Token-Specific Watermarking with Enhanced Detectability and Semantic Coherence for Large Language Models

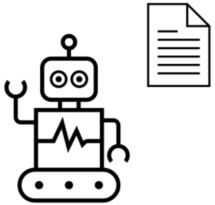
Sai Ashish Somayajula * Mingjia Huo* Youwei Liang Ruisi Zhang

Farinaz Koushanfar Pengtao Xie

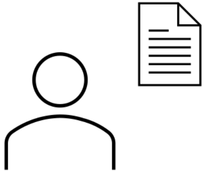
University of California, San Diego

* Equal contribution

Detecting LLM Generated Texts



LLM generated



Human generated

Detect

Academic dishonesty

Spam content

Misleading content

Training degeneration

Prior Methods

Indistinguishable methods [1]

- Tied to a sampling strategy such as multinomial sampling, top-k sampling etc.
- Restrictive
- EXP, EXP-Edit

Prior Methods: EXP

Implemented on top of multinomial sampling by casting it as exponential minimum sampling

Standard multinomial sampling

- Given the unnormalized logits over vocabulary, $[l_1, l_2, l_3, \ldots., l_V]$, where V is the vocabulary size
- Convert to probabilities via softmax; $p_i = \frac{e^{l_i}}{\sum_{j=1}^{V} e^{l_j}}, i = 1, \ldots, V$
- Draw next token based on these probabilities (multinomial sampling)

Prior Methods: EXP

Exponential minimum sampling (trick)

- For each token i in the vocabulary, draw a uniform random variable $U_i \sim Uniform(0,1)$
- Convert into exponential: $X_i = \frac{-\log(U_i)}{e^{l_i}}$
- Select the token i^* with the smallest X_i over all tokens in the vocabulary

Prior Methods: EXP

Why is this equivalent to multinomial sampling?

- Observe $X_i \sim Exp(e^{l_i})$
- $X_1, X_2, X_3, \dots, X_V$ are independent exponential random variables with rates $e^{l_1}, e^{l_2}, e^{l_3}, \dots, e^{l_V}$, respectively
- $P(X_i = \min\ (X_1, X_2, X_3, \dots, X_V)) = \frac{e^{l_i}}{\sum_{j=1}^{V} e^{l_j}}$
- The derived probability exactly equals the softmax probability!!

Summary - Exponential minimum sampling

- The sampled next token is given by the expression, $\arg \min_{i \in \{1,2,3,\dots,V\}} \frac{-\log(U_i)}{e^{l_i}}, \quad U_i \sim Uniform(0,1)$

Prior Methods: EXP

Embedding a watermark

- Convert the pseudo-random sampling process into a deterministic one using a watermark key
- Given a watermark key (setting random seed in python), sampled U_i is deterministic making the generated sentence deterministic
- Observe, a larger U_i most likely results in next token as the i^{th} token (which is useful for detection) from the sampling strategy: $\arg\min_{i \in \{1,2,3,\ldots,V\}} \frac{-\log(U_i)}{e^{l_i}}, \quad U_i \sim Uniform(0,1)$
- Given the watermark key, check whether the chosen token in the generated text is in the higher end of the spectrum of U_i at that position

Prior Methods: EXP

Detecting a watermark

- Determine whether a given text was generated using a hidden watermark key
- Each position t in the text is associated with a uniform random draw U^t
- Given watermark key, U^t is deterministic
- A large draw U_i^t (closer to 1) makes token i more likely to be selected at position t; Check if $text_t$ is in that set of higher U_i^t's
- Calculate $\mathrm{expCost} = \sum_{t=1}^{len(text)} \log(1 - U_{text_t}^t)$, where $U_{text_t}^t$ is draw corresponding to the token at position t in generated text
- If the text used the watermark key, the chosen tokens typically have larger $U_{text_t}^t$
- Larger $U_{text_t}^t \Rightarrow$ more negative $\log(1 - U_{text_t}^t) \Rightarrow$ lower expCost
- A very low expCost strongly suggests the text is watermarked

