

# Is the Larger the Better? An Exploratory Study into Human-Large Language Model Collaboration

Jie Tao  
Fairfield University  
[jtao@fairfield.edu](mailto:jtao@fairfield.edu)

Xing Fang  
Illinois State University  
[xfang13@ilstu.edu](mailto:xfang13@ilstu.edu)

Lina Zhou  
The University of North  
Carolina at Charlotte  
[lzhou8@uncc.edu](mailto:lzhou8@uncc.edu)

## Abstract

*Large language models (LLMs) have garnered considerable attention in both academics and industry. Given an array of LLMs available, one of the primary challenges lies in their selection and adaptation strategies. Although LLMs are generally large, they still vary significantly in size and larger LLMs consume significantly more computing resources. This prompts the inquiry into whether larger models perform better. In addition, there is a widespread recognition of the power of LLMs in performing open-ended or generative tasks. However, how to use LLMs to address a closed-ended problem remains under explored. The exploration of human-LLM collaboration on close-ended problem has been even more sparse. This research aims to address the above limitations by comparing different types of state-of-the-art adaptation strategies for LLMs, including in-context learning and fine-tuning. Moreover, it employs multi-class multi-label classification - a close-ended problem to empirically evaluate those adaptation strategies. The research findings provide valuable insights and recommendations for human users considering deploying LLMs for close-ended problems.*

**Keywords:** Large Language Models, Generative AI, Adaptation strategies, Aspect Based Sentiment Analysis, Human-LLM Collaboration

## 1. Introduction

Large Language Models (LLMs) are characterized by their ability to understand and generate language by training billions of parameters on massive amounts of text data (Minaee et al., 2024). LLMs have been employed to address various research and practical problems, ranging from natural language understanding (Karaniokolas et al., 2023), question answering (Louis et al., 2024), code generation (J. Liu et al., 2024), to education. In addition to individual users, organizations also strategically leverage LLMs aimed at capitalizing on advanced AI capabilities to enhance workforce efficiency, employee productivity, and improve customer experience.

There is a misconception that LLMs are merely chatbots, and expect them to return a satisfactory answer when a user query is provided to them. As a matter of fact, there are currently multiple popular families of LLM models, which include, but are not limited to, Generative Pre-trained Transformer (GPT) (Yenduri et al., 2023), Text-to-Text Transfer Transformer (T5) (Raffel et al., 2020), and Large Language Model Meta AI (LLaMA) (Touvron et al., 2023). For instance, OpenAI's ChatGPT and GPT-4 can be used for not only natural language processing (NLP), but also empowering Microsoft's Copilot systems as general task solvers. The latter can follow human instructions in performing multi-step reasoning for complex new tasks. Additionally, the T5 model has gained significant popularity in the business world for its innovative approach to NLP. By framing all NLP tasks as text-to-text generation challenges within a unified framework, T5 has streamlined the application of transfer learning in this field. The LLaMA suite of models, unveiled in February 2023, has quickly risen to prominence in the business sector for its advanced capabilities in language understanding and generation. Notably, the open-source LLaMA-13B model has demonstrated superior performance over the larger, proprietary GPT-3 (175B) model across numerous benchmarks. This remarkable efficiency positions LLaMA as an ideal baseline for cutting-edge LLMs research and a potent tool for businesses seeking to gain a competitive edge. To leverage the potential of LLMs, a fundamental question arises: how to use those models effectively?

We categorize existing adaptation strategies for LLMs into two main groups: in-context learning and fine-tuning. In context learning requires LLMs to generate contents based on a user query (i.e., prompt), which can be further categorized into zero- and few-shot learning depending on whether the prompt contains a few examples or not. Fine-tuning is a common procedure in employing transformer-based models for certain downstream tasks. The sizes of parameters in LLMs are much larger than those in traditional transformer-based models. For instance, RoBERTa base model contains 355 million parameters, which is about one-third the

size of a small LLMs (e.g., Bloom/Bloomz (Muenighoff et al., 2023), which contains 1.1 billion parameters. Consequently, LLMs require more efficient fine tuning techniques rather than full fine tuning. Parameter-Efficient Fine Tuning (PEFT) (H. Liu et al., 2022) recently emerged to meet the above requirements. PEFT consists of two main types: adapter-based and prompt-tuning-based methods. In light of the different adaptation strategies available, a more focused inquiry emerges: how can we adapt LLMs effectively and efficiently?

Although LLMs have been primarily used in generating open-ended content, as powerful as LLMs are, they can be used beyond that. For instance, LLMs can be used to solve close-ended problems, such as generating discrete labels from a given set (Loukas et al., 2023; Yu et al., 2023). This type of tasks is similar to the traditional classification problems. The classification problem can vary significantly in terms of complexity, ranging from binary classification to multi-class classification, and then to multi-class multi-label classification. However, not only is the application of LLMs to multi-class multi-label problems under explored, the strategies for human-LLM collaboration on these issues remain insufficiently studied. In this study, we choose a close-ended problem, and more specifically, multi-label classification problem to examine different adaptation strategies of LLMs.

