Pushing The Limit of LLM Capacity for Text Classification

Yazhou Zhang
yzhou_zhang@tju.edu.cn
Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing University
Tianjin University
Tianjin, China

Mengyao Wang
wangmengyao516@outlook.com
Zhengzhou University of Light Industry
Zhengzhou, China

Qiuchi Li*
qiuchi.li@di.ku.dk
Copenhagen University
Copenhagen, Denmark

Prayag Tiwari
prayag.tiwari@ieee.org
Halmstad University
Hulmstad, Sweden

Jing Qin*
harry.qin@polyu.edu.hk
The Hong Kong Polytechnic University
Hong Kong, China

Abstract

In this era of open-ended language modeling, where task boundaries are gradually fading, an urgent question emerges: *have we made significant progress in text classification with the full benefit of LLMs?* To answer this question, we propose RGPT, an adaptive boosting framework tailored to produce a specialized text classification LLM by recurrently ensembling a pool of base learners. The base learners are constructed by adaptively adjusting the distribution of training samples and iteratively fine-tuning LLMs with them. Such base learners are then ensembled to be a specialized text classification LLM, by recurrently incorporating the historical predictions from the previous learners. Through a comprehensive empirical comparison, we show that RGPT significantly outperforms 8 state-of-the-art (SoTA) PLMs and 7 SoTA LLMs on four benchmarks by 2.90% on average. Further evaluation experiments reveal a clear superiority of RGPT over average human classification performance[1].

CCS Concepts

• **Computing methodologies** → **Information extraction**; **Natural language generation**.

Keywords

Text classification, Large language model, Boosting

ACM Reference Format:
Yazhou Zhang, Mengyao Wang, Qiuchi Li, Prayag Tiwari, and Jing Qin. 2025. Pushing The Limit of LLM Capacity for Text Classification. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3701716.3715528

*Corresponding authors.

[1]Our codes are available at https://github.com/annoymity2024/RGPT_2024

WWW Companion '25, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1331-6/2025/04
https://doi.org/10.1145/3701716.3715528

1 Introduction

Recently, remarkable advances in LLMs, e.g., ChatGPT[2], GPT-4, etc., have demonstrated their outstanding performance across downstream NLP tasks. Sustained efforts have been primarily dedicated to two directions: (1) general LLMs capable of providing encyclopaedic domain knowledge and performing well across a range of tasks; (2) specialized LLMs tailored for vertical domains such as healthcare [1], education [4], etc. Strong LLMs intertwined with sophisticated optimization approaches are propelling LLM research to new heights [3, 5].

Despite the spotlight shining brighter on complicated tasks and exquisite domains, text classification languishes in the shadows with limited attention. Hence, an urgent research question emerges:

RQ: *have we made significant progress in text classification with the full benefit of LLMs?*

To answer this question, we present **RGPT**, an adaptive boosting framework designed to investigate the limit of LLMs' classification ability. The main distinction from the recent text classification approaches, e.g., CARP [2], etc., is that RGPT does not directly optimize the prompt space but instead builds a specialized LLM by adjusting sample distribution, thus demonstrating less sensitivity to prompts and stronger stability across various tasks (see Sec. 3.2).

In particular, the base learners are constructed by iteratively fine-tuning LLMs with training samples. The distribution of training samples will be adaptively adjusted based on the error rates of the base learners. The misclassified samples will be given more weight, where the weights of correctly classified samples will be decreased. Such base learners are then ensembled to be a specialized LLM, by taking the prediction and error rate of the previous learner as the contexts to prompt the current learner. This chain-like nature ensures that subsequent learners can improve and complement upon the existing knowledge.

We offer a comprehensive evaluation of the proposed RGPT model across four benchmarks and compare the results against 8

[2]https://chat.openai.com/

SoTA PLMs and 7 SoTA LLMs. The experimental results show the effectiveness of RGPT with the margin of 1.71%, 2.87%, 2.50% and 4.52% for four datasets. Our study reveals that RGPT with only 7 iterations has achieved the state-of-the-art results with performance continuing to grow as the number of iterations increases. A series of sub-experiments also prove that RGPT can universally boost varies base model structures. Hence, our study comes to a clear conclusion: *our approach has pushed the limit of LLM capacity for text classification.* The main contributions are concluded as follows:

- We make the first attempt to explore the ongoing research value of text classification in the era of LLMs.
- We propose RGPT, an adaptive boosting framework to push the limit of LLMs' classification ability.
- Comprehensive experiments demonstrate the effectiveness of RGPT in zero-shot text classification.

