Reflections on a Remote AI Training Function: A Job in Eliminating Jobs

CS4991 Capstone Report, 2024

Hunter McGuire
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
hhm9vn@virginia.edu

ABSTRACT

While working remote asynchronous may seem like a dream, the reality at a remote AI training company is far more complex. The job offered exposure to a diverse array of new topics within the field of computer science that I hadn't yet touched and allowed me to contribute to the training of new large language AI models (LLMs). While from a content perspective the job was complex and a great learning experience, culturally the job left much to be desired. The experience, most notably, involved an incredibly impersonal, anonymous work atmosphere, where the absence of supportive structures and interpersonal relationships made navigating the complexities of the job very difficult. This, coupled with the scarcity and subjectiveness of feedback, made the job very difficult to get accustomed to. While the company culture left much to be desired, the job did help me practice hard skills in languages I had only worked with in school, and helped me practice navigating ambiguous and uncertain work situations. Finally, the experience largely is accompanied with two main feelings: first of awe that I was able to help contribute to the training of LLMs that I may use in the future, and second of fear that these LLMs I helped train may have the potential to replace their trainers such as myself.

1. INTRODUCTION

"We regret to inform you that due to low accuracy and speed issues you have been automatically disabled from [project]. This decision is not reversible". After a short application process, a technical interview, and an hour-long onboarding workflow, within the first hour of working remotely for my new job that short email popped up in my inbox. Naturally, my heart sank. Was I being fired already? What had I done wrong? Anxiety and self-doubt were paramount. I was confused, disheartened, and insecure. Throughout my next few weeks, this self-criticism quickly gave way to disappointment and disillusionment with the nature of the job I had taken: AI Code Training.

The company I worked for is an online earnings platform that allows English-speaking individuals to earn money while working completely remote and asynchronously. The company is a workforce-facing subsidiary of a major Silicon Valley AI startup—a corporate structure designed specifically for the delegation of training AI models. While the Silicon Valley startup is a small, 600-some person company, the remote company for which I work conversely boasts a global community of 240,000+ [1][2]. The clients include Microsoft, Meta, Nvidia, OpenAI, the US government, as well as countless other fortune 500 companies. My subsidiary offers these clients the ability to train their large language models (LLMs) by dividing and

outsourcing training projects to the contractor's workforce.

AI still has a long learning process ahead, a process that relies heavily on tedious human training and reinforcement that often comes in the form of tagging, labeling, reviewing, and correcting. The needs of these AI products are immense, and the tasks that make up this training and reinforcement can vary greatly in difficulty—from labelling a car as a car in every single frame of a video, to analyzing and debugging code. My job was the latter. My disillusionment with the position came largely from one realization: that the platform and its entire ecosystem is designed to treat its workers as a renewable, fungible, and ultimately expendable resource.

2. RELATED WORKS

The state of AI continues to reach unprecedented levels of sophistication and utility. While AI's rapid evolution is largely driven by advancements in machine learning algorithms, computational power, and the availability of massive datasets, it is still also largely reliant on human aid. Two of the most important sub-fields of AI right now are that of computer vision (CV) and natural language processing (NLP). The first involves lots of image classification, as CV AIs learn to detect and understand their surroundings. The second pertains to exactly as it reads-- that of understanding and using natural language. While both are important, their needs throughout their pre-training and fine tuning differ greatly [3].

Classification-focused AI models, such as those involved in image analysis, require immense amounts of tagged data for their supervised learning. If an AI wants to learn what a horse looks like, it relies on a training set full of images either with or without a horse, that are pre-labelled by humans as such. The AI can then decipher and pick up on patterns in the images, developing a mathematical model in vector form of what a horse looks like in pixels. This model can then be used to analyze images outside of the training set and decipher whether a horse is present. These types of classification focused AI models are called Convolutional Neural Networks (CNNs), and as described above, they rely directly on the correct human classification of their training sets to provide accurate and useful detection models [3][4].

NLP models on the other hand, are often created through unsupervised training. Through the analysis of billions of parameters and input text, the models can pick up on natural language patterns and infer natural order of words in language. The most prevalent of these models is that of Chat-GPT, developed by Open AI. GPT stands for Generative Pretrained Transformer, and the model is based largely around the concept of transformers and their self-attention mechanisms. Transformers operate sequentially, analyzing and placing the next word in a sentence without being able to go backwards. While they rely on immense training sets as well, NLP models—unlike in CV—do not require human intervention in their training sets. Instead, NLP models such as GPT rely on reinforcement learning from human feedback (RLHF). RLHF operates like any other feedback mechanism and is used to fine-tune LLMs toward more desirable outputs [4].

While both fields require human involvement, they differ greatly in their specific human needs. While CV relies on the mindless labelling of photos, NLP relies on intelligent and thoughtful response reviewing and grading. The company I worked for was unfortunately structured the same for both.

