

# Corrective In-Context Learning: Evaluating Self-Correction in Large Language Models

Mario Sanz-Guerrero<sup>1</sup> and Katharina von der Wense<sup>1,2</sup>

<sup>1</sup>Johannes Gutenberg University Mainz, Germany

<sup>2</sup>University of Colorado Boulder, USA

{msanzgue, k.vonderwense}@uni-mainz.de

## Abstract

In-context learning (ICL) has transformed the use of large language models (LLMs) for NLP tasks, enabling few-shot learning by conditioning on labeled examples without finetuning. Despite its effectiveness, ICL is prone to errors, especially for challenging examples. With the goal of improving the performance of ICL, we propose *corrective in-context learning* (CICL), an approach that incorporates a model’s incorrect predictions alongside ground truth corrections into the prompt, aiming to enhance classification accuracy through self-correction. However, contrary to our hypothesis, extensive experiments on text classification tasks demonstrate that CICL consistently underperforms standard ICL, with performance degrading as the proportion of corrections in the prompt increases. Our findings indicate that CICL introduces confusion by disrupting the model’s task understanding, rather than refining its predictions. Additionally, we observe that presenting harder examples in standard ICL does not improve performance, suggesting that example difficulty alone may not be a reliable criterion for effective selection. By presenting these negative results, we provide important insights into the limitations of self-corrective mechanisms in LLMs and offer directions for future research.<sup>1</sup>

## 1 Introduction

In-context learning (ICL; Brown et al., 2020) has emerged as a powerful paradigm for leveraging large language models (LLMs) for various NLP tasks, including text classification. Unlike traditional approaches that require finetuning on task-specific data, ICL allows models to make predictions based on a small number of examples presented in the prompt, effectively transforming LLMs into flexible tools for few-shot learning. This paradigm has demonstrated remarkable performance in numerous scenarios, often approaching

<sup>1</sup>Code and data are available at <https://github.com/mario-sanz/CICL>.

Standard In-Context Learning	Corrective In-Context Learning
Text: Oh, great, another Monday. Label: negative	Text: Oh, great, another Monday. Predicted label: positive Correct label: negative
Text: This is the worst movie ever! Label: negative	Text: This is the worst movie ever! Predicted label: negative Correct label: negative
Text: I've seen worse. Label:	Text: I've seen worse. Predicted label: negative Correct label:

Figure 1: CICL prompt example. The model is tasked with predicting the correct label based on its own prediction, using examples that include both the prediction and its correction.

or surpassing finetuned models on specific tasks (Brown et al., 2020).

Although effective, ICL is susceptible to errors, particularly with difficult examples. We set out to further improve ICL by introducing a novel extension of it, which we term *corrective in-context learning* (CICL). Our approach is based on the idea that providing the model with its initial predictions alongside the correct ground truth labels can serve as a feedback mechanism, enabling the model to refine its understanding and improve subsequent predictions.

CICL builds on the intuition that LLMs, when exposed to their own errors in conjunction with the correct answers, might learn from these mistakes within the confines of a single interaction. For instance, if a model predicts “positive” for a sentence that is actually “negative,” presenting this prediction–error pair could prompt the model to recalibrate its internal representations and make better-informed predictions for similar inputs.

However, our empirical evaluation reveals a different story. While the approach is theoretically promising, our experiments show that CICL fails to deliver the anticipated improvements. In some cases, performance even deteriorates compared to

standard ICL. This underscores the complexity of self-correction in LLMs and the importance of rigorously evaluating intuitive extensions to ICL.

In this paper, we present a detailed investigation of CICL for text classification tasks. We outline our methodology, describe the experimental setup, and analyze the results, focusing on understanding why this approach falls short of expectations. By sharing these negative results, we aim to contribute to the growing discourse on the limitations of LLMs and to inspire future research to develop more effective self-correction mechanisms.

## 2 Related Work

**In-Context Learning** Introduced by Brown et al. (2020), ICL enables LLMs to perform tasks via few-shot prompting without parameter updates. Subsequent work has explored factors influencing ICL performance, including example selection (Liu et al., 2022) and ordering (Lu et al., 2022), raising questions about its robustness to errors or ambiguous examples. To enhance ICL, prompt tuning methods (Lester et al., 2021; Li and Liang, 2021) optimize soft prompts, improving adaptation while maintaining frozen model parameters, bridging the gap between ICL and finetuning.

**Self-Correction in LLMs** Prior work explores self-correction through iterative refinement (Madaan et al., 2023), finetuning on self-generated data (Huang et al., 2023), or reinforcement learning (Kumar et al., 2024). These approaches, however, require multi-step processes, parameter updates, or external rewards. In contrast, we investigate whether LLMs can self-correct *in-context* by directly incorporating corrections into the prompt. Closest to our setting, Monea et al. (2024) show that LLMs struggle to improve from binary reward signals (correct/incorrect) in ICL scenarios. While their feedback is implicit, we extend this observation to *explicit ground truth corrections* and similarly find degraded performance.

