# SURVEYING 1 MILLION DIVERSE WILDCHAT CONVERSATIONS: IN THE WILD USE CASES OF CHATGPT

By Jason Leung and Sharan Giri

# ABOUT US

Jason Leung and Sharan Giri – both 1st year Master's of Data Science students interested in ChatGPT and Natural Language Processing (NLP) but with little to no experience in Big Data and especially implementing Machine Learning and NLP

# WHAT IS THE WILDCHAT DATASET?

| | Full Toxic Dataset | Non-toxic | Non-toxic + English Only |
|---|---|---|---|
| Total Turns of Conversations | 2+ million | 2+ million | 1+ million |
| Number of Conversations | 1+ million | 800,000+ | 500,000+ |
| File Size (in Gigabytes) | 5 GB | 3.3 GB | 2+ GB |
| OpenAI Moderation | Hate/violence/ sexual/self-harm/ obscene | (PI) Personal Identifying Info | |
| Core Data Fields | Location, Conversation, Toxic | Redacted (of PI), Primary Language | |
| Secondary Data Fields | 2023-2024 timestamp, Hashed_IP | Turns in conversation | |

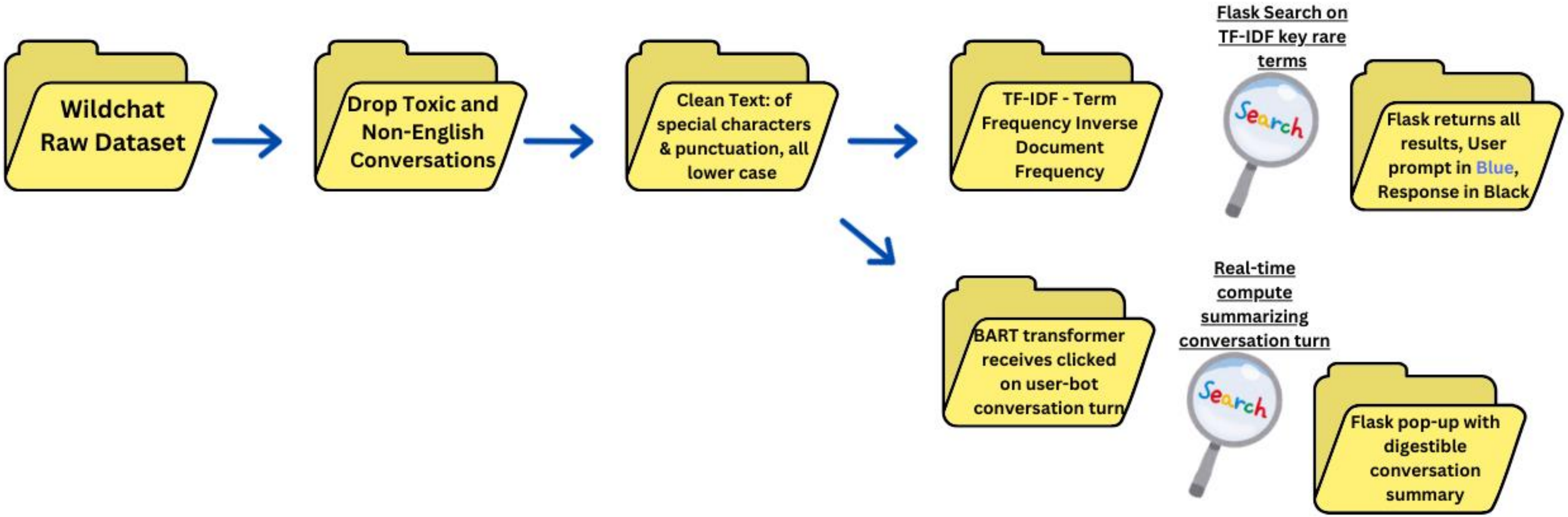
# PROBLEM

**The challenge of Big Data: How do we code and process Wildchat?**

**What is the most interesting but realistic and manageable project idea for this dataset and our two month timeline?**

- Scale Down from 1 Million: Focus on 2 types of conversation categories:
    - Journalism / Plagiarism
    - Programming / Coding
- Utilize PySpark's parallel processing and pre-process as much as possible especially NLP, to have a smooth and demo friendly UI
- Can't do everything: Jailbreaking is intriguing but too complicated. Most NLP will take longer than 2 months

# PROJECT STAGES - FLOWCHART



Wildchat Raw Dataset

Drop Toxic and Non-English Conversations

Clean Text: of special characters & punctuation, all lower case

TF-IDF - Term Frequency Inverse Document Frequency

Flask Search on TF-IDF key rare terms

Search

Flask returns all results, User prompt in Blue, Response in Black

BART transformer receives clicked on user-bot conversation turn

Real-time compute summarizing conversation turn

Search

Flask pop-up with digestible conversation summary

# JOURNALISM – FROM EMBELLISHING ARTICLES TO PLAGIARISM

-Reproducing some of the findings of paper "Case Studies of Generative AI's use in Journalism"

