Improving In-Context Learning with Prediction Feedback for Sentiment Analysis

Anonymous ACL submission

Abstract

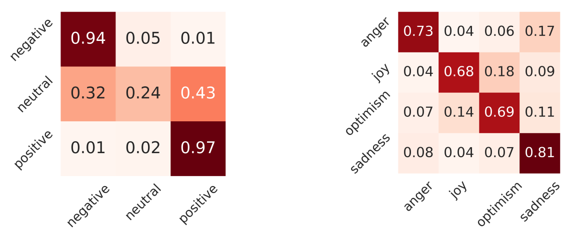
Large language models (LLMs) have achieved promising results in sentiment analysis through the in-context learning (ICL) paradigm. However, their ability to distinguish subtle sentiments still remains a challenge. Inspired by the human ability to adjust understanding via feedback, this paper enhances ICL by incorporating prior predictions and feedback, aiming to rectify sentiment misinterpretation of LLMs. Specifically, the proposed framework consists of three steps: (1) acquiring prior predictions of LLMs, (2) devising predictive feedback based on correctness, and (3) leveraging a feedback-driven prompt to refine sentiment understanding. Experimental results across nine sentiment analysis datasets demonstrate the superiority of our framework over conventional ICL methods, with an average F1 improvement of 5.95%.

1 Introduction

Sentiment analysis aims to detect subjective opinions within texts automatically (Medhat et al., 2014), covering tasks such as sentiment classification, aspect-based sentiment analysis, and emotion detection (Zhang et al., 2018).

Previous research proposed many supervised methods for sentiment analysis (Xu et al., 2019; Li et al., 2021a). To avoid their reliance on large amounts of human-annotated data, some studies attempted to use limited data to recognize sentiment yet obtained mediocre results. With the advent of large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023a), studies have revealed that LLMs can yield promising performance on sentiment analysis via in-context learning (ICL) paradigm (Li et al., 2023; Wang et al., 2023), which utilizes only few-shot input-label pairs selected from a candidate example pool.

Despite achieving favorable results, the conventional ICL paradigm still faces a concerning limitation. Namely, through the provided examples,



(a) Rest dataset.

(b) TwEmo dataset.

Figure 1: Normalized confusion matrices on two sentiment analysis datasets. Results are from ChatGPT.

LLMs fail to differentiate subtle sentiments effectively. Consequently, they would predict plausible yet incorrect sentiment labels. As depicted in Figure 1a, although LLMs can clearly distinguish between *positive* and *negative* polarities, they often mistakenly categorize *neutral* into others. In addition, as shown in Figure 1b, LLMs frequently mislabel fine-grained sentiments as relevant but wrong labels, such as *joy* and *optimism*, stemming from their incapacity to understand nuanced sentiments with similar contexts.

Inspired by the human learning process, where individuals initially make plans based on prior knowledge and adjust their understanding through actual feedback (Bélanger, 2011), we propose to integrate feedback on prior predictions into ICL, aiming to rectify sentiment misunderstandings of LLMs. Specifically, our framework first yields prior predictions for each candidate example using traditional ICL. We then categorize examples into two sub-pools based on correctness and exploit feedback to illustrate differences between prior predictions and human annotations. Finally, during inferring, we select relevant examples from each sub-pool and utilize a specific feedback-driven prompt to wrap input, label, prediction, and feedback. Unlike conventional ICL, where LLMs only see correct labels, our framework effectively directs LLMs to adjust their sentiment understanding and reason-

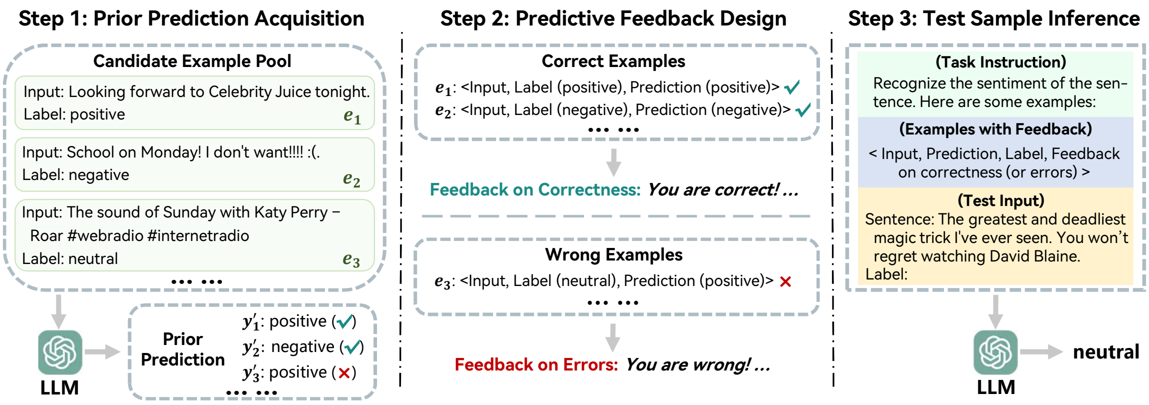


Figure 2: Overview of our framework.

ing to align more closely with label perception through prediction and feedback.

Experimental results on nine sentiment analysis datasets show that our framework outperforms existing ICL baselines by 5.95% in average F1. Further discussions indicate its effectiveness and robustness. Moreover, when extended to other tasks, our framework also yields competitive results.

