DYNAMIC TAGGING: TRACKING ENERGY CONSUMPTION ON LARGE LANGUAGE MODELS

ACTOR-NETWORK THEORY ANALYSIS ON SONY'S HOME VIDEO RECORDER: BETAMAX

A Thesis Prospectus
In STS 4500
Presented to
The Faculty of the
School of Engineering and Applied Science
University of Virginia
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Science

By
Tu-Yen Dang

December 13, 2024

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

ADVISORS

Ben Laugelli, Department of Engineering and Society

Rosanne Vrugtman, Department of Computer Science

Introduction

As the world continues to advance, the pursuit for sustainability grows even more critical. In the technological sphere, the growing normalization of artificial intelligence (AI) usage has led to an increase in energy-intensive computational demands (Kemene et al., 2024). Achieving sustainability in this domain requires innovative approaches, including transparent and instantaneous tools that provide software developers opportunities to understand the energy impacts of their programs. With this in consideration, I propose a Dynamic Tagging System to track and display energy consumption of programs on a granular level with instantaneous results. Obtaining this information will allow software developers to gain valuable insights to make more sustainable decisions.

Creating meaningful change with technical designs must consider the interactions of technical innovations and societal factors. To best account for this intersection and ensure a successful product, I will perform a study on the downfall of Sony's Betamax: a home video recorder and tape. This analysis will apply the Actor-Network Theory (ANT) and look into the actor network that contributed to Betamax's failure in the consumer market. To best address the energy consumption issue at hand, it is important to understand the intersection between technical and social factors in order to reflect on previous mistakes and avoid failed systems. Because the challenge of creating more sustainable and energy-efficient software is sociotechnical in nature, it requires attending to both its technical and social aspects to accomplish successfully. In what follows, I set out two related research proposals: a technical project proposal for developing a dynamic energy tracking system for computationally powerful systems and an STS project proposal for examining consumer habits with competitive products when it came to the downfall of Sony's Betamax recorder.

Technical Project Proposal

Recently, there has been a growing search for faster, stronger, and more efficient computational power, especially with the rise of artificial intelligence (AI). In fact, the AI market size is projected to reach $407 billion by 2027, which is astronomical compared to the estimated value of $86.9 billion in 2022 (Haan, 2024). This growth however, is accompanied by a considerable increase in environmental costs. It was found that training a large-scale AI model could produce over 600,000 pounds of carbon dioxide – nearly five times the lifetime emissions of an average US manufactured car (Hao, 2019). Considering the environmental footprint of data-heavy software, there is a critical need for energy and emissions tracking to make informed, sustainable decisions in the software development process.

Though software energy tracking may be a fairly recent concept, products exist to provide information on this topic. One modern-day application relevant to today's software sustainability plans is a Python package called CodeCarbon. CodeCarbon can be used in a program to estimate the carbon emissions a program uses. It measures both local and cloud-based programs and can even split the program into different tasks so users can see which specific part of their code can produce the most emissions ("CodeCarbon," 2021). However, the package relies on data center estimations based on the region and hardware assumptions that may not be fully accurate, and only records carbon emissions. This disregards the main need of tracking accurate information energy usages to offer optimal solutions for users.

AWS's Customer Carbon Footprint Tool addresses some of codecarbon's limitations by measuring emissions, offering a user-friendly, detailed review of program emission rates, and even providing forecasting and future planning services ("Understanding the Customer Carbon Footprint Tool," 2024). Though this product is powerful and seems to have almost every key

point desired, it provides emission reviews aggregated over a monthly span. This may be considered a flaw for developers who prefer more immediate, real-time data. Instantaneous insights would be more effective for users querying AI models and tracking each interaction individually, rather than relying on periodic summaries.

This project aims to address gaps in existing tools by developing a dynamic monitoring system that provides real-time energy usage data and logging capabilities. However, it will still take into account the seamless usability and digestibility of prior products. The proposed method is a Dynamic Tagging system, which will track energy usage and emissions, allowing programmers to monitor workflow energy rates in real time. This system will tag and track each desired task on a program, and output its estimated rate of energy usage in a logging file. Additionally, algorithmic features will be included to suggest users more optimal paths and choices for companies to make more environmentally-conscious choices on where they are running their programs.

The tagging system will be developed using Python, a programming language known for its extensive capabilities in data processing, machine learning, and artificial intelligence—all of which require significant computational power (Yadoshchuk, 2024). The project will follow a Scrum agile framework to ensure continuous progress and meet development goals. To support this approach, a comprehensive requirements document will outline task distribution and expected deliverables. The software will be structured around a class-based system to effectively simulate object specifications of a program.

This system will collect data on each program's task run, and return the calculated energy usage rates for each of the following: CPU, GPU, RAM, and Storage based on the computing instance it is run on. Since this system's innate purpose is to log and store data, this information

will be logged in a file accessible to users. Once run on the code, users can access this information and tag it on specific instances, store it, or perform their own analyses on it to produce more summarized results. This data can also ensure the system's viability by verifying the values make common sense. If a certain instance is expected to use more computational power, it should output higher energy usage values. Overall, the data output will quantifiably validate these predictions.

STS Project Proposal

Sony is a company and global leader in technological products and services, specializing in electronics. In 1975, the company created Betamax: a take-home friendly video tape recorder. This was released only one year prior to the launch of the well-known VHS tape. They were extremely similar, and though Betamax was better in audio/video quality, it lacked in other aspects like cost and recording limit. While being one of the first releases in the market, Betamax lost popularity in the 1990s and fully shut down around 2002 due to lack of popularity when faced with more affordable and compatible products (Boucher, 2024).

