Pushing The Limit of LLM Capacity for Text Classification

Anonymous ACL submission

Abstract

The value of text classification's future research has encountered challenges and uncertainties, due to the extraordinary efficacy demonstrated by large language models (LLMs) across numerous downstream NLP tasks. 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 strong 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 SOTA PLMs and 7 SOTA LLMs on four benchmarks by 1.36% on average. Further evaluation experiments reveal a clear superiority of RGPT over average human classification performance[1].

1 Introduction

Text classification aims to assign pre-defined categories to a given informative text, including sentiment analysis, topic labeling, news classification, etc. It has always been an active task across the eras of knowledge engineering and feature engineering (Cunha et al., 2023; Minaee et al., 2021). Recently, remarkable advances in LLMs, e.g., ChatGPT[2], GPT-4 (OpenAI et al., 2023), ChatGLM 2 (Zeng et al., 2023), LLaMA 2 (Touvron et al., 2023), etc., have demonstrated their outstanding performance across downstream NLP tasks. Through instruction fine-tuning and in-context learning, LLMs have possessed marvelous language understanding, generation and reasoning abilities.

Sustained efforts and investments from both academia and industry have been primarily dedicated to two directions: (1) general LLMs capable of providing encyclopaedic domain knowledge and performing well across a range of tasks, such as Mistral (Jiang et al., 2023), LLaMA series, etc.; (2) specialized LLMs tailored for vertical domains such as healthcare (Chen et al., 2023; Singhal et al., 2023), law (Cui et al., 2023), finance (Wu et al., 2023), education (Milano et al., 2023), etc., via task-specific architectures and knowledge. Additionally, arming LLMs with strategies such as mixture-of-experts (MoE) (Shen et al., 2023), tool learning (Qin et al., 2023) or modularization (Ye et al., 2023) have also garnered considerable attention. Strong LLMs intertwined with sophisticated optimization approaches are propelling LLM research to new heights.

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, it is important to investigate whether specialized text classification LLM can create substantial value over the existing approaches. We thus 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 (Xiaofei et al., 2023), QLFR (Wu et al., 2024) and PromptBoosting (Hou et al., 2023) is that RGPT does not directly optimize the prompt space but instead builds a specialized LLM by adjusting sample distribution and recur-

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

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

rently ensembling strong base learners, thus demonstrating less sensitivity to prompts and stronger stability across various tasks (see Sec. 4.1 and 4.2).

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 benchmark datasets and compare the results against 8 SOTA PLMs (e.g., DeBERTa, ERNIE, T5, etc.) and 7 SOTA LLMs (e.g., ChatGLM 2, LLaMA 2, GPT-4, etc.). The experimental results show the effectiveness of RGPT with the margin of 0.88%, 1.21%, 1.47% and 1.88% for four datasets. The study reveals that RGPT with only 7 iterations achieves the state-of-the-art results with performance continuing to grow as the number of iterations increases. Further human evaluation experiments demonstrate a clear surpassing of RGPT over average human classification. 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 over four datasets demonstrate the effectiveness of RGPT in zero-shot text classification.

2 Preliminaries

2.1 Problem Definition

Text classification is transformed as a conditional generative task, where the ouput $\mathcal{Y}$ will be the labels. Given a set of input documents $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ where each document x_i is augmented with a designed prompt $Prompt_i \in \mathcal{P}$ that provides contextual guidance, i.e., $Prompt_i = INS_i \oplus x_i$, where INS_i represents the task instruction, $\mathcal{P}$ represents the prompt set. Our task is to learn a text classification LLM $\mathcal{M}(\theta)$ which maps an input document to its target label: $\mathcal{M}(\mathcal{X}, \mathcal{P}, \theta) \rightarrow \mathcal{Y}$, where $\mathcal{Y} = \{y_1, y_2, \ldots, y_N\}$ denotes the label sequence generated by the LLM $\mathcal{M}(\theta)$ based on its comprehension of the documents and the provided prompts and $y_i \in R^c$, where c is the class of y_i. We formulate the classification problem as:

$$\mathcal{M}(\theta) = \arg \underset{c}{max} \prod_i Prob\,(y_i = c|x_i, Prompt_i, \theta) \quad (1)$$

2.2 Algorithm Overview

The recent LLM based approaches focus on elaborating prompts to improve classification performance. However, the performance gains from prompt engineering are limited, and the potential of classification performance for LLMs has not been fully investigated.

In contrast, RGPT is able to quickly generate a large pool of strong base learners through adjusting the distribution of training samples and fine-tuning LLMs, and proposes a recurrent ensembling approach to harnesses their complementarity, leading to improved effectiveness and generalization (see Sec. 4.2). As shown in Fig. 1, RGPT consists of the following key steps.

