Aurojit Ghosh

**Report: Week 2 Module 2 LLMs**

**Video Summary Number 1 Lab Session – 3D**

For Video Summary Number 1 Lab Session - 3D, some of the important points covered are

* Agents
* Platforms for use cases
* Unusable platform
* Demo deck
* Google calendar
* ZaubaCorp
* Creator Pulse, capstone project
* AI Readiness Score
* Social Media Agency
* Niche Agents
* Flipkart.com

  Agents are used for buying, direct messaging multiple people on LinkedIn, finding phone numbers, and utilizing them for marketplace research. Use Reddit for looking up new use cases, as well as Twitter (a somewhat usable platform). Agents can be used for VFX Artists and the Marketplace. The level begins with in-depth research and doesn't proceed through the cap chart. Whenever you log in, deep research ignores it. For Google Calendar, put a summary of what they have done, what they like, or what happened last week, or a problem statement. Research the people you're meeting with and provide three questions they can ask. For example, having a meeting with the smile group. In Google Calendar, summarize research on everything that they've done last week, what kinds of posts they're looking for, and problem statements. We can build niche agents by becoming domain experts. Over time, companies become bureaucratic. You have to picture yourself spending 3 weeks gaining domain knowledge and building niche solutions. To make a business, a wide range of developed skills needs to be developed. If you're good at numbers, numbers go up. For Zaubacorp, go there and find out how much money the company's making. Discover their subsidiaries and their financial performance. What is the founder's background, and provide a comprehensive summary of the company? As an advantage of deep research, you can bypass the chart and login pages. Use flipkart.com and think like a web page tester and find all issues. Build an AI readiness score and test it in an independent third-party setting, including prompt testing. Giving a scope out of 10 is necessary. Give an objective score on how to design a test or a rubric. The number of chatboxes and AI-first applications available for children is not currently available.

**Video Summary Number 2 Live Lecture 23 LLM Evaluation Techniques and Deployment Recording 1**

For Video Summary Number 2 Live Lecture 23 LLM Evaluation Techniques and Deployment Recording 1

Some key points are

* Process of fine tuning
* Evaluating LLMs
* Challenges with Evaluating LLMs
* LLM Evaluation is Subjective + Fuzzy
* Automated Metrics Fall Short
* Model Architecture

This video focused on the process of fine-tuning and evaluating large language models (LLMs), discussing the challenges involved, public benchmarks along with their pros and cons, model architecture, and the use of both automated and human evaluations. Fine-tuning LLMs requires careful evaluation because these models can be unpredictable. Even the best models may hallucinate facts, be overly vague or overly confident, and struggle with formatting. A few good outputs do not provide definitive proof of a model's effectiveness. In traditional machine learning, metrics such as accuracy, precision, and recall are clear indicators, but LLMs operate in a fuzzier domain. Some biases present in LLMs include a tendency to favor longer outputs, a propensity to agree with their previous answers, difficulty in justifying their scores, and challenges in handling style or nuance effectively. Automated evaluation of LLMs typically utilizes two main tools: auto-evaluators and bias warning systems. Human evaluations also have their advantages and disadvantages. The pros include capturing nuance, tone, and overall vibe. For challenges in evaluating LLMs, traditional ML is easier to measure. LLM Evaluation is subjective and fuzzy, Automated Metrics Fall Short, and human evaluation isn't Perfect Either.

**Video Summary Number 3 Live Lecture 23 LLM Evaluation Techniques Recording 2**

For Video Summary Number 3 Live Lecture 23 LLM Evaluation Techniques Recording 2

* Running evals
* Writing evals
* Push Transformers Model

In this video, users are shown how to run OpenAI evaluations. Examples discussed include .github, docs, evals, and exams. To run evaluations locally without contributing new ones, you can install the evals package using the command `pip install evals`. The video outlines a process for building an eval, which is documented in "build-eval.md". To write evals, walk through the process for building an eval build-eval.md, exploring an example of implementing custom and logic: custom\_eval.MD, writing your own completion functions completion\_fns.md, and reviewing our starter guide for writing evals: Getting Started with OpenAI evals. It also covers how to implement custom logic in evaluations through "custom\_eval.md", writing your own completion functions as detailed in "completion\_fns.md", and suggests reviewing the starter guide for writing evaluations available in "Getting Started with OpenAI Evals". Additionally, the video mentions that Transformers is an open-source Python library that provides a consistent interface for working with language models, such as FLAN, GPT-J, GPT Neo, LLAMA, and BLOOM, which are fine-tuned for specific tasks with small training data. This video covered the process of fine-tuning, evaluating LLMs, challenges in assessing LLMs, public benchmarks and their pros and cons, model architecture, automated evaluation using LLMs, and human evaluation, along with their pros and cons. For the process of fine-tuning, we use data and evaluation. We should evaluate LLMs because they are unpredictable and can produce unexpected results. Even top models can hallucinate facts, be overly vague or overly confident, or struggle with formatting. One or two good outputs don't prove anything. In traditional ML, we had accuracy, precision, and recall, but LLMs are fuzzy. Some of the biases that LLMs have are that they tend to prefer longer outputs, might agree with themselves, can't always justify their scores, and may not handle style or nuance well. Automated Evaluation using LLMs has two main tools: an auto-evaluator and a bias warning. Human Evaluation has its pros and cons. Pros are capturing nuance, tone, and vibe. Transformers is an open-source Python library that provides a consistent interface for language models. Transformers has nine steps: create a model, install cog, initialize your project, configure dependencies, customize your predictor, download weights, run your model, push your model, and use your model.