

Some tips on choosing a final
project

Possible project types

- There are many possible project types. We will give some suggestions.
- Note: This is not an exhaustive list.
- 1- Use an existing approach for a new task or application and suggest how we can improve it without significant changes.
 - Example: Can we use Chain-of-thought prompting to improve medical question answering?
 - Example: How to improve in-context learning for mathematical reasoning?
- 2- Propose a new approach or an improvement to an existing approach for a particular task or application
 - Example: A new training objective for mathematical reasoning.
 - Example: An approach for more efficient finetuning.
 - Example: Automate prompt engineering

Possible project types

- 3- Analyze the behavior of a model to provide some new insights.
 - E.g., Investigate hypotheses like “language models are more likely to hallucinate content if they are presented with unfamiliar examples”.
 - Another example: Investigations into the data. How training data can impact some behavior of LLMs.
 - Another example: Investigating the bias and trustworthiness of LLMs
- 4- Find an interesting shortcoming or weakness of existing LLMs.
 - For these projects you can play around with GPT3 or ChatGPT, look at the data, and find some interesting patterns. These will help form hypotheses that you can investigate.
 - E.g., LLMs do not usually perform well for X where X is a task that we care about.
 - Ideally these type of project should provide some hypotheses or insights on why LLMs are not great at performing a particular task, or if there’s a way to mitigate that problem.

Possible project types

- 5- Suggest a novel modeling approach for a task or set of tasks to improve some functionality.
 - You shouldn't think of proposing a completely model architecture!!
 - Instead, you should think of “lightweight” methods to that can be used alongside a main model to improve their performance, or a desired functionality
 - E.g., For in-context-learning how we can identify better in-context examples using KNN clustering.
 - E.g., For chain-of-thought prompting how can we use document retrieval to find good examples
 - E.g., How to improve the decoding methods to prevent models from hallucinating content.
 - E.g., How can we make models more efficient?
- 6- Theoretical analysis: Show some interesting and new theoretical analysis of an existing approach.
 - This type of project is only recommended for folks who have a strong theoretical background
 - E.g., prove that sparse attention models are universal function approximators.

Ok great, But do you have more suggestions on how to find concrete example projects?

Few suggestions to get more concrete ideas

- Check out the latest conferences in NLP (ACL, EMNLP, NAACL) and quickly scan through the papers to find topics of your interest
 - <https://aclanthology.org/> (only focus on ACL, EMNLP, NAACL, and findings)
 - EMNLP 2022 is here: <https://preview.aclanthology.org/emnlp-22-ingestion/volumes/2022.emnlp-main/>
 - Also checkout NLP related papers in ML conferences like [ICLR](#), NeurIPS, ICML
- You can also check out "custom projects" section of [Stanford NLP course](#) for more inspiration
- Look at an interesting interdisciplinary problems,
 - This is a perfect project for those who have other backgrounds in addition to CS
 - E.g., Legal document summarization
 - E.g., Adapting LLMs for applications in medical domain
 - E.g., Using efficient transformers to build better music language models
 - E.g., Applications to mental health

Practical tips

- Use Huggingface (<https://huggingface.co/>) as your starting point
 - For models, training scripts, downloading datasets
 - Majority of the models that we care about are already on Huggingface

```
from transformers import AutoTokenizer, AutoModel, T5ForConditionalGeneration
tokenizer = AutoTokenizer.from_pretrained('t5-small')
model = T5ForConditionalGeneration.from_pretrained('t5-small')
inputs = tokenizer(["translate English to German: That is good."], return_tensors='pt')
tokenizer.decode(model.generate(**inputs)[0])
```

```
Out[34]: '<pad> Das ist gut.</s>'
```

Practical tips

- If your project is analysis of LLMs or you are interested in a particular phenomena in LLMs, first try playing around with GPT3 and ChatGPT

<https://platform.openai.com/playground>

Practical tips

- Data: Typically, two possible situations
 - Your project requires some amount of labeled data
 - 1- You already have labeled data either from existing datasets or some dataset that you have access to -> Great!
 - 2- You have the unlabeled data for your problem -> Try quickly annotating a small set. Make sure your scope of project is in few-shot / low-resource training. Generally, we recommend avoiding project that requires significant data annotation work
 - 3- You don't have any data or any easy way to get data -> We don't encourage working on a problem that requires labeled data but doesn't have a clear way to get the data.
 - Your project doesn't need any labeled data but needs some general unlabeled data either for analysis or training
 - 1- Data needed for the project mostly exist on public datasets (e.g., Wikipedia, C4, etc) -> Great!
 - 2- Data is private, but it is possible to get it after signing data usage agreements (e.g., patient data) -> If it is quick, then sure. If the logistics are complicated, we don't recommend it
 - Most people use existing datasets

Practical tips

- Compute:
 - Yale High Performance Computing Servers:
<https://research.computing.yale.edu/services/high-performance-computing>
 - Your own compute resources
 - Cloud compute
 - We may be able to get limited cloud credits (TBD)
- Generally, avoid projects that require significant amount of compute
 - Unless you already have access to some large compute cluster
 - E.g., Don't propose a new pretraining objective or don't perform any large scale finetuning or instruction tuning project

Questions?