CS685 Quiz 3: ***attacking AI-generated text detectors***
Released 4/21, due 4/28 on Gradescope (please upload a PDF!)
Please answer each question in 2-4 sentences.

1. Let's say OpenAI releases a plagiarism detection service to verified teachers that allows them to check whether or not a student's submission was generated by ChatGPT. The service compares a candidate submission to every piece of text that ChatGPT has ever generated, and returns a **1** if the submission very closely resembles one of the generations and a **0** otherwise. Now assume one of the teachers is malicious and wants to abuse this service to perform membership inference attacks [*read/skim the paper to understand the idea behind these attacks!*]. First, explain how this teacher might set up their attacks, and what kind of information they might be able to access.

Answer: A membership inference attack is a type of privacy attack where an adversary, given a model's output, attempts to infer whether a specific instance was part of the training set. A teacher trying to set up a membership inference attack on ChatGPT (Target Model, aka model whose information we are trying to access) might proceed as follows:

1. **Create Shadow Models**: The teacher would first need to create several shadow models that mimic the behavior of ChatGPT.Large Language models like BERT, GPT-3, GPT-4, Bard, BLOOM, and LLaMA can be examples of shadow models in our case. These models would ideally be trained on datasets the teacher suspects were part of ChatGPT's training data. The key purpose of shadow models is to learn the target model's behavior on its training data versus its behavior on non-training data.
2. **Generate Outputs and Careful Inference**: The teacher would then use both training and non-training data to generate outputs from the shadow models. A malicious teacher could potentially conduct a membership inference attack by querying the service with carefully crafted inputs, attempting to deduce whether specific content has been generated by the model in the past. They might try to submit known outputs from the model and then compare the responses to other queries. They might also submit very unique or specific text snippets in an attempt to see if the model has generated that exact text before. It is also possible to give variations of paraphrased inputs generated by using the DIPPER Model (Paraphraser Model) with BM25 Retriever on original inputs.
3. **Train Attack Models**: Using these outputs of the shadow model, the teacher would train an attack model (a collection of **n** models, where n is the number of output classes of the target model) to distinguish between outputs generated from training versus non-training data. Essentially, these models learn to predict whether a given output from the target model is likely to have come from a training or non-training input. Note: Here, we can restrict n = 10, where the 10 class labels can be (English, Maths, Biology, Physics, Chemistry, Computer Science, Geography, History, Environment Science, Commerce). The inputs for the attack models are the confidence levels, along with the label in or out. *(in: part of training data and out: not part of training data).* Teachers use the confidence levels given by the shadow models to discriminate *in* samples from *out* samples. It gives

instances from which they can learn the differences in confidence between *in* and *out* samples, to infer the membership of a sample on the target model. As teachers control which samples are used in the training of a shadow model and which are not, they are able to generate instances of confidence levels belonging to the *in* group and others belonging to the *out* group. For each class of the target problem, an attack model is trained that learns to discriminate *in* from *out* samples. In *this* problem, the attacker uses ten attack models. The samples produced by the shadow models are dispatched to build different attack datasets regarding their real class/label. When this process is finished, the attack datasets are prepared, and all that remains is to train the attack models. These are simple discriminators (often multilayer perceptrons).

4. **Test on Target Model**: Finally, the teacher would use the attack model to analyze outputs from the target model (ChatGPT) and make inferences about whether certain inputs were part of ChatGPT's training data.

However, the practical feasibility of such an attack on models like GPT-3 and GPT-4 is low because models like these (GPT-3) do not retain specific inputs or outputs, and they are trained on such a vast and diverse range of data that distinguishing training from non-training data would be challenging and difficult.

As for the kind of information teachers might access, successful membership inference attacks could, in theory, provide some insight into the types of data or sources that were used for training. However, in this scenario, they would not reveal personal/sensitive/individual data about specific individuals or the exact documents in the training set. Given the nature of GPT models, a membership inference attack is likely to yield results like the **core concept/summarisation of a methodology that may be private data but does not reveal the personal data in it.** For example, "Stating diseases having a cold, nausea, headache, vomit symptoms" yet not recommending a personalized suggestion/diagnosis to a user. (Diplomatic tone of generation of words) Another example summarising the methodology of a research paper that is not publically accessible. These are tested examples of ChatGPT.

Another interesting take is that an adversary with typical API access to a Large Language Model like GPT-3 can steal the type and hyperparameters of its decoding algorithms at very low monetary costs as demonstrated in **Stealing the Decoding Algorithms of Language Models** paper.