Step 1: Initialization. Assign each training sample the same weight: $\frac{1}{N}$, and select a general LLM as initial base learner $\mathcal{LM}_0$[3].

Step 2: Constructing K base learners $\mathcal{LM}_K$. The k^{th} base learner, $\mathcal{LM}_k$, is optimized under its respective loss function, which is essentially a weighted loss over training samples with larger weights on those that are misclassified by the previous learner $\mathcal{LM}_{k-1}$.

Step 3: Integrating K base learners using a recurrent ensembling approach. More details will be provided in Sec. 3 and Algorithm 1 in App.B.

3 The Proposed Framework: RGPT

3.1 Initialization and Base Learner Selection

To lay the groundwork for subsequent base learner construction and ensembling, we commence with

[3] It has been proven that boosting can also effectively combine strong base learners (Wyner et al., 2017).

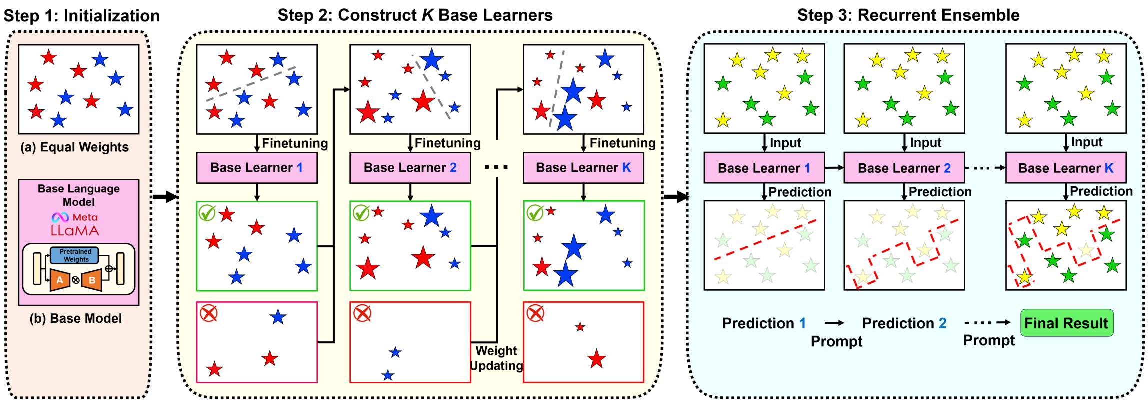

Figure 1: Overview of RGPT.

initialization. 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. In boosting, base learner can not only be a simple model (e.g., decision tree), but also be a strong learner that has yet considerable room to achieve optimal performance, such as DCNN (Moghimi et al., 2016).

We prove that our model works almost equally well on different base learners such as PLMs (i.e., RoBERTa) and LLMs (i.e., Alpaca[4], 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. 4.5).

3.2 Constructing Base Learners

The construction of K base learners involves (1) prompt construction; (2) fine-tuning LLMs with training samples; and (3) iteratively updating the weight distribution of training samples.

We follow the zero-shot prompting paradigm for text classification tasks. At each iteration k, the zero-shot prompt template $Prompt_i$ 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'].* "

The k^{th} base learner $\mathcal{LM}_k$ involves fine-tuning a general LLM using the 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:

$$\mathcal{LM}_k = \arg\min_{\theta^{(k)}} \sum_{\mathcal{D}^{(k)}} w_i^{(k)} \cdot \mathcal{L}(y_i^{(k)}, f(x_i^{(k)}; \theta^{(k)})) \quad (2)$$

where $\theta^{(k)}$ represents the parameters, $\mathcal{L}$ is the loss function, $f(\cdot)$ is a general LLM (e.g., LLaMA 2).

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:

$$\begin{aligned}
\epsilon^{(k)} &= Pr_{i \sim \mathcal{D}^{(k)}}\left[\mathcal{LM}_k\left(x_i^{(k)}\right) \neq y_i^{(k)}\right] \\
\alpha^{(k)} &= log\frac{1-\epsilon^{(k)}}{\epsilon^{(k)}} + log\,(c-1) \\
\mathcal{W}^{(k+1)} &= \frac{\mathcal{W}^{(k)}}{Z_k} \times \left\{\begin{matrix} e^{-\alpha^{(k)}} & if\ \mathcal{LM}_k\left(x_i^{(k)}\right) = y_i^{(k)} \\ e^{\alpha^{(k)}} & if\ \mathcal{LM}_k\left(x_i^{(k)}\right) \neq y_i^{(k)} \end{matrix}\right\}
\end{aligned} \quad (3)$$

where c denotes the number of class, Z_k represents the normalizing factor. Eq. 3 will assign higher weights to samples with higher errors, and ensure that subsequent learners address the weaknesses of the current learner. After K iterations, we construct K complementary and strong base learners $\{\mathcal{LM}_1, \mathcal{LM}_2, ..., \mathcal{LM}_k\}$ (More explanations are provided in App. A).

