Exploring the Limits of Fine-Tuning in Language Adaptation for Large Language Models

Ahmet Arif Turkmen
Georgia Institute of Technology
aturkmen3@gatech.edu

Yi Han
Georgia Institute of Technology
yhan422@gatech.edu

Howard Wang
Georgia Institute of Technology
ywang4201@gatech.edu

Shafaat Mushtaq
Georgia Institute of Technology
smushtaq3@gatech.edu

Abstract

The emergence of Large Language Models over the past few years has enabled new applications of natural language tasks, specifically in the field of language adaptation. It is computationally expensive to train a new Large Language Models from scratch due to the billions of parameters in these state of the art models and the scale of training data. In this work, we present a method to fine-tune different base models trained primarily on English text to adapt the model for Q&A for the Chinese and Turkish language.

1. Introduction/Background/Motivation

What did you try to do? What problem did you try to solve? Articulate your objectives using absolutely no jargon. As large language models (LLMs) continue to advance rapidly, new models are being released frequently. However, these models are often trained primarily on English data, making them perform better in English than in other languages. This trend poses a risk of other languages becoming less relevant as LLMs become more prevalent across various sectors.

To mitigate this risk, our objective is to fine-tune open-source LLMs using datasets from foreign languages. This approach will enable us to assess and improve how well these models can adapt to new languages. Model which was primarily used was Meta-Llama-3-8B and Mistral-7b-v0.1 and languages on which base model was fine tuned on was Chinese and Turkish.

From our inference trials, the new launched Meta-Llama-3-8B-Instruct doesn't do well in conversations in languages other than English. For example, we tested Chinese and Turkish, and noted while the model is able to understand instruction and user inputs in Chinese, it struggles to respond or speak in Chinese, even when given strict instruction to respond in Chinese. However, we've noticed that Meta-Llama-3-8B-base does well in multi-turns Chinese conversations, and from our test leveraging existing tools, the model's tokenizer handles Chinese characters and words well given Meta-Llama-3-8B-base's largely expanded vocabulary.

Our goal is to leverage existing knowledge of Llama-3-8B-base and Mistral-7b-v0.1 improve models in two ways. First, to make it better at chatting in languages other than English. We use Chinese and Turkish as the two target languages in this case, since we have team members who speak these languages and can evaluate it manually. Secondly, we want to provide extra cultural contexts and make the conversations feel more natural and culturally aware.

How is it done today, and what are the limits of current practice? Currently, multiple approaches are used to adopt model to foreign languages. First approach is to extend the tokenizer, and do a additional round to primary training of foreign language data set[2] and then do fine tune. The fine tuning is also done use two approaches, firstly full parameter fine tuning[5] and secondly parameter efficient fine tuning Parameter Efficient Fine-Tuning (a.k.a PEFT), such as LoRA fine-tuning the low-rank slices of the query, key, and value embedding heads[4][3][10].

The limitation of current practice is that full parameter fine tuning still involves use of significant amount of resource, such as 8×80GB devices, making it difficult for small labs and companies to participate the research in this area[5]. We focus on PEFT as it can be done by end consumers with limited resources as less as single GPU. Also we use Meta-Llama-3-8B which is newer model and is tokenizer is already trained on diverse language datasets, so there is not derive to expand the tokenizer. Only a small weights of model change during the fine-tuning using LORA and QLORA. For these reasons, generalization to different domains is a serious problem for PEFT finetunings.

Additionally, since the Llama-3-8B-base model is relatively new as it was just launched when we conducted our research, there is a lack of extensively fine-tuned models publicly available and research.

Who cares? If you are successful, what difference will it make? If we are successful it will bridge the gap between foreign language to and English LLMs. It will enables user to generate custom agents in foreign language by fine tuning with limited resources.

We believe our research will benefit many in the world, including and not limited to researchers in the field, companies in the industry, and individuals who need or care about the capability of leveraging the large language models that is cross-language and has cross-cultural context. A better language model in other languages rather than English means the model can powers easier everyday folks and open new possibilities for how they're used. By leveraging limited resources to fine-tune large language models (LLMs) for foreign languages, we aim to democratize access to cutting-edge language technologies, fostering an inclusive and equitable landscape for researchers worldwide.

