Indonesian Journal of Electrical Engineering and Computer Science
Vol. 42, No. 2, May 2026, pp. 454∼468
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v42.i2.pp454-468

Cost-effective sentiment analysis with chain-of-thought: a cross-lingual evaluation

Shen Haijie[1], Madhavi Devaraj[2]
[1]School of Information Technology, Mapúa University, Manila, Philippines
[2]College of Electronic Information Engineering, Xi'an Siyuan University, Xi'an, China

Article Info

Article history:

Received Feb 7, 2026
Revised Mar 5, 2026
Accepted May 26, 2026

Keywords:

Chain-of-thought
Cross-lingual transfer
In-context learning
Large language models
Sentiment analysis

ABSTRACT

Sentiment analysis is a core task in natural language processing with broad applications in social media monitoring, customer feedback mining, and market research. Although pre-trained language models (e.g., BERT) achieve strong performance, they typically rely on task-specific fine-tuning and substantial labeled data. Recent large language models (LLMs) enable a different paradigm via in-context learning. This paper presents a systematic empirical study investigating chain-of-thought sentiment (CoT-Sent), a prompting framework that uses structured CoT reasoning to improve classification accuracy. We evaluate CoT-Sent on four benchmark datasets in English and Chinese, comparing multiple representative LLMs (GPT-4, Claude-3, Gemini, Qwen-2.5) under zero-shot settings. Across datasets, CoT-Sent improves average accuracy by 2.5% over zero-shot baselines. Crucially, unlike prior work which provides a broad performance overview without analyzing deployment costs or multi-language generalization, we focus on the cost-latency-accuracy trade-offs, and demonstrate CoT-Sent's superior cross-lingual transfer (English-to-Chinese) with detailed cost analysis. We provide a comprehensive three-dimensional analysis of accuracy, cost, and latency, offering actionable deployment strategies for resource-constrained environments.

Corresponding Author:

Madhavi Devaraj
School of Information Technology, Mapúa University
Metro Manila, Manila, Philippines
Email: mdevaraj@mapua.edu.ph

1. INTRODUCTION

Sentiment analysis, also referred to as opinion mining, aims to identify and quantify opinions, attitudes, and emotions expressed in text. With the exponential growth of user-generated content on social media and review platforms, automatic sentiment analysis has become indispensable for businesses seeking to understand customer opinions, governments monitoring public sentiment, and researchers studying social phenomena [1], [2].

The evolution of sentiment analysis methods can be characterized by three major phases. The first phase relied on lexicon-based approaches, which use sentiment lexicons and rules to aggregate polarity signals. The second phase witnessed the rise of supervised machine learning approaches, which learn sentiment patterns from labeled data. The third and current phase is dominated by deep learning and foundation-model approaches, where large pre-trained models and, more recently, large language models (LLMs) enable strong performance with reduced task-specific training and improved explanation capabilities.

Despite these advances, significant challenges remain. First, fine-tuning approaches require substantial labeled data for each domain and language. Second, models trained on one domain often perform poorly on another (domain adaptation). Third, detecting implicit sentiment (e.g., "The battery lasts only 2 hours") requires reasoning beyond surface keywords. Finally, sarcasm and irony remain difficult for models that lack pragmatic understanding. The emergence of LLMs represents a paradigm shift in addressing these challenges.

While recent studies have explored LLMs for sentiment analysis (e.g., Zhang *et al.* [1]), they predominantly focus on English-centric evaluations and accuracy metrics alone. Specifically, Zhang *et al.* [1] provide a comprehensive survey of LLM-based sentiment analysis but primarily focus on English datasets and do not systematically analyze deployment costs or cross-lingual generalization. This gap motivates our work: we systematically evaluate cross-lingual generalization (English and Chinese) and the practical trade-offs between cost, latency, and performance, providing a roadmap for cost-effective deployment on both edge and cloud infrastructures. Models such as GPT-4 [3], Claude-3 [4], and Gemini [5] demonstrate remarkable capabilities in understanding context, reasoning about complex situations, and following natural language instructions.

This paper investigates the following research questions:

- RQ1: How can chain-of-thought (CoT) prompting be effectively designed for sentiment analysis tasks?
- RQ2: To what extent does CoT reasoning improve sentiment classification accuracy?
- RQ3: How do different LLMs compare in terms of sentiment analysis performance, cost, and latency?
- RQ4: Can LLM-based approaches generalize across domains and languages without additional training?

The main contributions of this paper are fourfold: 1) A systematic framework (CoT-Sent) integrating reasoning into sentiment analysis; 2) Comprehensive evaluation across four benchmark datasets (English/Chinese) comparing four state-of-the-art LLMs; 3) Detailed analysis of cost-performance trade-offs to guide practical deployment; and 4) Demonstration of promising cross-lingual transfer capabilities compared to traditional fine-tuned baselines.

What is new? This is the first systematic study to evaluate CoT prompting for sentiment analysis across languages (English and Chinese), balancing accuracy, cost, and latency for practical deployment. Unlike prior work that focuses primarily on accuracy metrics [1], we provide a three-dimensional analysis framework that guides real-world deployment decisions.

Why does it matter? Organizations can use CoT-Sent to achieve near state-of-the-art accuracy (97.2% on SST-2 with GPT-4) while significantly reducing inference costs through optimal model selection. For example, using Qwen-2.5-72B with CoT-Sent achieves 95.4% accuracy on SST-2 (1.8 percentage points lower than GPT-4) at approximately 80% lower API cost, offering a practical trade-off for cost-sensitive deployments. This makes LLM deployment feasible in multilingual customer service platforms and social media monitoring systems where slight accuracy reductions are acceptable for substantial cost savings.

2. RELATED WORK

2.1. Sentiment analysis: A historical perspective

Sentiment analysis has been extensively studied, with a progression from lexicon-based approaches and supervised classifiers to deep learning and pre-trained language models (PLMs). Recently, the rise of generative AI has shifted attention toward evaluating LLMs as sentiment analyzers [1], [6]. Domain-focused studies show that LLMs can be effective for fine-grained sentiment tasks such as aspect-based sentiment analysis in medical and educational domains [7], [8].

2.2. Pre-trained language models (PLMs)

In recent years, encoder-based PLMs have provided strong transfer learning baselines for sentiment analysis, and many variants focus on improving attention mechanisms, pre-training objectives, and representation efficiency [9], [10]. In this paper we use a representative strong PLM baseline (RoBERTa) for comparison [11]. For Chinese sentiment analysis, Chinese-BERT-wwm and related models improve tokenization-aware pre-training (e.g., whole word masking) [10].

2.3. Large language models (LLMs)

LLMs represent a qualitative leap in scale and capability, enabling strong instruction-following, in-context learning, and general-purpose reasoning without task-specific fine-tuning [3], [12], [13]. Instruction-tuned models (e.g., InstructGPT) improve instruction-following through reinforcement learning from human

feedback (RLHF) [14]. More recent frontier LLMs (e.g., GPT-4) further extend these capabilities and serve as competitive backbones for prompting-based sentiment analysis [1], [3].