3.3 Recurrently Ensembling the Base Learners

We propose a recurrent ensembling approach, which selectively leverages the historical outputs

[4] https://crfm.stanford.edu/2023/03/13/alpaca.html.

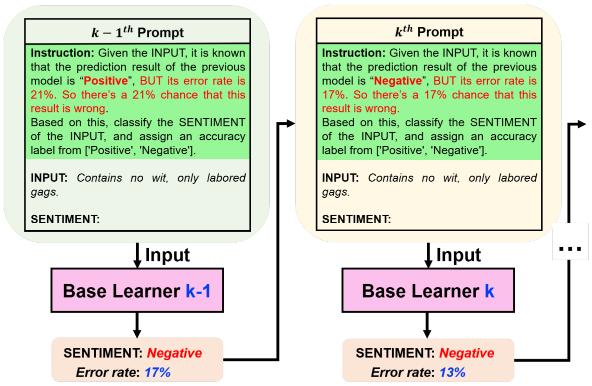

Figure 2: Recurrent ensembling K base learners.

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

Table 1: Dataset statistics.

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 current learner $\mathcal{LM}_k$, which can be written as:

$$Prompt_i = INS_i \oplus x_i^k \oplus \{\hat{y}_i^{k-1}, \epsilon^{(k-1)}\} \quad (4)$$

where $\hat{y}_i^{k-1}$ 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 $\mathcal{LM}_{k-1}$, as shown in Fig. 2.

This chain-like nature ensures that each subsequent learner can improve and complement upon the existing knowledge and producing a knowledge accumulation effect. Finally, a strong, specialized LLM $\mathcal{M}(\theta)$ is constructed.

4 Experiments

4.1 Experiment Setups

Datasets. Four benchmarking datasets are selected as the experimental beds, *viz.* SST-2 (Socher et al., 2013), MR (Pang et al., 2002), AG News (Zhang et al., 2015), Ohsumed[5]. The statistics for each dataset are shown in Table 1.

Baselines. A wide range of SOTA baselines are included for comparison. They are: (1) **RoBERTa** (Liu et al., 2019), (2) **XLNet** (Yang et al., 2019), (3) **RoBERTa-GCN** (Lin et al., 2021), (4) **DeBERTa** (He et al., 2020), (5) **ERNIE** (Sun et al., 2021) and (6) **T5** (Raffel et al., 2020) are six strong PLMs for text classification via masked language modeling and pretrained representations. (7) **E2SC-IS** (Cunha et al., 2023) selects the most representative documents for training classification model. (8) **ContGCN** (Yao et al., 2018) focuses on the misclassified training samples as the target for explainable text classification. (9) **BBTv2** (Sun et al., 2022), (10) **PromptBoosting** (Hou et al., 2023) and (11) **CARP** (Xiaofei et al., 2023) are three SOTA prompt based approaches that focus on how to find the best prompts given a specific classification task. (12) **ChatGLM 2**, (13) **LLaMA 2** and (14) **GPT-4** are three SOTA LLMs that have broad domain knowledge and outstanding performance across various NLP tasks. (15) **QLFR** (Wu et al., 2024) decomposes the text classification task into four distinct reasoning steps and presents a fine-tuned LLaMA 2-13B model.

Implementation. Training a base learner will cost about 1 hours on 8 × A100-SXM4-40GB GPUs. The micro batch size, batch size, the number of epoch and learning rate are set to be 8, 128, 10 and 3e-4 respectively. In the process of updating sample weights, we control the weights of samples by increasing or decreasing the number of samples. For a misclassified sample x_i^k, whose weight should increase to w_i^{k+1} (see Eq.3), we proportionally augment its quantity. To improve generalization and avoid overfitting, we utilize ChatGPT to generate additional samples similar to x_i^k.

4.2 Main Results

We report both **Accuracy** and **Macro-F1** results for RGPT and baselines in a zero-shot setting in Table 2. The mean and variance over 5 runs are calculated. We observe that RGPT consistently achieves state-of-the-art performance on four datasets, i.e., 0.88%↑, 1.21%↑, 1.47%↑, 1.88%↑ respectively. It outperforms PLMs based, prompt based and standard fine-tuning approaches. Despite that LLMs (i.e., ChatGLM 2, LLaMA 2, GPT-4) have shown extraordinary efficacy across general-domain tasks, their weak adaptation into text classification is also proved, in view of their worst classification performance. Among them, GPT-4 performs better than another two. By fine-tuning LLaMA 2-13B or optimizing in prompt space, QLFR, BBTv2, PromptBoosting and CARP gain significant improvements over general LLMs. QLFR, BBTv2 and PromptBoosting have been trading victories on different

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

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	**98.68**$_{\pm 0.2}$	**98.67**	**97.61**$_{\pm 0.3}$	**97.52**	**77.41**$_{\pm 0.2}$	**73.68**	**94.27**$_{\pm 0.5}$	**94.15**	**96.85**	**91.99**
Gain △	0.88%	1.49%	1.21%	2.34%	1.47%	0.76%	1.88%	3.45%	1.46%	2.98%

Table 2: Performance on four datasets. Bold and blue indicate the best and second-best results for each dataset.