2 Preliminary

Sentiment analysis aims to predict the sentiment label y' of an input text x. Here, different tasks may have different label spaces $\mathcal{C}$ and inputs[1]. In ICL paradigm, given an input x and k-shot in-context examples $\{(x_i, y_i)\}_{i=1}^{k}$ retrieved from a pre-defined candidate pool $\mathcal{P}$ (its size is relatively small), a frozen LLM $\mathcal{M}$ is used to predict y'.

$$y' = \operatorname*{argmax}_{y \in \mathcal{C}} \mathcal{M}(y|(x_1, y_1), ...(x_k, y_k), x) \quad (1)$$

Here, we ignore the task instruction and example template for simplicity. Meanwhile, we employ a constrained decoding strategy to avoid redundant and irrelevant outputs, ensuring only label words within $\mathcal{C}$ can be generated.

3 The proposed Framework

As shown in Figure 2, our framework consists of three steps: 1) *prior prediction acquisition*, 2) *predictive feedback design*, and 3) *test sample inference*. Below is a detailed description of each step.

Step 1: Prior Prediction Acquisition. This step focuses on acquiring the prior prediction y_i' on each candidate example x_i for subsequent feedback provision. To this end, examples from $\mathcal{P}$ are treated as inference targets. Following the traditional ICL, we randomly select other four input-label pairs from the candidate pool as demonstrations[2], which are then combined with task instructions to prompt LLMs for predictions (see Appendix A for more details). We refer to these predictions as *prior predictions* because they serve to reflect the prior sentiment understanding of LLMs.

Step 2: Predictive Feedback Design. The correctness of the prior predictions directly indicates whether LLMs can accurately grasp the sentiment of the corresponding examples. To elicit self-adjustments of LLMs in understanding and reasoning, we first classify the examples into two sub-pools, $\mathcal{P}_c$ and $\mathcal{P}_w$, where the former includes correctly classified examples, and the latter contains wrong ones. We then provide each sub-pool with feedback in the natural language form:

feedback on $\mathcal{P}_c$: *You are correct! Stay determined and keep moving forward.*

feedback on $\mathcal{P}_w$: *You are wrong! Make sure your prediction is accurate.*

Step 3: Test Sample Inference. To complete the inference for the given test input, we first retrieve $k/2$ examples from each candidate sub-pool[3]. In addition, we develop a feedback-driven prompt template to wrap the input, prediction, label, and feedback of each selected example into a quadruple. Subsequently, These quadruples are organized by $\mathcal{P}_w$ examples before $\mathcal{P}_c$ ones and sorted by descending relevance. Finally, the test sentence is wrapped in the standard example template, with the label position left blank for prediction.

[1]For example, aspect sentiment classification task needs to consider the effect of aspects on labels (Pontiki et al., 2014).

[2]The reason for selecting four is to strike a trade-off between contextual richness and computational efficiency.

[3]Our framework is retrieval-mode agnostic so any example retrieval technique can be employed here.

Method	Sentiment Classification				Aspect Sentiment Classification			Emotion Detection	
	SST-2	TwSenti	Poem	Finance	Rest	Laptop	Twitter	EmoC	TwEmo
BERT-FT†	84.69	54.54	72.55	89.41	64.59	69.03	56.40	47.73	62.53
Random	89.82	55.27	55.08	75.34	68.77	73.02	54.95	45.20	47.77
+ Ours	$91.65_{+1.83}$	$60.33_{+5.06}$	$64.37_{+9.29}$	$78.64_{+3.30}$	$71.16_{+2.39}$	$72.80_{-0.22}$	$57.64_{+2.69}$	$52.50_{+7.30}$	$60.91_{+13.14}$
BM25	90.26	55.35	49.99	56.13	68.99	70.29	50.99	44.89	48.44
+ Ours	$91.85_{+1.59}$	$59.20_{+3.85}$	$61.27_{+11.28}$	$66.94_{+10.81}$	$71.76_{+2.77}$	$71.67_{+1.38}$	$56.22_{+5.23}$	$51.63_{+6.73}$	$62.88_{+14.44}$
SBERT	87.96	50.13	47.41	47.12	68.21	65.72	50.60	46.28	48.58
+ Ours	$91.57_{+3.61}$	$55.08_{+4.95}$	$56.42_{+9.01}$	$58.21_{+11.09}$	$71.29_{+3.08}$	$69.56_{+3.84}$	$56.07_{+5.47}$	$50.30_{+4.02}$	$61.22_{+12.64}$
MMR	89.64	50.80	49.74	54.51	68.30	66.72	51.07	43.72	49.94
+ Ours	$92.65_{+3.01}$	$56.84_{+6.04}$	$63.38_{+13.64}$	$59.85_{+5.34}$	$69.76_{+1.46}$	$69.23_{+2.51}$	$55.57_{+4.50}$	$49.31_{+5.59}$	$61.74_{+11.80}$
K-Means	88.74	56.26	51.39	76.14	71.01	73.68	55.20	45.71	46.93
+ Ours	$92.23_{+3.49}$	$61.32_{+5.06}$	$68.70_{+17.31}$	$78.44_{+2.30}$	$71.10_{+0.09}$	$73.11_{-0.57}$	$57.78_{+2.58}$	$53.72_{+8.01}$	$61.89_{+14.96}$

Table 1: Main results in F1% (see Acc% results in Appendix C.1). Fine-tuning methods are marked by †.

4 Experiments

4.1 Experimental Setup

Dataset. We conduct experiments across three sentiment analysis tasks using nine distinct datasets, including **Sentiment Classification** (SC): SST-2 (Socher et al., 2013), TwSenti (Rosenthal et al., 2017), Poem (Sheng and Uthus, 2020), and Finance (Malo et al., 2014); **Aspect Sentiment Classification** (ASC): Rest and Laptop (Pontiki et al., 2014), and Twitter (Dong et al., 2014); **Emotion Detection** (ED): EmoC (Chatterjee et al., 2019) and TwEmo (Barbieri et al., 2020). Detailed statistics are listed in Appendix B.1.

