[CSCI 1470 Deep Learning Final Project - HackMD]

Intermediate Project Report

Due: November 21, 10pm

DO NOT USE ANY AI TOOLS FOR WRITING. IT MUST BE YOUR OWN WORK. WE WILL PUT IT THROUGH TURNITIN AND GPTZERO!

Title: Summarizes the main idea of your project.

Gemedi: Realistic Medical Data Generation

Who: Names and logins of all your group members.

Yumian Cui, ycui39
Yongcheng Shi, yshi119
Zejun Zhou, zzhou190
Ruoqian Zhang,rzhan221

Introduction: What problem are you trying to solve and why?

- If you are implementing an existing paper, describe the paper's objectives and why you chose this paper.
- If you are doing something new, detail how you arrived at this topic and what motivated you.
- What kind of problem is this? Classification? Regression? Structured prediction? Reinforcement Learning? Unsupervised Learning? Etc.

We chose this topic because real medical notes containing PHI are extremely restricted in today's data community. However, real PHI patterns are essential for training practical clinical NLP systems, especially for medical chatbot applications. Existing de-identified datasets cannot capture realistic PHI such as names, dates, unit number, and other PHI structures. To safely approximate real-world data, we want to generate synthetic notes that include PHI but contain no real patient information.

Our problem is not a simple classification or regression task. We use a generator to create synthetic PHI-included notes, and a discriminator to score them on two dimensions, which are realism and PHI difficulty. These scores are then fed back into the system as a kind of reinforcement-style signal. They help us select good examples, fine-tune the models, and refine prompts. Conceptually, it is a hybrid generative modeling with adversarial and reinforcement learning inspired optimization problems aimed at shaping a realistic distribution of PHI-rich notes.

Related Work: Are you aware of any, or is there any prior work that you drew on to do your project?

- Please read and briefly summarize (no more than one paragraph) at least one paper/article/blog relevant to your topic beyond the paper you are re-implementing/novel idea you are researching.
- In this section, also include URLs to any public implementations you find of the paper you're trying to implement. Please keep this as a "living list"–if you stumble across a new implementation later down the line, add it to this list.

Because our project aims to do something new, there are no papers that perfectly match our overall project architecture. However, there is one paper on prompt engineering for text generation that provided inspiration on how to generate structured text containing protected health information (PHI) with a fixed level of difficulty.

The CTRL: A Conditional Transformer Language Model for Controllable Generation（Keskar et al., 2019）paper proposes a large-scale Transformer model that achieves controllable text generation through "control codes." The core idea is to add control tags representing style, topic, or task to the training data, allowing the model to learn language distribution while simultaneously learning to generate text with specific styles and structures based on these control codes. This method is entirely based on supervised learning, it achieves highly controllable generation results through a simple prefix design, laying the foundation for subsequent research in controllable text generation.

Reference:

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., & Socher, R. (2019). *CTRL: A conditional transformer language model for controllable generation*. arXiv. https://doi.org/10.48550/arXiv.1909.05858

Data: What data are you using (if any)?

- If you're using a standard dataset (e.g. MNIST), you can just mention that briefly. Otherwise, say something more about where your data come from (especially if there's anything interesting about how you will gather it).
- How big is it? Will you need to do significant preprocessing?

Our approach is to expose the discriminator to both high-quality and low-quality medical notes at different difficulty levels. By learning what makes notes good versus bad, the discriminator can provide better feedback, which helps the generator produce more realistic outputs that match real-world data.

We initially planned to use the MIMIC-IV dataset from PhysioNet, which contains real clinical notes with PHIs. But this would require each team member to obtain PhysioNet credentials and complete training, which is infeasible given our timeline. Thus, we switched to a 1000 de-identified MIMIC texts from Kaggle instead. Since the Kaggle data is de-identified, we needed to fill PHI fields with realistic, synthetic values by LLMs. We also truncated texts to only focus on discharge summaries to reduce the foreseeable computational demands. For PHI generation, we split the work by using Llama 3.1 (via Oscar) 50% of time and using Faker (a Python library for synthetic data) 50% of time. For negative examples, we reused the same templates but intentionally inserted contradictory or incorrect information. We're starting with an 80/20 split of good to bad data, adjustable based on our training performance.