## 3 Methodology

### 3.1 Standard In-Context Learning

ICL leverages the few-shot capabilities of LLMs by conditioning predictions on a prompt constructed from a small set of labeled examples. Formally, given a dataset of examples  $\{x_i, y_i\}$ , where  $x_i$  represents a text input and  $y_i$  is a class label verbalized as one of the possible labels in the set  $\mathcal{L}$ , the model

$M$  predicts the label for a query  $x$  using a context prompt  $C_k$  of  $k$  examples:

$$\hat{y} = \arg \max_{y \in \mathcal{L}} P_M(y \mid C_k, x) \quad (1)$$

ICL operates under the assumption that the few-shot examples encapsulate task-relevant patterns, allowing the model to generalize to unseen queries. This paradigm has demonstrated remarkable versatility and competitive performance across numerous tasks and datasets (Brown et al., 2020). However, its effectiveness is heavily dependent on the quality and representativeness of the examples in the prompt (Liu et al., 2022). Furthermore, biases in the selection or ordering of examples, as well as the inherent biases of the model, can significantly influence predictions, posing challenges for consistent and reliable performance (Zhao et al., 2021).

In this study, standard ICL is the baseline against which we compare the proposed CICL approach, providing insights into whether iterative feedback mechanisms can address some of these challenges. The structure of the ICL prompt, along with an example, is provided in Appendix B.1.

### 3.2 Corrective In-Context Learning

CICL extends the standard ICL paradigm by introducing a second round of predictions informed by feedback from the first round. The algorithm is as follows:

1. *Initial prediction*: A prompt is constructed from  $k$  randomly selected examples from the training data, each consisting of an input-output pair  $(x_i, y_i)$ , for  $i \in 1, \dots, k$ . This prompt is used to generate prediction  $\hat{y}$  for the test input  $x$  via standard ICL.
2. *Feedback incorporation*: We perform ICL for the  $k$  examples selected in Step 1, getting the model’s predictions  $\hat{y}_i$  for each example  $(x_i, y_i)$ , with the remaining  $k - 1$  examples acting as few-shot examples. Using these predictions, a feedback-augmented prompt (CICL) is constructed from triplets of the form  $(x_i, \hat{y}_i, y_i)$ , where  $x_i$  is the input text,  $\hat{y}_i$  is the predicted label, and  $y_i$  is the true label. These triplets explicitly highlight the model’s errors (when  $\hat{y}_i \neq y_i$ ) and correct predictions (when  $\hat{y}_i = y_i$ ), providing the model with a context to learn from its earlier outputs.

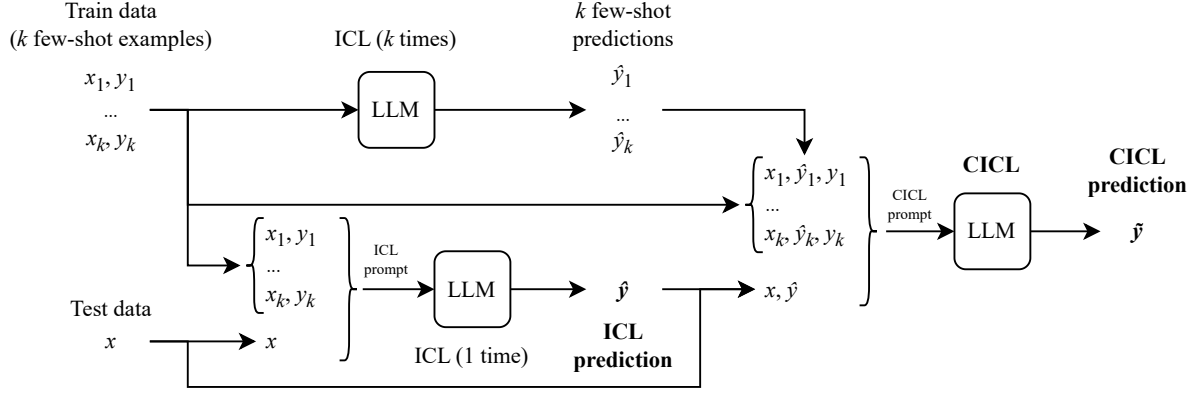


Figure 2: Diagram of the CICL methodology. For each test instance  $x$ ,  $k$  few-shot examples  $(x_1, y_1), \dots, (x_k, y_k)$  are selected. Standard ICL generates predictions  $\hat{y}_1, \dots, \hat{y}_k$  for these examples, which are used to build the CICL prompt. This, combined with the initial ICL prediction  $\hat{y}$  for  $x$ , allows the model to predict the corrected label  $\tilde{y}$ .