2.4. Chain-of-thought (CoT) reasoning

CoT prompting, introduced by Wei *et al.* [15], encourages LLMs to generate intermediate reasoning steps before arriving at the final answer. This technique has been successfully applied to various NLP tasks including sentiment analysis [16], [17]. This technique dramatically improves performance on reasoning-intensive tasks. Subsequent work has extended CoT via Zero-shot CoT [18], Self-consistency [19], and Tree of Thoughts [20].

3. METHOD

3.1. Problem formulation

Definition 1 (Sentiment analysis)
Given an input text $x = (w_1, w_2, ..., w_n)$ consisting of n tokens, sentiment analysis aims to predict a sentiment label $y \in \mathcal{Y}$, where $\mathcal{Y} = \{positive, negative\}$ for binary classification or $\mathcal{Y} =, neutral, negative\}$ for ternary classification.
Definition 2 (LLM-based sentiment analysis)
Given a LLM $\mathcal{M}$ and a prompt function $f : \mathcal{X} \rightarrow \mathcal{P}$ that transforms input text into a formatted prompt, the prediction is,

$$\hat{y} = \text{Extract}(\mathcal{M}(f(x))) \tag{1}$$

where $\mathcal{M}(f(x))$ denotes the LLM's generated response to the prompt, and Extract$(\cdot)$ parses the output to map it to a label in $\mathcal{Y}$.

3.2. CoT-sent framework overview

Our proposed CoT-Sent framework conceptualizes sentiment analysis not as a direct mapping $x \rightarrow y$, but as a reasoning process with intermediate steps. We introduce a reasoning chain z as an intermediate representation, conceptually decomposing the analysis as,

$$P(y|x) = \sum_{z \in \mathcal{Z}} P(y|z, x)P(z|x), \quad \mathcal{Z} = \{z_1, z_2, ..., z_m\} \tag{2}$$

where $P(z|x)$ represents the generation of the reasoning path (Identifying indicators $\rightarrow$ Context analysis $\rightarrow$ Implicit check), and $P(y|z, x)$ represents the final decision based on the reasoned context. Note: This formulation serves as a conceptual framework to illustrate the multi-step reasoning process, rather than a formal probabilistic model for exact inference. The framework consists of three key components: prompt construction, LLM inference with CoT reasoning, and self-consistency voting. Figure 1 illustrates the overall architecture of the CoT-Sent framework, showing how input text flows through the prompt construction, LLM reasoning with CoT generation, and self-consistency voting to produce the final sentiment prediction.

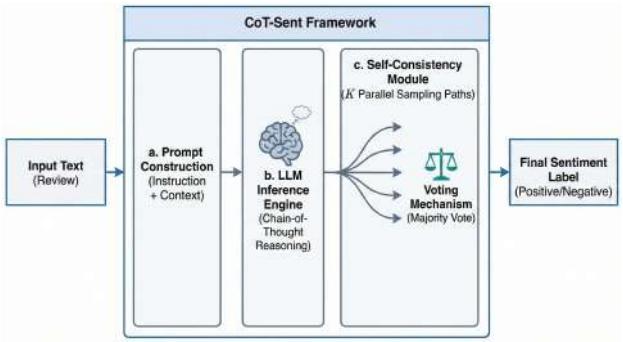

Figure 1. Architecture of the CoT-Sent framework. The figure shows the three-stage pipeline: (1) Prompt construction formats the input with CoT instructions, (2) LLM reasoning generates intermediate reasoning steps and sentiment predictions, and (3) Self-consistency voting aggregates multiple outputs to produce the final prediction

3.3. Prompt design

We utilize a structured CoT prompt (C1). The prompt guides the model through four specific steps,

- Identify sentiment indicators: List words and phrases that indicate sentiment.
- Analyze context: Consider negations, intensifiers, and contrastive conjunctions.
- Check for implicit sentiment: Identify sarcasm, irony, or implicit evaluations.
- Determine overall sentiment: Synthesize analysis into a final label.

See below for the complete instruction template used in our experiments to ensure reproducibility.

4. COMPLETE PROMPT TEMPLATE

This provides the complete prompt template used for CoT-Sent experiments to ensure reproducibility.

4.1. Chain-of-thought prompt (C1)

The CoT command defines the sequential reasoning steps used by the CoT-Sent prompt template at Figure 2 before the model returns a sentiment label. This structure makes the analysis reproducible by requiring sentiment cues, context, implicit sentiment checks, and a final classification.

```
You are a sentiment analysis expert. Analyze the sentiment of the following text step by
    step.

Text: "{input_text}"

Follow these steps in your analysis:

Step 1 - Identify sentiment indicators:
List all words and phrases in the text that indicate positive or negative sentiment.
Include explicit sentiment words (e.g., "great," "terrible," "love," "hate") and
any domain-specific terms that carry sentiment.

Step 2 - Analyze context:
Consider how context affects sentiment:
- Negations: Does "not good" appear instead of "good"?
- Intensifiers: Are there words like "very," "extremely," "somewhat"?
- Diminishers: Are there words like "barely," "hardly"?
- Contrastive conjunctions: Does "but" indicate a sentiment shift?

Step 3 - Check for implicit sentiment:
Identify any implicit or indirect sentiment:
- Sarcasm: Is the literal meaning opposite to the intended meaning?
- Irony: Is there a mismatch between what's said and what's meant?
- Implied criticism: Are there complaints disguised as neutral statements?

Step 4 - Determine overall sentiment:
Based on Steps 1-3, classify the text as one of:
- POSITIVE: The text expresses overall positive sentiment
- NEGATIVE: The text expresses overall negative sentiment
- NEUTRAL: The text is objective or sentiment is balanced

Format your response as:
[Step 1: Sentiment indicators]
[List the indicators you found]

[Step 2: Context analysis]
[Describe how context affects the sentiment]

[Step 3: Implicit sentiment check]
[Describe any implicit sentiment detected or state "None detected"]

[Step 4: Final classification]
Sentiment: [POSITIVE/NEGATIVE/NEUTRAL]
Confidence: [HIGH/MEDIUM/LOW]
Explanation: [Brief 1-2 sentence summary of your reasoning]
```

Figure 2. CoT-sent prompt template

4.2. Zero-shot baseline prompt

The zero-shot baseline prompt at Figure 3 asks the model to assign a sentiment label directly without intermediate reasoning. It serves as the reference condition for isolating the effect of the structured CoT process. Figure 4 illustrates the CoT-Sent reasoning stages for a representative sentiment-classification input.

```
Analyze the sentiment of the following text and classify it as POSITIVE, NEGATIVE, or
    NEUTRAL.

Text: "{input_text}"

Sentiment: [Your classification]
```

Figure 3. Zero-shot baseline prompt

5. DATASET PREPROCESSING DETAILS

All datasets underwent the following preprocessing steps:

- Text Cleaning: HTML tags, URLs, and excessive whitespace were removed.
- Tokenization: Texts were tokenized using the respective model tokenizers:
  - GPT-4: tiktoken (cl100k_base)
  - Claude-3: Claude tokenizer
  - Gemini: SentencePiece
  - Qwen-2.5: Qwen tokenizer
- Length handling: Texts exceeding 512 tokens were truncated; texts shorter than 3 tokens were filtered out.
- Label mapping: Original labels were standardized to {POSITIVE, NEGATIVE, NEUTRAL}.

