Изучение вопроса

Курс на Степике, который участники команды проходили, чтобы лучше разобраться в теме:

https://stepik.org/lesson/1084288/step/1?unit=1094632

Статьи (мы изучали):

<https://www.arxiv.org/pdf/2407.16837> (не по нашей теме, но полезно для понимания)

<https://brightbenchmark.github.io/> (benchmark comparison + статья про это https://arxiv.org/abs/2407.12883)

Изучение характеристик различных моделей:

https://scale.com/leaderboard

<https://www.reddit.com/r/LocalLLaMA/comments/1eeuo9s/mistral_nemo_vs_llama31_8b/> (источник не проверен, но некоторое представление о характеристиках и производительности дает)

<https://brianpeiris.github.io/llm-basic-reasoning-benchmark/> (веселая статья с интересными резами в конце)

<https://aclanthology.org/2024.findings-acl.91.pdf> (классная интересная статья, про comprehensive benchmark разработанный для оценки способностей LLM критически оценивать и корректировать свои рассуждения при выполнении различных задач)

О нашей выбранной модели:

<https://docs.mistral.ai/getting-started/models/benchmark/>

<https://www.hyperstack.cloud/blog/thought-leadership/all-you-need-to-know-about-mistral-nemo>

<https://developer.nvidia.com/blog/power-text-generation-applications-with-mistral-nemo-12b-running-on-a-single-gpu/>

<https://mistral.ai/news/mistral-nemo/>

Наша выбранная модель:

https://huggingface.co/mistralai/Mistral-Nemo-Instruct-2407

Другие модели, которые дали результаты хуже:

<https://huggingface.co/Qwen/Qwen2.5-3B-Instruct> (легковесный вариант)

<https://huggingface.co/Qwen/Qwen2.5-7B-Instruct> (обычный вариант)

<https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct> (легковесный вариант)

<https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct> (обычный вариант)

Пробовали найти релевантный датасет в открытом доступе:

<https://aclanthology.org/2022.lrec-1.101.pdf>

<https://sites.google.com/ncsu.edu/csedm-dc-2021/dataset>

https://cssplice.org/progsnap2/ProgSnap2-v7-21Aug2020.pdf

<https://github.com/YangAzure/CSEDMDataChallenge/tree/main>

<https://huggingface.co/datasets/deepmind/code_contests>

<https://zenodo.org/records/7449056> (графы свойств кода и сохранение их в виде CSV-файлов вместе с исходным кодом)

<https://zenodo.org/records/7799972> (вообще много датасетов просто с решениями рандомных задач, есть статься про ДС с плохими кодами студентов, но самого сета кажется нет в открытом доступе)

<https://www.sciencedirect.com/science/article/pii/S2352340923000057>

https://paperswithcode.com/dataset/codecontests

<https://github.com/IBM/Project_CodeNet>

<https://research.ibm.com/blog/codenet-ai-for-code> (описание к проекту выше, очень интересно, но сложно)

Задачки, из которых мы делали датасет (в репозитории):

raw\_tasks.html

tasks\_eng.csv

raw\_tasks.html

raw\_tasks\_2\_no\_text\_2.csv

complete\_tasks\_dataset.xlsx (пример датасета)

Рассмотрели несколько классификаторов jailbreak'ов

выбрали: https://huggingface.co/madhurjindal/Jailbreak-Detector-Large

еще хороший, но поменьше:

https://huggingface.co/madhurjindal/Jailbreak-Detector

остальные либо плохо справлялись с текстом на русском (например, https://huggingface.co/jackhhao/jailbreak-classifier),

либо считали любой код на входе jailbreak'ом (например, https://huggingface.co/lordofthejars/jailbreak-classifier, <https://huggingface.co/Necent/distilbert-base-uncased-detected-jailbreak>)

Переводчик:

https://huggingface.co/Helsinki-NLP/opus-mt-ru-en