3. WORK REVIEW

I didn't have a normal workday, I didn't have defined hours, I didn't have to come into a physical office. Instead, my work experience was flexible, asynchronous, and remote. The flexibility of the job allowed me to work

largely on my own schedule. In terms of concrete work, I was largely grading AI model responses on coding questions. I'd ask the LLM a code related question in a required language with certain other required parameters, and then evaluate, fact-check, grade, and compare multiple model responses. While "AI Coding Trainer" sounds nicer, a more accurate job title is "AI Grader". The work wasn't fascinating, it wasn't my dream job, but it did offer lots of practice in many coding languages (C, C++, JavaScript, SQL, Shell, and Python mainly). Additionally, as AI is already at a pretty sophisticated level, the coding questions often had to be about topics that I didn't fully understand; thus, I got pushed to deeper understand concepts, libraries, and functions I had never seen before. I had to review and execute code to make sure it compiled and worked as requested, review the code according to proper standards and documentation, correct errors when necessary, and provide feedback on the contextual strength of the accompanying description and explanation. I was not expected to have a vast breadth of knowledge of libraries and complex implementations of all languages, and instead was relied upon to be fast, reliable, and correct in my fact-checking and grading. This meant that I had to practice information gathering and internet querying a lot. Altogether, the job helped me practice my coding skills in various languages across a diverse array of topics and helped immensely in my ability to work through uncertain situations.

While in terms of concrete skills the job had lots to offer, I struggled a lot with the company culture. Going back to before, the company was structured the same way for CV labelling taskers as it was those doing code review. This structure seemed intentionally designed to treat us as employees as renewable, fungible, and expendable. For my first two weeks on the job there was no interpersonal contact, nor managerial contact except for automated emails. After that, I was allowed to join a company Slack channel with moderators that were spread to thin and often unavailable. None of the three questions I asked on Slack ever got answered. This structure makes sense to me for labelling. With mindless tasks, support and community is unnecessary. On the contrary, with AI code review, there was frequent ambiguity which was hard to get through.

A good example of this was in one of the first training tasks I ever received on the platform: to review LLM responses to an already-asked data structures and algorithm question. Instead of a DSA question, instead the prompt said, in broken English, that it *will* ask a DSA question. In the first model response, the LLM produced a long explanation of how it might complete a DSA question, complete with an example. The second LLM response simply prompted back asking for the specific DSA question. I chose the second, thinking the first was verbose and unnecessary. I was wrong. In reality I was right, and the grading rubric was wrong, but due to "low accuracy and speed issues" I was removed from that first project. This was within the first hour of my time at the company, and without any support or community to turn to I was left disheartened and confused. As it turns out, weeks later when I went back to my task history the grading rubric had changed to align with what I had answered. I was never added back to that specific project.

4. RESULTS

While the review system for us reviewers was often frustrating, in the end the job did help me immensely in navigating ambiguous work environments and helped me gain hard skills and knowledge in coding languages I had mostly just used in school such as C, C++, JavaScript, Python, and SQL, and Shell. While there were challenges, ultimately, they have helped me realize that I am not well-suited for

remote work and want a close work community wherever I end up in my professional life.

5. CONCLUSION

Key takeaways from my experience at the company largely involves the skills I gained throughout the process, including but not limited to increased proficiency in the languages I used, as well as improved search and discovery skills in obtaining new information, and practice at navigating ambiguous work situations. Additionally, the work experience provided valuable insight into the type of community and professional environment in which I hope to work in the future. Finally, the experience helped me realize that I was not well suited to remote employment. When looking back at the job, although I was working in a very compartmentalized role, I find comfort in knowing that my contributions helped train LLMs that will become ubiquitous and might even be tools I use in the future. Simultaneously, there is also a feeling of uneasiness with knowing that these LLMs that I have helped train have the future potential to surpass their aids like me and replace jobs.

6. FUTURE WORK

Artificial Intelligence for natural language processing still has a long way to go before it becomes self-sufficient [4]. For companies to get highly accurate and useful LLM training, I recommend that they prioritize fostering a helpful community for those that contribute to the reinforcement learning. My work experience here has taught me that community is vital to my working happiness, and I will be striving to prioritize that as well myself in whatever work I end up doing in the future.

REFERENCES

1. Remotasks. (n.d.). *Home page*. Retrieved April 26, 2024, from https://www.remotasks.com/en
2. Scale AI. (2024). *About Scale AI*. Retrieved April 26, 2024, from https://scale.com/about
3. Chai, J., Zeng, H., Li, A., & Ngai, E. W. T. (2021). Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications, 6*, 100134. https://doi.org/10.1016/j.mlwa.2021.100134
4. Yenduri, G. et. al. (2023). Generative Pre-trained Transformer: A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions. *Cornell University*. https://doi.org/10.48550/arXiv.2305.10435