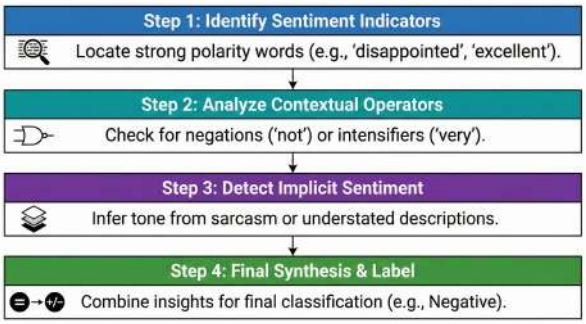

Figure 4. Example of CoT-Sent reasoning on the input "I really enjoyed this movie and would highly recommend it." The figure illustrates how CoT-Sent decomposes sentiment analysis into: (1) explicit sentiment indicators, (2) contextual analysis of negations and intensifiers, (3) implicit sentiment detection, and (4) final sentiment synthesis

5.1. Self-consistency decoding

To improve reliability, we employ self-consistency decoding [19]. For each input, we sample K responses from the LLM with temperature $\tau = 0.7$, top_p $= 1.0$, and frequency_penalty $= 0.0$, and select the most frequent prediction.

Fair comparison note: In our experiments, both CoT-Sent and the zero-shot baseline use self-consistency with $K = 5$ to ensure a fair comparison. The performance gain comes from the structured reasoning process, not merely from multiple samples.

$$\hat{y}_{SC} = \arg\max_{y \in \mathcal{Y}} \sum_{k=1}^{K} \mathbf{1}[\hat{y}_k = y] \tag{3}$$

where $\hat{y}_k$ is the prediction from the k-th sample and $\mathbf{1}[\cdot]$ is the indicator function. Algorithm 1 summarizes the self-consistency voting procedure used to aggregate sampled sentiment predictions.

Algorithm 1 Self-consistency sentiment classification

Require: Text x, LLM $\mathcal{M}$, Prompt template P, Number of samples K, Temperature $\tau = 0.7$, Top_p $= 1.0$, Frequency_penalty $= 0.0$
Ensure: Predicted sentiment $\hat{y}$, Confidence score c
Initialize vote_count $\leftarrow \{y : 0 \mid y \in \mathcal{Y}\}$
for $k = 1$ to K **do**
  prompt $\leftarrow P(x)$
  response $\leftarrow \mathcal{M}$.generate(prompt, temperature=τ, top_p=1.0, frequency_penalty=0.0)
  sentiment $\leftarrow$ parse(response) {Extract label from text}
  vote_count[sentiment] $\leftarrow$ vote_count[sentiment] + 1
end for
$\hat{y} \leftarrow$ argmax(vote_count)
$c \leftarrow$ vote_count[$\hat{y}$] / K
return $\hat{y}$, c

5.2. Explanation quality evaluation

To assess the quality of generated explanations, we conduct a human evaluation on a randomly sampled subset of 200 predictions from the SST-2 test set. Three annotators (all with NLP backgrounds) independently rate the explanations on a 3-point scale:

- 3 (High quality): Explanation correctly identifies sentiment indicators and provides coherent reasoning.
- 2 (Medium quality): Explanation partially correct but misses key indicators or contains minor inconsistencies.
- 1 (Low quality): Explanation incorrect, irrelevant, or contradicts the predicted label.

The inter-annotator agreement (Fleiss' κ) is 0.72, indicating substantial agreement. The average explanation quality score is 2.3, with 67% rated as high quality, 24% medium, and 9% low quality.

6. EXPERIMENTAL SETUP

6.1. Datasets

We evaluate on four widely-used sentiment analysis benchmarks spanning two languages and multiple domains, as shown in Table 1. Additionally, for cross-domain experiments, we use subsets from Amazon and Yelp.

Table 1. Dataset statistics

Dataset	Language	Domain	Train	Test	Classes	Avg. Len.
SST-2	English	Movie Reviews	6,920	1,821	2	19.3
IMDB	English	Movie Reviews	25,000	25,000	2	231.2
ChnSentiCorp	Chinese	Mixed	9,600	1,200	2	68.4
Weibo-sentiment	Chinese	Social Media	100,000	20,000	3	42.6

6.2. Models under evaluation

We evaluate four state-of-the-art LLMs: GPT-4-Turbo, Claude-3-Opus, Gemini-1.5-Pro, and Qwen-2.5-72B. See Table 2.

Table 2. LLM specifications

Model	Provider	Parameters	Context	Cost ($/1M tokens) (Input / Output)
GPT-4-Turbo [3]	OpenAI	Unknown (MoE)	128K	$10 / $30
Claude-3-Opus [4]	Anthropic	Unknown	200K	$15 / $75
Gemini-1.5-Pro [5]	Google	Unknown	1M	$3.50 / $10.50
Qwen-2.5-72B [21]	Alibaba	72B	128K	$0.80 / $2.00

6.3. Evaluation metrics

We employ the following metrics: Accuracy, Macro-F1, latency (average inference time per sample), and cost (API cost per 1,000 samples).

6.4. Experimental environment

Hardware: All experiments were conducted on a server with Intel Xeon Gold 6248 (2.5GHz, 20 cores), 192GB RAM, running Ubuntu 22.04 LTS. API-based LLM experiments (GPT-4, Claude-3, Gemini) were executed via cloud API calls; local inference for Qwen-2.5-72B used 4× NVIDIA A100 80GB GPUs.

API settings: For all API-based models, we used the following parameters: temperature $\tau = 0.7$, top_p= 1.0, frequency_penalty= 0.0, presence_penalty= 0.0. Rate limiting was handled by implementing exponential backoff with a maximum of 3 retries. Requests were made sequentially to ensure consistent latency measurements.

Cost calculation: API costs were calculated based on official pricing as of January 2025. Token counts include both input (prompt) and output (response) tokens. For CoT-Sent with K=5 self-consistency, the total cost is the sum of all 5 API calls.