This study makes several research contributions. First, we compare the effectiveness of different adaptation strategies of LLMs, providing insights on when and where a particular strategy might be preferred when human users applying LLMs to close-ended problems. Second, moving beyond the commonly used open-ended tasks for LLMs, we apply LLMs to a close-ended NLP problem and empirically evaluate their performances. Given resource constraints in businesses or organizations, such as financial limitations, hardware availability, and expertise accessibility, obtaining versatile models that address various types of problems can offer both short-term and long-term competitive advantages. The observations presented in this paper may facilitate more effective collaboration between less experienced users and LLMs. Third, we present findings from an empirical investigation of different characteristics of LLMs, which can serve as adaptation guides for future research and practices.

## 2. Related work

### 2.1. Large Language Models

Large Language Models (LLMs) are language models with enormous parameter sizes with superior ca-

pabilities to capture contextual linguistic cues and generate human-like texts (Chang et al., 2024). These models are built on the Transformer architecture (Vaswani et al., 2017), which used the self-attention mechanism as the kernel and are pre-trained on vast amounts of domain-independent texts. The Transformer architecture consists of inter-connected encoder and decoder components.

Generative LLMs, such as GPT (Yenduri et al., 2023) and LLaMA (Touvron et al., 2023), are essentially decoder-based models that estimate token probabilities based on the preceding  $t$  time steps:  $P(x_{t+1}|x_1, \dots, x_t)$ . In other words, generative LLMs (i.e., inferencing) are expected to generate texts based on certain given contexts or prompts. Generative LLMs have been widely used for NLP tasks such as Natural Language Inference (Gubelmann et al., 2024), summarization (Eigenschink et al., 2023), question answering (Huo et al., 2023), and so forth. In contrast, encoder-based models, such as BERT (Devlin et al., 2018) and RoBERTa (Y. Liu et al., 2019), are capable of performing token and sequence classifications given the whole context, such as masked language modeling and sentiment analysis (Tao & Fang, 2020).

As a summary, the decoder based models have recently received much attention because of their generative functionalities, which encoder based models fell short of. In this study, we investigate the potential of using *decoder based* models for text classification tasks, which are traditionally carried out by encoder-based models in deep learning. If decoder based models demonstrate comparable or even superior performances in text classification, researchers and practitioners may prioritize their use for addressing a variety of NLP tasks, albeit the models are usually more expensive to train than the encoder-based counterparts.

### 2.2. Text classification with LLMs

LLMs are largely used for open-ended problems such as information retrieval, synthesis of text and code (Yu et al., 2023), and data enrichment (Fernandez et al., 2023), which have been under-studied for closed-ended problems such as text classification.

One trend of using LLMs for text classification problems is the use of close-sourced models. For instance, Loukas et al. (2023) used OpenAI’s GPT-3.5, GPT-4 and Anthropic’s Claude 2 to perform multi-class classification on banking data. The paper also discussed the “overlapping labels” issue, and thus it essentially addressed a multi-label classification problem, which is a more challenging problem. Li et al. (2023) compared open- and close-sourced LLMs on different NLP tasks in the finance domain, including classification tasks such as sentiment analysis and headline classification.

Within the in-context learning paradigm, few-shot learning outperforms zero-shot learning on a variety of tasks. Sun et al. (2023) proposed a prompting strategy, named Clue And Reasoning Prompting (CARP), with GPT-3 models, on a variety of text classification tasks. The study claimed that the proposed CARP strategy is on par with full fine-tuning on smaller encoder-based models. Lopez-Lira and Tang (2023) used ChatGPT to predict the polarity of financial news.

LLMs have received serious criticism since they sometimes hallucinate. Compared to open-ended generations (e.g., question answering), it becomes an even more serious issue when solving close-ended problems (e.g., classification) since hallucinated labels do not exist, which lead to a failure in the classification problem (Yu et al., 2023).

The related studies on using LLMs for text classification largely leverage prompt engineering techniques. Although carefully engineered prompts have the potential to address hallucinations, the traditional prompt engineering process typically relies on trial and error, often resulting in labor-intensive efforts. Even for recent developments in prompt engineering like Chain-of-Thoughts (CoTs) (Wei et al., 2022), they require carefully designing the conversation (chain), a task demanding expertise. Additionally, to improve the performance of CoTs, it is suggested to use multiple datasets with CoT annotations, which would increase the requirements for both data and knowledge (Chung et al., 2024).

The current application of LLMs for text classification barely reached the paradigm of fine-tuning. However, the literature suggests that further fine tuning on open-sourced LLMs can achieve better performances. For instance, Loukas et al. (2023) and Yu et al. (2023) explored open-sourced models (e.g., LLaMA-2) for similar text classification problems, to improve the model performance or cost efficiency.

### 2.3. Aspect Based Sentiment Analysis

In this work, we focus on exploring applying LLMs to solve a multi-label text classification task, known as Aspect Based Sentiment Analysis (ABSA).