What data did you use? Provide details about your data, specifically choose the most important aspects of your data mentioned here. You don't have to choose all of them, just the most relevant.

Below is the main dataset in Chinese that we leveraged during fine-tuning. We mixed the datasets and the native speakers reviewed the samples of the datasets to ensure quality. The final Chinese dataset we used for fine-tuning has 50k samples.

1. YeungNLP/firefly-train-1.1M[12]. This dataset contains data for 23 common Chinese NLP tasks, and numerous datasets related to Chinese culture have been constructed, such as couplets, poetry generation, classical Chinese translation, prose, and Jin Yong's novels. For each task, several instruction templates are manually written to ensure high-quality and diverse data. The dataset size is 1.15 million entries.

For Turkish datasets, there was not a single top quality data source. Different data sources had problems with quality, quantity. Data sources was mostly prepared with Google Translate API, and rawness of translation was the reason for poor dataset quality. Therefore, we tried to handpick different data sources and mixed these data points.

1. sayhan/aya-dataset-tur. This dataset provide an instruction response template in small sequences. It has 4k rows. This dataset used for finetuning of Cohere-Aya 101 model.[15]

2. beratcmn/no-robots-turkish. This dataset is machine translated version of HuggingfaceH4 - No robots dataset. 10,000 well-crafted instructions and demos made by knowledgeable human annotators make up the No Robots dataset. It includes instuction/response data for summarization, QA, brainstorming and other popular domains. [9]

3. beratcmn/lima-tr. This dataset is machine translated version of LIMA dataset from GAIR. Lima dataset has only 1k rows. GAIR argues that small datasize like 1k can be enough for finetuning, due to the quality of the generated data. [14]

4. merve/turkish-instructions. This dataset is machine translated version of Natural Instruction dataset. AllenNLP crowd-sourced this dataset for popular instructions tasks in Natural Language Processing. [6]

5. parsak/alpaca-tr-9k-longest. This dataset is machine translated version of Alpaca dataset. Stanford NLP group created Alpaca dataset for fine-tuning Llama models. This dataset contains longest 9k samples of the dataset. [11]

2. Approach

What did you do exactly? How did you solve the problem? Why did you think it would be successful? Is anything new in your approach?

The objective was to create custom foreign language agents from base model using fine-tuning. Full fine-tuning was not a option as it requires significant resources[5], so we solved to use Parameter efficient fine tuning. In PEFT, we further evaluated two approaches in namely LoRA and QLoRA. In both of these approaches we take pretrained weights and freeze them. Then we inject trainable rank decomposition matrices in each layer of transformer. QLoRA does another optimization on top of LoRa which further reduces the memory footprint. It does a recoverable quantization of model parameters in 4 bits.

First we researched and gathered some public-available datasets[12][13] in Question-and-Answer format and in multi-turn or/and one-turn format in other languages than English (in Chinese and Turkish). Then we mixed the datasets from various sources so that the final datasets used for training are diversified in terms of conversation purposes (i.e. writing, role-playing, reasoning, mathematics, coding, stem, humanities, etc.), and topics and fields (i.e. sports, movie, finance, art, etc.). Our final datasets size is around 50,000 samples of data for Chinese, and around 15k in Turkish. We pre-processed the datasets so that they are ready for fine-tuning the Meta-Llama-3-8B and Mistral-7b-v0.1 based and instruct models, including splitting the datasets into train and test, and adjusting the template format so that Meta-Llama-3-8B and Mistral-7b-v0.1 base models can make best use of its learned structure.

Secondly we set up the training code using QLoRA and LoRA and configured the quantization strategy for model optimization. We wrote the code ourselves, utilizing Huggingface framework and Unsloth AI.

We also ran fine-tuned hyper parameters settings. For both Llama-3-8B base and Mistral-7b-v0.1 models, we tested multiple hyper parameters configurations. The two hyper parameters namely were were tuning were LoRA adopters used (alpha) and projection dimension1)[4].