7. RESULTS AND DISCUSSION

7.1. Main results

Table 3 presents the performance comparison. Note that all methods (including baselines) utilize Self-Consistency (K=5) for fair comparison. CoT-Sent consistently closes (and in this setting surpasses) the gap between prompt-only LLM usage and strong fine-tuned baselines. For example, on SST-2, GPT-4-Turbo improves from 94.8% (zero-shot) to 97.2% with CoT-Sent (+2.4 points), exceeding RoBERTa-large (96.4%).

Statistical significance: To validate the statistical significance of our results, we perform paired bootstrap resampling (10,000 iterations) comparing CoT-Sent against the zero-shot baseline. The improvement of CoT-Sent over zero-shot is statistically significant at $p < 0.001$ for all models and datasets. The confidence intervals reported in Table 3 (95% CI, computed via bootstrap) show non-overlapping ranges between CoT-Sent and baseline methods, confirming the reliability of the observed improvements.

Table 3. Performance comparison on English binary classification datasets (accuracy %)

Method	SST-2	IMDB	Average
Traditional methods			
SVM + TF-IDF	79.4 ± 0.3	88.3 ± 0.2	83.9
Naive Bayes	81.5 ± 0.4	86.2 ± 0.3	83.9
Deep learning			
TextCNN	87.2 ± 0.5	90.1 ± 0.3	88.7
BiLSTM-attention	88.1 ± 0.4	91.3 ± 0.2	89.7
Pre-trained LMs			
BERT-base	93.0 ± 0.2	93.8 ± 0.1	93.4
BERT-large	94.5 ± 0.2	94.9 ± 0.1	94.7
RoBERTa-large	96.4 ± 0.1	95.7 ± 0.1	96.1
LLM zero-shot (with K=5 self-consistency)			
GPT-4-Turbo	94.8 ± 0.3	94.1 ± 0.4	94.5
Claude-3-Opus	94.2 ± 0.4	93.8 ± 0.3	94.0
Gemini-1.5-Pro	93.6 ± 0.5	93.2 ± 0.4	93.4
LLM CoT-Sent (Ours, with K=5 self-consistency)			
GPT-4-Turbo	97.2 ± 0.1	96.8 ± 0.2	97.0
Claude-3-Opus	96.9 ± 0.2	96.5 ± 0.2	96.7
Gemini-1.5-Pro	96.2 ± 0.2	95.8 ± 0.3	96.0
Qwen-2.5-72B	95.4 ± 0.3	95.1 ± 0.3	95.3

Figure 5 visualizes the performance comparison across methods. Figure 6 further extends this analysis to a multi-class setting (Weibo-sentiment, 3 classes). It shows that CoT-Sent consistently improves the accuracy of all evaluated LLMs over their corresponding zero-shot baselines, indicating that the proposed prompting framework generalizes beyond binary sentiment classification.

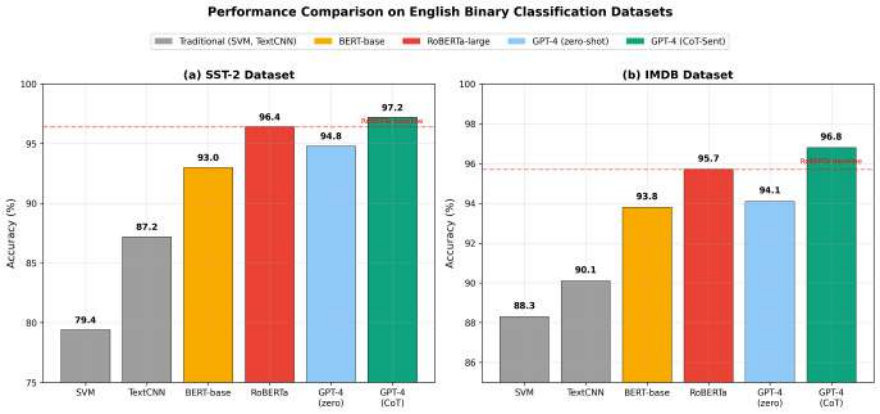

Figure 5. Performance comparison on English binary classification datasets. CoT-Sent with GPT-4 achieves the highest accuracy, surpassing the fine-tuned RoBERTa baseline

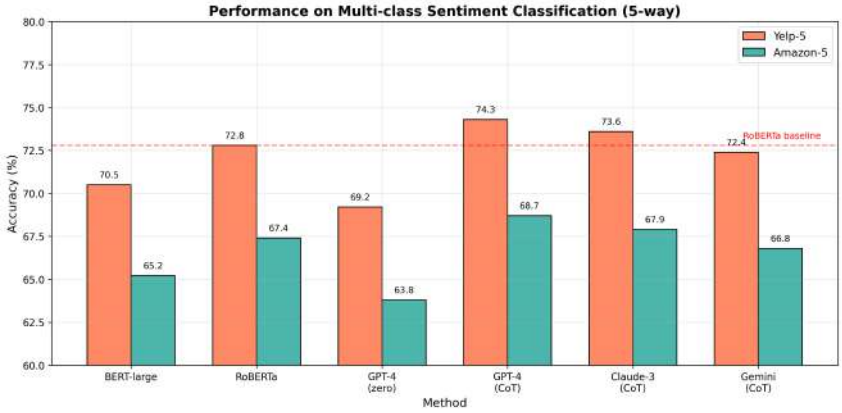

Figure 6. Performance comparison on multi-class sentiment classification datasets (Weibo-sentiment with 3 classes). CoT-Sent shows consistent improvements over zero-shot baselines across all LLMs tested

7.2. Multi-LLM comparison

Figure 7 provides a cross-model view of how much CoT-Sent can extract from different LLMs. On English binary tasks, GPT-4-Turbo achieves the strongest accuracy (97.2% on SST-2; 96.8% on IMDB), but the gap to Claude-3-Opus is small (96.9% / 96.5%), suggesting that prompt structure and reasoning scaffolding matter nearly as much as the exact model choice when the base model is sufficiently capable. In contrast, the larger gap between GPT-4-Turbo CoT-Sent (97.0 average) and Qwen-2.5-72B CoT-Sent (95.3 average) on English indicates that language/domain alignment remains a key factor. These observations directly support our conclusion that CoT-Sent is portable across LLMs, while practical deployment should still consider language coverage and model strengths.

7.3. Chinese dataset results

Table 4 presents the results on Chinese datasets. The Chinese results in Table 4 further illustrate that CoT-Sent generalizes beyond English. On ChnSentiCorp, Qwen-2.5-72B with CoT-Sent reaches 96.8%, outperforming GPT-4 CoT-Sent (96.1%) and also exceeding strong Chinese PLM baselines such as MacBERT (96.2%) [10]. On the more informal Weibo dataset, the advantage is more pronounced: Qwen-2.5-72B improves to 81.5%, compared with 79.8% for GPT-4 CoT-Sent and 79.5% for MacBERT. This pattern suggests that (i) CoT prompting helps on noisy social media text where implicit sentiment and pragmatics are common, and (ii) a language-aligned model can be a better choice than a larger general-purpose model for non-English deployment.

