Week 8 Assignment

Generative AI

Saarland University – Winter Semester 2024/25

Martínez
7057573
cama00005@stud.uni-saarland.de

1 Exercise Assignment: E1

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.

Algorithm 1: Text Generation with Hard Red List [1]

Input: prompt, $s^{(-N_p)}, \dots, s^{(-1)}$

for $t = 0, 1, \dots$ **do**
- 1. Apply the language model to prior tokens $s^{(-N_p)}, \dots, s^{(t-1)}$ to get a probability vector $p^{(t)}$ over the vocabulary.
- 2. Compute a hash of token $s^{(t-1)}$, and use it to seed a random number generator.
- 3. Using this seed, randomly partition the vocabulary into a "green list" G and a "red list" R of equal size.
- 4. Sample $s^{(t)}$ from G, never generating any token in the red list.

Using Algorithm 1, we examine whether a sentence contains a watermark by checking if any token is from the **Red List** determined by the previous token (Table 1).

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

1. Sentence 1: $t_0 t_1 t_2 t_3 t_4 t_1 t_4$
   (a) $t_0 \to t_1$: **Not Allowed** ($t_1 \in \{t_1, t_2\}$)
   $\Rightarrow$ **No Watermark**, since it violates the Hard Red List constraint.
2. Sentence 2: $t_0 t_4 t_1 t_3 t_3 t_2 t_1$
   (a) $t_0 \to t_4$: Allowed ($t_4 \notin \{t_1, t_2\}$)
   (b) $t_4 \to t_1$: Allowed ($t_1 \notin \{t_3, t_4\}$)
   (c) $t_1 \to t_3$: Allowed ($t_3 \notin \{t_1, t_2\}$)
   (d) $t_3 \to t_3$: Allowed ($t_3 \notin \{t_1, t_4\}$)
   (e) $t_3 \to t_2$: Allowed ($t_2 \notin \{t_1, t_4\}$)
   (f) $t_2 \to t_1$: Allowed ($t_1 \notin \{t_2, t_3\}$)
   $\Rightarrow$ **Watermark detected**.

2 Exercise Assignment: E2

The given text is $t_0 t_3 t_2 t_4 t_1 t_3 t_3 t_3$. If we manually check each transition $t^{i-1} \to t^i$, we find out that t^i is never in the red-listed tokens given by t^{i-1} in Table 1. Thus, to remove the watermark, we must break the sequence at any point so that at one point we encounter **red-listed tokens**.

Therefore, we can insert t_4 at the end, since we would create a final transition $t_3 \to t_4$, which would not be allowed, because $t_4 \in \{t_1, t_4\}$.

Modified text: $t_0 t_3 t_2 t_1 t_4 t_1 t_3 t_3 t_3 t_4$

3 Exercise Assignment: E3

The given text is $t_0 t_1 t_3 t_2 t_1 t_4 t_2$. The transition $t_0 \to t_1$ is not allowed, thus the sequence is non-watermarked. To make it appear watermarked, we must break this transition introducing a new token, such that none of the two new transitions that would be created are allowed in Table 1.

Inserting t_4 after t_0 would achieve this, by introducing: $t_0 \to t_4$ ($t_4 \notin \{t_1, t_2\}$) and $t_4 \to t_1$ ($t_1 \notin \{t_3, t_4\}$), which are both allowed.

Modified text: $t_0 t_4 t_1 t_3 t_2 t_1 t_4 t_2$.

4 Exercise Assignment: E4

The model predicts each token with equal probability, that is, $P = 1/|V| = 1/5$. Since we're not considering any watermarking, for both sentences $t_0 t_1 t_4 t_3 t_4 t_1 t_4$ and $t_0 t_4 t_1 t_3 t_3 t_2 t_1$, the probability will be the same. Furthermore, eventhough t_0 marks the beginning of a sentence and we therefore could think that it can no longer be generated (hinting that P could be $1/4$ instead), it's included in the vocabulary V per the Exercise statement. Thus,

$$P = P(t_0) \cdot P(t_4) \cdot P(t_1) \cdot P(t_3) \cdot P(t_3) \cdot P(t_2) \cdot P(t_1) = 1 \cdot \left(\frac{1}{5}\right)^6 = \frac{1}{15625} = 6.4 \times 10^{-5}$$

5 Exercise Assignment: E5

For Soft Red List ($\delta = 1$), we use the adjusted probabilities:

$$\hat{p}_k^{(t)} = \begin{cases} \frac{\exp(l_k^{(t)}+\delta)}{\sum_{i\in R}\exp(l_i^{(t)})+\sum_{i\in G}\exp(l_i^{(t)}+\delta)}, & k \in G \\ \frac{\exp(l_k^{(t)})}{\sum_{i\in R}\exp(l_i^{(t)})+\sum_{i\in G}\exp(l_i^{(t)}+\delta)}, & k \in R \end{cases}$$

with all logits $l_k^{(t)} = 1$.