Conventional sentiment analysis focuses on determining the overall sentiment at the sentence or document level. ABSA has received increasing attention over the past decade. ABSA aims at extracting sentiments expressed toward certain entities or characteristics of the entities in a sentence or document (Tao & Fang, 2020). For instance, “Bread is great but waiting time is too long” expresses a positive sentiment (“great”) toward the food aspect (“bread”) of a restaurant, but a negative sentiment (“too long”) toward the service aspect (“waiting time”). While ABSA can provide additional values to its stakeholders, it is a more

difficult problem compared to conventional sentiment analysis. In ABSA, the models must not only identify the aspect(s) but also determine the sentiment expressed toward them. Moreover, ABSA is a multi-label classification problem, given that a text segment may contain multiple aspect-sentiment pairs (labels), thus introducing another layer of complexity.

While extant studies on ABSA have yielded promising results (Dhanith & Prabha, 2023), the state-of-the-art models are mostly encoder-based. There are a few recent attempts to incorporate LLM-like methods or even generative models into the ABSA task. For instance, Liu et al. (2021) proposed a sequence-to-sequence method to tackle the ABSA problem as a text generation task using the Bart model. Additionally, Song et al. (2023) proposed a label prompting based model for multi-label classification problems, which is not necessarily an ABSA problem; however, it still uses the encoder-based model (i.e., BERT) as the underlying model.

The models used for ABSA need to go through full fine-tuning to optimize model performances. Full fine tuning often requires larger amounts of data, and higher computational costs with an increased size of language models (Loukas et al., 2023). This is feasible for most encoder-based models given their relatively small parameter sizes (e.g., 110M in Bert-base and 355M in RoBERTa-base); however, it becomes much more challenging when considering decoder-based models which usually have billions, if not trillions, of parameters. Previous studies have suggested using PEFT techniques, such as LoRA, for classification tasks (Yu et al., 2023).

## 3. Adaptation strategies

We conceptualize “adaptation strategies” as methods for adjusting an LLM to better fit specific requirements and constraints for human-LLM collaboration in completing a task. In this study, we focus on two types of adaptation strategies to address the above considerations: in-context learning and parameter efficient fine-tuning. We identify representative yet distinct state-of-the-art techniques for each of the above categories. For instance, we choose zero-shot learning and few-shot learning to represent in-context learning, and Low Rank Adaptation (LoRA) (Hu et al., 2021) and prompt tuning (Lester et al., 2021) to represent parameter efficient fine-tuning strategies. More specifically, we consider different methods for selecting examples in the zero/few-shot learning.

### 3.1. In-context Learning

Zero-Shot Learning is an adaptation strategy that uses a model for a downstream task without providing

any prior examples of the task or any fine tuning. It largely relies on the model's existing knowledge gained through the training process and its ability to generalize from related tasks. In contrast, few-shots learning is an adaptation strategy that requires prompting the model with a very small number of examples (typically 1 - 5) for a particular downstream task. It also relies on the model to generalize from pre-existing knowledge and the prompted examples to new instances.

Figure 1 illustrates the process of both zero-shot and few-shot learnings. In zero-shot learning, a query is directly presented to a generative AI model, whereas, in few-shot learning, the query along with several examples, which are retrieved from an example dataset, is sent to the model. The process of retrieving examples requires each instance in the sample dataset to be transformed into a vector space embedding through an embedding model. The embeddings are then saved into a vector database. When the query is ready, it will also be transformed into a vector by the same embedding model. Finally, a semantic similarity score will be computed between the query and each example based on their embedding vectors, and the examples that are semantically most similar to the query will be selected.

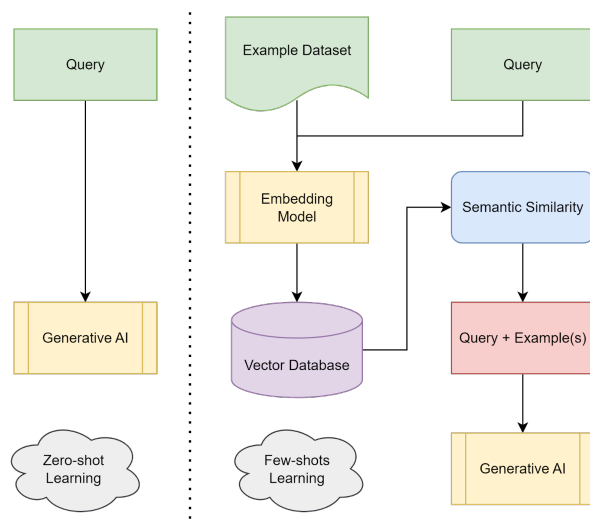


Figure 1. Processes of In-context Learning

The benefits of using both zero-shot and few-shot learning include that they can work with close-sourced models (e.g., GPT-3/4), in which the model architectures and weights are unknown to end-users (Brown et al., 2020). In addition, since in-context learning only requires the inference capabilities of the models, it can alleviate high hardware requirements. It is also interesting to discuss knowledge required for in-context learning. While in-context learning does not require deep technical knowledge as does any fine tuning based adaptation strategies (e.g., PEFT, full fine tuning, see Section

3.2), it often requires prompt engineering to get desirable results. Prompt engineering in itself is a knowledge and labor intensive step.