Algorithm 1 Recurrent ensemble Learning of RGPT

Require: $\mathcal{D}^{(0)}$: Training dataset with N samples $(x_i^{(0)}, y_i^{(0)})$, $\mathcal{LM}_0$: LLaMA 2, K: Number of base learners
Ensure: $\mathcal{M}_{ensemble}$: Recursively ensembled model
1: Initialize weights $\mathcal{W}^{(0)} = \left\{w_1^{(0)}, \ldots, w_N^{(0)}\right\}$, where $w_i^{(0)} = \frac{1}{N}$ for all $i \in N$
2: **for** $k = 1, 2, \ldots, K$ **do**
3: Construct prompt $\text{Prompt}_i^{(k)} = INS_i \oplus x_i^{(k)}$
4: Fine-tune $\mathcal{LM}_k$ with weighted samples: $\mathcal{LM}_k = argmin_{\theta^{(k)}} \sum_{\mathcal{D}^{(k)}} w_i^{(k)} \cdot \mathcal{L}(y_i^{(k)}, \text{f}_k(x_i^{(k)}; \theta^{(k)}))$
5: Compute error $\epsilon^{(k)}$ and weight coefficient $\alpha^{(k)} = \log \frac{1-\epsilon^{(k)}}{\epsilon^{(k)}} + \log(c-1)$
6: Update data weights:
$$\mathcal{W}_i^{(k+1)} = \frac{w_i^{(k)}}{Z_k} e^{-\alpha^{(k)}} \quad \text{if correctly classified, else} \quad \frac{w_i^{(k)}}{Z_k} e^{\alpha^{(k)}}$$
7: **for** $k = 1, 2, \ldots, K$ **do**
8: Forward prompt through $\mathcal{LM}_k$ and obtain $\hat{y}_i^{(k)}$
9: Update prompt for next iteration:
$$\text{Prompt}_i^{(k+1)} = \text{Prompt}_i^{(k)} \oplus \{\hat{y}_i^{(k)}, \epsilon^{(k)}\}$$
10: **return** $\mathcal{M}_{ensemble} = F(\mathcal{LM}_1, \mathcal{LM}_2, \ldots, \mathcal{LM}_K)$

2 The Proposed Framework: RGPT

The RGPT model is shown in Fig. 1 and Algorithm 1.

2.1 Initialization and Base Learner Selection

Let $\mathcal{D}^{(0)}$ be the initial training set including N samples. Each sample $(x_i^{(0)}, y_i^{(0)}) \in \mathcal{D}^{(0)}$, where $x_i^{(0)} \in \mathcal{X}$ is an input document and $y_i^{(0)} \in \mathcal{Y}$ its corresponding label.

(1) Weight initialization. Suppose $\mathcal{W}^{(0)} = \left\{w_1^{(0)}, w_2^{(0)}, ..., w_N^{(0)}\right\}$, where $\mathcal{W}^{(0)}$ represents the weight distribution of the initial training samples. Each sample will be initialized as the same weight, i.e., $w_i^{(0)} = \frac{1}{N}$, where $\mathcal{W}^{(0)} \sim U\left(\frac{1}{N}, \frac{1}{N}, ..., \frac{1}{N}\right)$. These weights will later be updated based on the error rate of the base learner.

(2) Initial base learner selection. We prove that our model works almost equally well on different base learners such as PLMs (i.e., RoBERTa) and LLMs (i.e., Alpaca, LLaMA 2, ChatGLM 2). LLaMA 2 is selected as an initial base learner $\mathcal{LM}_0$, in view that it empirically yields the best result (see Sec. 3.2).

2.2 Constructing Base Learners

(1) Prompt construction. We follow the zero-shot standard IO prompting paradigm. At each iteration k, the zero-shot prompt template consists of two components: task instruction INS_i and input document $x_i^{(k)}$. Task instruction INS_i provides specifications for a text classification target and states the output constraint, e.g., "*Classify the SENTIMENT of the INPUT, and assign an accuracy label from ['Positive', 'Negative'].* ".