Baseline. To evaluate the effectiveness of the proposed framework, We combine it with various training-free example retrieval baselines for comparison, including **Random**, **BM25** (Robertson et al., 2009), **SBERT** (Reimers and Gurevych, 2019), **MMR** (Ye et al., 2023), and **K-Means** (Zhang et al., 2023). Furthermore, we introduce **BERT-FT**, where BERT-base model is fine-tuned directly on candidate pool examples. See Appendix B.2 for their specific settings.

Implementation Details. We utilize Llama-2-13B-Chat (Touvron et al., 2023b) as the backbone LLM due to its moderate scale and excellent ICL performance. We set the number of in-context examples to 4. The candidate pool is formed by sampling 300 label-balanced examples from each training set. We experiment with three different random seeds and present the average outcomes. All experiments are conducted with NVIDIA RTX A6000 GPU. More details are shown in Appendix B.3.

4.2 Main Results

Results shown in Table 1 indicate that our framework substantially enhances baseline performance across nearly all datasets. For instance, augmenting K-means with our framework results in an average F1 increase of 5.91%, exhibiting its superiority. Meanwhile, compared with BERT-FT, our approach demonstrates outstanding performance on the majority of datasets, highlighting its efficacy in resource-limited and training-free scenarios.

Additionally, our framework notably excels in the ED task, where subtle sentiment detection is critical. Comparatively, ASC involves more complex aspect-based contextual understanding, constraining our framework's effectiveness in this task.

4.3 Ablation Study

We perform the ablation study to explore the effect of each component. Results are shown in Table 2. When removing task instructions (Inst), we see performance drops except for Rest, indicating its insensitivity to instructions with information-rich inputs. Additionally, both the removal of labels (Label) and

Inst	Label	Pred	Feed	Poem	Rest	TwEmo
✓	✓	✓	✓	**68.70**	71.10	**61.89**
✗	✓	✓	✓	55.97	**71.70**	61.13
✓	✗	✓	✓	51.47	67.05	52.78
✓	✓	✗	✓	59.47	69.94	48.53
✓	✓	*R*	✓	67.00	70.14	60.71
✓	✓	*Z*	✓	64.71	70.18	60.91
✓	✓	✓	✗	63.46	70.49	60.02

Table 2: Ablation study based on K-Means (see more in Appendix C.2). We apply other sources of Pred including random errors (R) and zero-shot prompting (Z).

	Conventional In-context Learning	In-context Learning w/ Prediction Feedback
Prompt	*Task Instruction*: ... *Examples*: $\{(x_i, y_i)\}$	*Task Instruction*: ... *Examples*: $\{(x_i, y'_i, y_i, fd)\}$
Input	@user Wishing you well sir... you are an extremely straightforward and jovial person...	
Output	optimism (✗)	joy (✓)
Reason	The sentence conveys a positive and uplifting message. The use of the word "jovial" suggests a **lighthearted and cheerful tone, which is consistent with the label "optimism"**. Additionally, the phrase "wishing you well" implies good intentions and a **positive attitude towards the recipient, further supporting the label "optimism".**	The sentence is addressed to a specific user and expresses well-wishes, which suggests a positive and friendly tone. The description of the user as "extremely straightforward and jovial" further **reinforces the positive sentiment. Based on the examples and feedback provided, the label "joy" best captures the emotion of the sentence.**

Table 3: Case study of conventional ICL and our framework. More cases are shown in Appendix C.4.

prior predictions (Pred) cause a notable decline, by averages of 10.13% and 7.92%, respectively, highlighting the significance of their combination in our framework. Besides, employing alternative prediction sources or excluding feedback (Feed) also leads to a slight decrease.

4.4 Discussions[4]

Effect on Subtle Sentiments. To demonstrate the impact of our framework on subtle sentiments, we visualize the prediction distributions as depicted in Figure 3. We can observe an obvious change of distribution in *neutral*, whose correct rate increases by 32%, while the other two categories are relatively stable. These results suggest that integrating predictive feedback could make more accurate distinctions between subtle sentiments.

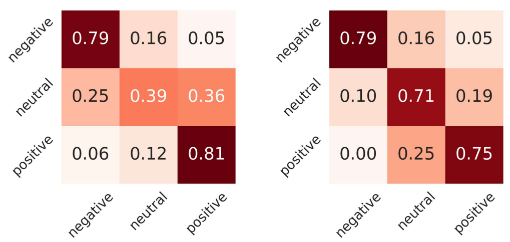


Figure 3: Normalized confusion matrices for the Poem dataset: K-Means (left) and K-Means+Ours (right).

Task Generalization. To demonstrate that our framework is not confined to adjusting sentiment understanding of LLMs, we conduct experiments on other three datasets: P-Stance (Li et al., 2021b) for Stance Detection, TwIrony (Van Hee et al., 2018) for Irony Detection, and MNLI (Wang et al., 2019) for NLI. Results are illustrated in Table 4.

Method	P-Stance	TwIrony	MNLI
Random	70.94	62.29	49.63
+ Ours	**73.31**$_{+2.37}$	**65.44**$_{+3.15}$	**55.21**$_{+5.58}$
BM25	72.23	60.06	50.68
+ Ours	**72.98**$_{+0.75}$	**64.29**$_{+4.23}$	**56.65**$_{+5.97}$
K-Means	71.17	61.47	50.60
+ Ours	**73.60**$_{+2.43}$	**65.72**$_{+4.25}$	**55.09**$_{+4.49}$

Table 4: Results of task generalization (F1%).