### 3.2. PEFT

Due to hardware and other resource limitations, traditional full fine tuning is less efficient when adapting LLMs to specific domains and tasks. Thus, PEFT becomes more popular. We select two specific adaptation strategies from the PEFT category in this study. LoRA is an adaptation strategy that adapts a pre-trained model to a new task domain by updating a low-rank decomposition of the weight matrix (Hu et al., 2022). For any single layer  $L$ , we denote the weights after pre-training as  $W_L$ , which is updated during model fine-tuning for a specific downstream task,  $W_L$  as shown in Eq (1):

$$W_L = W_L + \Delta W \quad (1)$$

Note that  $\Delta W$  shares the same shape as  $W_L$ . A low-rank decomposition would render  $\Delta W$  as the dot product of matrices  $A$  and  $B$  ( $\Delta W = AB$ ), where  $rank(A) \ll rank(\Delta W)$  and  $rank(B) \ll rank(\Delta W)$ . Accordingly, the LoRA fine tuning updates the weights, as shown in Eq (2):

$$W_L = W_L + AB \quad (2)$$

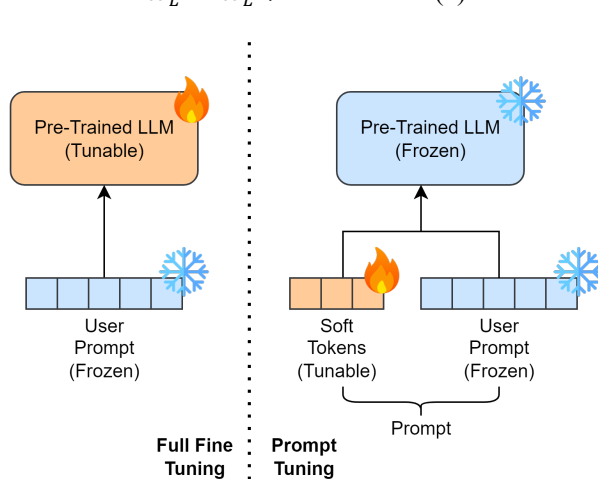


Figure 2. A Comparison of Full Fine-Tuning and Prompt Tuning

Compared to the full fine tuning, which updates  $\Delta W$ , LoRA is more computationally efficient since it only need to update the lower-ranked  $A$  and  $B$  with  $W_L$  frozen during back propagation. In other words, the number of trainable parameters are much smaller compared to the total number of parameters in the model. Since LLMs are pre-trained to capture the representations of diverse tasks, only a subset of the learned weights needs to be updated when adapting them to a specific downstream task. Additionally, given that only

a subset of weights are updated, LoRA is better at handling the “catastrophic forgetting” issue, where the model “forgets” certain skills obtained via pre-training after being adapted to a specific domain.

In contrast to LoRA, which insert adapters to layers (low-rank projection of the weight matrix in that layer), prompt tuning is another adaptation strategy that insert trainable parameters to the input of the model (i.e., prompts) to guide the model’s predictions or generations to a certain task domain. These additional parameters, or “soft prompts,” are optimized during fine-tuning to improve performance on specific tasks without altering the underlying model weights. This strategy is more flexible in multi-task learning since a set of soft tokens is learned for the specific task. In order to adapt the model to a different task, users can switch the “soft tokens” learned to that task without the need of making any changes to the model.

Compared to the in-context learning strategies, PEFT strategies provide better control of the model output. The generated results are better suited to the task domain, reducing the need for extensive post-processing and mitigating issues such as hallucination and labor-intensive editing of the generated contents. However, PEFT strategies necessitate a deeper understanding of the model architecture and weights, as well as greater hardware and resource requirements.

## 4. An Empirical Investigation - ABSA

### 4.1. Problem Formulation and Setup

Among different ABSA tasks, compound ABSA, which aims at detecting aspect (e.g., “food”) and sentiment polarity (e.g., “positive”) simultaneously, is considered more difficult than simple ABSA tasks (e.g., detecting aspects or sentiment separately) (Chauhan et al., 2023; Zhang et al., 2023). In this study, we focus on compound ABSA tasks. The detection of aspect categories is a multi-class multi-label classification problem, and the detection of sentiment polarity is a multi-class classification problem - since each review can contain multiple aspects, yet each aspect is associated with only one of many polarities.