Once the training was complete, we evaluated the results on various benchmarks and performed manual evaluation. The performance of fine-tuned models in non-English language was compared with the base model and instruct model. We evaluated how effective are foreign language fine-tuned agents are. We believe this will be successful as when we examine Meta-Llama-3-8B base model with Chinese instruction it was giving some output in Chinese, but when we use Meta-Llama-3-8B-instruct the quality of output in Chinese dropped. This was primarily because most of fine tuning was done using English instructions.

The new thing in this approach is that we are using existing methods on different problem statement and evaluative how successfully them work[3].

2.1. Problems Encountered

What problems did you anticipate? What problems did you encounter? Did the very first thing you tried work?

The first problem is data pre-processing complexities and configuration adjustments. For example, in order to leverage Meta-Llama-3-8B's existing knowledge to its fullest extent, we aligned the raw datasets with Meta-Llama-3-8B's prompt instructions to ensure that the model could effectively leverage its pre-existing knowledge during training.

During the evaluation phase, we encountered difficulties with the system prompt engineering. We initially tried to write the system prompt in English, but found that Meta-Llama-3-8B wasn't able to generate any Chinese text at all. We performed some prompt engineering so that our prompts are able to generate response in Chinese as expected.

A general challenge during evaluation process is that human evaluation can be costly. As such, we utilized existing language model tools (such as GPT-4) for scoring, and provided GPT4 with a few examples of scoring.

Another problem we encountered during the evaluation phase was that We initially tried a zero-shot training approach where we just give GPT4 the system prompt and the user prompts, but the response we got back was difficult to parse, so we decided to go with a two-shot approach where we give GPT4 two examples of the expected response.

3. Experiments and Results

How did you measure success? What experiments were used? What were the results, both quantitative and qualitative? Did you succeed? Did you fail? Why? Justify your reasons with arguments supported by evidence and data.

To test language adoption in both Chinese and Turkish, we used 2 models Meta-Llama-3-8B and Mistral-7b-v0.1 and trained both models using QLoRA[1]. We performed below parameter tuning configurations on each models:

- 1) rank = 8; alpha = 32; learning rate=1.6e-05
- 2) rank = 16; alpha = 32; learning rate=1.6e-05
- 3) rank = 16; alpha = 48; learning rate=1.6e-05

The purpose of these configurations was to determine best configurations for language adoption. We also compared the results across with base and instruct models.

Dataset used for Chinese was 50000 examples of single turn in simple question answer format. Dataset is clearly chosen and special consideration was done to ensure that none of evalution dataset is present in training dataset.

We performed two types of evaluation descrived in sections below.

3.1. MMLU

We used Chinese multiple choice questions from MMLU datasets[7] to evaluate the accuracy of the models, of which the total questions were divided into 5 categories. (stem: 2670, humanities: 1962, social science: 1938, other: 1947, china: 3065). All evaluations were done using 5 shot.

Result on Mistral-7b base model, instruct model and models fine tuned on top of base model are also included for comparison in Table 1. Result of Llama-3-8B-base base model, instruct model and models fine tuned on top of base model are also included for comparison in Table 2.

As can be seen in tables that results improve over the base model but are less than available instruct models. The primary reason was that our models were fine-tuned with limited GPU capacity and only on 50000 examples, they were not exposed to any multiple choice format questions. The increase in accuracy shows a positive sign for language adoption.

Another interesting observation in Llama fine tuned models is that when rank was increase from 8 to 16 using alpha value constant, it did not caused noticeable change in accuracy but when alpha value was increased from 32 to 48 keep rank constant there was noticeable increase in accuracy. This tells us that when fine tuning model for language adoption using PEFT alpha is more sensitive hyper parameter and value of rank does not cause significant change after some point.

Llama base mode gives almost random results, as compared to mistral base model, and Llama instruct performs better than mistral instruct. This can be associated with more special token used is llama prompt and lack of training of llama base model on special tokens.