Notably, the enhancements with our framework are also evident on these datasets, with F1 increasing up to 4.25% for TwIrony and 5.97% for MNLI. These suggest that the adaptability of prediction feedback can extend to a broader scope of language understanding tasks.

Case Study. To gain a deeper insight into the advantages of our framework, we conduct the case study on LLM's output and explanation, which is illustrated in Table 3. In this case, our framework outputs *joy* instead of the plausible yet incorrect *optimism*, and offers a more fitting explanation that aligns with the implied emotion of input. Therefore, it proves that our framework promotes self-adjustment of the LLM in sentiment analysis, refining both output and reasoning accuracy.

5 Conclusion

In this paper, we propose a novel ICL framework that utilizes prediction feedback akin to human learning. It improves ICL by incorporating both prior predictions and corresponding feedback into examples, tackling the difficulties LLMs encounter when identifying subtle sentiments. Experiments across various datasets confirm the advantage of our framework compared to traditional ICL, as well as its potential for broader applications.

[4]We present more analyses in Appendix D, including The Sensitivity of Feedback Prompt, Impact of the Ratio of Error Examples, Impact of the Quantity of Examples, Impact of the Order of Examples, and Language Model Generalization.

Limitations

Although our research significantly enhances the performance of conventional ICL and provides in-depth analyses about the adjustment for sentiment understanding in LLMs, the inner working mechanisms of the framework remain elusive due to the black-box nature of language models. Besides, our research primarily focuses on sentiment analysis and other text classification tasks in NLU, leaving more complex realms of language comprehension and generation unexplored, such as text summarization and commonsense generation. We aim to broaden the scope of our framework in future research, delving into more profound insights and wider applicability.

References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of ACL*, pages 8857–8873.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of EMNLP*, pages 1644–1650.

Paul Bélanger. 2011. *Theories in adult learning and education*. Verlag Barbara Budrich.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in NIPS*, 33:1877–1901.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of SemEval*, pages 39–48.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent Twitter sentiment classification. In *Proceedings of ACL*, pages 49–54.

Caoyun Fan, Jidong Tian, Yitian Li, Hao He, and Yaohui Jin. 2023. Comparable demonstrations are important in in-context learning: A novel perspective on demonstration selection. *arXiv preprint arXiv:2312.07476*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b.

Ruifan Li, Hao Chen, Fangxiang Feng, Zhanyu Ma, Xiaojie Wang, and Eduard Hovy. 2021a. Dual graph convolutional networks for aspect-based sentiment analysis. In *Proceedings of ACL*, pages 6319–6329.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. Unified demonstration retriever for in-context learning. In *Proceedings of ACL*, pages 4644–4668.

Yingjie Li, Tiberiu Sosea, Aditya Sawant, Ajith Jayaraman Nair, Diana Inkpen, and Cornelia Caragea. 2021b. P-stance: A large dataset for stance detection in political domain. In *Findings of ACL-IJCNLP*, pages 2355–2365.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of ASIS&T*, 4(65):782–796.

Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113.

OpenAI. 2023. Gpt-4 technical report. *CoRR, abs/2303.08774*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in NIPS*, 35:27730–27744.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of SemEval*, pages 27–35.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of EMNLP-IJCNLP*, pages 3982–3992.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of SemEval*, pages 502–518.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of NAACL-HLT*, pages 2655–2671.

Emily Sheng and David Uthus. 2020. Investigating societal biases in a poetry composition system. In *Proceedings of Workshop on Gender Bias*, pages 93–106.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pages 1631–1642.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Attentional encoder network for targeted sentiment classification. In *ICANN*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. SemEval-2018 task 3: Irony detection in English tweets. In *Proceedings of SemEval*, pages 39–50.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.

Zengzhi Wang, Qiming Xie, Zixiang Ding, Yi Feng, and Rui Xia. 2023. Is chatgpt a good sentiment analyzer? a preliminary study. *arXiv preprint arXiv:2304.04339*.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of NAACL-HLT*, pages 2324–2335.

Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Veselin Stoyanov, Greg Durrett, and Ramakanth Pasunuru. 2023. Complementary explanations for effective in-context learning. In *Findings of ACL*, pages 4469–4484.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *WIREs: DMKD*, 8(4):e1253.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *ICLR*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *ICLR*.

Appendix for "Improving In-Context Learning with Prediction Feedback for Sentiment Analysis"

We organize the appendix into four sections:

- Prompts used in the proposed framework are presented in Appendix A;
- Additional details of datasets, baselines, and implementation are presented in Appendix B;
- Additional experimental results in different settings, such as more metrics, baselines, and datasets are presented in Appendix C; and
- More discussions about the proposed framework are presented in Appendix D.

A Prompt Design

We present the task instructions and prompt templates utilized in our framework for each task in Table 14. Besides, for conventional ICL, the examples and test input are wrapped in the template that removes prediction and feedback, and the formatting word "Correct Label:" is replaced by "Label:".

We divide the feedback prompt into two parts, namely, Feedback on Correctness and Feedback for Analysis, abbreviated as FC and FA. Two manually designed feedback prompts are illustrated in Table 13 for further discussion (see D.1).