3. *Corrective prediction:* To perform the CICL prediction for the input  $x$ , the feedback prompt built in Step 2 is expanded to include  $x$  along with its initial prediction  $\hat{y}$  (obtained in Step 1). The task for the model is then to predict a corrected label  $\tilde{y}$  for  $x$ , leveraging the feedback triplets to generalize how errors were corrected in the few-shot examples.

This iterative setup allows the model to “see” its own mistakes and explicitly incorporate the correct answers during the second round, hypothesizing that this feedback mechanism can improve its performance. Figure 2 shows a diagram of the proposed methodology for CICL, and the structure of the CICL prompt, along with an example, is provided in Appendix B.2.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets** We evaluate CICL using 17 text classification datasets widely adopted in previous work. These datasets span a variety of tasks, including sentiment analysis, topic classification, and more. Further details are provided in Appendix A.

**Models** To explore the effectiveness of CICL across different models and ensure findings are not model-specific, we use four recent LLMs: Llama-3.1 (8B; Dubey et al., 2024), GPT-J (6B; Wang and Komatsuzaki, 2021), Mistral 7B v0.3 (Jiang et al., 2023) and Qwen2.5 (7B; Qwen et al., 2024). The choice of these relatively small-sized models allows for extensive experimentation while maintaining computational feasibility. However, preliminary experiments with the larger 70B version

of Llama-3.1 yielded similar results to the smaller versions.

**Implementation Details** Following prior work on ICL for text classification, we use simple and unified templates for all datasets and do not include task instructions, keeping human engineering to a minimal level (Min et al., 2022; Fei et al., 2023). Also following prior work, we set  $k = 8$  few-shot examples, which enables incorporating a fair number of corrections in the prompt while keeping computational costs manageable. Preliminary experiments with larger  $k$  values showed similar results, so we stick with  $k = 8$  for simplicity.

To assess how CICL performs with different levels of corrected examples, we introduce varying proportions of corrected examples in the CICL prompt, ranging from 0% (no corrected examples) to 100% (all examples corrected) in increments of 25%. Each proportion determines how many of the  $k$  examples in the CICL prompt are corrected (i.e., their initial ICL prediction was incorrect). This approach allows us to evaluate how the ratio of corrected feedback influences the model’s ability to refine its predictions. To minimize the impact of randomness in the results, every experimental configuration is run using 5 different random seeds.

**Metric** We compare standard ICL and CICL performance using macro-F1 score, which accounts for class imbalance.

### 4.2 Results

Figure 3 compares the performance of standard ICL and CICL across different models. The results present the mean and standard deviation of

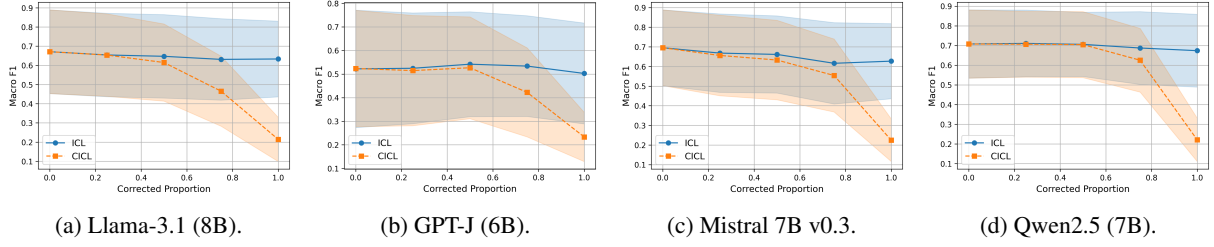


Figure 3: Mean Macro-F1 ( $\pm$  Std Dev) across all datasets for each model, comparing standard ICL (blue) and CICL (orange). As the proportion of corrected examples increases ( $x$ -axis), the performance of CICL decreases ( $y$ -axis).

Macro-F1 scores across all datasets, evaluated for varying proportions of corrected examples in CICL. Detailed results are available in Appendix C.

Contrary to our hypothesis, CICL consistently underperforms standard ICL. When the proportion of corrected examples is 0%—or even 25% for some models—both methods yield equivalent performance, as the “corrective” task essentially reduces to replicating the previously predicted label. However, as the proportion of corrected examples increases, CICL’s performance deteriorates further, highlighting the model’s struggle to integrate feedback from corrected examples effectively.

Our findings suggest that the corrective nature of CICL introduces confusion rather than guiding the model towards improved predictions. Swapping (correcting) labels in the few-shot examples appears to disrupt the model’s internal representations, making it harder to generalize and refine predictions, especially when encountering harder examples.

