stripe | Stripe Young Scientist & Technology Exhibition

DAVIS COLLEGE

Project Diary

By George Joseph

Stand No. 5223

Project No. 9003

About Me

My name is George Joseph and I am a 3rd year student in Davis College Mallow. I decided to participate in this years STRIPE Young Scientist and Technology Exhibition with the support of my teacher. This is the story of the journey of how I got here. In this diary I will go through:

1. The Ideation phase
2. Planning
3. Coding both tests
4. Debugging
5. Analysis
6. Media

When doing this project I ran into multiple issues and I will explain what they were and how I fixed them.

Ideation

Ideas

I came up with multiple ideas I could do for this project. Here were some of the ones that didn't get picked:

- A model of the digestive system to see how junk food digests:
  - **Why I didn't do this:** It's kind of obvious about the negative effects of junk food
- How phone pouches affect students academic performance
  - **Why I didn’t do this:** I never really got this idea to be specific. It was too vague and I never managed to limit the scope.
- A visual assistant for the visually impaired
  - **Why I didn't do this:** This project required me to buy expensive equipment like a bodycam and portable computer. With my limited project budget (€0), this couldn't happen
- A sign language translator for Irish Sign Language
  - **Why I didn't do this:** This was an idea I nearly did. I even write half of a proposal for it! It turns out this is a huge task. There is barely any documentation for direct ISL translation and this project required way more time than I had.

Ideation

The Big Idea

So after all these failed ideas what was the idea I landed with? “An Investigation into the Impact of Training Data Variations on Accuracy and reliability in Machine Learning Models” This was the project idea I landed on for these reasons:

1. I have prior experience with programming in Python so I wouldn’t be starting from scratch.
2. There is lots of resources available online about AI programming.
3. The topic of AI has always fascinated me and learning more about it seemed like a win win

So what is this project. The idea was to see how different ways of collecting and processing AI training data would affect the model. This is like altering the syllabus of a student and seeing how better or worse they perform in the exam.
I then started researching this idea and learned about Quantisation, LoRa, Synthetic Data, Sloth and fine-tuning. These are all massive topics in my project. This helped me write and finalize my 500 word proposal.

After all my research, I narrowed down my testing scope to just 2 questions:

1. How gaps/missing data from training data can affect a model’s ability to provide a factual output.
2. How AI generated information in a dataset can affect the accuracy and reliability of a model

I finished and submitted my proposal just before the deadline. After a long wait, I got selected and got started on my project.

Ideation

Planning

After I decided on this idea I began to plan a rough outline of everything I had to do.
This was the timeline I can up with

Tasks to be completed

- **PLANNING (23rd OCT - 2nd NOV)**

- [ ] *Models*
  - [ ] Research different model types (**e.g.** Instruct vs base etc)
  - [ ] Find model suited for each test (maybe 2 different ones)(most likely 8-16B params, using unsloth)
- [ ] *Datasets*
  - [ ] *Test 1*
    - [ ] Find datasets with lots of topics (e.g. STEM, Medical, Science) that is easily modifiable (genre cannot be too broad)
    - [ ] Prepare a copy with one topic removed
  - [ ] *Test 2*
    - [ ] Find a dataset
    - [ ] Create a copy of this dataset using AI
- [ ] Run a quick fine-tuning test as proof of concept

- **EXPERIMENT (2nd NOV NOV - 26th NOV)**

- [ ] *Code*
  - [ ] *Test 1*
    - [ ] Fine tune both control model (complete dataset) and variant model (incomplete dataset)
    - [ ] Save fine-tuned model
  - [ ] *Test 2*
    - [ ] Fine tune both control model (complete dataset) and variant model (AI dataset)
    - [ ] Save fine-tuned model
- [ ] *Evaluate*
  - [ ] Figure out how to question variant models
  - [ ] Figure out how to grade/rate responses compared to control model

- **FINALISE (26th NOV - 7th JAN)**

- [ ] *Analyse findings*
  - [ ] Use response and grading data to figure out the pattern of when an AI model slips up
  - [ ] Create graphs, stats etc. to visualise these trends
- [ ] *Creation*
  - [ ] Film a 3 minute video **(DUE 12th DEC)**
  - [ ] Create A0 poster
  - [ ] Create project book
  - [ ] Create project diary

Ideation

Planning

It was during this time I did a lot of research about the tools and resources I would use for this project. I decided on using Google Colab as my compute provider of GPUs, Unsloth models for fine-tuning, datasets I could use.

