CSC490 A2

Benson Li	1007815376
Yuchen Wang	1010441595
Xuanyi Lyu	1009343266
Zheyu Zheng	1008979580

October 2, 2025

Part One: Aspirational Datasets

1. **Human Text Corpus**
This will be a large, highly diverse collection of human writing, primarily focused on social media posts. While initially envisioned to include a variety of sources such as Twitter, Reddit, and Thread, the dataset will now be sourced exclusively from various social media platforms. This ensures a wide range of writing styles, tones, and topics, but within the context of online interactions. The goal is to capture a broad spectrum of social media discourse to improve model generalization across different online settings.

- **Desired Schema:**

  `text_id`: A unique ID for each text.

  `text_content`: The actual text.

  `source_type`: e.g., 'Tweet', 'Reddit', 'Thread'.

  `author_id`: An anonymous ID for the writer.

  `timestamp`: When it was written.

  `topic_tags`: Keywords like 'politics', 'tech', 'sports'.

  `label`: `human`.

2. **LLM Output Collection**
This dataset would serve as the AI counterpart to our human dataset. Because different LLMs exhibit distinct language styles and may reveal hidden patterns, it is valuable to include a diverse range of models, including proprietary LLMs (GPT-5, GPT-4, Claude, Cohere, Gemini, PaLM) and open-source LLMs (LLaMA, Falcon, Mistral, Mixtral). Using the same prompts and topics from the human dataset allows for direct comparisons. We would also vary generation settings, such as temperature, to capture a wide spectrum of outputs, ranging from "boring and robotic" to "creative and chaotic."

- **Desired Schema:**

  `text_id`: Unique ID.

  `text_content`: The generated text.

`source_prompt`: The prompt used to create the text.

`generating_model`: e.g., 'GPT-4-Turbo', 'Llama3-70b'.

`model_parameters`: The settings used, like {`"temperature": 0.8, "top_p": 1.0`}.

`label`: `ai`.

3. **The Adversarial Evasion Dataset**
This dataset would be filled with AI-generated text that has been deliberately tweaked to try and fool detectors. We'd include text that has been run through paraphrasing tools like QuillBot, text with subtle typos added, and maybe even text where sentences are rearranged to break the typical AI flow.

- **Desired Schema:**

`text_id`: Unique ID.

`original_ai_text`: The raw text from the LLM.

`modified_text`: The tweaked, evasive text.

`evasion_technique`: e.g., 'paraphrasing', 'style_transfer', 'typo_injection'.

`tool_used`: e.g., 'Quillbot', 'manual_edit'.

`label`: `ai_adversarial`.

4. **Fine-tuned LLM Output Collection**
The general LLM models may not be well-aligned with the specific domain of our project, which focuses on Twitter posts. Therefore, we will create a fine-tuned dataset using open-source LLMs (Mistral, LLaMA, Falcon, Deci-LM, T5, Pythia, BLOOM). We will leverage the following methods to fine-tune the models:

Instruction tuning: Instructions are composed using various metadata, such as topic hashtag, post engagement score (likes/retweets), user type, and post length. The responses are the corresponding Twitter posts.

One-topic held out: LLMs are fine-tuned on Twitter posts with one hashtag or topic held out. During generation, only posts for the held-out topic are produced, encouraging the model to generate novel posting styles for unseen topics.

Span-wise generation: Posts are generated one segment at a time (e.g., sentence or clause), conditioned on the rest of the post content. This helps the model maintain coherence while varying style and tone.

- **Desired Schema:**

`text_id`: Unique ID.

`text_content`: The generated text.

`source_prompt`: The prompt used to create the text.

`generating_model`: e.g., 'LLaMA', 'Falcon'.

`fine-tuning methods`: e.g., 'Instruction tuning','One-topic held out'.

`label`: `ai`.

5. **Slang and Informal Communication Corpus**
Twitter posts and similar platforms frequently contain non-standard language such as slang, acronyms,

memes, deliberate misspellings, and emoji-based expressions. This dataset would explicitly capture these linguistic patterns, ensuring the model can distinguish informal human quirks from AI outputs, which often struggle to imitate chaotic or creative slang.

- **Desired Schema:**

  `text_id`: Unique ID.

  `text_content`: The slang-heavy or informal text.

  `slang_type`: e.g., 'acronym', 'meme', 'intentional typo'.

  `language_mix`: e.g., 'Emoji substitution'.

  `label`: `human`.

Part Two: Reality Check

After looking at the datasets we actually found on Kaggle and other platforms, here's a rundown of what's available and how it fits into our project.