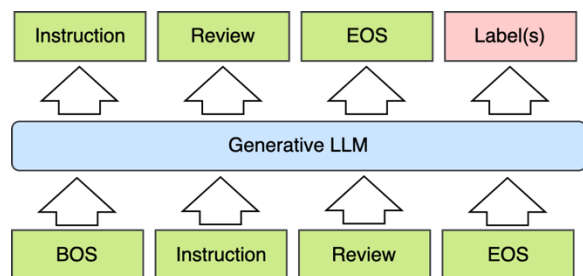


Figure 3. Inference on Generative LLMs

Figure 3 depicts the inference phases of applying generative LLMs to ABSA. The prompt consists of four parts, including a begin-of-sequence (BOS) special token, an instruction, a review, and an end-of-sequence (EOS) special token, and the generative model is prompted to generate labels in an autoregressive (token by token) fashion. For few-shot learning, we use a pre-trained BERT (Reimers & Gurevych, 2019) to generate the vector database in order to select examples that have the highest semantic similarities to an incoming review. The examples are then inserted in between the instruction and review to form the final prompt.

For LoRA, the ground truth labels are appended after the EOS special token for each review during the fine tuning phase. For prompt tuning, soft tokens are inserted between the BOS special token and instruction, and the ground truth labels are appended after the EOS special token. For instance, below is the prompt we used for the PEFT fine tuning (with the ground truth label appended at the end) in this study, in which `<s>` is the BOS special token and `</s>` is the EOS special token.

```
[INST] <s> Below is an instruction that describes a task. Write a response that appropriately completes the request. [INST] Given the review <review> But the staff was so horrible to us. </review>, returning one or more from <labels> ['price#neutral', 'service#positive', 'service#conflict', 'ambience#positive', 'price#positive', 'service#neutral', 'ambience#conflict', 'food#neutral', 'service#negative', 'ambience#negative', 'anecdotes/miscellaneous#conflict', 'food#conflict', 'price#conflict', 'anecdotes/miscellaneous#negative', 'food#positive', 'anecdotes/miscellaneous#neutral', 'food#negative', 'ambience#neutral', 'anecdotes/miscellaneous#positive', 'price#negative'] </labels>. [INST] </s> ['service#negative']
```

### 4.2. Data and Models

We use the well-known SemEval 2014 Task 4 (ABSA) dataset (Pontiki et al., 2014), specifically the restaurant reviews, as a case to study the different adaptation strategies. The data is pre-split into training and test sets, which contains 3,041 and 800 English reviews, respectively. The dataset contains 5 aspect categories (i.e., food, service, price, ambience, and anecdotal/miscellaneous) and 4 sentiment polarities (i.e., positive, negative, neutral, and conflict). Thus, we have a total of 20 labels or label pairs, as shown in the `<labels>` section in the sample prompt from the previous section.

Our model selection is based on two criteria: generalizability and relatedness to text classification. As a result, we select generative models from different families and with different-sized parameters, including:

Bloomz 1.1B, GPT-Neo 2.3B, and LLaMA-2 7B. We provide a brief introduction to each model family below.

- Bloomz is finetuned from Bloom, which is a pre-trained multilingual language model. Bloomz is capable of following human instructions in dozens of languages to perform zero-shot learning.
- GPT-Neo (Black et al., 2021) is an open-source alternative to GPT-3, designed to provide similar capabilities. GPT-Neo is trained on the Pile, a diverse and extensive dataset, aiming to achieve strong performance on various NLP tasks.
- LLaMA-2 is a family of LLMs developed by Meta AI. They range from 7 billion to 70 billion parameters and are optimized for dialogue use cases. The models are auto-regressive and

have been fine-tuned using supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF) to align with human preferences for helpfulness and safety.

It is also worth noting that we perform 4-bit quantization on LoRA and limit the max length of input sequences to 64 in prompt tuning during the fine-tuning phase, which is a common practice to reduce the hardware requirements for training LLMs. Other hyperparameters used in the fine tuning and inference phases are shown in Table 1. In this study, the in-context learning adaptation strategies, such as zero-shot and few-shot learning, follow the same inferencing hyperparameters as the PEFT strategies, because they do not require fine-tuning. We use classification accuracy as the evaluation metric.

**Table 1. Hyperparameters Settings**

	LoRA	Prompt Tuning
<b>Fine Tuning</b>	Training epochs: 3 Learning rate: 2e-4 Max_grad_norm: 0.03 Weight decay: 0.001 Alpha: 16 R: 64 Dropout: 0.1 Warm up ratio: 0.03 Max_length: None Quantization: 4bit (nf4) LR scheduler: constant Optimizer: paged AdamW 32bit	# of virtual tokens: 20 Training epochs: 30 Max length: 64 Learning rate: 3e-2 Batch size: 8 LR scheduler: Linear Optimizer: AdamW
<b>Inferencing</b>	Task: Causal Language Modeling Max new tokens: 40	

### 4.3. Result and discussion

Table 2 reports the performance of in-context learning. The table shows, zero-shot learning has the worst performance compared to other strategies despite that the former is most commonly adopted in using LLMs. In particular, Bloomz and GPT-Neo failed to predict any labels correctly. For few-shot learning, we include up to three examples in a prompt. All these models outperform the zero-shot learning counterpart. In addition, compared with a random selection of examples for the prompt, choosing examples based on semantic similarities consistently leads to a better performance. Specifically, Bloomz and LLaMA-2 achieve the best accuracy under the two-shot learning and GPT-Neo has the best accuracy in the one-shot learning setting.