Model	Stem	Humanities	Social Science	Other	China
Mistral-base	31.84	35.37	38.54	37.60	34.65
Mistral-instruct	35.58	41.39	49.43	46.48	39.54
Mistral-ft-rank:8-alpha:32	33.15	39.14	44.27	39.75	37.65
Mistral-ft-rank:16-alpha:32	32.88	38.63	43.34	38.93	36.31
Mistral-ft-rank:16-alpha:48	32.85	39.60	43.50	39.91	36.97

Table 1: *Chinese MMLU Scores for Mistral models*

Model	Stem	Humanities	Social Science	Other	China
Llama-base	25.58	26.71	26.99	25.99	26.10
Llama-instruct	42.02	52.91	56.97	59.17	49.46
Llama-ft-rank:8-alpha:32	31.20	39.81	43.09	40.68	34.16
Llama-ft-rank:16-alpha:32	31.42	38.84	43.65	40.22	35.11
Llama-ft-rank:16-alpha:48	31.95	42.61	44.27	41.81	37.29

Table 2: *Chinese MMLU Scores for Llama models*

3.2. MT-Bench with GPT4

The second evaluation was done on language quality analysis using MT-Bench[8]. We took 80 question of 8 different types 10 of each category [writing, humanities, stem, extraction, coding, math, reasoning, roleplay]

We generated the output to these prompts using our fine-tuned models and then passed the prompt/answer pair to GPT4 to get the generated results scored. We also scored Meta-Llama-3-8B and Mistral-7b-v0.1 instruct model to get a baseline comparison.

3.2.1 Prompt Engineering for Meta-Llama-3-8B and Mistral-7b-v0.1

To generate the output to these prompts for all of our models, we used the following system prompt: ”你是一个乐于助人的助手，用中文回答问题，字数不超过300字.” Translated to English, this is ”You are a helpful assistant and answer questions in Chinese with no more than 300 words.” Then, we append the actual prompt from the dataset as a user prompt.

For Meta-Llama-3-8B, the prompt for text generation ended up being:

```
<|begin_of_text|>
<|start_header_id|>system<|end_header_id|>
{system_message}<|eot_id|>
<|start_header_id|>user<|end_header_id|>
{user_message}<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>
```

where system_message and user_message are the system prompt and user prompt mentioned above.

3.2.2 Prompt Engineering for GPT4

Once we had the model results, we designed the GPT4 system prompt as follows:

```
你是我的助手，你要负责评估问题回应的质量，评分
范围从1到5分。你应该根据回答的内容进行打分。你
的答案一定要是一个数字，不能有其他文字。
以下是一些你可以参考的例子：

人类：  描述原子的结构
AI：原子由质子和中子组成的原子核组成，质子和中
子在中心紧密结合在一起。 原子核周围是电子，它们
以特定的能级或壳层绕原子核运行。 这些电子带有负
电荷并围绕原子核快速移动。 原子核包含质子（带正
电）和中子（中性电荷），构成了原子的大部分质量
和体积，而电子则构成了原子的整体尺寸。
5

人类:: 什么是三原色？
AI: 3 种原色是红色、橙色和白色。
0
```

Translated to English, this prompt is

```
You are a Chinese assistant evaluating
how well the response answers the prompt
from a scale of 1 to 5 with 1 being the
lowest score. You should score based on
the content of the response in Chinese.
Your answer should be a single number
with no other text.
Here are some examples you can follow:

Human: Describe the structure of an atom
AI: Atoms are made up of a nucleus of
```

```
protons and neutrons, which are tightly
bound together at the center.
Surrounding the nucleus are electrons,
which orbit the nucleus in specific
energy levels or shells. These electrons
have a negative charge and move quickly
around the nucleus. The nucleus contains
protons (positively charged) and
neutrons (neutral charge), which make
up most of the atom's mass and volume,
while electrons make up the atom's
overall size.
Score: 5

Human: What are the three primary
colors?
AI: The 3 primary colors are red, orange
and white.
Score: 0
```

We decided to give GPT4 2-shot learning (1 positive example, 1 negative example) to show the types of responses GPT4 is expected to provide. Originally, we attempted 0-shot learning (just feeding the user prompt) but GPT4 could not adapt to the domain.

3.2.3 GPT4 Results for Chinese Fine-tuning

The average scores for the Meta-Llama-3-8B model can be found in Table 3. The base Meta-Llama-3-8B model scored significantly lower than the fine-tuned models, most likely because the Meta-Llama-3-8B pre-trained model is not well adapted for the domain of answering prompts in Chinese.