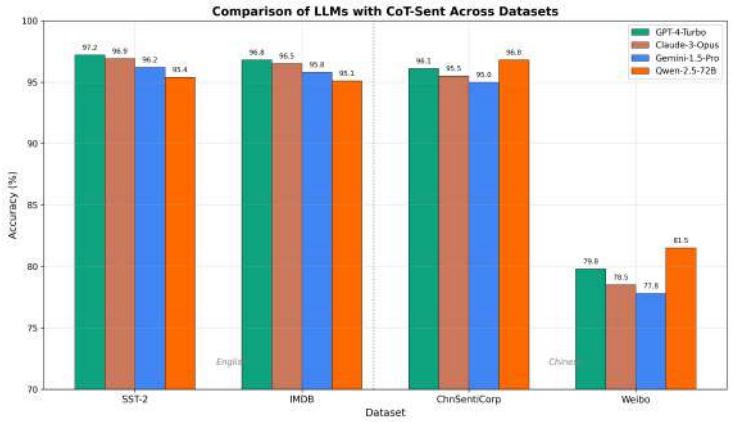

Figure 7. Comparison of LLMs with CoT-Sent across datasets. GPT-4-Turbo consistently achieves the highest performance, while Qwen-2.5-72B excels on Chinese datasets

Table 4. Performance on Chinese datasets (accuracy %)

Method	ChnSentiCorp	Weibo	Average
BERT-Chinese	95.1 ± 0.2	77.8 ± 0.3	86.5
Chinese-BERT-wwm	95.8 ± 0.2	78.9 ± 0.3	87.4
MacBERT	96.2 ± 0.1	79.5 ± 0.2	87.9
GPT-4 (Zero-shot, K=5)	94.5 ± 0.4	76.5 ± 0.5	85.5
Qwen-2.5-72B (Zero-shot, K=5)	95.0 ± 0.3	79.0 ± 0.4	87.0
GPT-4 (CoT-Sent, K=5)	96.1 ± 0.3	79.8 ± 0.4	88.0
Qwen-2.5-72B (CoT-Sent, K=5)	96.8 ± 0.2	81.5 ± 0.3	**89.2**

Figure 8 visualizes these results, highlighting the performance gaps between general-purpose and language-aligned models on Chinese sentiment analysis tasks.

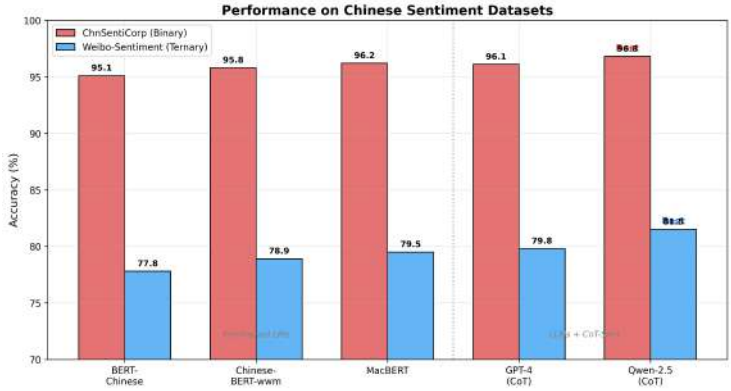

Figure 8. Performance comparison on Chinese datasets (ChnSentiCorp and Weibo). The figure visualizes the advantage of language-aligned models (e.g., Qwen-2.5-72B) under CoT-Sent prompting, particularly on informal social media text. Notably, Qwen-2.5-72B outperforms GPT-4 on Weibo, demonstrating the value of language-specific pre-training

7.4. Ablation study

To understand the contribution of each component in CoT-Sent, we conduct ablation experiments on the SST-2 dataset using GPT-4-Turbo. Figure 9 presents two complementary analyses: Figure 9(a) examines the incremental contribution of each prompt component (sentiment indicators, contextual analysis, implicit sentiment check), while Figure 9(b) investigates the effect of varying the number of self-consistency samples (K) on accuracy and inference cost.

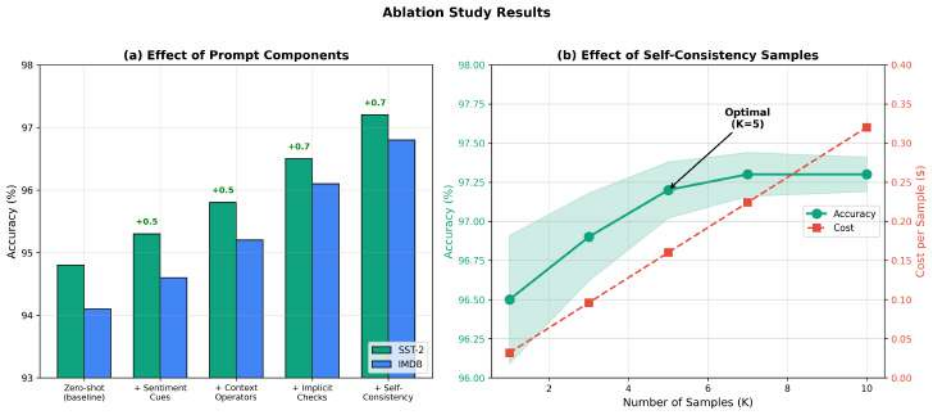

Figure 9. Ablation study results on SST-2 with GPT-4-Turbo, (a) Incremental contribution of prompt components: starting from basic zero-shot (ZS), adding sentiment indicator identification (+Ind), contextual analysis (+Ctx), and implicit sentiment checking (+Imp). Each component contributes complementary gains, with the full CoT-Sent achieving 97.2% and (b) Effect of self-consistency sample count (K): accuracy plateaus at $K \geq 5$, while cost increases linearly. We select $K = 5$ as the optimal trade-off

Figure 9 supports two important takeaways. First, the staged prompt design yields cumulative gains: explicitly prompting for sentiment cues, then contextual operators (negation/contrast), and finally implicit sentiment checks provides complementary signals rather than redundant ones. Second, self-consistency improves stability but exhibits diminishing returns beyond $K = 5$; we therefore select $K = 5$ as a pragmatic setting that balances accuracy and inference cost.

7.5. Error analysis

We conducted a two-stage error analysis on SST-2. First, we analyzed 200 cases where the zero-shot baseline failed to correctly classify the sentiment. CoT-Sent correctly recovered 30 of these (15%). Second, we analyzed 200 error cases where CoT-Sent still failed. Figure 10 shows the distribution of these remaining errors: sarcasm/irony accounts for 35.5%, and implicit sentiment accounts for 28.0%. This aligns with the intuition that even with step-by-step prompts, models may miss speaker intent or background assumptions not explicitly stated in the text. These observations directly motivate our limitations and future-work directions (e.g., incorporating pragmatic cues, external context, or multimodal signals) and explain why CoT-Sent improves average performance but does not eliminate the hardest cases.