### 4.3 Statistical Analyses

To assess the significance of these performance differences, we conduct statistical tests. A Shapiro-Wilk test for normality reveals no statistical evidence supporting normality in the distributions of ICL and CICL results. Therefore, we employ non-parametric tests to evaluate the impact of correction proportions on performance.

A Wilcoxon signed-rank test is used to determine the threshold at which ICL becomes statistically superior to CICL. As shown in Table 1, from a correction proportion of 25% onward, ICL demonstrates statistically significant superiority over CICL, reinforcing our conclusion that CICL’s self-corrective mechanism is ineffective.

Additionally, a Kruskal-Wallis test is conducted to examine variation in performance across correction proportions (for CICL) and example conditions (for ICL). Table 2 highlights significant

Corrected Proportion	Statistic	P-value
0%	178.00	0.7916
25%	9453.00	4e-03*
50%	20439.00	7e-04*
75%	13553.00	1e-16*
100%	984.00	9e-54*

Table 1: Wilcoxon signed-rank test results for different correction proportions. \* indicates significant differences at the 0.01 level.

Method	Statistic	P-value
ICL	12.18	0.02
CICL	677.54	3e-145*

Table 2: Kruskal-Wallis test results for ICL and CICL. \* indicates significant differences at the 0.01 level.

variation in CICL performance as the proportion of corrected examples increases, whereas ICL shows minimal variation across differing example conditions. These results underscore the robustness of ICL across diverse contexts, contrasting with CICL’s instability under varying correction levels.

## 5 Impact of Harder Examples on ICL

Although not the primary objective of this study, an intriguing finding emerges from our results. As shown in Figure 3, increasing the proportion of incorrectly classified examples (i.e., higher “corrected proportions”) in the few-shot context does not improve the performance of standard ICL. In fact, for some models, performance slightly declines as the corrected proportion increases.

Intuitively, one might expect that presenting the model with harder examples—those it initially misclassified—would enhance performance. The rationale is that such examples could help refine the model’s decision boundaries and establish clearer



classification thresholds, improving overall accuracy. However, our results do not support this.

When examining the performance trends of standard ICL (blue lines in Figure 3), we observe that including harder examples in the few-shot context fails to yield any consistent improvement. In some cases, performance remains unchanged or even decreases slightly. Furthermore, statistical analysis using the Kruskal-Wallis test reveals no significant variation in standard ICL performance across different corrected proportions (Table 2, first line). These findings suggest that simply exposing the model to harder examples does not provide the anticipated benefits for few-shot classification tasks.

This observation warrants further investigation into how example difficulty interacts with ICL, particularly in understanding why harder examples fail to contribute to improved model calibration or decision-making in this context.

## 6 Conclusion

In this paper, we introduced CICL, a novel approach aimed at leveraging incorrect model predictions to improve performance through self-correction. By including misclassified examples along with their correct labels in the prompt, CICL sought to refine the model’s predictions in text classification tasks. However, contrary to our initial hypothesis, CICL consistently underperformed ICL across all models and datasets. Our findings revealed that the corrective nature of CICL often led to confusion rather than improvement. The swapping of labels in the few-shot examples disrupted the model’s understanding of the task, resulting in degraded performance as the proportion of corrected examples increased.

Additionally, we explored an auxiliary finding regarding the impact of harder few-shot examples on standard ICL performance. Despite the expectation that presenting harder, misclassified examples could enhance the model’s decision-making, our results showed no significant improvement across varying proportions of examples that required correction. This challenges the assumption that harder examples inherently contribute to better generalization in few-shot learning.

Overall, our study highlights the challenges of incorporating self-corrective mechanisms into LLMs through ICL and demonstrates that harder examples are not necessarily more useful than easier ones in standard ICL. These findings provide

valuable insights for the development of more robust and effective few-shot learning methods with LLMs.

## Limitations

Despite the intuitive appeal of CICL, our findings indicate that it performs worse than standard ICL for text classification tasks. Several factors may contribute to this result. First, our experiments are restricted to small-scale open-source LLMs due to computational constraints. Larger models may exhibit stronger reasoning and adaptation capabilities, potentially improving CICL performance. While preliminary experiments with the 70B version of Llama-3.1 yielded similar results to the smaller models, the impact of even larger models remains an open question. Second, CICL may be more effective for tasks requiring multi-step reasoning, e.g., via chain-of-thought prompting, where the model can benefit from explicit corrective feedback to refine intermediate steps. Third, LLMs are highly sensitive to prompt design. It is possible that alternative prompt formats or different ways of structuring corrective feedback could lead to better results.

## Acknowledgments

The research in this paper was funded by the Carl Zeiss Foundation, grant number P2022-08-009 (MAINCE project).