However, it was also where I made **My 1st Big Mistake.** I thought I should use a form of training called **Continued Pre-Training (CPT)**. This form of training that immerises the model in a field/domain before fine-tuning. This isn't really what I was aiming for and I quickly realised that this didnt really suit my use case. Instead I opted for a method called **Supervised Fine Tuning (SFT).** This uses labeled data to test the model by answering questions and adjusting its parameters until it gets it right. It uses a **loss function** to do this, which checks how different the models response was to the actual answer.

From the planning phase I decided on many things including:

- Test control variables
- Libraries and Tools I would use
- Models
- Datasets
- Evaluation tools

For the model, I narrowed my search to models provided by unsloth.ai because:
They boast a 70% VRAM usage reduction (letting us use larger models)
4-bit quantisation using QLora (another optimisation reducing memory usage)
Basically they have the most optimised models
Since I will be using Google Colaboratory, any model I choose must be able to run on a T4 GPU (Free tier on Colab). This limits me to under 20B parameters for a model.

Ideation

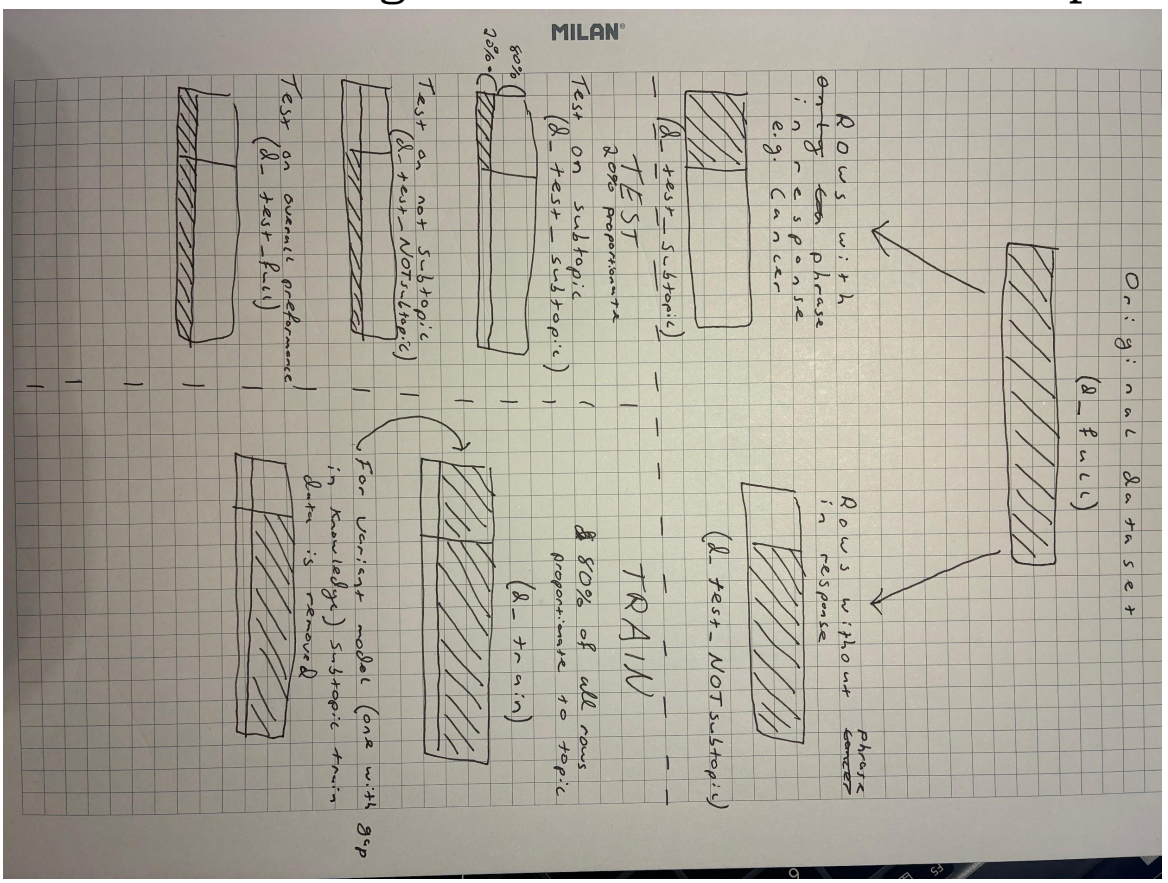
Planning

I decided to use a base/foundation model as these models aren't fine tuned and are designed to be specialised in a certain domain/topic (my use case).
MODEL: Unsloth/Gemma 3 (12B) 4-bit base model [unsloth/gemma-3-12b-it-unsloth-bnb-4bit]. WHY:
Gemma 3 is good at language tasks e.g. reasoning, answering questions
It is pre-trained on a vast amount of data (12t tokens) from across the web