To generate explanations for the case study (in Section 4.4), we structure the input as instructive forms to stimulate interaction[5] and employ prompts: *Provide the correct label for the following sample and explain your answer based on the above examples (and feedback).*

B Detailed Settings of Experiments

B.1 Dataset and Metrics

We provide detailed statistics of each investigated dataset by listing its associated task type, number of samples, number of classes, and labels, as summarized in Table 12. When establishing the candidate pool, we select instances only from the training set. Additionally, for the Finance dataset in which no standard split is provided, we randomly partition the dev set and the test set each at a rate of 20% of the total samples. For the MNLI dataset that does not offer publicly available test set labels, we employ the dev set for evaluation.

Across all sentiment analysis datasets utilized in this study, we uniformly apply two metrics for evaluation: Accuracy (**Acc**) and F1 score (**F1**). Specifically, we calculate the binary-F1 score for SST-2 and the macro-F1 score for all others. In the scenario of task generalization, we use binary-F1 for binary classification tasks, such as P-Stance and TwIrony, and macro-F1 for the MNLI dataset.

[5] https://github.com/huggingface/blog/blob/main/llama2.md

B.2 Baseline Details

(1) **Random** randomly selects k-shot examples from the candidate pool for each test sample. (2) **BM25** (Robertson et al., 2009) assesses relevance through keyword overlap and sentence length, used by (Agrawal et al., 2023). (3) **SBERT** (Reimers and Gurevych, 2019) is a semantic-based retrieval method, where we use "paraphrase-mpnet-basev2" following (Li et al., 2023). (4) **MMR** (Ye et al., 2023) leverages BERTScore (Zhang et al., 2019) with maximal-marginal relevance for complementary example selection. (5) **K-Means** (Zhang et al., 2023) performs k-means clustering to divide each dataset into four clusters. We then select examples randomly from each cluster. (6) **BERT-FT** (Devlin et al., 2019) fine-tunes "bert-base-uncased" using the candidate pool examples.

B.3 More Implementation Details

Due to limited computational resources, our assessments are restricted to a test subset of 2,000 examples across the tasks: TwSenti, EmoC, P-Stance, and MNLI. Additionally, to accelerate inference, we load LLMs using fp16 precision. During the generation, we directly use the label words for each class as the verbalizer, as illustrated in Table 12. In instances where label words are tokenized into multiple subtokens, we only utilize the first subword as the label word for prediction (e.g., using 'optim' as the label for 'optimism').

C Additional Results

C.1 Main Results in Accuracy

For a more comprehensive comparison with the performance of baseline methods, we show additional main results in Acc, as shown in Table 5. Contrary to previous studies (Rubin et al., 2022; Li et al., 2023), we find that semantic similarity retrievals like SBERT negatively impact the performance. We suppose it is due to the demonstration bias when solving simple sentiment analysis tasks (Fan et al., 2023) and the lack of example complementarity (Ye et al., 2023).

Method	Sentiment Classification				Aspect Sentiment Classification			Emotion Detection	
	SST-2	TwSenti	Poem	Finance	Rest	Laptop	Twitter	EmoC	TwEmo
BERT-FT†	85.02	54.87	75.64	90.80	73.19	74.37	56.31	65.12	65.87
Random	89.64	55.13	55.77	75.28	79.80	77.27	54.00	69.05	52.99
+ Ours	91.40$_{+1.76}$	60.37$_{+5.24}$	69.23$_{+13.46}$	78.44$_{+3.16}$	81.23$_{+1.43}$	77.27$_{+0.00}$	56.74$_{+2.74}$	74.97$_{+5.92}$	66.26$_{+13.27}$
BM25	90.06	55.15	49.68	53.13	80.25	75.05	50.05	71.23	52.69
+ Ours	91.70$_{+1.64}$	59.20$_{+4.05}$	66.03$_{+16.35}$	64.46$_{+11.33}$	81.83$_{+1.58}$	76.11$_{+1.06}$	55.20$_{+5.15}$	75.95$_{+4.72}$	66.92$_{+14.23}$
SBERT	87.65	49.98	47.76	44.44	78.08	69.57	49.76	72.33	50.88
+ Ours	91.32$_{+3.67}$	55.57$_{+5.59}$	62.82$_{+15.06}$	54.67$_{+10.23}$	80.28$_{+2.20}$	74.31$_{+4.74}$	55.30$_{+5.54}$	78.37$_{+6.04}$	64.34$_{+13.46}$
MMR	89.45	50.55	50.00	51.29	78.37	71.04	50.00	70.48	53.51
+ Ours	92.49$_{+3.04}$	56.98$_{+6.43}$	67.95$_{+17.95}$	56.36$_{+5.07}$	79.95$_{+1.58}$	74.00$_{+2.96}$	54.53$_{+4.53}$	76.28$_{+5.80}$	65.63$_{+12.12}$
K-Means	88.65	56.12	50.96	75.86	81.47	77.85	54.38	72.48	52.69
+ Ours	92.09$_{+3.44}$	61.37$_{+5.25}$	72.44$_{+21.48}$	78.15$_{+2.29}$	81.23$_{-0.24}$	77.69$_{-0.16}$	56.84$_{+2.46}$	76.55$_{+4.07}$	66.96$_{+14.27}$

Table 5: Main results in Acc%. Fine-tuning methods are marked by †.

C.2 Ablation results on more baselines

To comprehensively analyze the significance of each component within our framework, we conduct more ablation studies on two competitive baselines: Random and BM25. We report the results in Tables 6 and 7, respectively.