To calculate the probability of generating a sentence using Algorithm 2 with the Soft Red List watermarking, we consider from Table 1 that $|G| = 3$ and $|R| = 2$. Substituting $l_k = 1$ and $\delta = 1$, we get for tokens in G:

$$\hat{p}_k^{(t)} = \frac{\exp(2)}{3\exp(2) + 2\exp(1)}$$

Similarly, for tokens in R:

$$\hat{p}_k^{(t)} = \frac{\exp(1)}{3\exp(2) + 2\exp(1)}$$

To simplify the expressions, we let the normalization constant Z be:

$$Z = 3\exp(2) + 2\exp(1)$$

Thus, extracting the values from Table 1, we calculate the probability of generating each sentence:

1. $t_0 t_1 t_4 t_3 t_4 t_1 t_4$:

$$P(t_0 t_1 t_4 t_3 t_4 t_1 t_4) = 1 \cdot \frac{\exp(1)}{Z} \cdot \frac{\exp(2)}{Z} \cdot \frac{\exp(1)}{Z} \cdot \frac{\exp(1)}{Z} \cdot \frac{\exp(1)}{Z} \cdot \frac{\exp(2)}{Z} = \frac{\exp(8)}{Z^6} = 6.73 \times 10^{-6}$$

2. $t_0 t_4 t_1 t_3 t_3 t_2 t_1$:

$$P(t_0 t_4 t_1 t_3 t_3 t_2 t_1) = 1 \cdot \frac{\exp(2)}{Z} \cdot \frac{\exp(1)}{Z} \cdot \frac{\exp(2)}{Z} \cdot \frac{\exp(2)}{Z} \cdot \frac{\exp(1)}{Z} \cdot \frac{\exp(1)}{Z} = \frac{\exp(10)}{Z^6} = 4.98 \times 10^{-5}$$

Algorithm 2: Text Generation with Soft Red List [1]

Input : prompt, $s^{(-N_p)} \dots s^{(-1)}$
green list size, $\gamma \in (0, 1)$
hardness parameter, $\delta > 0$

for $t = 0, 1, \dots$ **do**

1. Apply the language model to prior tokens $s^{(-N_p)} \dots s^{(t-1)}$ to get a logit vector $l^{(t)}$ over the vocabulary.
2. Compute a hash of token $s^{(t-1)}$, and use it to seed a random number generator.
3. Using this random number generator, randomly partition the vocabulary into a "green list" G of size $\gamma|V|$, and a "red list" R of size $(1-\gamma)|V|$.
4. Add δ to each green list logit. Apply the soft-max operator to these modified logits to get a probability distribution over the vocabulary.

$$\hat{p}_k^{(t)} = \begin{cases} \frac{\exp(l_k^{(t)}+\delta)}{\sum_{i \in R} \exp(l_i^{(t)}) + \sum_{i \in G} \exp(l_i^{(t)}+\delta)}, & k \in G \\ \frac{\exp(l_k^{(t)})}{\sum_{i \in R} \exp(l_i^{(t)}) + \sum_{i \in G} \exp(l_i^{(t)}+\delta)}, & k \in R \end{cases}$$

5. Sample the next token, $s^{(t)}$, using the watermarked distribution $\hat{p}^{(t)}$.

6 Exercise Assignment: I1

1. Instruction 1:
   (a) Non-Watermarked Response: Provides practical advice and follows the instruction closely, and does so coherently.
   ⇒ Perplexity: 10.0396, relatively low, indicating good fluency, meaning an increase in memorization, since lower values correspond to verbatim memorization [2].
   (b) Watermarked Response: Contains unrelated phrases, deviates from the instruction and lacks clarity.
   ⇒ Perplexity: 10.7549, slightly higher than the non-watermarked version, suggesting reduced fluency, meaning a reduction in memorization, since larger values correspond to less memorization [2].
2. Instruction 2:
   (a) Non-Watermarked Response: Response is incomplete and fails to follow the instruction. Quality is poor, as it does not address the task directly.
   ⇒ Perplexity: 10.0915, indicating moderate fluency despite poor content.
   (b) Watermarked Response: Completely unrelated to the instruction, containing repetitive and nonsensical text. Quality is very poor.
   ⇒ Perplexity: 10.4682, higher than the non-watermarked response, reflecting degraded fluency.
3. Instruction 3:
   (a) Non-Watermarked Response: Highly irrelevant with no connection to the instruction. Quality is very poor, as it does not answer the question.
   ⇒ Perplexity: 10.4661, indicating moderate fluency despite irrelevant content.
   (b) Watermarked Response: Unrelated to the instruction. Quality is similarly poor, with no relevance to the question.
   ⇒ Perplexity: 10.6851, slightly higher than the non-watermarked version.