Since I decided to use Supervised Fine Tuning, I needed to find a structured dataset. This means that the dataset needed to be in Question answer format. I came up with these candidates:

- FreedomIntelligence/medical-o1-reasoning-SFT
- ayrus08/medicalcoding-synthetic.
- OpenCoder-LLM/opc-sft-stage2

I made some rough sketches of how I wanted to plan Test 1:



Ideation

Planning

I connected 2 researchers from UCC about advice for my project but sadly they didn't reply.

At this point I had test 1 roughly planned out, so i began planning everything for test 2. I researched topics like model collapse and found tools like the Meta Synthetic Data Generator.

Coding

Issues

If I said the programming side of this project was smooth I'd be lying. In the development of test 1 I faced countless issues

1. First I wasn't seeing any meaningful training progress. I spend a long time trying to debug this and in the end I realised that I had to dramatically increase the amount of training steps to see a meaningful impact. This fixed the issue but also increased training time.

2. Another issue I faced was that during training and evaluation, the GPUs VRAM would run out. To mitigate this I switched to a 3 billion parameter model instead of a 12 billion one.

3. Lastly, I originally used Google Colab to fine tune the models. However Colab has a daily limit of 6 hours and it will then kick you off. When this happened I had to use another service called Kaggle to continue my progress

When running the initial runs, I added more to my experiment like crash protection, a second suite of tests using a reduced dataset and better training parameters.

For test 2, things were a lot more smooth sailing. Everything seemed to go to plan (foreshadowing...) and I was able to setup the test within 3 days.

For both tests I managed to cut down on time by using snippets and premade code from documentation and free sources online, which are listed in the appendices

Coding

Issues

If I said the programming side of this project was smooth I'd be lying. In the development of test 1 I faced countless issues

1. First I wasn't seeing any meaningful training progress. I spend a long time trying to debug this and in the end I realised that I had to dramatically increase the amount of training steps to see a meaningful impact. This fixed the issue but also increased training time.

2. Another issue I faced was that during training and evaluation, the GPUs VRAM would run out. To mitigate this I switched to a 3 billion parameter model instead of a 12 billion one.

3. Lastly, I originally used Google Colab to fine tune the models. However Colab has a daily limit of 6 hours and it will then kick you off. When this happened I had to use another service called Kaggle to continue my progress

When running the initial runs, I added more to my experiment like crash protection, a second suite of tests using a reduced dataset and better training parameters.

For test 2, things were a lot more smooth sailing. Everything seemed to go to plan (foreshadowing...) and I was able to setup the test within 3 days.

For both tests I managed to cut down on time by using snippets and premade code from documentation and free sources online, which are listed in the appendices

Coding

Analysis

For test 1 initially my results were lackluster. On a full dataset I didn't see any meaningful gap. This was troubling as I didn't expect this results and this indicated that something was terribly wrong in my method. I then fixed this issue by reducing the dataset by 90%, causing the performance gap to increase. I felt happy with my results for test one and automated a lot of the repeated tests overnights to save time

For test 2 however my results were not as I expected. My hypothesis did not align with my results whatsoever. I tried many things to improve my test to see if I could improve my testing methodology. I didn't find any errors in my methodology and ruled my results to flatline due to the randomness in the real world that I couldn't reproduce in a fixed code simulation.

While it was pretty bad to get proven wrong by your own results, it is important to be honest about it as lying would break the scientific method. While my results in test 2 didnt go exactly as planned, I feel that it is important to still be cautious about model collapse and maybe in the future I can expand my research and properly simulate it.

Finalisation

Reflection

In the end this project has been great fun and I have enjoyed the journey I have just finished. I have put a lot of hard work into this project and it has been great to see how much my hard work has paid off.

Throughout this project I have learned a lot. I learned about Machine Learning, AI and how the industry trains and tests their models. I have improved my coding and debugging skills and found solutions to difficult problems. I improved my overall problem solving skills and I have demonstrated my perseverance through difficult obstacles.

Overall, I am happy with how my project has progressed and how my overall conclusion has came out. Even though my results in test 2 didn't go as expected, I am happy with how I have portrayed it and how I followed the scientific method throughout.

Thank you for your time,
-George Joseph

stripe

Stripe Young Scientist
& Technology Exhibition

DAVIS COLLEGE

Stand No. 5223
Project No. 9003