Table 3 presents the model performances using the PEFT adaptation strategies, namely LoRA and prompt tuning. The results show that Bloomz - the smallest-sized model in our selection performs the best (69%) with the prompt-tuning strategy, whereas LLaMA-2 - the largest-sized model performs the best (71%) with the LoRA strategy. It is worth noting that both are comparable to an encoder-based RoBERTa model (73%) with full fine-tuning.

Based on the above experiment results, we draw the following findings. These findings may inform guidelines for more effective human-LLM collaborations, particularly for less experienced users:

- **Model size and adaptation strategies:** The adaptation strategy of LLMs may vary with the model size. Specifically, the findings of our case study suggest that smaller models perform better under prompt tuning and larger models

perform better under LoRA. The ongoing trend of the adaptation strategies for in-context learning is that: larger models perform better than smaller ones. The findings of this study also suggest that there is no “one-size-fits-all”

fine tuning strategy for different sizes of models. We also expect that the choice of model family will impact how human users select the most appropriate fine tuning strategies.

**Table 2. In-context Learning Performance**

Model	Zero-shot	One-shot		Two-shot		Three-shot	
		Random	Semantic	Random	Semantic	Random	Semantic
<b>Bloomz 1.1B</b>	0%	6%	15%	8%	<b>22%</b>	6%	21%
<b>GPT-Neo 2.3B</b>	0%	2%	<b>4%</b>	1%	3%	1%	2%
<b>LLaMA-2 7B</b>	7%	22%	31%	34%	<b>41%</b>	33%	39%

**Table 3. PEFT Performance**

Model	LoRA	Prompt Tuning
<b>Bloomz 1.1B</b>	7%	<b>69%</b>
<b>GPT-Neo 2.3B</b>	29%	11%
<b>LLaMA-2 7B</b>	<b>71%</b>	25%

- **Number of shots in few-shot learning:** In few-shot learning, the models consistently perform better when the examples are selected based on semantic similarity compared with random selection. It is interesting to note that increasing the number of examples does not necessarily lead to better performance, as shown in Table 2. This is consistent with the findings from Loukas et al. (2023). Thus, human users should only consider the most relevant examples when engineering in context learning prompts.
- **Computational and knowledge requirements:** In addition to examining model sizes, it is valuable to explore the efficiency and knowledge requirements of various adaptation strategies, while considering model performances. For instance, while in-context learning strategies typically entail lower computational requirements because inferring is less resource demanding than fine-tuning, in-context learning may necessitate prompt engineering to achieve comparable

performances. This also suggests a knowledge trade-off for human users: if they are less experienced with LLMs and therefore more likely to choose in context learning, they need to be more familiar with the domain to formulate effective prompts.

- **Data requirements:** By design, in-context learning strategies require less data than PEFT strategies, and the latter in turn requires much less data than full fine-tuning. Therefore, when selecting different adaptation strategies, human users should consider data availability. For instance, if only a small amount of labeled data is available, in context learning might be more effective than PEFT. It would be interesting to perform different sensitivity analysis to test the impacts of data sizes on model performance. It is important to recognize that any sampling strategies employed may introduce biases inherent within the dataset. Therefore, techniques such as cross-validation may be utilized to mitigate this concern.
- **Insights on prompt engineering:** we conducted experiments to explore various types of prompts in the empirical investigation. These experiments range from simple prompts such as *Q: review A: label* (in which “review” and “label” represent the respective reviews and labels in the dataset) to the specific prompt (see section 4.1). While the selected prompt yielded the best overall performance, we noticed variations in the model behavior when presented with different prompts. For instance, Bloomz performed

best with simple prompts. It is worth studying how models react to different types of prompts, as it could inform the design of guidelines for prompt engineering to alleviate the labor/knowledge requirements while maintaining satisfactory performances. Additionally, we recommend that human users, particularly those with less experience, refer to the respective technical reports of the models for their specific prompt templates, since this can be a crucial factor influencing model performance.

## 5. Future Work and Concluding Remarks

Our research not only offers insights toward effective adaptation strategies of the appropriate utilization of LLMs, but also suggests potential avenues for enhancing their design. These insights can serve as starting points to guide more effective human-LLM collaborations, particularly for less experienced users. By selecting proper adaptation strategies, LLMs can achieve performance on par with encoder-based transformer models on close-ended problems. This, along with the proficiency of LLMs in performing generative tasks, indicates that LLMs possess remarkable versatility in addressing a wide array of tasks. In other words, human users have more flexibility when collaborating with different AI models on different tasks.