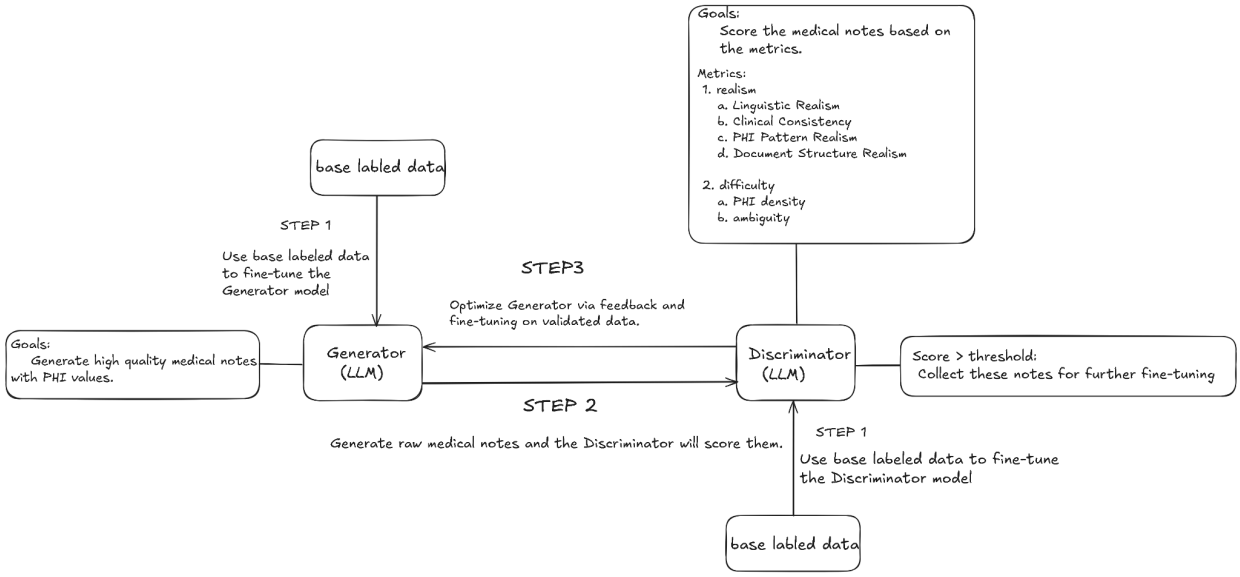
We also discussed and developed scoring metrics to evaluate our generated data, both as labels for training data and judging metrics for the discriminator . The realism score measures how authentic the notes appear, weighted as follows: 35% linguistic realism, 40% clinical consistency, 15% PHI pattern realism, and 10% document structure realism. The difficulty score measures generation complexity, combining 70% PHI density (frequency, how crowded is this one note with PHI?) and 30% ambiguity (how many PHIs it contains that could be interpreted in more than one way.)

Methodology: What is the architecture of your model?

- How are you training the model?

  We fine-tuned the Llama-3.1-8B-Instruct model using Supervised Fine-Tuning on the OSCAR cluster. And we used QLoRA . The base model was loaded in 4-bit precision to minimize footprint, and low-rank adapters were injected into all linear layers for parameter-efficient training.

- If you are doing something new, justify your design. Also note some backup ideas you may have to experiment with if you run into issues.



1. Overview:

We design a closed-loop generation framework which aims to generate high-quality synthetic medical notes containing Protected Health Information. This framework contains two Large Language Models(LLMs) Generator and Discriminator.The Generator is designed to generate high-quality synthetic medical notes and the Discriminator is designed to score the quality of the notes which Generator genetated.

The process can be divided into three steps:

2. Step1: Discriminator Initialization

This step aims to make the discriminator score the notes according to two dimensions: realism and difficulty. Realism consists of Linguistics Realism, Clinical Consistency, PHI Pattern Realism and Document Structure Realism and difficulty consists of PHI density and ambiguity.

Discriminator Fine-Tuning : The Discriminator model is fine-tuned using a dataset of "base labeled data." This dataset consists of verified medical notes with accurate labels. These labels contain the score of Realism and Difficulty.

Generator Initialization: The Generator model is fine-tuned using a dataset of "base labeled data". This dataset consists of high quality data and specific instruction.

3. Step 2: Generation and Evaluation

The Generator: The Generator LLM will produce "raw medical notes" with specific PHI values. Its goal is to simulate realistic medical notes.

The Discriminator: The generated notes will pass to the Discriminator for scoring. And the discriminator will score the notes based on the two primary metrics:

- **A. Realism Metrics:**

    - **Linguistic & PHI Pattern Realism:**

      Names must consist of at least two words (First and Last name); single-letter names are not allowed.Names must be alphabetic. "Strange characters", symbols, or non-standard punctuation are not permitted. Proper nouns must follow the standard capitalization rules. Physician names must be explicitly preceded by the honorific prefix "Dr.". All dates must follow the `Year/Month/Day` format. Generated dates cannot be in the future relative to the current system date. Dates cannot exceed a period of 105 years.