Dataset Name	Link	Description	Commentary
GenAI Tweets10k	Link	A dataset of over 10,000 AI-generated tweets produced by ChatGPT and Google Bard, spanning 21 diverse topics (about 500 tweets per topic) such as Climate Change, Election2020, and Bitcoin.	The dataset is well-structured, with balanced topical coverage and outputs from two major LLMs, which increases confidence in its representativeness of AI tweet styles. This makes it valuable for detecting stylistic and semantic markers of AI generation in diverse contexts. A limitation is that this dataset contains only AI-generated tweets.
TweepFake (Twitter Deepfake Text)	Link	A dataset comprising 25,572 tweets: 12,786 human-written and 12,786 machine-generated by 23 bots mimicking 17 human accounts. Bots utilized various generation techniques, including Markov Chains, RNN, RNN+Markov, LSTM, and GPT-2.	This dataset offers a balanced mix of human and machine-generated content, providing a robust foundation for training and evaluating detection models. Its use of multiple generation techniques enhances its relevance for identifying AI-generated tweets. A limitation is that it focuses on deepfake text and may not encompass all forms of AI-generated content [2].

Dataset Name	Link	Description	Commentary
MultiSocial: Multilingual Benchmark of Machine-Generated Text Detection	Link	A dataset comprising 472,097 texts across 22 languages and 5 social media platforms. Approximately 58,000 are human-written, with the remainder generated by 7 different multilingual LLMs.	This dataset addresses the gap in multilingual and cross-platform machine-generated text detection. Its diverse linguistic and platform coverage makes it invaluable for training robust, scalable models in real-world social media applications. For our project, we will process the dataset and use only tweets written in English. A limitation is that it may not encompass all forms of AI-generated content [4].
TweetEval Benchmark (Multiple Twitter Tasks)	Link	A benchmark dataset for 7 Twitter-related tasks: irony, hate speech, offensive language, stance, emoji usage, emotion, and sentiment. It provides standardized splits for each task, facilitating consistent comparisons.	This dataset is frequently referenced for fine-tuning models in social media tasks, especially related to sentiment and social sentiment analysis. It is particularly useful for learning and understanding informal, slang-heavy, and emoji-inclusive language used in tweets. A limitation is that it does not include AI-generated content.[1].
Sentiment140	Link	A widely-used dataset containing 1.6 million labeled tweets from 2009, classified as positive or negative.	Sentiment140 provides a large-scale, reliable baseline of human-written short-form social media text predating modern generative AI. Its widespread use in sentiment analysis research further supports its confidence. Given its age, it provides insight into human language prior to AI-driven text, making it a strong reference for 'human' behavior classification. For our project, we will sample from this dataset to obtain a manageable subset for training and evaluation. A limitation is that the dataset may contain outdated topics and language that do not fully reflect current social media usage [3].

Dataset Name	Link	Description	Commentary
LLM-based Synthesis	N/A (self-generate)	A custom dataset of Twitter posts synthesized using various state-of-the-art LLMs, including GPT-5, GPT-4, Claude, Cohere, Gemini, PaLM, LLaMA, Falcon, and Mistral. It spans a wide variety of topics and formats to provide diverse content for model evaluation.	This dataset is essential for testing detection models on modern AI-generated content in social media contexts. By including outputs from multiple proprietary and open-source LLMs, it captures a broad range of AI writing styles, improving model robustness. A limitation is that, as a self-generated dataset, it may not fully replicate all real-world human-AI interaction patterns or tweet styles.
Twitter API	Link to API	The official Twitter API allows real-time collection of tweets using filters such as keywords, hashtags, or user accounts. It supports large-scale, custom dataset creation for social media analysis.	Using the Twitter API ensures up-to-date, human-written content, making it ideal for constructing datasets that reflect current social media usage. It is widely adopted in social media research and for testing AI detection models in dynamic settings. A limitation is that rate limits and API access tiers may restrict the volume of data that can be collected in a given period.
Slangvolution	Link	A dataset containing tweets from 2010 and 2020, annotated with slang and non-slang words, designed to study the evolution of language on Twitter.	This dataset is valuable for analyzing linguistic trends and the adoption of slang over time, aiding in tasks like language modeling and sentiment analysis. A limitation is that it may not fully capture the context in which slang is used, potentially affecting the accuracy of analysis.

Table 1: Available Datasets for AI-Generated Text Detection

Part Three: Data-processing pipelines

Data Schemas

To handle the variety of our data sources, we have defined distinct schemas for each type of data ingested. These schemas will eventually be merged and transformed into a unified dataset for model training.

1. Main Dataset Schema

This schema represents the final, unified dataset that incorporates all text samples used to train and test the model.

Field	Description	Example
text_id	Unique identifier for each text entry	`001`
text	The text content (raw or cleaned)	"This is an essay example."
label	Classification: 1 = AI-generated, 0 = Human	`1`

2. Twitter Dataset Schema

This schema is designed for tweets scraped in real-time using the Twitter API, focusing on capturing relevant metadata.