## References

- Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. [TweetEval: Unified benchmark and comparative evaluation for tweet classification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650. Online. Association for Computational Linguistics.
- 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](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. [The Llama 3 Herd of Models](#). *Preprint*, arXiv:2407.21783.

- Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. 2023. [Mitigating label biases for in-context learning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14014–14031, Toronto, Canada. Association for Computational Linguistics.
- Minqing Hu and Bing Liu. 2004. [Mining and summarizing customer reviews](#). In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04*, page 168–177, New York, NY, USA. Association for Computing Machinery.
- Jiaxin Huang, Shixiang Gu, Le Hou, Yuxin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. [Large language models can self-improve](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics.
- 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](#). *Preprint*, arXiv:2310.06825.
- Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2024. [Training language models to self-correct via reinforcement learning](#). *Preprint*, arXiv:2409.12917.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, et al. 2021. [Datasets: A community library for natural language processing](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.
- Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. [What makes good in-context examples for GPT-3?](#) In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.
- Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. [Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. [Self-refine: Iterative refinement with self-feedback](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- 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 the Association for Information Science and Technology*, 65(4):782–796.
- Sewon Min, Xinxin Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. [Rethinking the role of demonstrations: What makes in-context learning work?](#) In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Giovanni Monea, Antoine Bosselut, Kianté Brantley, and Yoav Artzi. 2024. [Llms are in-context reinforcement learners](#). *Preprint*, arXiv:2410.05362v1.
- Bo Pang and Lillian Lee. 2004. [A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts](#). In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.
- Bo Pang and Lillian Lee. 2005. [Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales](#). In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.
- Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.

Emily Sheng and David Uthus. 2020. [Investigating societal biases in a poetry composition system](#). In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 93–106, Barcelona, Spain (Online). Association for Computational Linguistics.

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 the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Ellen M. Voorhees and Dawn M. Tice. 2000. [Building a question answering test collection](#). In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’00, page 200–207, New York, NY, USA. Association for Computing Machinery.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. <https://github.com/kingoflolz/mesh-transformer-jax>.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. [Character-level convolutional networks for text classification](#). In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. [Calibrate before use: Improving few-shot performance of language models](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

## A Datasets Details

Our experiments utilize 17 datasets spanning various text classification tasks, all of which are commonly used in prior research (Min et al., 2022; Lu et al., 2022; Zhao et al., 2021; Fei et al., 2023). All datasets are accessed through the HuggingFace Datasets library (Lhoest et al., 2021). For evaluation, we use the provided test sets when available. If no test set is provided, we create a stratified development set by sampling from the training data, ensuring the class distribution is preserved. Detailed dataset information, including task type, class counts, and data distribution, is summarized in Table 3.

As shown in Table 3, most datasets are imbalanced. To account for this, we use macro F1 as the evaluation metric, as it equally weighs all classes and ensures a fair assessment of the model’s performance across both frequent and rare classes.

## B Prompt Formats

### B.1 In-Context Learning Prompt

```
Text: {example_1}
Label: {ground_truth_1}

Text: {example_2}
Label: {ground_truth_2}
...
Text: {example_k}
Label: {ground_truth_k}

Text: {input_text}
Label:
```

Figure 4: Prompt format for standard ICL, showing ground truth labels for  $k$  examples.

### Example

Below is an example of a standard ICL prompt for the TREC dataset with  $k = 8$  few-shot examples.

Text: What is the name of the tallest mountain in the world?

Label: location

Text: How many eyes does a bat have?

Label: numeric

Text: What does Ms., Miss, and Mrs. stand for?

Label: abbreviation

Text: What does IQ stand for?

Label: abbreviation

Text: What were the achievements of Richard Nixon?

Label: entity

Text: What is the C programming language?

Label: description

Text: Who was considered to be the father of psychology?

Label: human

Text: What are the top five oil-producing countries in the world?

Label: location

Text: What are the stars made of?

Label:

Dataset	# Classes	Balanced
<i>Sentiment and topic classification</i>		
SST-2 (Socher et al., 2013)	2	✓
SST-5 (Socher et al., 2013)	5	✗
MR (Pang and Lee, 2005)	2	✓
CR (Hu and Liu, 2004)	2	✓
financial_phrasebank (Malo et al., 2014)	3	✗
poem_sentiment (Sheng and Uthus, 2020)	4	✗
Subj (Pang and Lee, 2004)	2	✗
AG News (Zhang et al., 2015)	4	✓
DBpedia (Zhang et al., 2015)	14	✓
TREC (Voorhees and Tice, 2000)	6	✗
<i>Detection</i>		
tweet_eval-hate (Barbieri et al., 2020)	2	✗
tweet_eval-irony (Barbieri et al., 2020)	2	✗
tweet_eval-offensive (Barbieri et al., 2020)	2	✗
tweet_eval-stance_atheism (Barbieri et al., 2020)	3	✗
tweet_eval-stance_feminist (Barbieri et al., 2020)	3	✗
hate_speech18 (Barbieri et al., 2020)	2	✗
ethos-binary (Barbieri et al., 2020)	2	✗

Table 3: Full dataset information.

