**Leveraging LLMs for data augmentation on sparse label problems**

**Objective**

Facebook Integrity uses ML models to help identify and remove content that violates its community standards, such as hate speech, graphic violence, and sexual exploitation. These models are trained on large datasets of examples of bad content and are able to recognize patterns and features that are associated with such content. This spans a variety of policies, including spam messages, fake accounts, and fraudulent activity.

However, these ground-truth datasets are gathered through human review, which is a slow and costly process. To accurately identify violations, human reviewers must have expertise in the relevant area. This expertise can be difficult to find and retain, and the cost of hiring and training qualified reviewers can be high. Additionally, with 3 billion monthly active users, Facebook's user base is massive. If even only 0.1% of users created a violating post each month, that is still 3 million violations to identify and address. Obviously, we are not able to manually review every post that may pose a violation risk.

Therefore, one of the biggest challenges within Integrity is finding high enough label volume to train a reliable model. With the improvements across LLMs, we can generate sample data at high volumes to train models to tackle very sparse problems.

**What is data augmentation?**

Data augmentation is a technique to generate new data that is similar in style and structure to the training data. This can be particularly useful for sparse problems, where there is a limited amount of training data available. In such cases, data augmentation can help to increase the size of the dataset, which can improve the performance of the model by reducing overfitting and improving the model's ability to generalize. Here, we propose using LLMs for data augmentation - however, performance is dependent on how well the LLM can generate high-quality text that is similar to the existing data. This relies on several independent factors including LLM quality and prompt design.

**Data**

1. Unfortunately, Llama2 will not easily directly generate samples of many Facebook violations, including hate speech, sexual exploitation, etc. Instead, we can proxy a sparse label problem using disaster data. This is both a binary and multi-target problem across a variety of potential outcomes.

Disaster Response Messages

* <https://huggingface.co/datasets/disaster_response_messages>

1. However, there are existing LLM workarounds to generating hate speech labels, and I have attached a dataset with labels in case you’d like to explore. This will need to be joined with general comment data ([example](https://www.kaggle.com/datasets/smagnan/1-million-reddit-comments-from-40-subreddits)) to proxy a sparse dataset.If you are able to find workarounds to generate hate comments - start with this dataset first.

Social Media Hateful & Toxic Comments

* <https://socialmediaarchive.org/record/19?ln=en>

**Understanding Questions**

* What is the general data overview? Most/least common labels, initial patterns, etc.
* How did you approach data transformation? Did you use embeddings or find alternatives? Why?
* What methodology for data augmentation works best? Pros/cons of particular prompt formats? What filters need to be added and/or clarified?
* How does data augmentation benefit the performance of your algorithms?
* What are the specific pros and cons of utilizing LLMs to augment data? How can this affect model performance - in the short and long term?
* Compare performance of your model against just asking the LLM to predict the label on a given piece of text. What is the performance/pros/cons, etc?
  + This space is transforming fast. OpenAI has new research [published here](https://openai.com/blog/using-gpt-4-for-content-moderation).
* If working with the hateful comment dataset - how does your model compare against SOTA models that detect hate speech/toxicity? E.g. <https://huggingface.co/facebook/roberta-hate-speech-dynabench-r4-target>

**Deliverables**

1. Notebook (Jupyter/Google Colab/Other) with all relevant code, including but not limited to all aspects of:
   1. Data Cleaning and Transformation
   2. Model Training
   3. Model Performance
2. Written processes (code and/or text) of data augmentation process, along with specific learnings that apply to augmenting data for this particular sparse use-case
3. Visualizations comparing model performance and outlining the research and development process

**Methodology**

1. Download, clean, and understand the data
2. Generate augmented data for one/many target labels
   1. Request access to Llama2 on [HuggingFace](https://huggingface.co/meta-llama/Llama-2-7b) and [MetaAI](https://ai.meta.com/llama/) and download
      1. Recommend the lightest 7B model, which should be runnable on Google Colab. Alternatives include Vicuna/other models that are not ethnic-locked and can generate hate speech.
      2. If possible, load the model locally or through Colab and write a script that can scalably generate labels for any given topic
   2. If Llama2/alternatives are infeasible due to compute constraints, students should find a lightweight LLM and explore scalable prompt-to-text scripts
   3. Otherwise, generate data manually through online sandboxes
3. Train several models, across the range of dependencies using and not using augmented data
4. Assess and compare model performance for both binary and multi-label tasks

**Success Criterion**

This project consists of two main technical tasks - 1) engineering the correct prompts to generate reliable data, and 2) developing and analyzing effective NLP models

* [Prompt Engineering] Provide a systematic approach to reliably generate augmented data for sparse label problems based on exploration and research on real-world textual data
* [ML Development] Deliver recommendations for training text classification models and propose robust methods to evaluate algorithm performance
* Demonstrate abilities in writing scalable code by wrapping up functions in a callable objects and annotating and formatting code blocks
* Craft a story that draws insights from data and models that illustrates industry value and effectiveness, complete with clean and digestible visual graphics
* Stretch Goal: Perform technically rigorous projects and run an LLM through local/cloud-based methods for more scalable data augmentation

**Student Learnings**

* Working with real-world data
* Using data and results to tell a story
* Automate evaluation of algorithms
* Gaps and benefits of leveraging LLMs
* Challenges of working with sparse datasets

**Sponsor Benefits**

Facebook is a leader in the LLM space and is committed to developing improved algorithms that can be directly applied to benefit cross-functional teams across the company. We want to evaluate the benefits and limitations of LLM application especially for human review use cases. These projects are crucial to make sure we continue to keep Facebook a safe and transparent place for people to form connections.

**References**

1. Local LLM performance comparisons <https://github.com/Troyanovsky/Local-LLM-Comparison-Colab-UI>
2. HuggingFace Llama2 Demo <https://huggingface.co/blog/llama2>