacing the backbone models with multilingual variants (e.g., mDeBERTa).
- **Decoding Strategy Optimization:** Further exploration of constrained beam search to ensure the output strictly adheres to the requested format.