Assignment A4 : Generative Fine-tuning and Memorisation

Individual assignment • 20% individual-report component Released: 10 Dec 2026 Due: 17 Dec 2026, 23:59

In class we saw that fine-tuning teaches a model new behaviour — but it can also quietly **memorise its training data**. In this assignment you will **plant secrets in a corpus, fine-tune GPT-2 on it, and then try to steal those secrets back**. The question to answer: *how often must something appear before the model memorises it, and what does that mean for training on private data?* This is the warm-up for the group project *Model Leakage*.

Data. We use the comment text from the Kaggle competition *Jigsaw Unintended Bias in Toxicity Classification* (https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification). Load it with `load_lm()` and `load_canaries()` from the `starter/` bundle (join the competition first and set up Kaggle credentials). *Content note: the comments can be offensive.* `load_canaries()` gives you unique secret strings ("canaries", several at each of a few repetition counts). Keep the fine-tune tiny (`max_len`=128, batch 2, a few hundred steps): this runs in ∼20–40 min on a laptop (`mps`); Colab is an optional fallback.

Ethics. The canaries are synthetic. Do not fine-tune on, or try to extract, any real personal data.

1. Tasks

1. **Plant and fine-tune.** Insert each canary into the training text at its given frequency, then fine-tune GPT-2 small (`gpt2`) on the result.
2. **Attack.** Prompt the fine-tuned model with the **first half** of each canary, greedily generate the rest, and check whether the secret suffix comes back. Compute the **extraction rate per repetition count**.
3. **Explain.** State at what repetition count memorisation begins, and what it implies for training LLMs on private data.

What to submit & how it's graded. Submit `a4_extraction.csv` (columns `frequency, extraction_rate`). It is checked for the right schema and the expected trend — more repetition should mean higher extraction. *Metric: extraction rate per canary frequency.*

2. Reflection (required, max 200 words)

Submit `reflection.md` (or `.pdf`). It must answer these three fixed questions and include the AI-use declaration. The reflection is graded for honesty and insight, not length.

1. **What did you know before?** — your starting point on this topic.
2. **What was hardest, and how did you get unstuck?**
3. **What does your result say, and what would you try next?**

AI-use declaration. State which parts (if any) were generated with AI tools, and *how you verified them.* Using AI is allowed; undisclosed or unverified use is not.

3. Submission

Upload a single `zip` to Brightspace named `A4_<studentnumber>.zip`, containing the required output file(s) listed above, your `code/` folder, and `reflection.md`. Run the format checker in

`starter/` on your zip first — it validates file names, schemas, and value ranges locally before you submit. Auto-grading scores your output file against the hidden reference; the format checker passing is a precondition for grading.