(2) Fine-tuning LLMs with training samples. The k^{th} base learner $\mathcal{LM}_k$ involves fine-tuning an initial base learner $\mathcal{LM}_0$ using the current training samples with the weight distribution, $\mathcal{W}^{(k)} = \left\{w_1^{(k)}, w_2^{(k)}, ..., w_N^{(k)}\right\}$, effectively adjusting the model's focus on challenging samples. The objective is achieved by minimizing the weighted loss function:

$$\mathcal{LM}_k = \arg \min_{\theta^{(k)}} \sum_{\mathcal{D}^{(k)}} w_i^{(k)} \cdot \mathcal{L}(y_i^{(k)}, f(x_i^{(k)}; \theta^{(k)})) \quad (1)$$

where $x_i^{(k)}$ denotes the i^{th} input document at k^{th} iteration, $\mathcal{D}^{(k)}$ represents the document set at k^{th} iteration, $\theta^{(k)}$ represents the parameters, $\mathcal{L}$ is the loss function, $f(\cdot)$ is an initial base learner $\mathcal{LM}_0$ (e.g., LLaMA 2 in this work).

(3) Updating the weight distribution. Then, we compute its error rate $\epsilon^{(k)}$ and weight coefficient $\alpha^{(k)}$, and thus update the distribution of training samples to guide the next iteration's focus towards misclassified samples:

$$\begin{aligned} \epsilon^{(k)} &= Pr_{i \sim \mathcal{D}^{(k)}} \left[\mathcal{LM}_k\left(x_i^{(k)}\right) \neq y_i^{(k)}\right] \\ \alpha^{(k)} &= log\frac{1-\epsilon^{(k)}}{\epsilon^{(k)}} + log\,(c-1) \\ \mathcal{W}^{(k+1)} &= \frac{\mathcal{W}^{(k)}}{Z_k} \times \begin{Bmatrix} e^{-\alpha^{(k)}} & if\ \mathcal{LM}_k\left(x_i^{(k)}\right) = y_i^{(k)} \\ e^{\alpha^{(k)}} & if\ \mathcal{LM}_k\left(x_i^{(k)}\right) \neq y_i^{(k)} \end{Bmatrix} \end{aligned} \quad (2)$$

where $Pr_{i \sim \mathcal{D}^{(k)}}$ represents the proportion of the number of misclassified samples to the total number of samples, $\alpha^{(k)}$ represents the importance of $\mathcal{LM}_k$. When $\epsilon^{(k)} \leq 0.5$, $\alpha^{(k)} \geq 0$, and $\alpha^{(k)}$ increases as $\epsilon^{(k)}$ decreases. c denotes the number of class (e.g., $c \in [0, 1]$ for sentiment classification task), Z_k represents the normalizing factor.

In practice, there are three common methods for updating the dataset distribution using weights: (1) samples with higher weights are more likely to be selected for training subsequent base classifiers; (2) during the optimization of the loss function, misclassified samples contribute more to the total loss based on their weights; (3) samples with higher weights are replicated proportional to their weights, approximating the effect of the first method.

This paper employs both the second and third methods. However, to avoid overfitting that might result from directly duplicating samples, this paper utilizes GPT-4 to generate stylistically similar samples instead of direct replication. This method offers three advantages: (1) it increases data diversity, reducing the model's over-reliance on specific samples; (2) it enriches the feature distribution of the dataset, improving the representativeness; (3) it enhances the model's robustness to input variations.

After K iterations, we construct K complementary and strong base learners $\{\mathcal{LM}_1, \mathcal{LM}_2, ..., \mathcal{LM}_K\}$.