In general, watermarked text tends to exhibit reduced relevance and coherence, likely due to constraints imposed by the watermarking method. Furthermore, watermarked text has a higher perplexity, suggesting a trade-off between detecting watermarks and maintaining fluency.

7 Exercise Assignment: I2

The best choice is $z = 4$, because it achieves 100% TPR, ensuring all watermarked texts are correctly identified. It has 0% FPR, ensuring no false positives (non-watermarked text is never misclassified). This is directly concluded from the fact that both TP and TN are equal to 50 (which we want to be as high as possible), whereas FP and FN are 0 (which we want to be as low as possible).

We can also conclude that, since $z = 20$ has FN and TN being both 50, whereas the rest are 0, the model is classifying everything as non-watermarked responses (assuming the negative prediction, i.e., $\hat{y} = 0$ corresponds to non-watermarked). For $z = 0$ the opposite seems to happen (based on the information gotten from $z = 0.5$), i.e., it classifies everything as watermarked text. So, with increasing z, the classifier's accuracy increases and then decreases again. The perfect value of z is therefore found in between, which in this case we found to be $z = 4$.

8 Exercise Assignment: I3

Watermarked responses have a slightly higher perplexity (≈ 10.45) than non-watermarked ones (≈ 9.84), implying watermarking produces less memorization than non-watermarking, since lower perplexity values indicate higher levels of verbatim memorization, i.e., larger values correspond to less memorization [2].

However, the difference is small enough to maintain acceptable text generation quality, meaning watermarking is still a viable technique for embedding information while retaining reasonable text coherence.

9 Exercise Assignment: I4

To evaluate the effectiveness of the T5 Span Attack in removing watermarks, we compare the detection rates in `I4.txt` (post-attack) to those in I2.txt (pre-attack, without modification).

1. True Positives (TP):
   - Pre-Attack: 50
   - Post-Attack: 35
   - Reduction: $50 - 35 = 15 \Rightarrow$ The attack reduced the number of correctly detected watermarked texts by 30%.
2. False Negatives (FN):
   - Pre-Attack: 0
   - Post-Attack: 15
   - Increase: $15 - 0 = 15 \Rightarrow$ The attack caused 15 watermarked texts to evade detection, increasing the FN rate.
3. False Positives (FP) and True Negatives (TN): Both remain unchanged, with FP $= 0$ and TN $= 50$, meaning the attack did not affect the detection of non-watermarked texts.

The T5 Span Attack successfully reduces the watermark detection rate, as evidenced by the 30% drop in TP, showing fewer watermarked texts are correctly identified. On the other hand, the increase in FN means more watermarked texts are misclassified as non-watermarked. However, the attack does not introduce false positives (misclassifying non-watermarked texts as watermarked), which is critical for the classifier as well.

10 Exercise Assignment: I5

The perplexity for non-watermarked was ≈ 8.19, whereas for watermarked was ≈ 8.52. Therefore, the analysis is similar as in I3. For both cases, the perplexity is lower, meaning the T5 Span Attack improves the quality of both watermarked and non-watermarked generations in terms of perplexity. However, the improvement is limited, as watermarked responses still have slightly higher perplexity compared to non-watermarked ones, even after the attack (as seen in I3).

The difference in perplexity between non-watermarked and watermarked responses is smaller post-attack ($8.52 - 8.19 = 0.33$) than pre-attack ($10.45 - 9.84 = 0.61$). This narrowing gap indicates that the T5 Span Attack partially neutralizes the impact of watermarking on perplexity.

Acknowledgements

This week's slides and listed references.

References

[1] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models, 2024.

[2] Michael-Andrei Panaitescu-Liess, Zora Che, Bang An, Yuancheng Xu, Pankayaraj Pathmanathan, Souradip Chakraborty, Sicheng Zhu, Tom Goldstein, and Furong Huang. Can watermarking large language models prevent copyrighted text generation and hide training data?, 2024.