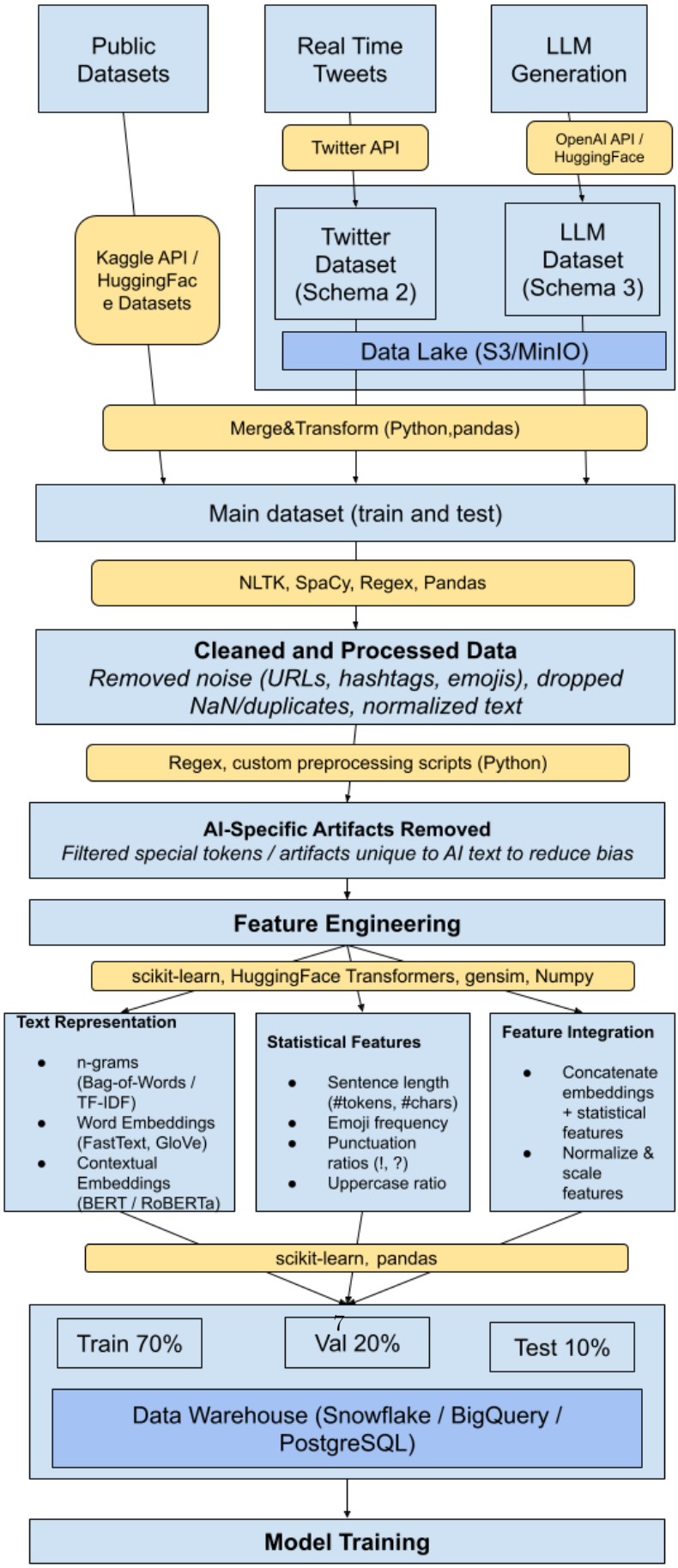
Field	Description	Example
text_id	Unique identifier for each tweet	`002`
text	Tweet content (raw or cleaned)	"AI writing is evolving fast."
label	Classification: 1 = AI-generated, 0 = Human	`0`
user_id	Unique identifier of the user	`1456789`
username	Twitter handle of the user	`@openai`
created_at	Timestamp of when the tweet was posted	`2025-09-30 18:00:00`
source	Data source / method of collection	`Twitter API v2`

Currently, we have generated several data points using the Twitter API. The code for data collection is available here, and the resulting dataset can be accessed here.

3. LLM Dataset Schema

This schema is tailored for content generated by different Large Language Models (LLMs), tracking the model and prompt used.