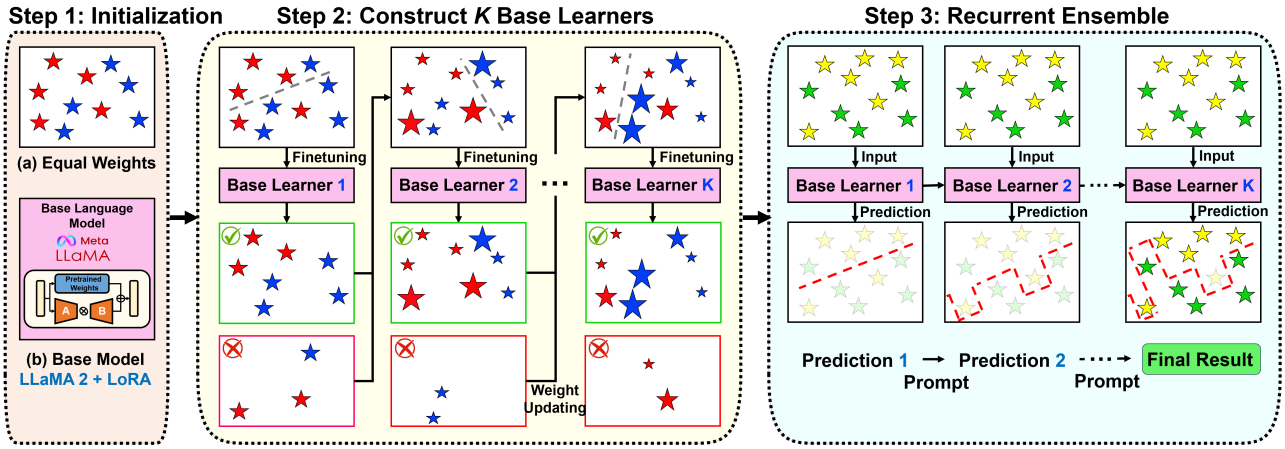

Figure 1: Overview of RGPT, where stars represent training samples and the size of the stars indicates the weights.

2.3 Recurrently Ensembling the Base Learners

We propose a recurrent ensembling approach, which selectively leverages the historical outputs generated by the previous learners. More specifically, the prediction result $\hat{y}_i^{(k-1)}$ of the previous learner $\mathcal{LM}_{k-1}$ along with its error rate $\epsilon^{(k-1)}$ will be incorporated into the input prompt for the next learner $\mathcal{LM}_k$. $\hat{y}_i^{(k-1)}$ is considered the supplementary knowledge for $\mathcal{LM}_k$. The error rate $\epsilon^{(k-1)}$ acts as a trustworthiness metric, determining whether to rely on and adopt the prediction result of $\mathcal{LM}_{k-1}$, which can be written as:

$$\hat{y}_i^{(k)} = \mathcal{LM}_k\left(\hat{y}_i^{(k-1)}, \epsilon^{(k-1)}, x_i^{(k)}\right) \tag{3}$$

The following is an example for sentiment classification:

Instruction: *Given the INPUT, it is known that the prediction result of the previous classifier is "Positive", BUT its error rate is 21%. So there's a 21% chance that this result is wrong. You can freely refer to or abandon this result.*
Based on the prediction result and error rate, classify the SENTIMENT of the INPUT, and assign an accuracy label from ['Positive', 'Negative'].
INPUT: *Contains no wit, only labored gags.*
SENTIMENT: (*Negative*)

Finally, a strong, specialized LLM is constructed by the above-mentioned recurrently ensembling approach:

$$\mathcal{M}_{ensemble} = Recurrent(\mathcal{LM}_1, \mathcal{LM}_2, \ldots, \mathcal{LM}_K) \tag{4}$$

3 Experiments

3.1 Experiment Setups

Datasets. We select four widely used benchmarks: SST-2, MR, AG News, Ohsumed[3]. The statistics are shown in Table 1.

Baselines. A wide range of SoTA baselines are included for comparison. They are: (1) **RoBERTa**, (2) **XLNet**, (3) **RoBERTa-GCN**, (4) **DeBERTa**, (5) **ERNIE** and (6) **T5** are six strong PLMs for text classification via masked language modeling and pretrained representations. (7) **E2SC-IS** selects the most representative documents for training classification model. (8) **ContGCN** focuses on the misclassifed training samples as the target. (9) **BBTv2**, (10) **PromptBoosting**, (11) **CARP** and (12) **QLFR** are four SoTA prompt based approaches for text classification. (13) **ChatGLM 2**, (14) **LLaMA 2** and (15) **GPT-4** are three SoTA general LLMs.

[3]http://davis.wpi.edu/xmdv/datasets/ohsumed.html

Table 1: Dataset statistics.