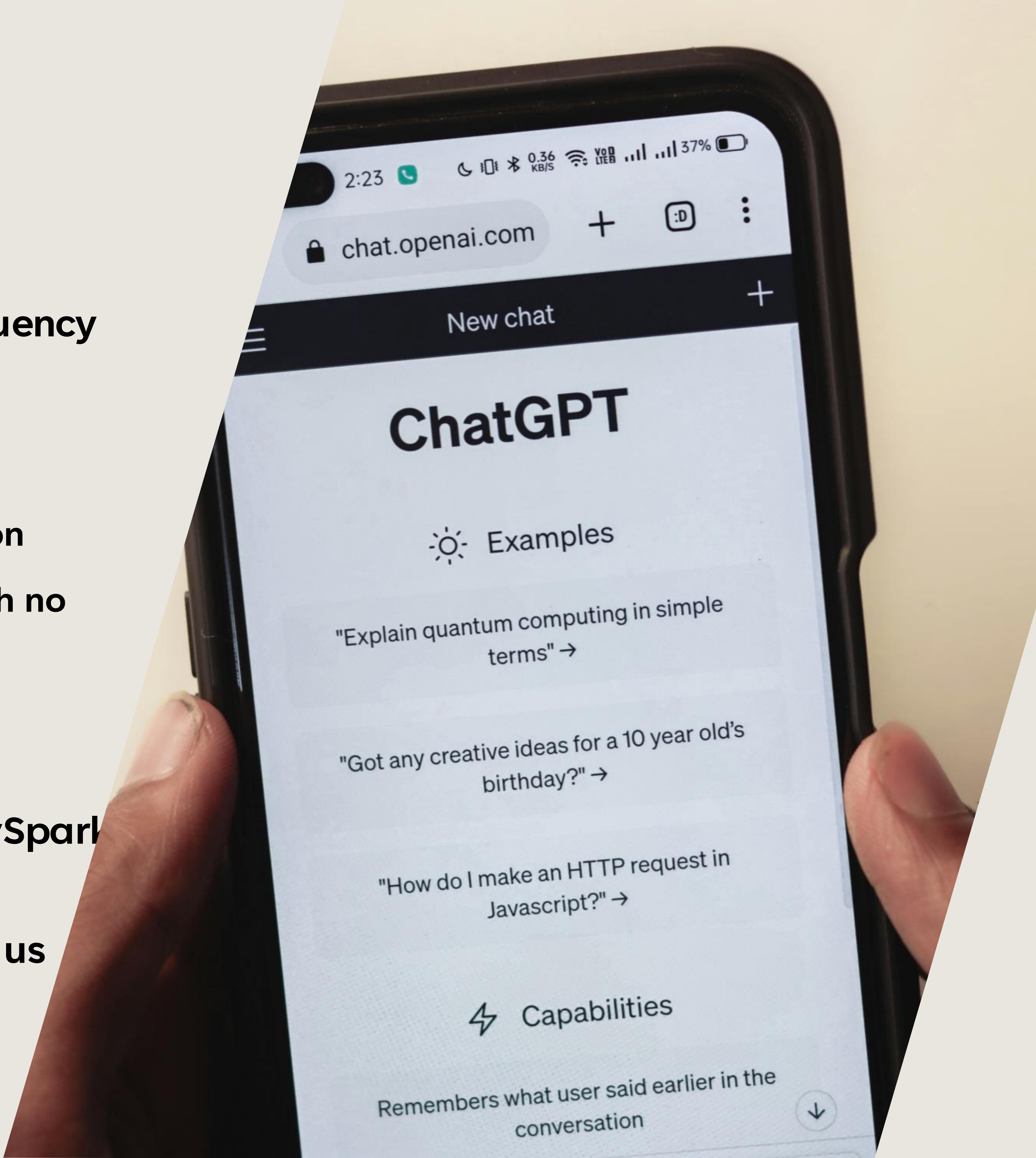
-Journalists use ChatGPT to find catchy headlines but also to auto-complete articles or even plagiarize articles from another news outlet

# CASE STUDY: MEDITERRANEAN NEWSPAPER TAKING THINGS TOO FAR?

- Important not to reveal info that could lead to the singular identification of the newspaper or worker
- Leave it at a smaller Mediterranean country, (probably) one person working at a news outlet used ChatGPT 20+ times over course of the year
  - For creating a catchy headline
  - Producing news articles from company press release
  - Creating articles from scratch using source material
  - Plagiarize articles from another news outlets and rephrasing to avoid detection

# DEMO: OUR FLASK UI AND KEYWORD SEARCH

- **Simple keyword search based on term frequency**
- **Search criteria includes:**
  - **Model version (ChatGPT 3.5 or 4.0 or both)**
  - **Country and State if have IP address location**
  - **Will only return non-toxic conversations with no personal identifying info**
- **We chose PySpark to handle the dataset's volume and real-time search querying**
- **We chose Flask as it integrates well with PySpark and we learned the basics in class.**
- **Try it out! – we want 2-3 volunteers to give us keywords they are interested in searching!**



"Explain quantum computing in simple terms" →

"Got any creative ideas for a 10 year old's birthday?" →

"How do I make an HTTP request in Javascript?" →

Remembers what user said earlier in the conversation

# PROGRAMMING / CODING CATEGORY

- Python, HTML, SQL, basically any programming language is found for ChatGPT

-Ask to fix syntax, complete code etc.

# WHAT IS EFFECTIVE USE OF CHATGPT FOR CODING? HOW DOES IT COMPARE TO DEEPSEEK?

-Example 1: Breadth-First Search (BFS), a fundamental search algorithm many of us learn about in CS5800

-Example 2 Prompt: What are the best algorithms to train neural networks?

# THANK YOU

- **Question and Answer session:**
- **Feel free to ask anything or email:**
- leung.jaso@northeastern.edu
- giri.sha@northeastern.edu
- https://github.com/sharan0276/WILDCHAT