## B.2 Corrective In-Context Learning Prompt

Text: {example_1}
Predicted label: {predicted_label_1}
Correct label: {ground_truth_1}
Text: {example_2}
Predicted label: {predicted_label_2}
Correct label: {ground_truth_2}
...
Text: {example_k}
Predicted label: {predicted_label_k}
Correct label: {ground_truth_k}
Text: {input_text}
Predicted label: {predicted_label}
Correct label:

Figure 5: Prompt format for CICL, showing predicted and ground truth (“correct”) labels for  $k$  examples.

### Example

Below is an example of a CICL prompt for the TREC dataset. There are  $k = 8$  few-shot examples, with 50% being corrected examples. This means 4 examples are correctly predicted by ICL (in positions 2, 4, 6, and 7), and 4 are corrected examples (in positions 1, 3, 5, and 8). The final

instance represents the current input, where ICL predicted the label “description.” The task of the LLM is to predict the correct label based on the input text, the predicted label, and the corrections made in previous examples. The true label for this instance is “entity,” and the goal of CICL is to make this correction.

Text: What is the name of the tallest mountain in the world?  
Predicted label: entity  
Correct label: location

Text: How many eyes does a bat have?  
Predicted label: numeric  
Correct label: numeric

Text: What does Ms., Miss, and Mrs. stand for?  
Predicted label: description  
Correct label: abbreviation

Text: What does IQ stand for?  
Predicted label: abbreviation  
Correct label: abbreviation

Text: What were the achievements of Richard Nixon?  
Predicted label: human  
Correct label: entity



Text: What is the C programming language?

Predicted label: description

Correct label: description

Text: Who was considered to be the father of psychology?

Predicted label: human

Correct label: human

Text: What are the top five oil-producing countries in the world?

Predicted label: numeric

Correct label: location

Text: What are the stars made of?

Predicted label: description

Correct label:

## C Detailed Results

Table 4 presents the detailed results for all combinations of models, datasets, corrected example proportions, and approaches (ICL and CACL). Each configuration is evaluated five times using different random seeds to ensure diverse selections of examples, minimizing the impact of randomness on the results. The table reports the mean and standard deviation for each configuration, providing a comprehensive view of the performance across variations.