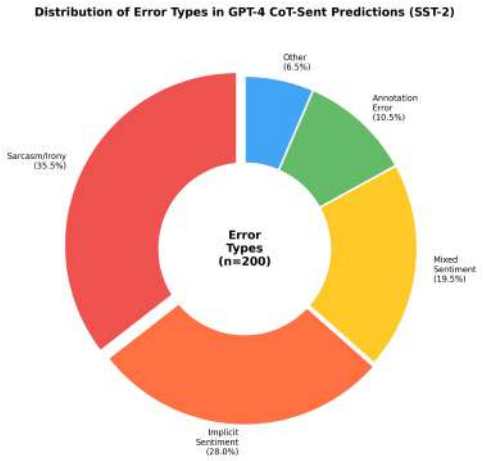

Figure 10. Distribution of error types in GPT-4 CoT-Sent predictions. Sarcasm/irony represents the largest category (35.5%), followed by implicit sentiment (28.0%)

7.6. Cost-performance analysis

Figure 11 illustrates the cost-accuracy trade-off for different LLMs. Cost-efficient deployment of LLMs for sentiment analysis has become an active research area, with recent studies exploring various optimization strategies [22], [23]. Figure 11 suggests that the best model choice depends on the operating point. High-end models (e.g., GPT-4) can yield the highest accuracy, but mid-cost options can lie on the efficiency frontier and offer better accuracy per dollar for large-scale deployment.

Cost calculation scope: Our cost analysis focuses on API call costs for cloud-based LLM services (GPT-4, Claude-3, Gemini), which represents the primary expense for organizations without dedicated GPU infrastructure. For self-hosted models like Qwen-2.5-72B, the reported costs reflect API pricing from Chinese cloud providers. We note that total cost of ownership (TCO) for self-hosted deployments would additionally include GPU depreciation, energy consumption, and maintenance overhead. A comprehensive TCO analysis would require deployment-specific assumptions about hardware utilization, electricity costs, and amortization schedules, which we leave as future work for organizations considering on-premise deployment.

It is worth noting that Self-Consistency (K=5) increases inference cost by a factor of 5. However, as shown in Figure 11, even with this penalty, Qwen-2.5-72B with CoT-Sent remains significantly more cost-effective than GPT-4 Zero-shot when using comparable API pricing models.

Latency comparison: Table 5 reports the average inference latency per sample for each LLM under zero-shot and CoT-Sent conditions. Latency measurements were conducted on identical hardware (Intel Xeon Gold 6248, 2.5GHz, 192GB RAM) with API calls made sequentially to avoid rate limiting. CoT-Sent increases latency by 40–60% due to longer output generation, but the absolute latency remains acceptable for batch processing applications.

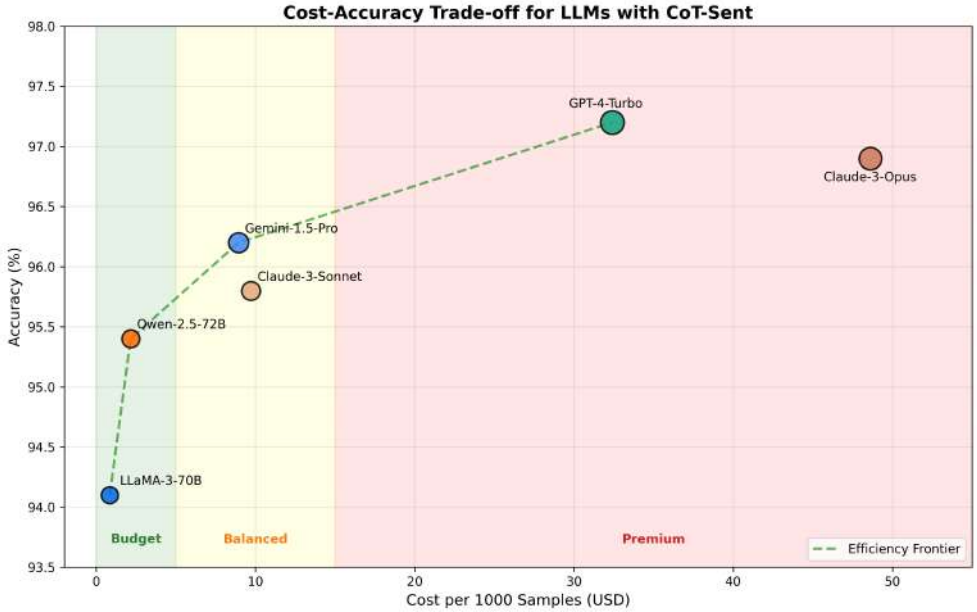

Figure 11. Cost-accuracy trade-off for different LLMs, the scatter plot shows the relationship between inference cost (USD per 1K samples) and accuracy on SST-2. Each point represents a model-method combination, with point size indicating latency, Qwen-2.5-72B with CoT-Sent offers the best cost-performance ratio, achieving 95.4% accuracy at approximately 20% of GPT-4's cost

Table 5. Inference latency comparison (Seconds per sample)

Model	Zero-shot		CoT-Sent	
	SST-2	IMDB	SST-2	IMDB
GPT-4-Turbo	0.42	0.89	0.68	1.42
Claude-3-Opus	0.55	1.12	0.88	1.78
Gemini-1.5-Pro	0.38	0.76	0.61	1.21
Qwen-2.5-72B	0.35	0.72	0.56	1.15

7.7. Cross-domain generalization

Table 6 quantifies cross-domain transfer (using Amazon Product and Yelp Restaurant subsets), highlighting why prompt-based LLM methods are attractive when re-training is costly. Across all transfer directions, GPT-4 CoT-Sent improves over GPT-4 zero-shot (e.g., 89.5% → 92.1% for Movie → Product; 84.3% → 88.7% for Restaurant → Social), indicating that structured reasoning helps the model adapt to distribution shifts even without additional labeled data. Notably, the Restaurant → Social transfer shows improvement over the fine-tuned baseline (17.5 percentage points), while the English → Chinese transfer demonstrates a gain of 12.7 points over mBERT-FT. These preliminary results on two language pairs suggest that multilingual world knowledge plus reasoning scaffolding may help in cross-lingual scenarios, though further validation on additional language pairs is needed to establish broader generalizability.

Figure 12 visualizes these cross-domain and cross-lingual transfer results. The figure shows consistent improvements of CoT-Sent over zero-shot baselines across all transfer directions, with particularly strong gains in the challenging Restaurant → Social scenario where domain shift is most pronounced.