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	98.18	96.90	76.99	93.71
RGPT	98.68	97.61	77.41	94.27

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

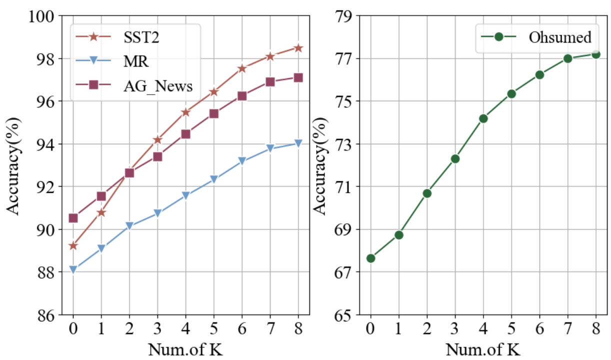

Figure 3: Performance of RGPT with increasing number of learners.

benchmarks, but they are inferior to other methods using PLMs, e.g., RoBERTa, DeBERTa, T5, etc. CARP achieves the best performance on AG News and MR datasets among all the baselines, and obtain comparable results against ERNIE on SST-2 dataset. This suggests that prompt learning indeed elicits LLMs to outperform traditional PLMs based approaches, but the design of prompts is critically important.

4.3 Ablation Study

Table 3 shows the result of ablation studies on four datasets. 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 small language model (namely RoBERTa) to be the backbone language model. For *w/o Recurrent ensemble*, we perform a weighted combination of K strong base learners according their coefficients $\alpha^{(k)}$. From the experiment results 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. In a summary, each module in our method contributes to the final performance.

4.4 Effect of K

In our main experiments, we adopt $K = 7$ due to its significant SOTA performance. Intuitively, a large learners pool increases the diversity of base learners which could improve the performance. We empirically present the relationship between the number of learners and the model performance in Fig. 3. 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, which slightly overcomes all the baselines. Further, when $K = 7$, the performance can be boosted to 91.99%, which significantly outper-

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

Table 4: The impact of different base learners.

forms others with performance continuing to grow as the number of iterations increase (e.g., $K = 8$). But the performance increase plateaus as the number of base learners rises from 7 to 8, suggesting that 7 base learners makes a good balance between performance and training cost.

4.5 How RGPT Varies With Different Base Learners

We select LLaMA 2-7B to the initial base model by default. In order to evaluate the effect of different base learners, we have also tried another two SOTA LLMs and one strong PLM, i.e., Alpaca, ChatGLM 2 and RoBERTa, as shown in Table 4. We notice that RGPT+RoBERTa performs the worst on four tasks, but still significantly outperforms the standard RoBERTa with the margin of 2.26% on average. Additionally, RGPT+Alpaca obtains slightly improvements over RGPT+RoBERTa, but is inferior to ChatGLM 2 and LLaMA 2. The reason is that latter models have adopted more advanced architectures and training methodologies. In addition, three standard SOTA LLMs perform very poorly without boosting, which implies that general LLMs are still insufficient to directly cope with various text classification tasks. But their performance significantly improves using RGPT, with an increase of over 21.0%$\uparrow$. Different base models can achieve comparable results using RGPT. We demonstrate that RGPT universally boosts varies base model structures.

4.6 Zero-shot v/s Few-shot Prompting

We perform zero-shot and few-shot experiments to evaluate whether RGPT can perform better when a limited number of contextual examples are available. The results are shown in Table 5. We design four k-shot settings: zero-shot, one-shot, five-shot, ten-shot. For each setting, we randomly sample $k = \{0, 1, 5, 10\}$ examples from the training set.

The impact of adding shots varies with the number of shots. The change from zero-shot to one-shot results in a slight improvement in classification performance. With the gradual increase in the number of shots, the performance drops down. This potentially arises from RGPT learning redundant information when handling too long contextual data. This implies that crudely increasing the number of extra shots does not necessarily result in a stable performance improvement.

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
3-shot	98.31	97.57	77.32	94.11
10-shot	97.95	96.60	76.85	93.52

Table 5: Few shot performance testing.