Dataset	Corrected Proportion	Llama-3.1		Mistral 7B		Qwen2.5		GPT-J	
		ICL	CICL	ICL	CICL	ICL	CICL	ICL	CICL
AG News	0%	87.31.6	87.31.6	85.51.0	85.51.0	83.41.7	83.61.6	73.94.4	73.94.4
	25%	83.74.0	84.03.6	86.61.0	86.50.7	85.31.1	85.71.2	74.20.8	74.20.8
	50%	84.13.4	83.04.4	85.02.3	76.413.0	83.22.4	82.02.4	73.97.3	70.57.8
	75%	68.720.2	56.024.0	80.93.9	47.622.8	80.06.1	70.48.7	75.92.4	47.67.3
	100%	83.62.8	13.52.1	85.03.2	12.22.2	85.11.9	15.35.1	77.13.8	14.62.0
CR	0%	93.20.5	93.20.5	93.20.4	93.20.4	93.20.5	93.10.5	87.62.0	87.62.0
	25%	92.01.3	91.81.5	92.00.7	91.50.7	91.81.8	92.00.8	87.82.7	84.87.9
	50%	93.50.7	83.67.3	91.71.9	89.51.0	91.41.4	91.50.9	87.12.3	84.55.4
	75%	91.11.9	55.718.9	87.26.7	71.814.1	89.25.7	75.318.2	84.35.6	58.220.3
	100%	91.11.0	8.00.8	91.41.0	7.80.6	84.37.5	14.95.3	86.25.8	11.94.4
DBpedia	0%	85.61.4	85.61.4	82.62.4	82.62.4	75.72.2	75.82.1	77.41.0	77.41.0
	25%	84.03.0	84.22.9	80.70.9	81.01.3	75.51.4	76.91.3	75.11.6	74.72.2
	50%	80.51.1	84.61.8	76.27.1	79.15.3	74.11.4	81.46.0	75.12.4	73.34.7
	75%	81.61.1	86.63.2	78.01.0	79.34.6	71.21.9	74.76.3	73.81.7	59.011.2
	100%	78.82.1	12.83.4	66.111.8	9.64.4	68.83.4	13.64.8	72.82.6	12.33.8
ethos-binary	0%	72.83.3	72.83.3	79.63.6	79.63.6	80.62.2	80.62.2	56.99.5	56.69.3
	25%	63.210.8	62.710.7	76.15.7	73.14.6	77.74.1	77.93.9	44.615.0	45.314.6
	50%	67.64.6	62.67.8	69.27.0	60.817.6	77.53.0	77.81.8	58.78.4	57.410.9
	75%	55.616.2	46.715.7	70.612.9	55.014.8	78.93.7	71.89.8	55.814.4	53.118.8
	100%	51.47.3	30.82.8	64.313.8	26.74.5	77.92.3	23.83.4	36.30.0	30.10.0
financial_phrasebank	0%	77.98.3	77.98.3	84.10.9	84.10.9	85.43.1	85.23.0	59.512.0	59.512.0
	25%	78.64.8	78.74.8	82.51.2	82.41.3	82.44.3	82.42.9	58.012.0	57.612.3
	50%	74.99.3	68.88.5	83.51.3	83.31.7	84.21.2	83.61.8	61.911.1	61.89.7
	75%	79.48.8	43.410.4	82.23.0	70.311.9	84.71.5	82.32.5	50.28.3	17.30.6
	100%	81.15.5	10.54.3	78.55.3	25.97.8	84.14.0	23.214.9	49.613.8	13.87.9
hate_speech18	0%	53.54.1	53.54.1	67.51.8	67.51.8	68.42.1	68.32.1	38.913.1	38.712.9
	25%	47.98.8	47.48.6	57.26.2	51.313.0	66.02.5	65.72.6	37.210.0	34.410.8
	50%	46.411.1	46.110.7	62.25.0	60.95.7	63.43.3	64.92.1	38.14.6	37.54.6
	75%	36.95.5	39.812.5	52.814.0	49.218.1	64.23.5	58.37.3	37.511.7	42.917.2
	100%	42.813.9	34.36.8	57.84.7	24.24.9	63.62.1	21.44.6	45.53.0	15.811.4
MR	0%	93.60.5	93.60.5	93.80.4	93.80.4	92.60.7	92.60.7	90.71.0	90.71.0
	25%	93.11.3	93.01.0	94.10.3	94.00.4	93.00.6	92.70.6	88.62.8	88.32.6
	50%	93.10.9	89.83.7	93.90.6	92.71.4	92.40.5	91.00.8	81.68.4	81.88.1
	75%	91.74.1	44.58.3	91.62.9	79.410.4	92.60.5	69.014.0	76.021.6	62.715.7
	100%	81.811.3	14.86.9	80.721.7	13.39.7	90.24.2	9.83.0	75.119.1	17.77.9
poem_sentiment	0%	39.43.6	39.43.6	54.05.6	54.05.6	54.03.5	54.03.5	13.04.2	13.04.2
	25%	42.27.1	42.27.1	53.411.3	53.910.3	50.96.6	51.46.3	20.711.0	22.111.1
	50%	41.712.8	40.612.6	57.86.1	57.24.9	53.65.9	54.55.6	22.58.0	28.02.2
	75%	47.212.0	25.19.0	49.511.1	50.39.7	51.57.9	47.06.1	40.44.9	27.712.5
	100%	54.78.6	9.83.6	57.711.6	17.87.8	52.07.0	21.64.2	25.411.7	28.55.0
SST-2	0%	91.61.3	91.61.3	92.41.5	92.41.5	93.81.0	93.81.0	83.09.0	83.09.0
	25%	91.21.1	90.71.8	91.10.9	91.10.8	91.92.8	92.72.6	72.517.6	73.918.4
	50%	90.21.5	86.34.0	89.82.1	89.02.6	92.41.1	94.50.5	80.88.4	80.75.0