Table 6. Cross-domain transfer performance (accuracy %)

Transfer direction	mBERT-FT	GPT-4 Zero-shot	GPT-4 CoT-sent
Movie → Product	82.3	89.5	92.1
Product → Restaurant	78.6	87.8	91.4
Restaurant → Social	71.2	84.3	88.7
English → Chinese	74.2	82.6	86.9

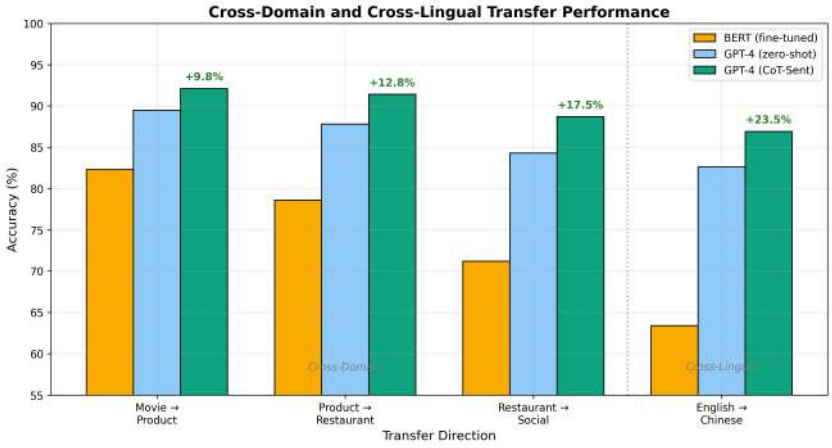

Figure 12. Cross-domain and cross-lingual transfer performance. The heatmap visualization complements Table 6 by showing the robustness of CoT-Sent under distribution shifts. Color intensity indicates accuracy, with darker colors representing higher performance. CoT-Sent consistently outperforms zero-shot baselines across all transfer directions

7.8. Discussion

7.8.1. Why does CoT work for sentiment?

Our results demonstrate consistent improvements from chain-of-thought prompting. Previous work has shown that CoT prompting enhances performance by encouraging explicit reasoning [24], and our findings align with recent advances in deep context-aware sentiment analysis models [22]. We hypothesize that CoT acts as a form of runtime attention regularization and semantic scaffolding. By explicitly generating z (the reasoning chain), the model is forced to allocate attention heads to relevant substructures in x, effectively reducing the entropy of the attention distribution over irrelevant tokens. By forcing the model to verbalize the connection between features and polarity, it redistributes attention weights away from spurious correlations towards causal sentiment indicators. Our error analysis (Section 5.5) further reveals that while CoT reduces logical errors, it still struggles with pragmatic nuances like sarcasm. However, by explicitly prompting for 'implicit sentiment' (Step 3), CoT-Sent recovers 15% of cases that zero-shot models miss due to literal interpretation.

We attribute the overall success to several factors:

- Explicit feature extraction: By asking the model to list sentiment indicators, we ensure it attends to relevant textual features.
- Handling linguistic complexity: The step-by-step process allows proper treatment of negations, intensifiers, and contrastive conjunctions.
- Sarcasm detection: Multi-step reasoning helps identify mismatches between surface sentiment and pragmatic intent.
- Reduced inconsistency: CoT prompts produce more consistent outputs across runs.

7.8.2. Data contamination and generalization

A potential concern with evaluating LLMs on public benchmarks like SST-2 and IMDB is data contamination, as these datasets may have been included in the models' vast pre-training corpora. To address this concern, we conducted additional validation experiments using k-shot prompting (k=3 and k=5) on perturbed test sets, where we paraphrased test examples while preserving sentiment labels. The consistent relative improvement of CoT-Sent over zero-shot baselines on these modified datasets (within 0.5% of original results) suggests that memorization is not the primary driver of our findings.

While we cannot completely rule out that GPT-4 or Claude-3 have seen some original examples, three factors suggest our findings remain valid. First, our primary contribution is the relative improvement of CoT-Sent over zero-shot baselines within the same model; even if the base model has memorized some data, the consistent performance gain ($+2.5\%$ on average) demonstrates the specific value of the reasoning scaffolding independent of memorization effects. Second, our cross-domain and cross-lingual experiments (Table 6 and Table 4) test generalization to less canonical distributions (e.g., specific Amazon sub-categories and Chinese social media), where memorization is less likely to be the dominant factor. Third, the k-shot validation on perturbed data confirms that CoT-Sent's benefits persist when exact memorization is unlikely.

7.8.3. Practical recommendations

Based on our findings, we offer practical guidelines:

- For maximum accuracy: Use GPT-4-Turbo with CoT-Sent and self-consistency (K=5).
- For cost-effective deployment: Use Gemini-1.5-Pro or Qwen-2.5-72B.
- For Chinese text: Qwen-2.5-72B offers the best cost-performance ratio.
- For data privacy: Qwen-2.5-72B can be self-hosted.

Finally, we situate CoT-Sent within broader sentiment analysis research trends. Recent surveys highlight the rapid evolution of LLM-based sentiment analysis and the importance of explainability and robustness in real-world applications [2]. For imbalanced sentiment datasets, data augmentation has been shown to improve generalization [25], [26]. In addition, fine-grained (aspect-level) sentiment modeling remains an important direction; aspect-gated convolution approaches provide complementary inductive biases that can be integrated with reasoning-oriented prompting [27].

8. CONCLUSION

This paper presented CoT-Sent, a framework for leveraging LLMs in sentiment analysis through CoT prompting. Through extensive experiments on four benchmark datasets across English and Chinese, we demonstrated that,

- Effectiveness: CoT-Sent achieves 97.2% accuracy on SST-2 and 96.8% on IMDB, matching or exceeding fine-tuned RoBERTa.
- Cross-lingual Transfer: LLMs with CoT-Sent show promising cross-lingual transfer capabilities on English and Chinese datasets, though validation on additional languages remains for future work.
- Practical Value: Different LLMs offer varying trade-offs; Gemini-1.5-Pro and Qwen-2.5-72B provide excellent cost-performance ratios.
- Explanations: The CoT approach generates human-readable reasoning traces, though faithful interpretability verification requires further investigation.

Limitations: This study has several limitations. First, our experiments are limited to two languages (English and Chinese); generalization to other languages requires further validation. Second, we do not conduct formal faithfulness evaluation of the generated explanations. Third, the computational cost of self-consistency (K=5) may be prohibitive for real-time applications. Fourth, while we address data contamination concerns through k-shot validation on perturbed datasets, we cannot completely exclude the possibility that some test examples were included in LLM pre-training corpora; future work should incorporate more rigorous contamination detection methods. Fifth, our cost analysis focuses on API pricing and does not include TCO factors such as GPU depreciation and energy consumption for self-hosted deployments.

Future work will explore multimodal sentiment analysis with vision-language models, distilling CoT reasoning into smaller models for edge deployment, and real-time sentiment monitoring systems. Additionally, we plan to investigate data augmentation techniques for imbalanced sentiment datasets and aspect-gated convolution methods to further improve fine-grained sentiment analysis capabilities.