4.7 Human v/s Machine

We create a new test set including 200 samples randomly sampled from three datasets, e.g., IMDB (Maas et al., 2011), R8[6] and DBPedia (Auer et al., 2007), where their proportion is 4:3:3. Then, we recruit three volunteers[7] to evaluate the sentiment, news and topic labels. We ask the first two annotators to proceed at their standard speeds, where the third annotator should annotate meticulously and conduct a double-check. Their classification scores and time costs will be compared with RGPT in Table 6. It can be seen that RGPT consistently outperforms two humans in terms of accuracy and efficiency. Despite that RGPT underperforms the third annotator, its time cost is $\frac{1}{7}$ of that of the third annotator. It is foreseeable that with the continuous improvement of future LLMs, their classification capabilities will further enhance. RGPT also surpasses the average performance of three annotators, proving that we have made much progress in text classification over the existing approaches.

4.8 Overfitting Study

To confirm that our model doesn't overfit when consistently adjusting the sample distribution, we adopt three strategies: (1) early stopping approach is used; (2) we present the learning curves to show how the training loss changes, as shown in Fig. 4. This visual representation helps us understand if

[6]https://www.cs.umb.edu/ smimarog/textmining/datasets/

[7]They all signed on the consent form before the study and were paid an equal \$5.0/hour. Prior to annotation, they received professional guidance covering the criteria for labeling, positive and negative examples, etc.

Method	Accuracy	Efficiency (minutes)
Human 1	89.21	53.3
Human 2	90.05	56.9
Human 3	96.59	80.6
Avg.	91.95	63.6
RGPT	92.54	10.9

Table 6: The human classification results against RGPT.

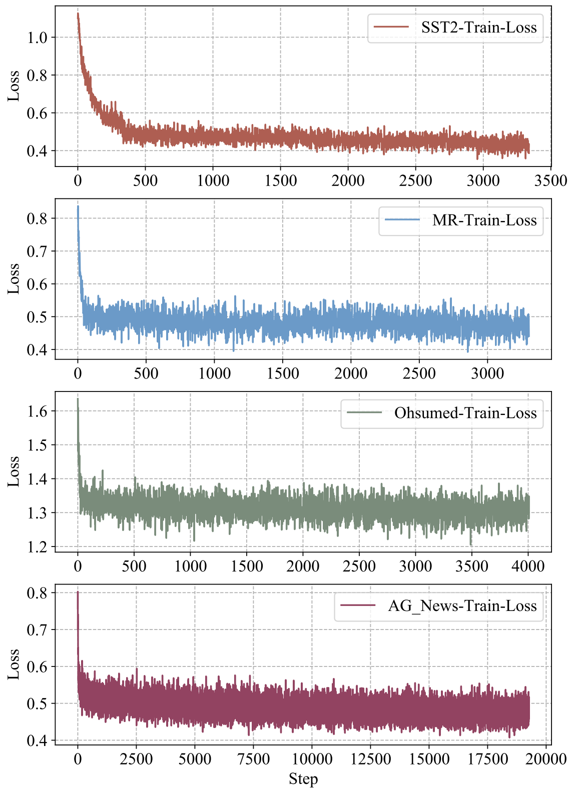

Figure 4: The training loss of RGPT.

the model's performance improves consistently on both new and seen data during training; (3) in addition, we do not directly increase (e.g., replicate) the number of those misclassified samples. Instead, we choose to increase similar style samples generated by ChatGPT. This strategy can improve the diversity of samples to avoid overfitting.

4.9 Data Visualization

We present a visual comparison chart between the distribution of testing set and the distribution of training set after $K = 7$ iterations, as shown in Fig. 5. We notice that the distribution of the training set at $K = 0$ differs significantly from the test set distribution, while at $K = 7$, the distribution of the training set becomes more aligned with the distribution of the testing set. This indicates that our RGPT method effectively adjusts the distribution of the training set to be more similar to the true distribution, thereby enhancing the classification performance of the model. In addition, the distribution of the training set, evolving through iterative adjustments in boosting, exhibits concentration around previously misclassified samples, indicating the algorithm's focus on challenging cases. The visualization provides a nuanced understanding of how the model adjusts the training data. This analysis aids in assessing the model's potential overfitting tendencies, and its ability to generalize effectively to new instances.

4.10 Error Analysis

The detailed error analysis is also conducted via the confusion matrices that are shown in Figure 6. Each cell (i, j) represents the percentage of class i is classified to be class j. Upon reviewing the classification results produced by RGPT on four datasets, we discover that imbalanced categories and the similarity across different categories are the key factors contributing to misclassification.

By examining the diagonal elements of the matrices, RGPT demonstrates effective true-positive categorization for most fine-grained categories across four datasets. However, it exhibits a tendency to misclassify the "negative" utterances to be "positive", particularly on the SST-2 and MR datasets. In addition, RGPT tends to misclassify "Bussiness" to be "World" and "Technology" on AGNews dataset. RGPT has high error rate on Ohsumed dataset. There are two possible reasons: (1) the highly unbalanced samples leads to the model's misclassification, e.g., C18, C20, etc.; (2) the similarity across several categories, e.g., C4, C11, C12, C13, etc., may pose a challenge for the model to accurately distinguish them.