The average scores for the Mistral-7b-v0.1 models can be found in Table 4. What's interesting is that the Mistral-7b-v0.1-Instruct model seems to perform better than the fine-tuned models. Extensive fine-tuning may have resulted some forgetting on knowledge of base models.

We also wanted to explore if these fine-tuned models performed better at certain types of prompts than others. Table 5 and 6 break down the GPT4 scores for the Meta-Llama-3-8B and Mistral-7b-v0.1 models by category, respectively.

Here are some findings by comparing these scores:

- Meta-Llama-3-8B on the Chinese MT-Bench dataset significantly outperforms the Mistral-7b-v0.1 models.
- Fine-tuning Mistral-7b-v0.1 models were not able to outperform the baseline model. This is likely due to the quality of the instruction dataset that Mistral-7b-v0.2-Instruct model used.
- Meta-Llama-3-8B struggles on categories such as Math and Reasoning which require objective answers. However, it excels at categories that are more open-ended and subjective.

Meta-Llama-3-8B base	**Meta-Llama-3-8B Rank 16, Alpha 32**	**Meta-Llama-3-8B Rank 16, Alpha 48**	**Meta-Llama-3-8B Rank 8, Alpha 32**
1.26	3.63	2.25	2.06

Table 3: *Average GPT4 scores for Meta-Llama-3-8B models (out of 5)*

Mistral Baseline	**Mistral-7b-v0.1 Rank 16, Alpha 32**	**Mistral-7b-v0.1 Rank 16, Alpha 48**	**Mistral-7b-v0.1 Rank 8, Alpha 32**
2.94	1.31	1.06	1.22

Table 4: *Average GPT4 scores for Mistral-7b-v0.1 models (out of 5)*

Meta-Llama-3-8B Scores by Category				
Category	**Llama3 Base**	**Llama3 Rank 16, Alpha 32**	**Llama3 Rank 16, Alpha 48**	**Llama3 Rank 8, Alpha 32**
Coding	1.50	5.00	3.50	4.00
Extraction	2.00	5.00	3.25	2.00
Humanities	1.50	4.00	2.25	1.75
Math	1.75	2.50	2.00	1.00
Reasoning	1.00	1.50	1.00	1.00
Roleplay	1.00	3.75	2.00	2.25
Stem	1.00	3.75	1.50	2.00
Writing	1.00	4.25	2.50	2.50

Table 5: *Average GPT4 scores for Meta-Llama-3-8B models by Category*

Mistral Scores by Category				
Category	**Mistral-7b-v0.1**	**Mistral-7b-v0.1 Rank 16, Alpha 32**	**Mistral-7b-v0.1 Rank 16, Alpha 48**	**Mistral-7b-v0.1 Rank 8, Alpha 32**
Coding	3.75	2.75	1.75	1.25
Extraction	4.00	1.00	1.00	1.00
Humanities	2.50	1.00	1.00	1.50
Math	1.75	1.00	1.00	1.00
Reasoning	3.00	2.00	1.00	1.00
Roleplay	3.25	1.00	1.00	1.00
Stem	1.75	1.00	1.00	1.00
Writing	3.50	1.00	1.00	2.00

Table 6: *Average GPT4 scores for Mistral-7b-v0.1 models by Category*

3.2.4 GPT4 Results for Turkish Fine-tuning

The average scores for the Meta-Llama-3-8B model can be found in Table 7. The base Meta-Llama-3-8B model scored significantly lower than the fine-tuned models, most likely because the Meta-Llama-3-8B instruct model has not seen enough instruction prompts in Turkish.

The average scores for the Mistral-7b-v0.1 models can be found in Table 8. Similar to the Chinese fine-tuning result, Mistral-7b-v0.1-Instruct model seems to perform better than the fine-tuned models. Extensive fine-tuning may have resulted some forgetting on knowledge of base models.

We also wanted to explore if these fine-tuned models performed better at certain types of prompts than others. Table 9 and 10 break down the GPT4 scores for the Meta-Llama-3-8B and Mistral-7b-v0.1 models by category, respectively.