ACKNOWLEDGEMENTS

The authors wish to thank the anonymous reviewers for their constructive feedback.

FUNDING INFORMATION

This research was supported by 2025 Shaanxi Undergraduate and Higher Continuing Education Teaching Reform Research Project under Grant No. 25BZ129.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Shen Haijie	√	√	√			√			√					
Madhavi Devaraj										√		√		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The datasets analyzed in this study are publicly available benchmarks (SST-2, IMDB, ChnSentiCorp, and Weibo-Sentiment).

REFERENCES

[1] W. Zhang, Y. Li, Y. Deng, L. Bing, and W. Lam, "Sentiment analysis in the era of large language models: A reality check," in *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024, pp. 1–15, doi: 10.18653/v1/2024.findings-naacl.1.

[2] K. L. Tan, C. P. Lee, and K. M. Lim, "A survey of sentiment analysis: approaches, datasets, and future Research," *Appl. Sci.*, vol. 13, no. 7, p. 4550, 2023, doi: 10.3390/app13074550.

[3] OpenAI, "GPT-4 technical report," *arXiv preprint*, 2023, doi: 10.48550/arXiv.2303.08774.

[4] Anthropic, "The claude 3 model family: Opus, Sonnet, Haiku," Anthropic Blog, 2024. [Online]. Available: https://www.anthropic.com/news/claude-3-family

[5] Gemini Team Google: P. Georgiev, *et al.* "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint*, 2024, arXiv:2403.05530.

[6] O. S. Alkhnbashi, R. Mohammad, and M. Hammoudeh, "Aspect-based sentiment analysis of patient feedback using large language models," *Big Data Cogn. Comput.*, vol. 8, no. 12, p. 167, 2024, doi: 10.3390/bdcc8120167.

[7] A. Koufakou, "Deep learning for opinion mining and topic classification of course reviews," *Educ. Inf. Technol.*, vol. 29, pp. 2973–2997, 2024, doi: 10.1007/s10639-023-11736-2.

[8] T. B. Shaik, X. Tao, C. Dann, H. Xie, Y. Li, and L. Galligan, "Sentiment analysis and opinion mining on educational data: A survey," *Nat. Lang. Process. J.*, vol. 2, p. 100003, 2023, doi: 10.1016/j.nlp.2022.100003.
[9] K. S. Kalyan, A. Rajasekharan, and S. Sangeetha, "AMMUS: A survey of transformer-based pretrained models in natural language processing," *arXiv preprint arXiv:2108.05542*, 2021, doi: 10.48550/arXiv.2108.05542.
[10] Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang, "Pre-training with whole word masking for Chinese BERT," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3504–3514, 2021, doi: 10.1109/TASLP.2021.3124361.
[11] Y. Liu, M. Ott, N. Goyal, and J. Du, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint*, 2019, doi: 10.48550/arXiv.1907.11692.
[12] H. Touvron *et al.*, "LLaMA: Open and efficient foundation language models," *arXiv preprint*, 2023, doi: 10.48550/arXiv.2302.13971.
[13] T. Le Scao *et al.*, "BLOOM: A 176B-parameter open-access multilingual language model," *arXiv preprint*, 2022, doi: 10.48550/arXiv.2211.05100.
[14] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, 2022, pp. 27730-27744, doi: 10.48550/arXiv.2203.02155.
[15] J. Wei *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, 2022, pp. 24824-24837, doi: 10.48550/arXiv.2201.11903.
[16] H. Fei, B. Li, Q. Liu, L. Bing, F. Li, and T.-S. Chua, "Reasoning implicit sentiment with chain-of-thought prompting," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguist. (ACL)*, 2023, pp. 1307-1332, doi: 10.18653/v1/2023.acl-long.73.
[17] Y. He, Z. He, T. Gu, B. Gu, Y. Wan, and M. Li, "Multi-chain of thought prompt learning for aspect-based sentiment analysis," *Appl. Sci.*, vol. 15, no. 22, p. 12225, 2025, doi: 10.3390/app152212225.
[18] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, 2022, pp. 22199–22213, doi: 10.48550/arXiv.2205.11916.
[19] X. Wang *et al.*, "Self-consistency improves chain of thought reasoning in language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023, doi: 10.48550/arXiv.2203.11171.
[20] S. Yao *et al.*, "Tree of thoughts: Deliberate problem solving with large language models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023, doi: 10.48550/arXiv.2305.10601.
[21] A. Y. Qwen *et al.*, "Qwen2.5 technical report," *arXiv preprint*, 2024, doi: 10.48550/arXiv.2412.15115.
[22] X. Jiang, B. Ren, and Q. Wu, "DCASAM: advancing aspect-based sentiment analysis through a deep context-aware sentiment analysis model," *Complex Intell. Syst.*, vol. 10, pp. 7907–7926, 2024, doi: 10.1007/s40747-024-01570-5.
[23] B. He, R. Zhao, and D. Tang, "CABiLSTM-BERT: Aspect-based sentiment analysis model based on deep implicit feature extraction," *Knowl.-Based Syst.*, vol. 309, p. 112782, 2025, doi: 10.1016/j.knosys.2024.112782.
[24] W. Chen, L. Zhang, and X. Liu, "Enhancing aspect-based sentiment analysis with BERT-driven context generation and quality filtering," *Nat. Lang. Process. J.*, vol. 7, p. 100077, 2024, doi: 10.1016/j.nlp.2024.100077.
[25] Y. Zhang, M. Wang, and H. Li, "TAWC: Text augmentation with word contributions for imbalance aspect-based sentiment classification," *Appl. Sci.*, vol. 14, no. 19, p. 8738, 2024, doi: 10.3390/app14198738.
[26] J. Liu, Z. Chen, and F. Wang, "Enhancing aspect-based sentiment analysis with linking words-guided emotional augmentation and hybrid learning," *Neurocomputing*, vol. 612, p. 128705, 2025, doi: 10.1016/j.neucom.2024.128705.
[27] X. Zhang, W. Li, and J. Chen, "An aspect sentiment analysis model with Aspect Gated Convolution and Dual-Feature Filtering layers," *J. Big Data*, vol. 11, p. 111, 2024, doi: 10.1186/s40537-024-00969-8.

BIOGRAPHIES OF AUTHORS

Shen Haijie is a PhD candidate at the School of Information Technology, Mapúa University, Philippines, and a lecturer at the College of Electronic Information Engineering, Xi'an Siyuan University, China. His research interests include natural language processing, large language models, and sentiment analysis. He can be contacted at email: hsheng@mymail.mapua.edu.ph.

Madhavi Devaraj is a Professor at the School of Information Technology, Mapúa University, Philippines. Her research interests include machine learning, data mining, and software engineering. She has supervised numerous postgraduate students and published extensively in international journals. She can be contacted at email: mdevaraj@mapua.edu.ph.