5 Related Work

In recent years, significant advancements in NLP have been attributed to the emergence of LLMs. OpenAI has achieved significant milestones with the creation of two groundbreaking models: ChatGPT and GPT-4. However, due to their proprietary nature, There has been numerous LLM variants featuring tens or even hundreds of billions of parameters (Zhao et al., 2023). We categorize these LLMs into two groups based on their specialization: general LLMs and specialized LLMs. General LLMs are designed for versatility across a wide spectrum of NLP tasks. Prominent examples of these models

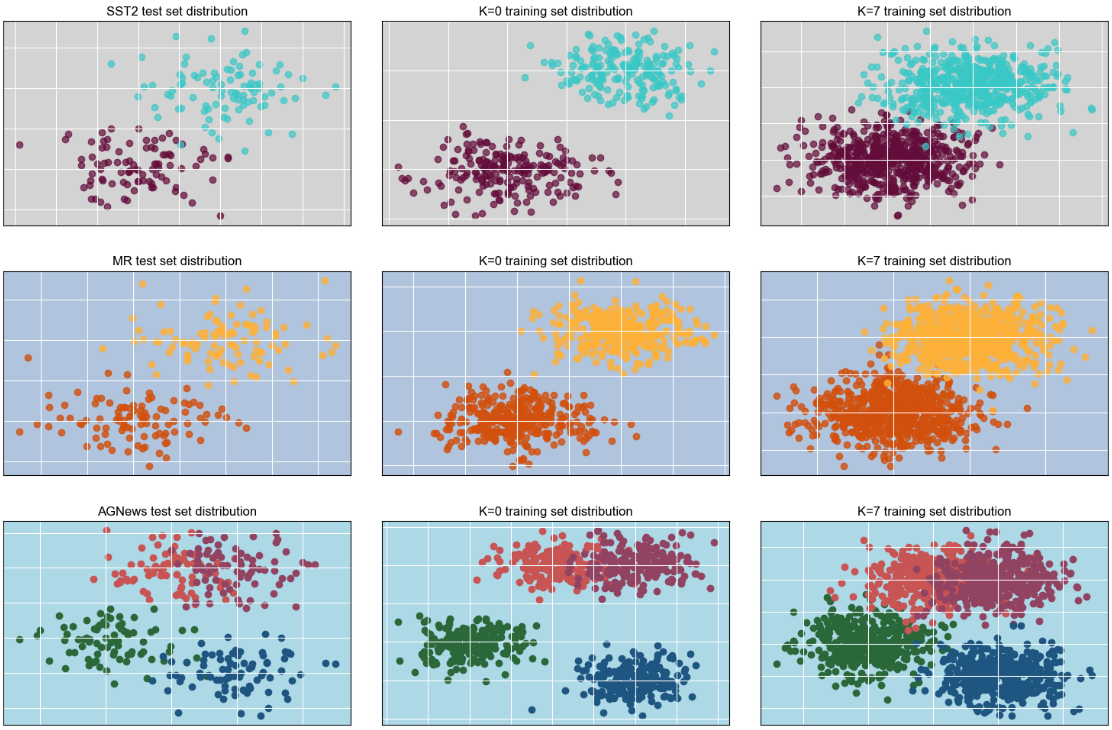

Figure 5: Distribution of training samples and initial test samples during K iterations.

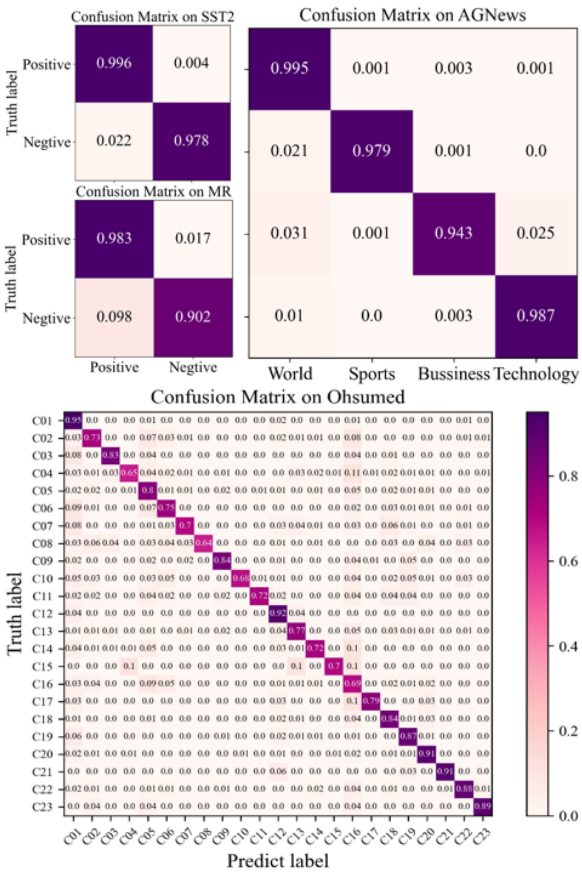

Figure 6: The normalized confusion matrices for RGPT across four datasets. The columns represent the truth label, where the rows represent the predicted labels.

