Week 8 Assignment [4 Points]

Generative AI

Saarland University – Winter Semester 2024/25

Submission due: 21 December 2024 by 6pm CET

genai-w24-tutors
genai-w24-tutors@mpi-sws.org

1 Reading Assignment

This week's reading assignment comprises of the following:

(R.1) A Watermark For Large Language Models [1]. Paper link: https://proceedings.mlr.press/v202/kirchenbauer23a/kirchenbauer23a.pdf

[OPTIONAL] Below, we are providing additional material covering content relevant to the lecture:

- Tree-Rings Watermarks: Invisible Fingerprints for Diffusion Images [2].
  Paper Link: https://openreview.net/pdf?id=Z57JrmubNl
- Safe RLHF : Safe Reinforcement Learning from Human Feedback [3].
  Paper Link: https://openreview.net/pdf?id=TyFrPOKYXw

2 Exercise Assignment

This exercise assignment aims to give you a deeper understanding of the watermarking of LLMs as described in the paper *A Watermark for Large Language Models* [1]. These questions should be answered in the PDF submission file – see instructions in Section 4.

More concretely, we consider a simplified vocabulary $V = \{t_0, t_1, t_2, t_3, t_4\}$, where t_0 is a special token that marks the beginning of a sentence.

Table 1: Red List for text generation

x_{i-1}	Tokens in Red List
t_0	t_1, t_2
t_1	t_1, t_2
t_2	t_2, t_3
t_3	t_1, t_4
t_4	t_3, t_4

(E.1) First, consider Algorithm 1 in [1], *Text Generation with Hard Red List*. The possible tokens in the Red List depend on the previous token, as shown in Table 1. Based on the presence of any red token in the sentence, which of the following sentences contains a watermark, and which does not?

- $t_0t_1t_2t_3t_4t_1t_4$
- $t_0t_4t_1t_3t_3t_2t_1$

(E.2) Consider the text $t_0t_3t_2t_4t_1t_3t_3t_3$, which was generated by a watermarked model. How could you remove the watermark by inserting a single new token? What would the resulting text look like?

(E.3) Consider the text $t_0t_1t_3t_2t_1t_4t_2$, which was generated by a non-watermarked model. How could you make this text look like it was generated by a watermarked model by inserting a single new token? What would the resulting text look like?

(E.4) Consider a model that predicts each token with equal probability, with all the logit values $l_{t_1}^{(i)} = l_{t_2}^{(i)} = \ldots = 1$ at any time step i. What would be the probability that the model generates the following sentences when not considering any watermarking?

- $t_0t_1t_4t_3t_4t_1t_4$
- $t_0t_4t_1t_3t_3t_2t_1$

Remark: Note that the model generates tokens one by one in order to generate a sentence (refer to week 2 slides). Here, t_0 marks the beginning of the sentence and you can set the probability of the first token being t_0 as 1.0.

(E.5) Now, consider Algorithm 2 in [1], *Text Generation with Soft Red List*. Use the same Red List as in Table 1. When using $\delta = 1$ and the same model as in the previous exercise, what is the probability that the following sentences are generated considering the Soft Red List watermarking method?

- $t_0t_1t_4t_3t_4t_1t_4$
- $t_0t_4t_1t_3t_3t_2t_1$

3 Implementation Based Assignment

In this week's implementation assignment, you will test the effectiveness of watermarking with the Hard Red List as described in Algorithm 1 of the paper *A Watermark for Large Language Models* [1]. Further, you will test the effectiveness of the T5 Span Attack as described in Section 7.1. We have provided you with the following files in `week8_implementation.zip`:

- `watermarking.py`: A script that implements watermarking with the Hard Red List and the T5 Span Attack. It will generate the required data for (I.1)-(I.5) in separate text files. Running this file once will generate all the necessary data, i.e., you do not need to modify this file.
- `requirements.txt`: This file contains all the required packages to run this week's implementation assignment. Refer to Section 5.2 for installing the requirements.

A watermark should be accurately detected while avoiding falsely classifying non-watermarked text – for this, we will analyze the rate of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). Furthermore, the watermark should not have a noticeable impact on the generation quality – for this, we will use the model's perplexity as a measure of the generation quality. The questions below should be answered in the PDF submission file – see instructions in Section 4.

(I.1) `I1.txt` contains 3 example generations with and without a watermark. How do they compare in terms of quality and perplexity?

(I.2) We will now do a more rigorous evaluation of the watermarking technique. `I2.txt` contains the detection rates for 50 watermarked and 50 non-watermarked responses. How does the choice of z impact the detection? Out of these options, what is the best choice for it?

(I.3) `I3.txt` contains the average perplexity for the generated watermarked and non-watermarked responses. How do watermarked responses compare to non-watermarked ones?

(I.4) Next, we will analyze the T5 Span Attack described in Section 7.1 of [1]. `I4.txt` contains the detection rates for watermarked and non-watermarked responses after they have been modified using the attack. Compared to `I2.txt`, how successful is the attack in removing watermarks?

(I.5) `I5.txt` contains the average perplexity of the generations after they have been modified using the T5 Span Attack. What impact does the attack have on the quality of generation?

4 Submission Instructions

Please submit the following file:

- **<Lastname>_<Matriculation>_week8.pdf**. This PDF file will report answers to the following questions: E.1, E.2, E.3, E.4, E.5, I.1, I.2, I.3, I.4, and I.5. The PDF file should be generated in LaTeX using NeurIPS 2024 style files; see further formatting instructions in Section 5.1. The PDF file size should **not exceed** 1**mb**. Please use the following naming convention: Replace <Lastname> and <Matriculation> with your respective Lastname and Matriculation number (e.g., Singla_1234567_week8.pdf).

5 Technical Instructions

5.1 Instructions for Preparing PDFs

Refer to Week 1 assignment.

5.2 Instructions for Implementation

For detailed instructions on how to use the GPU resources in Google Colab, please refer to the Technical Instructions in Week 3 assignment.

In `week8_implementation.zip`, we have provided a `requirements.txt` file. This file contains the list of all the packages needed to run this week's assignment and has to be uploaded in Google Colab. Please make sure that before you run the provided code, you first run the following command in Google Colab in order to install all the required packages:

```
!pip install -r requirements.txt
```

References

[1] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A Watermark for Large Language Models. In *ICML*, 2023.

[2] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-Rings Watermarks: Invisible Fingerprints for Diffusion Images. In *NeurIPS*, 2023.

[3] Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe Reinforcement Learning from Human Feedback. In *ICLR*, 2024.

Based on template from https://media.neurips.cc/Conferences/NeurIPS2024/Styles.zip