Field	Description	Example
text_id	Unique identifier for each entry	`003`
text	The generated text content	"AI writing is evolving fast."
label	Classification: 1 = AI-generated, 0 = Human	`1`
model	The backbone LLM used for generation	`gpt-4`
prompt_name	The prompt used to generate content	`persuade_prompt`


Public Datasets
Real Time Tweets
LLM Generation
Twitter API
OpenAI API / HuggingFace
Kaggle API / HuggingFac e Datasets
Twitter Dataset (Schema 2)
LLM Dataset (Schema 3)
Data Lake (S3/MinIO)
Merge&Transform (Python,pandas)
Main dataset (train and test)
NLTK, SpaCy, Regex, Pandas
Cleaned and Processed Data
Removed noise (URLs, hashtags, emojis), dropped NaN/duplicates, normalized text
Regex, custom preprocessing scripts (Python)
AI-Specific Artifacts Removed
Filtered special tokens / artifacts unique to AI text to reduce bias
Feature Engineering
scikit-learn, HuggingFace Transformers, gensim, Numpy
Text Representation
n-grams (Bag-of-Words / TF-IDF)
Word Embeddings (FastText, GloVe)
Contextual Embeddings (BERT / RoBERTa)
Statistical Features
Sentence length (#tokens, #chars)
Emoji frequency
Punctuation ratios (!, ?)
Uppercase ratio
Feature Integration
Concatenate embeddings + statistical features
Normalize & scale features
scikit-learn, pandas
Train 70%
Val 20%
Test 10%
Data Warehouse (Snowflake / BigQuery / PostgreSQL)
Model Training

Pipeline Stages

1. **Acquire Public Datasets:** Download relevant public datasets from sources like **Kaggle** and **Hugging Face** using their respective APIs.
2. **Collect Real-Time Tweets:** Use the **Twitter API** to gather real-time tweets from human users, which will serve as the human-generated text corpus.
3. **Generate Synthetic Tweets:** Utilize the **OpenAI API** or locally-hosted **Hugging Face** models (LLMs) to generate AI-based text.
4. **Store in Data Lake:** Format the data from different sources (real-time tweets, LLM-generated content) according to predefined schemas and store them in a **Data Lake** (e.g., **S3** or **MinIO**).
5. **Merge & Transform:** Using **Python** and **pandas**, merge the public datasets, tweet data, and LLM data from the Data Lake to create a unified Main Dataset.
6. **Data Cleaning & Preprocessing:** Perform in-depth cleaning on the main dataset using tools like **NLTK, SpaCy, Regex, and pandas**:
   - **Remove Noise:** Delete irrelevant information such as URLs, hashtags, and emojis.
   - **Handle Missing Values & Duplicates:** Drop NaN values and duplicate records.
   - **Normalize Text:** Standardize text (e.g., convert to lowercase).
   - **Tokenization:** Split cleaned text into individual tokens/words using tools like **NLTK, SpaCy, or Hugging Face Tokenizers**. This step ensures proper input for n-grams, embeddings, and contextual models.
7. **Remove AI-Specific Artifacts:** Filter and remove special characters or patterns that appear exclusively in AI-generated text to reduce model bias.
8. **Feature Engineering:**
   - **Text Representation:**
     - Traditional methods: n-grams (Bag-of-Words, TF-IDF).
     - Word embeddings: **FastText**, **GloVe**.
     - Contextual embeddings: **BERT**, **RoBERTa**.
   - **Extract Statistical Features:** e.g., text length, emoji frequency, punctuation ratio, uppercase ratio.
   - **Feature Integration:** Concatenate embeddings + statistical features, then normalize and scale.
9. **Split the Dataset:** Divide into **70% train, 20% validation, 10% test**.
10. **Load into Data Warehouse:** Store the final split datasets in a **Data Warehouse** such as **Snowflake**, **BigQuery**, or **PostgreSQL**.
11. **Model Training** Use the training and validation sets stored in the Data Warehouse to train and fine-tune machine learning or deep learning models.

Pipeline Use Cases

- **Public Datasets:** Downloaded and processed only once at the beginning of training, since these datasets are static and do not change over time.
- **Real-Time Tweets:** Scraped every 24 hours to capture fast-evolving internet trends and keep the model up to date.
- **LLM Generation:** Executed every month, or sooner if there are major updates to LLM models or the release of specific new features.

Initial Pipeline Code

The initial implementation of the pipeline is available in the GitHub repository: https://github.com/UofT-CSC490-F2025/TweetVerify

Next Steps

1. Currently, the project uses a local data lake and data warehouse. Migration to cloud-based solutions is needed to enable scalability and facilitate model training.
2. The LLM-based data synthesis is not yet complete. It is necessary to integrate APIs for online models and deploy local open-source models to generate and gather synthetic data.

References

[1] Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *Findings of EMNLP 2020*, 2020.

[2] Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. Tweepfake: about detecting deepfake tweets. *arXiv preprint arXiv:2008.00036*, 2020.

[3] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. In *CS224N Project Report, Stanford*, 2009.

[4] Dominik Macko, Jakub Kopal, Robert Moro, and Ivan Srba. Multisocial: Multilingual benchmark of machine-generated text detection of social-media texts. *arXiv preprint arXiv:2406.12549*, 2024.