are GPT-4, ChatGLM, LLaMA 2, PanGu-Σ (Ren et al., 2023), Falcon (Penedo et al., 2023), etc. In contrast, specialized LLMs are fine-tuned for specific tasks via task-specific architectures and knowledge, allowing them to achieve higher performance. An increasing number of studies are raging over medical, law, finance and education domains, e.g., HuaTuo (Zhang et al., 2023), FinGPT (Yang et al., 2023), ChatLaw (Cui et al., 2023), etc.

Different from the above-mentioned studies, we pioneer a specialized LLM by iteratively refining and integrating base LLMs, unlocking its untapped potential on text classification tasks.

6 Conclusions

In this work, we propose RGPT, an adaptive boosting framework tailored to produce a specialized text classification LLM. we efficiently 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 other PLM and prompt based approaches. Human evaluation experiments proves that RGPT can outperform average human performance.

7 Limitations.

The proposed RGPT model also has several limitations: (1) **High computational cost.** The iterative nature of its boosting-based mechanism, which involves multiple rounds of fine-tuning LLMs, leads to a significant computational cost. (2) **Limited testing sets.** RGPT has shown significant performance improvements across four benchmark datasets. However, the study does not thoroughly examine how well the model may work on a wider range of text classification tasks. (3) **Monotony of base learners.** 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 innovation and its potential for improvement. Ensembling different LLMs may enhance the adaptability and versatility of the approach when facing new challenges.

Potential Risks. Even though RGPT addresses overfitting by increasing similar samples instead of the misclassified samples themselves, there remains a risk of overfitting during the repeated fine-tuning of large language models. This risk becomes more prominent in situations with a small training set.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *international semantic web conference*, pages 722–735. Springer.

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*.

Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *arXiv preprint arXiv:2306.16092*.

Washington Cunha, Celso França, Guilherme Fonseca, Leonardo Rocha, and Marcos André Gonçalves. 2023. An effective, efficient, and scalable confidence-based instance selection framework for transformer-based text classification. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 665–674.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Bairu Hou, Joe O'connor, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2023. Promptboosting: Black-box text classification with ten forward passes. In *International Conference on Machine Learning*, pages 13309–13324. PMLR.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. BertGCN: Transductive text classification by combining GNN and BERT. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1456–1462, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Silvia Milano, Joshua A McGrane, and Sabina Leonelli. 2023. Large language models challenge the future of higher education. *Nature Machine Intelligence*, 5(4):333–334.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40.

Mohammad Moghimi, Serge J Belongie, Mohammad J Saberian, Jian Yang, Nuno Vasconcelos, and Li-Jia Li. 2016. Boosted convolutional neural networks. In *BMVC*, volume 5, page 6.

OpenAI, :, Josh Achiam, Steven Adler, and Sandhini Agarwal et al. 2023. Gpt-4 technical report.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only.

Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, Chaojun Xiao, Chi Han, et al. 2023. Tool learning with foundation models. *arXiv preprint arXiv:2304.08354*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Xiaozhe Ren, Pingyi Zhou, Xinfan Meng, Xinjing Huang, Yadao Wang, Weichao Wang, Pengfei Li, Xiaoda Zhang, Alexander Podolskiy, Grigory Arshinov, et al. 2023. Pangu-{\Sigma}: Towards trillion parameter language model with sparse heterogeneous computing. *arXiv preprint arXiv:2303.10845*.

Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, Tu Vu, Yuexin Wu, Wuyang Chen, Albert Webson, Yunxuan Li, Vincent Zhao, Hongkun Yu, Kurt Keutzer, Trevor Darrell, and Denny Zhou. 2023. Mixture-of-experts meets instruction tuning:a winning combination for large language models.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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.

Tianxiang Sun, Zhengfu He, Hong Qian, Yunhua Zhou, Xuan-Jing Huang, and Xipeng Qiu. 2022. Bbtv2: towards a gradient-free future with large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3916–3930.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation.

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

Hui Wu, Yuanben Zhang, Zhonghe Han, Yingyan Hou, Lei Wang, Siye Liu, Qihang Gong, and Yunping Ge. 2024. Quartet logic: A four-step reasoning (qlfr) framework for advancing short text classification.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Abraham J Wyner, Matthew Olson, Justin Bleich, and David Mease. 2017. Explaining the success of adaboost and random forests as interpolating classifiers. *The Journal of Machine Learning Research*, 18(1):1558–1590.