    - **Clinical Consistency:** The admission data must precede or be equal to the *Discharge Date*. Ensures relationship between patient Age/Gender and the illness.

    - **Document Structure Realism:** The first four lines of the document must match the pre-defined format. Medical keywords and section headers must be present in the generated note.

- **B. Difficulty Metrics:**

    - **PHI Density:** Measures the volume of sensitive entities (names, dates, locations) per document.

    - **Ambiguity:** The List of 100 Ambiguous words.

4. Step 3: Iterative Optimization (Step 3)

   Fine-Tuning on Validated Data:
   Notes that successfully pass the Discriminator's evaluation (where Score > Threshold) are collected into a "High-Quality" dataset. And then fine-tune the Generator by these "High-Quality" datasets to make the generator generate high quality notes.

Metrics: What constitutes "success?"

- What experiments do you plan to run?
- For most of our assignments, we have looked at the accuracy of the model. Does the notion of "accuracy" apply for your project, or is some other metric more appropriate?
- If you are implementing an existing project, detail what the authors of that paper were hoping to find and how they quantified the results of their model.
- If you are doing something new, explain how you will assess your model's performance.
- What are your base, target, and stretch goals?

Since our project is not a regular supervised learning task, accuracy seemed not really make sense for evaluating success. Instead, we care about whether the generator can produce high-quality medical notes that look realistic while also following a natural range of PHI difficulty.

To measure this, we will run experiments where the generator produces batches of 100, 500, and 1000 samples. We will first use our local rule-based scripts, which are the same ones used to label our initial training data to score each batch on realism and PHI-difficulty. For realism, we want to find out if the text looks like a real clinical note. For PHI-difficulty, we want to check if the generated PHI patterns follow the natural range of PHI difficulty.

Success means the average realism score goes up after training, and the PHI difficulty distribution becomes closer to a near-normal shape rather than being dominated by very easy cases.

To double check our results, we will also use an LLM-as-judge to rate a subset of samples. If both the local script testing and the LLM show similar improvements, we know the model is genuinely getting better.

Ethics: Choose 2 of the following bullet points to discuss; not all questions will be relevant to all projects so try to pick questions where there's interesting engagement with your project. (Remember that there's not necessarily an ethical/unethical binary; rather, we want to encourage you to think critically about your problem setup.)

- What broader societal issues are relevant to your chosen problem space?
- Why is Deep Learning a good approach to this problem?

- What is your dataset? Are there any concerns about how it was collected, or labeled? Is it representative? What kind of underlying historical or societal biases might it contain?
- Who are the major "stakeholders" in this problem, and what are the consequences of mistakes made by your algorithm?
- How are you planning to quantify or measure error or success? What implications does your quantification have?
- Add your own: if there is an issue about your algorithm you would like to discuss or explain further, feel free to do so.

What broader societal issues are relevant to your chosen problem space?

There is a big shortage of high-quality medical data containing PHIs available for model training. Because such data cannot be freely shared due to privacy and ethical restrictions, many medical NLP systems are trained only on de-identified text, which limits their ability to understand and handle PHI-related content in real-world clinical documentation. Through our project, the goal is to train a generator that generates realistic synthetic data to be used for model training. We hope this work demonstrates a viable approach for generating synthetic medical notes, which could help address data scarcity challenges in medical NLP research.

What is your dataset? Are there any concerns about how it was collected, or labeled? Is it representative? What kind of underlying historical or societal biases might it contain?

We used MIMIC data from Beth Israel Deaconess Medical Center (Boston, 2008-2022), which has some limitations. First, it only covers ER and ICU cases from Boston, so it may not represent general healthcare settings across the US. Second, since we used a de-identified version, the original de-identification process may be imperfect. Third, our manual PHI injection partially relying on LLMs could introduce biases or accidentally generate information resembling real individuals, since LLMs are trained on real-world data.

Division of labor: Briefly outline who will be responsible for which part(s) of the project.

Everyone participates in ideation, project discussions, and report write-ups.

Yongcheng Shi: Data preprocessing and regression testing

Yumian Cui: Data preprocessing, prompt refinement, regression testing

Zejun Zhou: Creating synthetic data labels, fine-tuning both generator and discriminator, optimizing prompt engineering.

Ruoqian Zhang: Fine-tuning both Generator and Discriminator models and optimizing prompt enginering.