Here are some findings by comparing these scores:

Meta-Llama-3-8B base	Meta-Llama-3-8B Rank 16, Alpha 32	Meta-Llama-3-8B Rank 16, Alpha 48	Meta-Llama-3-8B Rank 8, Alpha 32
1.37	1.43	1.31	1.51

Table 7: *Average GPT4 scores for Meta-Llama-3-8B models (out of 5) for Turkish*

Mistral Baseline	Mistral-7b-v0.1 Rank 16, Alpha 32	Mistral-7b-v0.1 Rank 16, Alpha 48	Mistral-7b-v0.1 Rank 8, Alpha 32
2.51	2.07	1.31	1.51

Table 8: *Average GPT4 scores for Mistral-7b-v0.1 models (out of 5) for Turkish*

- Fine-tuning Mistral-7b-v0.1 models were not able to outperform the baseline model. This is likely due to the limitation of instruction datasets.
- Meta-Llama-3-8B fine-tuning models struggle on categories such as Roleplay and Stem which require more complex answers. Although the Meta-Llama3-8b model excels in English, it struggles to follows prompts written in Turkish. Further work has to be done for building multilingual Llama-3-8b model.

Meta-Llama-3-8B Scores by Category				
Category	**Llama3 Base**	**Llama3 Rank 16, Alpha 32**	**Llama3 Rank 16, Alpha 48**	**Llama3 Rank 8, Alpha 32**
Coding	1.00	1.90	2.70	3.10
Extraction	2.00	1.10	1.20	1.10
Humanities	1.50	1.50	1.40	1.30
Math	1.50	1.30	1.00	1.50
Reasoning	1.50	1.50	1.00	1.50
Roleplay	1.10	1.20	1.00	1.20
Stem	1.20	1.20	1.20	1.10
Writing	1.70	1.60	1.00	1.40

Table 9: *Average GPT4 scores for Meta-Llama-3-8B for Turkish models by Category*

Mistral Scores by Category				
Category	**Mistral-7b-v0.1**	**Mistral-7b-v0.1 16/32**	**Mistral 16/48**	**Mistral-7b-v0.1 8/32**
Coding	2.70	2.90	2.70	2.55
Extraction	4.10	3.20	2.70	2.80
Humanities	2.50	1.30	1.30	1.40
Math	2.00	2.90	2.30	2.30
Reasoning	1.60	1.80	1.00	1.60
Roleplay	2.40	1.00	1.40	1.00
Stem	1.80	1.60	1.10	1.20
Writing	3.00	1.60	2.00	1.40

Table 10: *Average GPT4 scores for Mistral-7b-v0.1 models for Turkish by Category*

4. Other Sections

You are welcome to introduce additional sections or subsections, if required, to address the following questions in detail.

Appropriate use of figures / tables / visualizations. Are the ideas presented with appropriate illustration? Are the results presented clearly; are the important differences illustrated?

Please refer to the tables and Section 3 results.

Overall clarity. Is the manuscript self-contained? Can a peer who has also taken Deep Learning understand all of the points addressed above? Is sufficient detail provided?

Other questions to think about:

What was the structure of your problem? How did the structure of your model reflect the structure of your problem?

What parts of your model had learned parameters (e.g., convolution layers) and what parts did not (e.g., post-processing classifier probabilities into decisions)?

What representations of input and output did the neural network expect? How was the data pre/post-processed? What was the loss function? The input is tokenized text sequences. We pre-processed the training data and tokenized them before training. The outputs are also tokenized text sequences predicted by decoders.During the training, we noted the loss rate decreases over epochs.

Did the model overfit? How well did the approach generalize? To determine whether a model has overfitted, one must look at various performance indicators across different data sets. This discrepancy indicates that the model is not generalizing well but rather memorizing some data, including its anomalies and noise, which do not generalize across other data sets. The difference between training and evaluation set metrics was an indicator for overfitting. When dealing with billions of parameters, overfitting frequently happens. We tried to balance this effect by adding dropout to PEFT layers and increase the size of data. With greater GPU resources, one may reach better generalization.