	75%	85.29.3	40.58.6	86.17.1	68.821.6	92.03.0	86.05.6	68.920.1	41.223.6
	100%	86.05.0	12.93.8	87.94.5	11.43.3	90.61.1	9.90.4	71.77.8	21.04.2
SST-5	0%	45.42.4	45.42.4	42.64.3	42.64.3	42.02.1	42.02.1	31.59.8	31.59.8
	25%	43.72.1	43.61.5	39.04.6	38.94.7	41.52.2	41.62.2	27.43.4	23.67.2
	50%	39.04.4	40.04.7	40.54.3	39.83.9	42.34.2	38.13.4	34.56.3	31.98.6
	75%	43.32.4	39.43.9	38.84.6	37.32.7	39.16.0	42.03.5	33.23.1	28.37.2
	100%	41.42.2	26.31.2	44.34.1	31.94.2	41.16.6	29.61.9	28.24.5	20.95.2
Subj	0%	87.17.9	87.17.9	75.319.5	75.319.5	79.76.3	79.46.3	43.012.0	43.012.0
	25%	86.93.3	86.03.2	68.313.3	62.414.2	80.83.7	70.63.9	55.718.3	42.914.1
	50%	83.64.5	57.213.2	59.019.0	51.521.2	82.12.4	68.43.6	57.215.4	49.811.4
	75%	82.06.8	25.07.2	37.44.9	54.818.1	78.46.8	47.812.7	70.43.6	44.314.7
	100%	82.49.3	15.25.9	47.920.3	28.28.1	84.42.6	14.61.9	39.99.7	32.81.2
TREC-6	0%	75.07.3	75.07.3	56.67.6	56.67.6	74.73.6	74.73.6	51.27.7	51.27.7
	25%	74.67.4	74.57.7	56.912.2	57.212.2	81.63.9	81.13.6	46.83.6	47.03.3
	50%	75.83.5	75.62.9	60.14.9	59.43.0	74.85.0	77.13.3	49.97.1	49.42.2
	75%	77.73.0	69.68.4	49.916.4	52.26.7	81.71.9	63.910.3	48.77.7	44.79.9
	100%	66.89.0	13.46.0	50.613.6	22.67.2	77.02.6	9.14.3	42.96.4	20.91.5
tweet_eval_atheism	0%	21.210.1	21.210.1	27.610.3	27.610.3	34.04.3	34.04.3	12.11.3	12.11.3
	25%	26.64.8	28.33.1	33.211.5	33.211.5	37.15.9	37.75.6	20.110.6	21.57.7
	50%	31.59.0	34.48.1	33.76.5	35.88.0	40.97.9	37.96.0	16.65.6	23.110.7
	75%	42.57.0	38.410.8	37.47.3	30.07.1	36.611.7	36.611.1	18.46.1	23.04.3
	100%	41.08.1	22.54.9	39.010.9	16.83.6	35.28.1	23.53.2	17.57.1	29.14.4
tweet_eval_feminist	0%	48.710.1	48.710.1	56.36.0	56.36.0	55.64.2	55.64.2	25.68.5	28.77.2
	25%	39.610.8	39.610.8	41.87.0	42.77.8	62.13.7	61.63.1	25.57.7	26.17.0
	50%	37.76.5	41.25.2	41.116.4	43.913.9	56.83.7	57.73.3	28.47.7	23.810.0
	75%	43.58.5	47.36.0	43.412.2	51.68.9	56.74.1	56.92.9	27.78.8	21.06.8
	100%	44.45.6	33.812.0	49.79.5	29.111.7	47.36.1	33.96.8	37.77.5	16.92.1
tweet_eval_hate	0%	52.35.3	52.35.3	67.53.6	67.53.6	61.31.8	61.31.8	40.45.1	40.35.2
	25%	51.04.5	50.94.5	66.54.5	60.712.4	61.14.0	61.53.9	46.26.5	43.86.5
	50%	51.88.5	49.97.5	66.74.2	52.08.5	62.42.3	64.81.6	46.66.7	34.24.5
	75%	44.75.2	53.57.7	54.27.5	51.65.2	55.710.1	62.41.6	42.611.5	47.010.5
	100%	51.011.6	37.16.6	52.77.9	35.92.8	51.14.8	40.04.5	43.39.7	36.15.5
tweet_eval_irony	0%	57.01.9	57.01.9	58.93.2	58.93.2	63.62.6	63.62.6	49.62.6	49.62.6
	25%	51.07.3	50.97.2	56.48.6	55.510.2	63.03.8	63.23.4	51.13.1	52.29.9
	50%	48.66.8	43.58.1	51.212.4	43.010.7	62.63.6	65.32.4	47.98.2	46.47.8
	75%	46.77.2	40.35.5	53.89.8	43.98.3	59.36.3	55.38.4	47.25.1	43.64.7
	100%	43.78.1	32.31.3	51.68.0	35.32.4	52.19.3	34.70.7	44.56.4	43.06.4
tweet_eval_offensive	0%	59.44.8	59.44.8	64.51.4	64.51.4	65.01.8	65.11.8	53.25.3	53.25.4
	25%	62.72.0	62.72.0	60.04.5	59.94.5	66.20.9	66.41.1	60.35.9	62.73.0
	50%	60.24.9	58.04.8	62.82.2	61.53.2	65.71.9	67.01.7	60.46.8	61.64.3
	75%	55.013.8	39.45.7	53.512.0	48.75.8	56.513.9	63.23.7	56.614.4	56.97.7
	100%	54.49.1	35.92.6	61.12.9	32.92.7	60.53.7	36.44.3	60.96.6	31.53.4

Table 4: Results (Macro-F1) across datasets, models, and corrected proportions for ICL and CICL.