Dataset	Task	Class	Avg. Length	#Train	#Test
SST-2	Sentiment	2	17	6,920	1821
MR	Sentiment	2	20	8,662	2,000
AG News	News	4	47	120,000	7,600
Ohsumed	Topic	23	136	3,357	4,043

3.2 Results and Analysis

(1) Main results. We report both **Accuracy** and **Macro-F1** results in a zero-shot setting in Table 2. The mean and variance over 5 runs are calculated. We observe that RGPT ($K = 7$) consistently achieves state-of-the-art performance on four datasets, i.e., 1.53%↑, 2.46%↑, 1.00%↑, 3.80%↑ respectively. RGPT achieves better performance when $K = 14$, which are 98.84%, 97.91%, 74.83%, 94.80% respectively, by a significant margin of 2.90%↑ on average. Although LLMs (e.g., ChatGLM 2, LLaMA 2, GPT-4) excel in general-domain tasks, their adaptation to text classification remains limited. Fine-tuning LLaMA 2-13B or optimizing prompt space through methods like QLFR, BBTv2, PromptBoosting, and CARP improves performance. However, RGPT outperforms PLMs based, prompt based and other fine-tuning approaches. We have proven that the performance of RGPT will further improve as the number of iterations increases.

(2) Ablation Study. Table 3 shows the result of ablation studies. For *w/o Boosting*, we choose to directly fine-tune LLaMA 2-7B with initial training samples, removing the boosting strategy. For *w/o LLM*, we substitute LLaMA 2-7B with RoBERTa to be the backbone model. For *w/o Recurrent ensemble*, we choose the stacking approach to combine K strong base learners. From the experiment results

Table 2: Performance on four datasets. Bold and blue indicate the best and second-best results. ♠ represents significance improvement over the best baseline via t-test ($p < 0.05$).

Method	SST-2		AG		Ohsumed		MR		Avg. of Acc.	
	Acc.	Ma-F1	Acc.	Ma-F1	Acc.	Ma-F1	Acc.	Ma-F1	No Ohsumed	All
RoBERTa	96.40	96.23	94.69	94.35	72.80	72.57	89.42	-	93.50	88.32
XLNet	96.80	96.67	95.51	95.18	70.70	70.41	87.20	-	93.17	87.55
RoBERTa-GCN	95.80	-	95.68	-	72.94	-	89.70	-	93.73	87.53
DeBERTa	94.75	94.15	95.32	-	75.94	-	90.21	90.70	93.43	89.01
ERNIE	97.80	-	-	-	73.33	-	89.53	-	-	-
T5-11B	97.50	97.18	92.21	-	51.72	44.10	91.15	-	93.62	83.15
E2SC-IS	-	93.10	-	86.30	-	76.10	-	88.60	89.33	86.02
ContGCN	-	-	-	-	73.40	-	91.30	-	-	-
BBTv2	90.33	-	85.28	-	-	-	83.70	-	86.44	-
PromptBoosting	87.60	-	85.20	-	-	-	84.70	-	85.83	-
CARP	97.39	97.14	96.40	-	-	-	92.39	-	95.39	-
ChatGLM 2	81.36	80.11	83.67	83.67	54.33	41.84	74.39	74.27	79.57	74.01
LLaMA 2	60.50	61.08	79.40	80.67	48.08	40.21	71.49	71.03	62.69	64.89
QLFR	-	-	89.14	89.28	61.10	51.85	81.70	81.72	-	-
GPT-4	82.52	81.17	84.62	84.50	55.20	51.26	77.90	77.63	81.68	75.06
RGPT(k=7)	**98.68**$^{\spadesuit}_{\pm 0.2}$	**98.67**	**97.61**$^{\spadesuit}_{\pm 0.3}$	**97.52**	**77.41**$^{\spadesuit}_{\pm 0.2}$	**73.68**	**94.27**$^{\spadesuit}_{\pm 0.5}$	**94.15**	**96.85**	**91.99**$^{\spadesuit}$
Gain △	0.90%	1.53%	1.26%	2.46%	1.94%	1.00%	2.03%	3.80%	1.53%	3.35%
RGPT(k=14)	**99.12**$^{\spadesuit}_{\pm 0.3}$	**98.84**	**98.20**$^{\spadesuit}_{\pm 0.4}$	**97.91**	**79.36**$^{\spadesuit}_{\pm 0.3}$	**74.83**	**95.40**$^{\spadesuit}_{\pm 0.3}$	**94.80**	**97.57**	**93.04**$^{\spadesuit}$
Gain △	1.35%	1.71%	1.87%	2.87%	4.50%	2.50%	3.26%	4.52%	2.29%	4.53%

Table 3: Ablation study in a zero-shot setting.

