CS605 Project Instructions

1. Overview

The project aims to apply NLP techniques to some real-world tasks. This document specifies the detailed requirements, scopes, deliverables for the project, which is worth 30% of your overall grade. The topics of the project include four candidate projects and a custom project. You are free to choose one of them as your project. The evaluation of the project will break into three parts, proposal report (10%), final presentation (10%) and final report (10%).

2. Groups

Each group is up to 5 students. Workload for each should be distributed equally. At the end of final report for each group, there will be anonymous peer review. Group members by default will receive equal grades unless with considerably unequal contribution.

3. Custom project

Description:

Start by finding a problem of interest. Then formulate the problem to a task to be solved. Either collect your own data or retrieve it online for existing ones to support your experiments. Develop and implement models to address the task and provide analysis.

Detailed requirements:

- *Datasets*. Collect and make your own data (e.g. annotate small amount of data; seek different sources of supervised data; target at unsupervised data). Another option is to retrieve existing sources:
    - Huggingface datasets: https://huggingface.co/datasets
    - Paperswithcode datasets: https://www.paperswithcode.com/
    - Kaggle datasets: https://www.kaggle.com/datasets?fileType=csv
    - Research papers
    - Linguistic Data Consortium: https://catalog.ldc.upenn.edu/
- *Model*. You are required to build your own model and explain the rationale after each design. Implementation can be, but not limited, in Pytorch.
- *Experiments*. You are required to conduct abundant experiments to prove your assumption of your proposed solution.
- *Analysis*. You are required to conduct experimental analysis on (i) preprocessing operations, e.g. tokenization, normalization etc. (ii) model design. (iii) hyper-parameters.

4. Candidate project 1: GPT understanding.

Description:

The goal is to understand the knowledge from LLMs. Although recent LLMs are becoming omnipotent in various applications, such as text summarization and machine translation, how they store and manage the knowledge learnt during pre-training is still unclear. You will systematically evaluate and analyze LLMs by designing suitable tasks and datasets, thus not only provide insights for better understanding the working mechanism, but also to explore the limitations for improvement directions, e.g., when the LLMs fail [1]. Here are several papers for reference [2-3].

[1]. McKenzie, I. R., Lyzhov, A., Pieler, M., Parrish, A., Mueller, A., Prabhu, A., ... & Perez, E. (2023). Inverse Scaling: When Bigger Isn't Better. arXiv preprint arXiv:2306.09479.
[2]. Finding Skill Neurons in Pre-trained Transformer-based Language Models. Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, Juanzi Li. EMNLP2022.
[3]. Holistic Evaluation of Language Models. Percy Liang, at. el. Arxiv 2022
[4]. Is ChatGPT a General-Purpose Natural Language Processing Task Solver? Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, Diyi Yang. Arxiv 2023.

5. Candidate project 2: RAG Application

Description:

In this project, students will explore the innovative domain of Retrieval-Augmented Generation (RAG), an approach that integrates the retrieval of external knowledge into the process of generating text. This project aims to enhance understanding of how RAG models leverage retrieved documents to produce more informative and contextually relevant responses in NLP tasks. Some example references are provided below.

[1]. Xu, P., Ping, W., Wu, X., McAfee, L., Zhu, C., Liu, Z., ... & Catanzaro, B. (2023). Retrieval meets long context large language models. arXiv preprint arXiv:2310.03025.
[2]. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., ... & Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997.

6. Candidate project 3: Tool Learning

Description:

LLMs have demonstrated a great reasoning and generalization ability in various applications. However, they still have many weaknesses, such as lack of latest information, accurate calculation, time-awareness, hallucination, etc. A straightforward idea is to teach the LLMs learn to use external APIs, such as search engine and calculator. You may design suitable pipeline to construct an instruction tuning dataset to tune a LLMs (e.g., 7B LLaMA), or other effective method, empowering them an ability to generate code-like instructions towards real-world applications. Here are some paper for references.