LLMs are not necessarily better. One important finding of this study is that the effect of the size of LLM on their performances depends on the adaptation strategy. The model size can have a positive impact on the performance of in-context learning, which may not be the case for fine-tuning. Within the paradigm of fine-tuning, the performance of LoRA improves with the model size, whereas prompt tuning works better with smaller-sized models. We can offer a few explanations for why prompt tuning may not work as effectively with larger-sized LLMs. First, there might be overfitting issues with the model becoming too specialized on the training data and performing poorly on test data. Second, based on our observations, the prompt tuning strategy takes more epochs for smaller models to reach comparable performance to larger models using LoRA, and larger models are expected to take even more epochs to do so, which can be more resource-intensive. Third, the ratio of trainable parameters (i.e., soft token embeddings) in larger models is lower compared to smaller models, given that the number of trainable parameters is fixed across different models. This suggests a potential need to increase the number of soft tokens to enhance their effect on model performance. In addition, the generalizability of the above findings requires further validation in future

research, especially when accounting for increased problem complexity.

In addition, a larger example size does not necessarily lead to a better model performance for in-context learning, specifically few-shot learning. One possible explanation is that more examples can potentially introduce less relevant or even conflicting information regarding the target task. Given that prompt engineering is the most direct interaction in human-LLM collaboration, and selecting examples is one of the most important tasks in prompt engineering, it will be interesting to investigate how to select a set of examples that can collectively boost the performance of in-context learning.

While in principle in-context learning is more computationally efficient than fine-tuning, it has the ‘hidden’ cost associated with prompt engineering to optimize model performance. In addition, method design for example selection is critical for the performance of in-context learning, which can benefit from domain expertise and deserve a separate study. In other words, it is worth future investigation of how to better infuse human knowledge in human-LLM collaboration.

While we evaluated representative techniques of main adaptation strategies of LLMs, there are other emerging techniques that are beyond the scope of this study given the fast-evolving pace of this field. In addition, the findings based on a single empirical investigation, regardless how representative it is, is inherently limited. Future research should consider evaluating the generalizability of our findings by conducting a more comprehensive evaluation of the different types of LLMs on a wide range of tasks, which could potentially provide more guidelines for human-LLM collaboration..

## 6. References

- Brown, T. B., Kaplan, J., Ryder, N., Henighan, T., Chen, M., Herbert-voss, A., Ziegler, D. M., Krueger, G., Askell, A., Hesse, C., McCandlish, S., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., ... others. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS 2020)*, 33, 1877–1901.
- Black, S., Gao, L., Wang, P., Leahy, C., & Biderman, S. (2021). *GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow*. <https://doi.org/10.5281/ZENODO.5297715>
- Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., & Xie, X. (2024). A Survey on Evaluation of Large Language Models. *ACM Transactions on Intelligent Systems and Technology*, 15(3), 1–45.