Inst	Label	Pred	Feed	Poem	Rest	TwEmo
✓	✓	✓	✓	**64.37**	71.16	**60.91**
✗	✓	✓	✓	54.43	**71.18**	60.13
✓	✗	✓	✓	52.93	67.45	51.63
✓	✓	✗	✓	59.10	70.59	47.40
✓	✓	*R*	✓	63.35	69.21	60.66
✓	✓	*Z*	✓	64.08	68.58	60.76
✓	✓	✓	✗	61.32	70.54	60.13

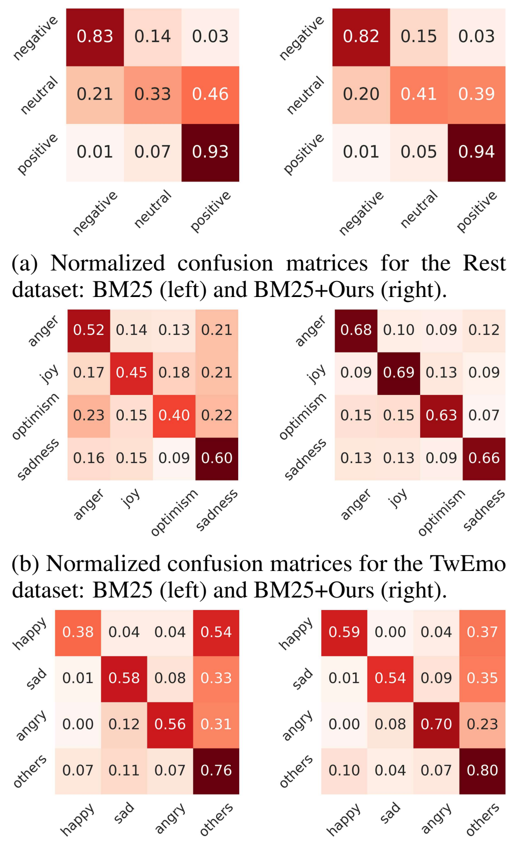
Table 6: Ablation study based on Random.

Inst	Label	Pred	Feed	Poem	Rest	TwEmo
✓	✓	✓	✓	**61.27**	**71.76**	**62.88**
✗	✓	✓	✓	53.86	71.73	61.80
✓	✗	✓	✓	50.42	67.74	51.95
✓	✓	✗	✓	52.45	70.05	50.32
✓	✓	*R*	✓	60.07	70.18	62.64
✓	✓	*Z*	✓	61.09	69.82	62.02
✓	✓	✓	✗	54.68	70.84	62.03

Table 7: Ablation study based on BM25.

C.3 Effect on Subtle Sentiments for More Datasets

To further illustrate how our framework corrects subtle sentiment understanding of the LLM and aligns predictions more closely with true labels, we visualize the improved prediction distributions on more datasets. The results are shown in Figure 4.



(a) Normalized confusion matrices for the Rest dataset: BM25 (left) and BM25+Ours (right).

(b) Normalized confusion matrices for the TwEmo dataset: BM25 (left) and BM25+Ours (right).

(c) Normalized confusion matrices for the EmoC dataset: K-Means (left) and K-Means+Ours (right).

Figure 4: Effect on subtle sentiments on three datasets.

C.4 More Case Studies

We contrast the outputs and reasoning of conventional ICL and the proposed framework on additional cases, as illustrated in Table 8. Observations reveal that incorporating prior predictions and feedback into examples not only corrects the sentiment labels but also yields more aligned explanations with human understandings.

	Conventional In-context Learning	In-context Learning w/ Prediction Feedback (Ours)
Prompt	*Task Instruction*: ... *Examples*: $\{(x_i, y_i)\}$	*Task Instruction*: ... *Examples*: $\{(x_i, y'_i, y_i, fd)\}$
Input	they beached their galley on the shore, and sought	
Output	positive (✗)	neutral (✓)
Reason	The sentence has a sense of adventure and exploration, with the use of the word "beached" which **implies a successful landing,** and the word "sought" which implies a deliberate and purposeful action. **The overall tone of the sentence is optimistic and confident, which aligns with the positive label.**	**The sentence does not convey a positive or negative sentiment.** In this sentence, the word "beached" is used in the sense of landing on the shore, **which is a neutral action.** Additionally, the word "sought" **does not convey any strong emotions or opinions, and is also a neutral word.** Therefore, the overall sentiment of the sentence is neutral.
Input	Another friend had to ask 3 times for parmesan cheese . What is the sentiment polarity of the aspect parmesan cheese ?	
Output	negative (✗)	neutral (✓)
Reason	The sentence implies that the friend had to request the parmesan cheese multiple times, which suggests that **the restaurant had issues with providing the cheese** or that the staff was not attentive to the customer's requests. **This negative experience with the parmesan cheese is evident in the sentence, and therefore, the sentiment polarity of the aspect "parmesan cheese" is negative.**	**The sentence does not convey any strong emotions, either positive or negative, towards the aspect "parmesan cheese".** The fact that the friend had to ask three times for it suggests that there may have been some inconvenience or delay in receiving the cheese, **but it does not convey any strong emotions or opinions about the cheese itself. Therefore, the sentiment polarity of the aspect "parmesan cheese" is neutral.**
Input	my wife did we all marry the same woman loudlycryingface	
Output	angry (✗)	sad (✓)
Reason	The sentence expresses a strong emotion, with the use of "loudly crying face" indicating that **the speaker is upset or angry** about something related to their wife. The sentence structure and word choice also **suggest a sense of frustration or disbelief, which are common characteristics of an angry emotion.**	**The sentence expresses a sense of disappointment and disbelief,** as the speaker seems to be questioning how they all ended up married to the same woman. The use of the word "loudlycryingface" also **suggests that the speaker is emotional and possibly upset. The tone is introspective and melancholic, which is consistent with the emotion of sadness.**

Table 8: More case studies of conventional ICL and our framework.