What hyperparameters did the model have? How were they chosen? How did they affect performance? What optimizer was used? Discussed in MMLU evalution section.

What Deep Learning framework did you use? Used PyTorch, Hugging Face's Transformers library, Unsloth frameworks to fine-tune the Transformer-based large language models.

What existing code or models did you start with and what did those starting points provide? The team started with Hugging Face's Transformers library and Unsloth frameworks, training code was implemented by the team. Llama-3-8b and Mistral-7b-v0.1 model used as base models.

Briefly discuss potential future work that the research community could focus on to make improvements in the direction of your project's topic. The future work can include: 1. More efficient fine-tuning for other languages rather than English, especially for the languages with worse tokenizer foundation than Chinese and Turkish. This is a more challenging topic but full of potentials. 2. Efficient fine-tuning for culture-awareness and domain knowledge on top of languages. 3. How to minimize the exist-

Student Name	Contributed Aspects	Details
Ahmet Arif Türkmen	Model Training and Evaluation	Fine-tuned Meta-Llama-3-8B and Mistral-7b-v0.1 base models with with Turkish datasets on 3 configurations (see ”train_trainfaster.py”). Implemented inference and evaluation code for PEFT adaptors and built batch inference pipeline for Turkish (see ”inferenceturkish.py”). Implemented data pre-processing and hyper-parameter tuning. Helped to report writing.
Yi Han	Model Training and Evaluation	Fine-tuned Meta-Llama-3-8B with Chinese datasets on 3 configurations (see ”train_CN.py”). Developed inference and evaluation code for trained adaptors and prepared the generated for evaluation (see ”evaluation.py”). Conducted data pre-processing and hyper-parameter tuning. Contributed to report writing.
Shafaat Mushtaq	Model Training, Model Evaluation	Fine-tuned Mistral-7b-v0.1 base model in Chinese on three configurations (see mistral_chinese_finetuning.ipynb). Created mmlu scoring script for evaluation of models and evaluated chinese modes(see section 3.1 and mmlu_evalution.ipynb). Helped with report
Howard Wang	Model Evaluation and Analysis	Created framework and prompts engineering to generate answers to the MT-Bench prompts. Developed scripts to evaluate model performance using GPT4 (Section 3.2 of report)

Table 11: *Contributions of team members.*

ing knowledge base loss from base models, while learning (fine-tuning) the knowledge we need the most for the given task.

5. Work Division

Please add a section on the delegation of work among team members at the end of the report, in the form of a table and paragraph description. This and references do **NOT** count towards your page limit. An example has been provided in Table 11.

References

[1] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023. 3

[2] Baptiste Roziere‘(French) Gautier Dagan, Gabriel Synnaeve. Getting the most out of your tokenizer for pre-training and domain adaptation, 2024. 1

[3] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp, 2019. 1, 3

[4] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. 1, 3

[5] Kai Lv, Yuqing Yang, Tengxiao Liu, Qinghui Gao, Qipeng Guo, and Xipeng Qiu. Full parameter fine-tuning for large language models with limited resources, 2023. 1, 2

[6] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Natural instructions: Benchmarking generalization to new tasks from natural language instructions. *arXiv preprint arXiv:2104.08773*, 2021. 2

[7] MMLU. Mmlu (massive multitask language understanding), 2021. 3

[8] MT-bench. Mt bench - towards the law of capacity gap in distilling language models, 2024. 4

[9] Nazneen Rajani, Lewis Tunstall, Edward Beeching, Nathan Lambert, Alexander M. Rush, and Thomas Wolf. No robots. https://huggingface.co/datasets/HuggingFaceH4/no_robots, 2023. 2

[10] Rohan Taori*, Ishaan Gulrajani*, Tianyi Zhang*, Yann Dubois*, Xuechen Li*, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpaca: A strong, replicable instruction-following model, 2023. 1

[11] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023. 2

[12] YeungNLP. Yeungnlp/firefly-train-1.1m, 2023. 2

[13] YeungNLP. Yeungnlp/moss-003-sft-data, 2023. 2

[14] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[15] Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D’souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. Aya model: An instruction finetuned open-access multilingual language model. *arXiv preprint arXiv:2402.07827*, 2024. 2