The main reason people believed Betamax failed was because of competition against VHS tapes. Some believed that Betamax did not suit the needs of Sony's wide audience, since quality was not as necessary compared to affordability when it came to at-home use (Brodowicz, 2024). Others believed that it was due to Sony's unwillingness to license their product to other companies to make compatible electronic viewing products; this was something VHS tapes were more open to (Boucher, 2024). In general, the majority of people discussing this downfall agreed that it was due to the overall competition VHS tapes brought to the table. Another perspective was that the smaller limit in recording time on the Betamax compared to the VHS tape was what drove them to be unsuccessful (Burr, n.d.).

While these elements played a role in the unpopularity of this device, it does not portray the full extent of what led to the failure of this product and how it resulted in this outcome. A technical project cannot fail solely because of one individual component of the build. Rather, the entire system of interactions between these components should also be considered to determine what led to Betamax's demise. These current perspectives fail to consider the full system built around Betamax that made it a failing product, and only focus on how it compared to the VHS tape's widespread success at the time. This oversimplifies the study of a technological failure, which can lead to repeating mistakes in future products as well.

In order to have a broader understanding on everything that could have affected Betamax's failure, I will use an approach that considers the entire system and all the interactions that play roles in the technology. With this, I believe that a combination of consumer habits, inadequate design choices, an incompatible market, and timing of launch led to the downfall of Sony's Betamax recorder. While one may lead to Betamax's decline, the interaction of all these qualities is what ultimately caused the product to have such a short-term lifespan.

For this analysis, I will use a science, technology, and society (STS) framework called the Actor-Network theory (ANT). ANT was proposed by philosopher Bruno Latour and developed by many others including John Law. This concept believes all engineering projects are structured around a network. Built by a network builder, the network is formed by the interactions between heterogeneous actors, both human and non-human. One key concept in ANT is the idea of translation, which "bridges the gap between the varied aspects that are combined in technology" (Cressman, 9). This interpretation of relationships between actors can help determine which relationships were successful, and which were not in building Betamax's network. To find sources to apply the framework to the Betamax study, I will look into news articles, consumer

reviews, online forums (like Quora's thread on "Was Betamax really that much better than VHS? Why did it fail?"), and any Sony official releases relevant to the topic in order to gather as much information and as many perspectives on the network and actors as possible (Currens, 2023).

Conclusion

This combined approach integrates a technical project aiming to provide a solution to the sociotechnical problem of rising AI energy usages. The project proposes an innovative energy-tracking tool that outperforms existing tools like *CodeCarbon* and AWS's Customer Carbon Footprint Tool by offering more accurate real-time data to help users and companies make sustainable decisions. By logging and analyzing specific code instances, this system will enable users to measure their energy impact and make more sustainable choices like optimizing resources. With this, the STS project investigates the failure of Sony's Betamax product and applies the actor-network theory (ANT) framework to identify the broader network of factors including consumer behaviors, design decisions, and market competition. Performing this analysis will provide insight into what led this product to its downfall from a societal perspective. Understanding the causes of this failure can help develop the technical project in a way that will avoid Sony's mistakes that led to its downfall. By understanding these social and technical dynamics, the combined project aims to encourage sustainable technology adoption and create tools that are both impactful and relevant to users' needs.

References

Boucher, V. (2024, June 12). *Fun Facts About Betamax.* EverPresent. https://everpresent.com/fun-facts-about-betamax/.

Brodowicz, M. (2024, June 5). *Reasons why Sony Betamax Fails in the Market and How to Avoid it*. Aithor. https://aithor.com/essay-examples/reasons-why-sony-betamax-fails-in-the-market-and-how-to-avoid-it

Burr, S. (n.d.). *Betamax vs VHS: The Differences & History*. Legacybox. https://legacybox.com/blogs/analog/vhs-beat-betamax?srsltid=AfmBOoqWEq262AWz7G_dVpFKYrFnJozNCXlMKat90L7p0hxE3e8YKREZ

CodeCarbon: Track and Reduce CO2 Emissions from your Computing. (2021). CodeCarbon https://codecarbon.io/

Cressman, D. (2009). A Brief Overview of Actor-Network Theory: Punctualization, Heterogeneous Engineering & Translation. *ACT Lab/Centre for Policy Research on Science & Technology.* 1-17. https://summit.sfu.ca/item/13593

Currens, D. (2023). *Response to Was Betamax really that much better than VHS? Why did it fail?* Retrieved from Douglas Curren's answer to Was Betamax really that much better than VHS? Why did it fail? https://www.quora.com/Was-Betamax-really-that-much-better-than-VHS-Why-did-it-fail

Haan, K. (2024, October 16) *24 Top AI Statistics and Trends in 2024.* Forbes. https://www.forbes.com/advisor/business/ai-statistics/

Hao, K. (2019, June 6) . *Training a Single AI Model can Emit as Much Carbon as Five Cars in Their Lifetimes.* MIT Technology Review.

https://www.technologyreview.com/2019/06/06/239031/training-a-single-ai-model-can-emit-as-much-carbon-as-five-cars-in-their-lifetimes/

Kemene, E., Valkhof, B., & Tladi, T. (2024, July 22). *AI and energy: Will AI help reduce emissions or increase demand? Here's what to know*. World Economic Forum. https://www.weforum.org/stories/2024/07/generative-ai-energy-emissions/

Understanding the Customer Carbon Footprint Tool. (2024). Amazon Web Services. https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/ccft-overview.html

Yadoshchuk, V. (2024, April 29). *Why use python for AI and machine learning*. Waverley. https://waverleysoftware.com/blog/python-for-ai-and-ml/#:~:text=In%20general%2C%20Python%20is%20used,AI%2C%20according%20to%20Upwork's%20research.