D More Discussions

D.1 The Sensitivity of Feedback Prompt

To investigate the sensitivity of feedback design, we leverage the two hand-crafted feedback prompts shown in Table 13 and experiment with four combinations. As presented in Table 9, we find that on both datasets, the average variance in performance is within a 1% margin. These results indicate that the performance is not highly sensitive to different feedback prompts.

Combination	Rest		TwEmo	
	Acc	F1	Acc	F1
FC-1+FA-1	81.83	71.76	66.92	62.88
FC-1+FA-2	$82.31_{+0.48}$	$72.51_{+0.75}$	$67.00_{+0.08}$	$62.86_{-0.02}$
FC-2+FA-1	$82.22_{+0.39}$	$71.71_{-0.05}$	$68.30_{+1.38}$	$64.20_{+1.32}$
FC-2+FA-1	$82.10_{+0.27}$	$72.08_{+0.32}$	$67.98_{+1.06}$	$63.93_{+1.05}$

Table 9: Results of different feedback (BM25+Ours).

D.2 Impact of the Ratio of Error Examples

To consider the effect of the erroneous example, we fix k at 8, and vary the number of examples selected from P_w: 0, $k/4$, $k/2$, $3k/4$, and k. The results are presented in Figure 5. We find that the framework tends to underperform with no error examples in the TwEmo dataset and an excess (8/8) in the Rest dataset. As the quantity of error examples increases, the performance initially rises and then declines, indicating that a relative balance of error to correct examples is beneficial in our predictive feedback approach.

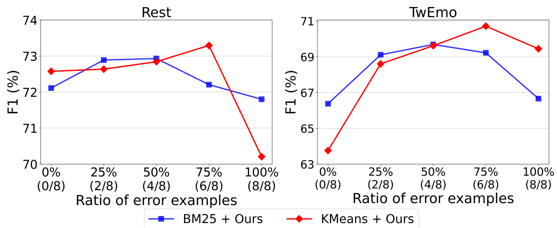


Figure 5: Impact of the ratio of error examples.

D.3 Impact of the Quantity of Examples

To analyze the effect of the example quantity, we perform experiments varying the number of examples ($k = 2, 4, 6, 8, 12$), as depicted in Figure 6. First, we observe a consistent progressive trend of F1 as the number of examples increases. Second, our method generally yields significant performance gains on both datasets compared to the baselines, except for the 12-shot scenario on TwEmo. This suggests the proposed framework can positively influence the LLM to understand and analyze sentiment within an optimal context length.

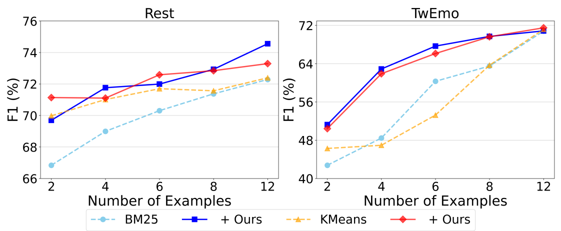


Figure 6: Effect of the quantity of in-context examples.

D.4 Impact of the Order of Examples

To investigate the impact of example ordering, we first categorize three strategies: prioritizing wrong examples (wrong first), prioritizing correct examples (correct first), and alternating between the two. We then subject each strategy to both ascending and descending orders based on retrieval scores. On this basis, we experiment with five additional permutations, as shown in Table 10. We find that the performance of Rest remains stable regardless of permutations, with a negligible standard deviation of 0.51 in F1. Conversely, on TwEmo, descending ordering generally outperforms ascending ones, while the effect is more stable across different strategies. These findings suggest that although our framework is robust against the variability of ordering strategy, the consideration of specific arrangement methods can also be important.

D.5 Language Model Generalization

To thoroughly assess the effectiveness and generalizability of our framework, we conduct model generalization experiments across various LLMs. Specifically, we select three capable and prominent models: Mistral-7B-inst (Mistral-7B-instruct-v0.2) (Jiang et al., 2023), GPT-3.5 Turbo (gpt-3.5-turbo-0301) (Ouyang et al., 2022), and GPT-4 (gpt-4-0613) (OpenAI, 2023). Our experiments compare the performance variation of three methods (Random, BM25, and K-means), with an exception for GPT-4 where only K-means is utilized due to API cost considerations. Additionally, we explore the effect of supervised learning with the BERT-base model. Here, FT (All) denotes fine-tuning on a complete dataset, while FT (CP) indicates fine-tuning using just a candidate pool, similar to the scenario in the ICL approach.

The results illustrated in Table 11 show that incorporating our framework consistently enhances the performance of ICL, illustrating its generalizability. This is particularly evident with GPT-3.5 Turbo, where the average F1 improvement is 4.72%. Notably, when applied to LLMs with advanced language understanding, our framework significantly surpasses the supervised method in similar resource settings. For instance, by leveraging GPT-3.5 Turbo on TwEmo, we observe a 9.59% increase in F1 over FT (CP). Furthermore, using prediction feedback on the Rest dataset, GPT-3.5 Turbo even outperforms fine-tuning BERT on the full training set, and GPT-4 demonstrates more substantial improvements. These results highlight the effectiveness of the proposed framework.

Type	Sort	Rest		TwEmo	
		Acc	F1	Acc	F1
Wrong First	Desc‡	81.83	71.76	66.92	62.88
	Asc	81.35	71.12	65.94	62.18
Correct First	Desc	81.77	70.04	66.97	62.57
	Asc	82.08	70.77	65.47	60.90
Alternating	Desc	81.74	71.05	66.71	63.11
	Asc	82.10	70.83	66.60	62.11

Table 10: Effect of the order of examples (BM25+Ours). Desc means descending order and Asc means ascending order. The standard setting is marked by ‡.