- <https://doi.org/10.1145/3641289>
- Chauhan, G. S., Nahta, R., Meena, Y. K., & Gopalani, D. (2023). Aspect based sentiment analysis using deep learning approaches: A survey. *Computer Science Review*, 49, 100576. <https://doi.org/10.1016/j.cosrev.2023.100576>
- Chung, H. W., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Chi, E. H., Dean, J., Devlin, J., ... Le, Q. V. (2024). Scaling Instruction-Fine-tuned Language Models. *Journal of Machine Learning Research*, 25, 1–53.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, *abs/1810.0*. <http://arxiv.org/abs/1810.04805>
- Dhanith, P. R. J., & Prabha, K. S. S. (2023). A critical empirical evaluation of deep learning models for solving aspect based sentiment analysis. In *Artificial Intelligence Review* (Vol. 56, Issue 11). Springer Netherlands. <https://doi.org/10.1007/s10462-023-10460-0>
- Eigenschink, P., Reutterer, T., Vamosi, S., Vamosi, R., Sun, C., & Kalcher, K. (2023). Deep Generative Models for Synthetic Data: A Survey. *IEEE Access*, 11(March), 47304–47320. <https://doi.org/10.1109/ACCESS.2023.3275134>
- Gubelmann, R., Katis, I., Niklaus, C., & Handschuh, S. (2024). Capturing the Varieties of Natural Language Inference: A Systematic Survey of Existing Datasets and Two Novel Benchmarks. *Journal of Logic, Language and Information*, 33(1), 21–48. <https://doi.org/10.1007/s10849-023-09410-4>
- Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LORA: LOW-RANK ADAPTATION OF LARGE LANGUAGE MODELS. *ArXiv Preprint*. <https://github.com/microsoft/LoRA>.
- Huo, S., Arabzadeh, N., & Clarke, C. (2023). Retrieving Supporting Evidence for Generative Question Answering. *SIGIR-AP 2023 - Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, 11–20. <https://doi.org/10.1145/3624918.3625336>
- Karanikolas, N., Manga, E., Samaridi, N., Tousidou, E., & Vassilakopoulos, M. (2023). *Large Language Models versus Natural Language Understanding and Generation*. 13. <https://doi.org/10.1145/3635059.3635104>
- Lester, B., Al-Rfou, R., & Constant, N. (2021). The Power of Scale for Parameter-Efficient Prompt Tuning. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, 3045–3059. <https://doi.org/10.18653/v1/2021.emnlp-main.243>
- Li, X., Chan, S., Zhu, X., Pei, Y., Ma, Z., Liu, X., & Shah, S. (2023). Are ChatGPT and GPT-4 General-Purpose Solvers for Financial Text Analytics? A Study on Several Typical Tasks. *EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Industry Track*, 408–422. <https://doi.org/10.18653/v1/2023.emnlp-industry.39>
- Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., & Raffel, C. (2022). Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning. *Advances in Neural Information Processing Systems*, 35, 1950–1965. <https://github.com/r-three/t-few>
- Liu, J., Steven Xia, C., Wang Lingming, & Zhang, Y. (2024). Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. *Advances in Neural Information Processing Systems*, 36. <https://github.com/evalplus/evalplus>
- Liu, J., Teng, Z., Cui, L., Liu, H., & Zhang, Y. (2021). Solving Aspect Category Sentiment Analysis as a Text Generation Task. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, 4406–4416. <https://doi.org/10.18653/v1/2021.emnlp-main.361>
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 1. <http://arxiv.org/abs/1907.11692>
- Lopez-Lira, A., & Tang, Y. (2023). Can ChatGPT Forecast Stock Price Movements? Return Predictability and Large Language Models. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4412788>
- Louis, A., van Dijck, G., & Spanakis, G. (2024). Interpretable Long-Form Legal Question Answering with Retrieval-Augmented Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(20), 22266–22275. <https://doi.org/10.1609/AAAI.V38I20.30232>
- Loukas, L., Stogiannidis, I., Diamantopoulos, O., Malakasiotis, P., & Vassos, S. (2023). Making LLMs Worth Every Penny: Resource-Limited Text Classification in Banking. *ICAIF 2023 - 4th ACM International Conference on AI in Finance, Mlm*, 392–400. <https://doi.org/10.1145/3604237.3626891>
- Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2024). *Large Language Models: A Survey*. <http://arxiv.org/abs/2402.06196>
- Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Le Scao, T., Bari, M. S., Shen, S., Yong, Z. X., Schoelkopf, H., Tang, X., Radev, D., Aji, A. F., Alnubarak, K., Albanie, S., Alyafeai, Z., Webson, A., Raff, E., & Raffel, C. (2023). Crosslingual Generalization through Multitask Finetuning. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 1, 15991–16111. <https://doi.org/10.18653/v1/2023.acl-long.891>
- Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., & Manandhar, S. (2014). SemEval-2014 Task 4: Aspect Based Sentiment Analysis. *8th International Workshop on Semantic Evaluation, SemEval 2014 - Co-Located with the 25th International Conference on Computational*

- Linguistics, COLING 2014, Proceedings*, 27–35.  
<https://doi.org/10.3115/V1/S14-2004>
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21, 1–67.  
<http://jmlr.org/papers/v21/20-074.html>.
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 3982–3992. <https://doi.org/10.18653/v1/d19-1410>
- Song, R., Liu, Z., Chen, X., An, H., Zhang, Z., Wang, X., & Xu, H. (2023). Label prompt for multi-label text classification. *Applied Intelligence*, 53(8), 8761–8775. <https://doi.org/10.1007/s10489-022-03896-4>
- Sun, X., Li, X., Li, J., Wu, F., Guo, S., Zhang, T., & Wang, G. (2023). Text Classification via Large Language Models. *Findings of the Association for Computational Linguistics: EMNLP 2023*, 8990–9005.  
<https://doi.org/10.18653/v1/2023.findings-emnlp.603>
- Tao, J., & Fang, X. (2020). Toward multi-label sentiment analysis: a transfer learning based approach. *Journal of Big Data*, 7(1), 1–26.  
<https://doi.org/10.1186/s40537-019-0278-0>
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). *LLaMA: Open and Efficient Foundation Language Models*.  
<http://arxiv.org/abs/2302.13971>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, L. (2017). Attention is all you need. *In Advances in Neural Information Processing Systems*, 5998–6008.  
<https://doi.org/10.1109/2943.974352>
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35(NeurIPS), 1–43.
- Yenduri, G., M, R., G, C. S., Y, S., Srivastava, G., Maddikunta, P. K. R., G, D. R., Jhaveri, R. H., B, P., Wang, W., Vasilakos, A. V., & Gadekallu, T. R. (2023). Generative Pre-trained Transformer: A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions. *IEEE Access*, 12(April), 54608–54649. <https://doi.org/10.1109/ACCESS.2024.3389497>
- Yu, S., Fang, C., Ling, Y., Wu, C., & Chen, Z. (2023). LLM for Test Script Generation and Migration: Challenges, Capabilities, and Opportunities. *IEEE International Conference on Software Quality, Reliability and Security, QRS*, 206–217.  
<https://doi.org/10.1109/QRS60937.2023.00029>
- Zhang, W., Li, X., Deng, Y., Bing, L., & Lam, W. (2023). A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges. *IEEE Transactions on Knowledge and Data Engineering*, 35(11), 11019–11038.  
<https://doi.org/10.1109/TKDE.2022.3230975>