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*, pages 8990–9005.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2018. Graph convolutional networks for text classification. *ArXiv*, abs/1809.05679.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023. mplug-owl: Modularization empowers large language models with multimodality.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130b: An open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Jianquan Li, Guiming Chen, Xiangbo Wu, Zhiyi Zhang, Qingying Xiao, et al. 2023. Huatuogpt, towards taming language model to be a doctor. *arXiv preprint arXiv:2305.15075*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

A Explanations of The Complementary and Robustness Across Base Learners

The complementarity among multiple base learners, as observed in ensemble learning frameworks like boosting, refers to the ability of different foundational models to recognize and process distinct features or patterns within the data. For RGPT, which employs LLMs as base learners, this complementarity is manifested in several aspects:

(1) Feature space coverage. Each fine-tuned LLM may exhibit varying degrees of understanding and capturing capabilities for different semantic, syntactic structures, or contextual information in the input text. For instance, one LLM may excel at handling long-distance dependency relationships, while another may demonstrate greater accuracy in understanding domain-specific terms.

(2) Error distribution. As the sample weights are adjusted based on the prediction errors of the preceding weak learners during each iteration, subsequent learners focus more on the previously misclassified samples. Consequently, even if the foundational architectures of all LLMs are similar, they address and correct different subsets of data, creating complementarity.

(3) Randomness and robustness. Despite fine-tuning for the same task, different initialization states and random factors during the training process (such as the stochastic nature of gradient descent) may lead LLMs to produce distinct decision boundaries. These boundaries may intersect or misalign in complex data distributions, enhancing the overall robustness and generalization performance of the ensemble model.

(4) Model capacity. While LLMs possess high capacity, a single model may not fully leverage all its parameters to adapt to complex tasks, especially with limited training data. Through multiple rounds of fine-tuning and ensemble combination, the model potential can be better explored, allowing each learner to focus on specific aspects of the task, resulting in overall optimization.

B Recurrent Ensembling The Base Learners: Algorithm and Illustration

Here, we present further details of RGPT in Algorithm 1 and the overall architecture of ensembling in Fig. 2

Algorithm 1 Recurrent ensemble Learning of RGPT

Require:
Input:
  $\mathcal{D}^{(0)}$: Original training dataset with N samples $(x_i^{(0)}, y_i^{(0)})$
  $\mathcal{LM}_0$: LLaMA 2 as initial base learner
  K: Number of base learners
Ensure:
Output:
  $\mathcal{M}_{ensemble}$: Recursively ensembled model
Training:
  Initialize data weights $\mathcal{W}^{(0)} = \left\{w_1^{(0)}, \ldots, w_N^{(0)}\right\}$ where $w_i^{(0)} = \frac{1}{N}, \forall i \in N$
for $k = 1, 2, \ldots, K$ **do**
  Construct prompt $\text{Prompt}_i^{(k)} = INS_i \oplus x_i^{(k)}$
  Fine-tune $\mathcal{LM}_k$ with weighted training samples:

$$\mathcal{LM}_k = argmin_{\theta^{(k)}} \sum_{\mathcal{D}^{(k)}} w_i^{(k)} \cdot \mathcal{L}(y_i^{(k)}, \mathrm{f}_k(x_i^{(k)}; \theta^{(k)}))$$

  Compute error rate $\epsilon^{(k)}$ of $\mathcal{LM}_k$
  Calculate weight coefficient $\alpha^{(k)} = \log \frac{1-\epsilon^{(k)}}{\epsilon^{(k)}} + \log(c-1)$
  Update data weights for $k+1^{th}$ iteration:

$$\mathcal{W}_i^{(k+1)} = \begin{cases} \frac{w_i^{(k)}}{Z_k} e^{-\alpha^{(k)}} & \text{if } \mathcal{LM}_k(x_i^{(k)}) = y_i^{(k)} \\ \frac{w_i^{(k)}}{Z_k} e^{\alpha^{(k)}} & \text{if } \mathcal{LM}_k(x_i^{(k)}) \neq y_i^{(k)} \end{cases}$$

  Normalize weights by Z_k to ensure $\sum_{i=1}^{N} w_i^{(k+1)} = 1$
Inference:
for $k = 1, 2, \ldots, K$ **do**
  Forward the prompt through k^{th} base learner $\mathcal{LM}_k$
  Obtain the classification result $\hat{y}_i^{(k)}$
  Update prompt for next iteration:

$$\text{Prompt}_i^{(k+1)} = \text{Prompt}_i^{(k)} \oplus \{\hat{y}_i^{(k)}, \epsilon^{(k)}\}$$

return $\mathcal{M}_{ensemble} = F(\mathcal{LM}_1, \mathcal{LM}_2, \ldots, \mathcal{LM}_K)$