Language Model	Method	Rest		TwEmo	
		Acc	F1	Acc	F1
BERT-base†	FT (ALL)†	84.46*	76.98*	81.28	77.84
	FT (CP)†	73.19	64.59	65.87	62.53
Mistral-7B-inst	Random	81.19	73.66	76.00	66.30
	+ Ours	$83.78_{+2.59}$	$75.05_{+1.39}$	$76.53_{+0.53}$	$68.50_{+2.20}$
	BM25	80.25	72.03	75.37	68.47
	+ Ours	$83.07_{+2.82}$	$73.47_{+1.44}$	$76.07_{+0.70}$	$69.74_{+1.27}$
	K-Means	81.68	73.64	75.55	66.07
	+ Ours	$82.93_{+1.25}$	$74.08_{+0.44}$	$76.88_{+1.33}$	$68.04_{+1.97}$
GPT-3.5 Turbo	Random	83.32	68.46	73.92	69.97
	+ Ours	$85.17_{+1.85}$	$77.87_{+9.41}$	$75.79_{+1.87}$	$72.11_{+2.14}$
	BM25	83.87	70.15	72.41	68.17
	+ Ours	$82.13_{-1.74}$	$74.34_{+4.19}$	$75.21_{+2.80}$	$71.19_{+3.02}$
	K-Means	84.23	70.15	73.56	69.68
	+ Ours	$85.22_{+0.99}$	$77.29_{+7.14}$	$75.89_{+2.33}$	$72.12_{+2.44}$
GPT-4	K-Means	88.83	81.90	82.13	77.80
	+ Ours	**89.45**$_{+0.62}$	**82.29**$_{+0.39}$	**82.27**$_{+0.14}$	**78.10**$_{+0.30}$

Table 11: Results of language model generalization. Fine-tuning approaches are marked by †. Results with ∗ are from (Song et al., 2019). The best scores across all methods are in bold.

Task	Dataset	Train	Dev	Test	Classes	Labels
Sentiment Classification	SST-2	6,920	872	1,821	2	positive, negative
	TwSenti	45,615	2,000	12,284	3	positive, negative, neutral
	Poem	843	105	104	3	positive, negative, neutral
	Finance	1,358	453	453	3	positive, negative, neutral
Aspect Sentiment Classification	Rest	3,608	454	1,119	3	positive, negative, neutral
	Laptop	2,282	283	682	3	positive, negative, neutral
	Twitter	6,248	-	692	3	positive, negative, neutral
Emotion Detection	EmoC	30,160	-	5,509	4	happy, sad, angry, others
	TwEmo	3,257	374	1,421	4	anger, joy, optimism, sadness
Stance Detection	P-Stance	17,756	2,282	2,207	2	favor, against
Irony Detection	TwIrony	2,862	955	784	2	irony, non-irony
NLI	MNLI	263,789	3,000	9,796	3	entailment, contradiction, neutral

Table 12: The statistics of investigated datasets.

	Feedback on correct examples	Feedback on wrong examples
FC-1	You are correct!	You are wrong!
FA-1	Make sure your prediction is accurate.	Stay determined and keep moving forward.
FC-2	The answer is accurate.	The answer is incorrect.
FA-2	Please keep up the good work.	Please adjust to ensure the prediction is correct.

Table 13: Different feedback prompts. FC denotes feedback on correctness and FA denotes feedback for analysis.

Task	In-context Learning Prompts
SC	**Instruction:** Recognize the sentiment of the sentence. Here are some examples: **Examples:** ... Sentence: *text* x_i Prediction: *prior prediction* y'_i Correct Label: *label* y_i *feedback on correct (or wrong) examples* ... **Test Input:** Sentence: *text* x Correct Label:
ASC	**Instruction:** Recognize the sentiment polarity for the given aspect term in the sentence. Here are some examples: **Examples:** ... Sentence: *text* x_i What is the sentiment polarity of the aspect *aspect* ? Prediction: *prior prediction* y'_i Correct Label: *label* y_i *feedback on correct (or wrong) examples* ... **Test Input:** Sentence: *text* x What is the sentiment polarity of the aspect *aspect* ? Correct Label:
ED	**Instruction:** Recognize the emotion of the sentence. Here are some examples: **Examples:** Same as SC **Test Input:** Sentence: *text* x Correct Label:
Stance Detection	**Instruction:** Recognize the stance of the sentence to the given target. Here are some examples: **Examples:** ... Sentence: *text* x_i What is the attitude of sentence toward target *target* ? Prediction: *prior prediction* y'_i Correct Label: *label* y_i *feedback on correct (or wrong) examples* ... **Test Input:** Sentence: *text* x What is the attitude of sentence toward target *target* ? Correct Label:
Irony Detection	**Instruction:** Determine whether the sentence is ironic or not. Here are some examples: **Examples:** Same as SC **Test Input:** Sentence: *text* x Correct Label:
NLI	**Instruction:** Recognize textual entailment between the 2 texts. Here are some examples: **Examples:** ... Premise: *text1* x_{i1} Hypothesis: *text2* x_{i2} Prediction: *prior prediction* y'_i Correct Label: *label* y_i *feedback on correct (or wrong) examples* ... **Test Input:** Premise: *text1* x_{test1} Hypothesis: *text2* x_{test2} Correct Label:

Table 14: The prompt and format of in-context learning for each task.