Method	SST-2	AG News	Ohsumed	MR
w/o Boosting	89.23	90.53	67.73	88.08
w/o LLM	97.47	95.84	74.70	93.28
w/o Recurrent ensemble	98.61	97.24	76.17	93.52
RGPT	98.68	97.61	77.41	94.27

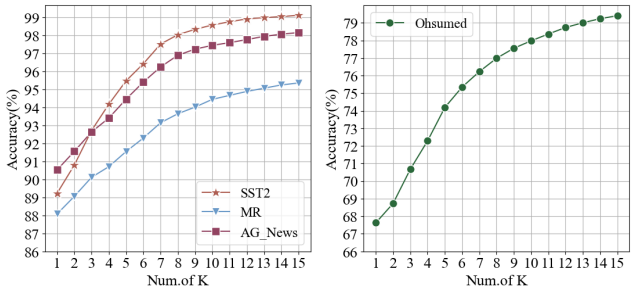

Figure 2: RGPT performance with increasing learners.

above, we highlight the following conclusions: (a) boosting LLM making the greatest contribution in improving the classification performance; (b) LLMs demonstrating greater advancedness over PLMs for text classification; (c) the effectiveness of our proposed recurrent ensembling approach.

(3) Effect of K. We empirically present the relationship between the number of learners and the model performance in Fig. 2. As we have discussed in Table 3, an individual fine-tuned LLM performs very poorly (i.e., 83.89% accuracy on average). However, by using our recurrent boosting framework, the performance can be boosted to 90.67% when 6 base learners are provided. Further, when $K = 7$, the performance can be boosted to 91.99%, which significantly outperforms others. When $K = 14$, the performance has achieved to 93.04% on average (1.14%↑, as compared to $K = 7$). But the performance increase plateaus as the number of base learners rises from 7 to 14, suggesting that 7 base learners makes a good balance between performance and training cost.

(4) How RGPT Varies with Different Base Learners. We test Alpaca, ChatGLM 2 and RoBERTa (Table 4) to evaluate the effect of different base learners. RGPT significantly improves the performance of each model, with RoBERTa improving by 2.26% on average, and the state-of-the-art large language model's average performance improved by more than 21.0% after using RGPT. RGPT could enhance both high-performance models and cost-effective models, demonstrating its flexibility and broad applicability to a variety of text classification tasks.

(5) Zero-Shot *vs.* Few-Shot Prompting. Table 5 shows that the impact of adding shots varies with the number of shots. The change from zero-shot to one-shot results in a slight improvement. With the gradual increase in the number of shots, the performance

Table 4: The impact of different base learners.

Method	SST-2	AG News	Ohsumed	MR
RoBERTa	96.40	94.69	72.80	89.42
RGPT$_{+RoBERTa}$	97.47	95.84	74.70	93.28
Alpaca	57.81	71.23	46.55	53.78
RGPT$_{+Alpaca}$	97.81	96.45	75.26	93.55
ChatGLM 2	81.36	83.67	54.33	74.39
RGPT$_{+ChatGLM\ 2}$	98.10	96.77	75.16	93.02
LLaMA 2	60.50	79.40	48.08	71.49
RGPT$_{+LLaMA\ 2}$	98.68	97.61	77.41	94.27

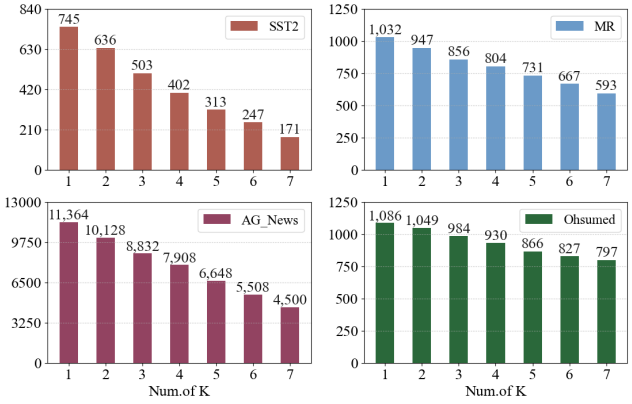

Figure 3: Number of misclassified samples at each iteration.

drops down. This potentially arises from RGPT learning redundant information when handling too long contextual data.