References:

1. https://medium.com/disaitek/demystifying-the-membership-inference-attack-e33e510a0c39#:~:text=The%20Membership%20Inference%20Attack%20is,trained%20ML%20model%20or%20not.
2. https://arxiv.org/pdf/1610.05820.pdf
3. http://128.84.21.203/pdf/2303.04729%201

2. **What are some counter-measures OpenAI can take to prevent the malicious teacher's attacks from succeeding?**

Answer: OpenAI could implement various measures to mitigate the risk of membership inference attacks on such a service. Some potential countermeasures are:

1. **Rate Limiting**: Impose a limit on the number of queries a user (here, teacher) can make within a specific time frame which can prevent them from making a large number of queries in a short period of time, which could potentially be used to gather more information than intended.
2. **Data Anonymization:** Redacting Private Information is a strong countermeasure against membership inference attacks, especially in scenarios involving training data. However, in the scenario suggested, where a service checks whether a submission closely resembles a text previously generated by the ChatGPT, the role of redaction might be less direct. In a system where data is stored and used for checks, redacting private information before storing it can prevent any accidental leakage of personally identifiable information (PII). This means even if someone tries to misuse the system, they cannot retrieve sensitive information because it has been removed beforehand. Redaction can be implemented in various ways:
    - **Manual Redaction**: This is labor-intensive and time-consuming but may be necessary for particularly sensitive datasets.
    - **Automated Redaction**: Some tools can automatically detect and redact PII. These use machine learning or rule-based systems to identify and mask names, addresses, phone numbers, and other sensitive information.
    - **Differential Privacy**: By adding noise to the results, OpenAI could prevent an attacker (teacher) from gaining precise/personal/sensitive/individual information while allowing for useful analysis. Differential privacy is a mathematical framework that aims to get valuable results from queries on a dataset while minimizing the risk of exposing individual data points.

    In the case of an AI like ChatGPT, the training data is a mix of licensed data, data created by human trainers, and publicly available data. However, specific documents are not remembered or stored by the model. Hence the model doesn't know specifics about which documents were in its training set, making redaction less relevant to the AI's output. However, in the case of the described plagiarism detection service, it would be crucial to ensure that any data stored for the purpose of this service is thoroughly anonymized and redacted of any potentially sensitive information to prevent any chance of misuse.

3. **Result Generalization**: Instead of returning a binary 0 or 1 for whether the submission closely matches a past generation, OpenAI could yield more generalized statements (e.g., "The submission appears original" or "The submission may contain elements

common in AI-generated text") which would provide qualitative information for the intended purpose (detecting plagiarism) without revealing precise matches.

4. **Model Generalization**: Overfitting can make models more susceptible to membership inference attacks since models that overfit to their training data behave more distinctly on that data. Using techniques to improve model generalization, such as regularization, dropout, early stopping, or data augmentation, can help protect against these attacks.
5. **Restrict the prediction vector to top k classes:** When the number of classes is large, many classes may have very small probabilities in the model's prediction vector. However, the model will still be helpful if it only outputs the probabilities of the most likely k classes. (Add a filter to the last layer of the model). The smaller k is the less information the model leaks. In the extreme case, the model returns only the label of the most likely class without reporting its probability.
6. **Coarsen precision of the prediction vector:** To implement this, we round the classification probabilities in the prediction vector down to d floating point digits. The smaller d is, the less information the model leaks.
7. **Increase entropy of the prediction vector:** One of the signals that membership inference exploits are the difference between the prediction entropy of the target model on its training inputs versus other inputs. As a mitigation technique for neural network models, we can modify (or add) the softmax layer and increase its normalizing temperature t > 0. This technique would increase the entropy of the prediction vector. Note that the output becomes almost uniform and independent of its input for a very large temperature, thus leaking no information.
8. **Give detector access to verified users only:** Since the detector is public, attackers can iteratively improve their perturbation model, hence use verification.
9. **Retrieve over training data as well:** Detect False positives due to training data memorization

References:

1. https://people.cs.umass.edu/~miyyer/cs685/slides/security.pdf Slide 51
2. https://arxiv.org/pdf/1610.05820.pdf Section Mitigation Strategies

AI Disclosure

AI1: Did you use any AI assistance to complete this homework? If so, please also specify what AI you used.
Yes

(only complete the below questions if you answered yes above)

AI2: If you used a large language model to assist you, please paste *all* of the prompts that you used below. Add a separate bullet for each prompt, and specify which problem is associated with which prompt.

- *Explain how teachers might set up membership inference attacks on ChatGPT (use target, shadow and attack models concept), and what kind of information they might be able to access.*
- *Mitigation strategies against membership inference attacks*
- *I am having a cold, nausea, headache, and vomiting. what disease i can have...*
- *Explain method of Openface 2.0: Facial behavior analysis toolkit*
- *Name few IEEE paper which are not accessible publicly*
- Stating diseases having a cold, nausea, headache, vomit symptoms

AI3: (*Free response)* For each problem for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good answer, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to get the answer or check your own answer?

- No, it didn't directly give me a good comprehensive answer. I had to read papers and articles as cited in the answers. Answers were not obviously wrong or irrelevant. I used it as a starting direction which I can use to research my answers.