[1]. Toolformer: Language Models Can Teach Themselves to Use Tools. Timo Schick, at. el. Arxiv 2023.
[2]. HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in HuggingFace. Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, Yueting Zhuang. Arxiv 2023.
[3]. Augmented Language Models: a Survey. Gregoire Mialon, at el. Arxiv 2023.

7. Candidate project 4: LLMs continual learning

Description:

As more and more LLMs are used in real-world applications, they can access much valuable information from users' feedback, behavior, latest news articles, etc. By incorporating these data, LLMs can learn persona, new data, or domain-specific knowledge. How to improve LLMs' ability from continual data, while keeping their abilities in old tasks, is a major challenge. You may explore a novel way, e.g., to include an external memory, to this end. Here are some papers and project for references.

[1]. Long-term memory for LLM via self-replicating prompt.
[2]. Large Language Models with Controllable Working Memory. Daliang Li, at el. Arxiv 2023.

8. Project proposal (1-2 pages)

For candidate projects, introduce the task you have selected (task definitions and literature review). Introduce the fundamentals of required datasets. Give brief demonstration of proposed methods.

For custom projects, present the background, motivation behind your selected problem. Give brief introduction of your problem formulation analysis, data source and general solution.

9. Project presentation (15mins)

Each team will have a change to present your project and results in class. The rest groups will make peer review and ask questions. Peer review will affect the project grading. So, try to make your project interesting and important!

10. Final report (6-8 pages, excluding references)

Your report should follow the similar format as NLP research paper. Consider using below the section structure, but you can use different ones too.

Abstract. An abstract should briefly motivate the problem, describe your aims, describe your contribution, and highlight your main findings.

Introduction. The introduction explains the problem, why it's difficult, interesting, or important, how and why current methods succeed/fail at the problem, and explains the key ideas of your approach and results. Though an introduction covers similar material as an abstract, the introduction gives more space for motivation, detail, references to existing work, and to capture the reader's interest.

Related work. This section helps the reader understand the research context of your work, by providing an overview of existing work in the area.

Approach. This section describes your approach to the problem. For example, this is where you describe the architecture of your neural network, and any other key methods or algorithms.

Experiments. This section contains the following.

- Data: Describe the datasets you are using (give references).
- Evaluation method: Describe the evaluation metrics you use.
- Experimental details: Report how you ran your experiments (e.g. model configurations, learning rate, training time, etc.)

Results. Report the quantitative results that you have obtained. Use tables or plots to compare results and compare against baselines.

Analysis. This section is the qualitative evaluation. Provide reasons for the results you obtained. Explain why certain methods work or not work, how the results support/contradict your initial assumptions and/or model design.

Conclusion. Summarize the main findings of your project and what you have learned. Provide limitations of your method and possible future work.

Reference. Literatures that help you along the way.

11. Codes

Zip your codes and submit along with final report.

- Include all project code written or adapted by you.
- Don't include the source codes you used by calling off-the-shelf packages without adaptation.
- For include the data (with a shared link if large).

12. Submission

a. Project proposal should be in PDF format.

- Zip your codes and final report in PDF format with the name of group number, eg. "group3.zip"
- Submit your zip file via eLearn. If upload cannot complete, then upload the file to Google drive and email the link to our instructor Jean (jeanchen@smu.edu.sg) and professor Cao Yixin (yxcao@smu.edu.sg). The email will be replied within 48 hours in acknowledgement of the receipt. If you don't receive it, consider sending it again or contact the professor.
- Submission deadline:
  - Project proposal: 30 May 2023 11:59PM
  - Final report: Sunday of Exam week, 11:59PM
  - Late submissions are allowed but will be penalized by -0.5% every calendar day (until zero). Can submit for multiple times, only the latest submission will be graded and time-stamped.
  - Do not forget the presentation in Week11 including recess week.