Table 5: Few shot performance testing.

Prompt	SST-2	AG News	Ohsumed	MR
0-shot	98.68	97.61	77.41	94.27
1-shot	**98.97**	**98.01**	**77.83**	**94.65**
5-shot	98.31	97.57	77.32	94.11
10-shot	97.95	96.60	76.85	93.52

(6) Weight Adjustment of Wrongly Classified Samples. We perform misclassification sample tracking to discuss how the proposed weight adjustment mechanism dynamically affects the performance. We record the number of misclassified samples at each iteration as shown in Fig. 3, and also report the proportions of initially misclassified samples that are correctly classified after weight updating, as shown in Fig. 4. We notice a distinct decreasing trend in overall misclassifications as iterations proceed.

4 Conclusions

In this work, we propose RGPT, an adaptive boosting framework tailored to produce a specialized text classification LLM. we efficiently

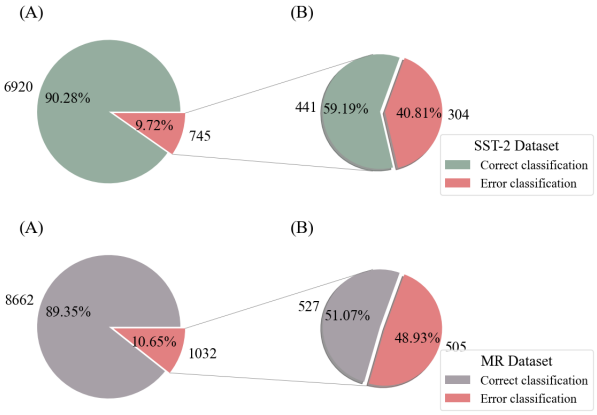

Figure 4: (A) represents the proportion of samples classified after the first iteration, and (B) shows the proportion of misclassified samples corrected in the second iteration.

train a pool of strong base learners by adjusting the distribution of training samples and iteratively fine-tuning LLMs with them. Such base learners are then recurrently ensembled to be a specialized LLM. We offer a comprehensive evaluation and our model achieves the state-of-the-art results. This proves that boosting LLMs will yield significant improvements over the existing approaches.

Limitations. Base learner should not only be homogeneous, but also can be heterogeneous. Limiting the RGPT framework's base learners solely to LLaMA 2 may hinder the method's performance.

Acknowledgements

This work is supported by a grant under the Collaborative Research with World-leading Research Groups scheme in The Hong Kong Polytechnic University (project no. G-SACF) and a General Research Fund under Hong Kong Research Grants Council (project no. 15218521). This work is also supported by Natural Science Foundation of Henan Province of China (242300421412), Foundation of Key Laboratory of Dependable Service Computing in Cyber-Physical-Society (Ministry of Education), Chongqing University (PJ.No: CPSDSC202103).

References

[1] Junying Chen, Xidong Wang, Anningzhe Gao, Feng Jiang, Shunian Chen, Hongbo Zhang, Dingjie Song, Wenya Xie, Chuyi Kong, Jianquan Li, et al. 2023. HuatuoGPT-II, One-stage Training for Medical Adaption of LLMs. *arXiv preprint arXiv:2311.09774* (2023).

[2] Sun Xiaofei, Li Xiaoya, Li Jiwei, Wu Fei, and et al. 2023. Text Classification via Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 8990–9005.

[3] Ben Yao, Yazhou Zhang, Qiuchi Li, and Jing Qin. 2024. Is Sarcasm Detection A Step-by-Step Reasoning Process in Large Language Models? *arXiv preprint arXiv:2407.12725* (2024).

[4] Peiyi Zhang, Yazhou Zhang, Bo Wang, Lu Rong, and Jing Qin. 2024. Edu-Values: Towards Evaluating the Chinese Education Values of Large Language Models. arXiv:2409.12739 [cs.CL] https://arxiv.org/abs/2409.12739

[5] Yazhou Zhang, Chunwang Zou, Zheng Lian, Prayag Tiwari, and Jing Qin. 2024. Towards evaluating large language models on sarcasm understanding. *arXiv e-prints* (2024), arXiv–2408.