Prior Methods: EXP-edit

Embedding the watermark is the same as EXP

Detecting a watermark

- Further includes Levenshtein distance [1] to make the detection more robust

Limitations

[5] argues that indistinguishability is not necessary and imposes restrictions

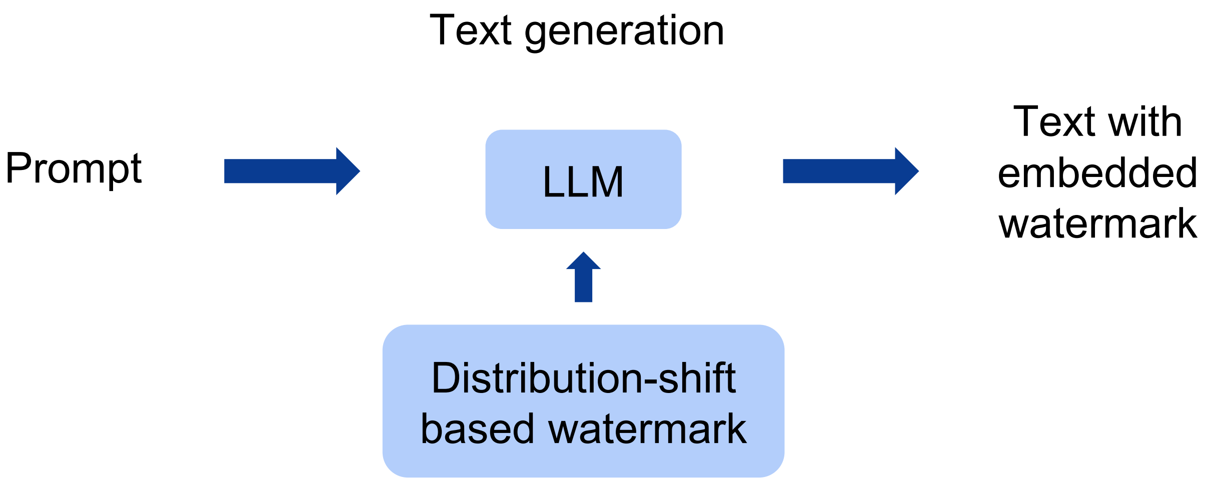
- Restriction on the sampling strategy; for instance, cannot be used with beam search where there is no pseudo random sampling process

Prior Methods

Distribution-shift based methods [2, 3, 4]

- Shift the output distribution towards a subset of tokens in the vocabulary
- Statistically estimate the likelihood that the probability distribution has shifted
- Can be used with any sampling strategy such as beam search
- KGW, SWEET
- [5] claims these methods are simpler, easiest-to-detect algorithm, and often match or surpass the performance of indistinguishable watermarking methods.

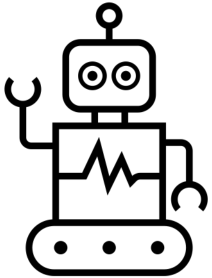
Prior Methods: Distribution-Shift Based Methods



Prior Methods: KGW

During the generation of tth token,

cat jumped on the | bed



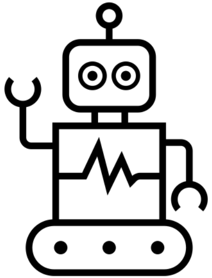
moon

sofa

pen

Prior Methods: KGW

cat jumped on the | bed moon sofa pen

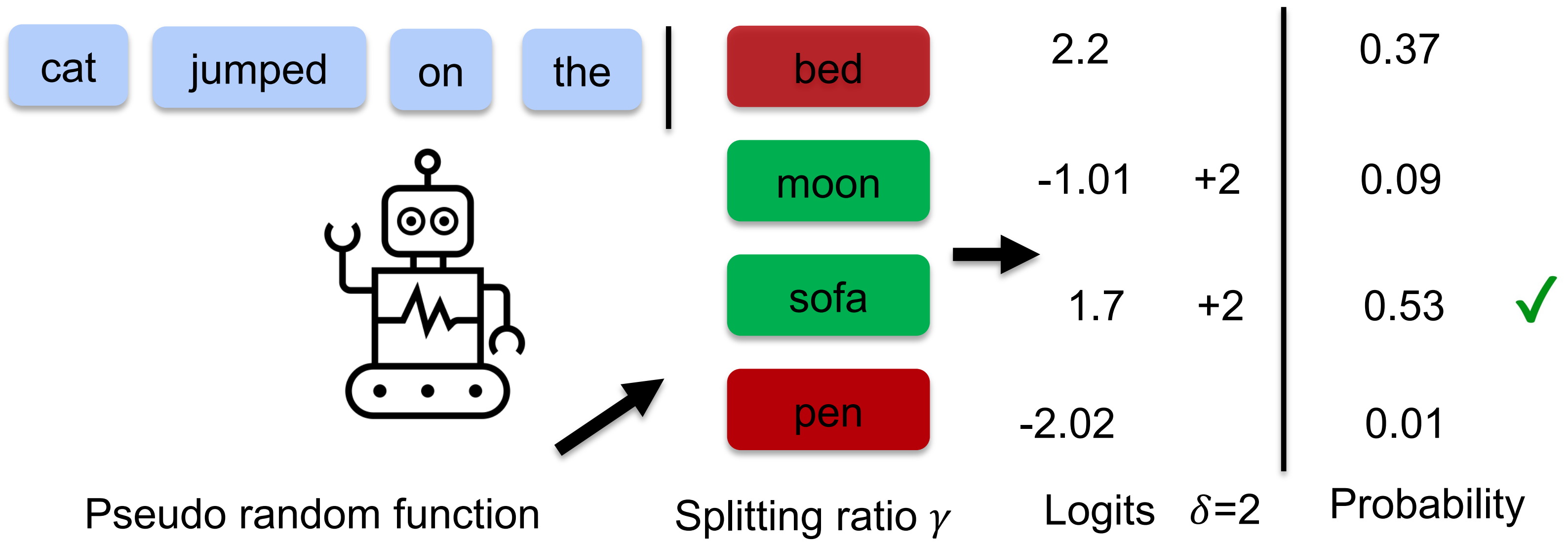


Splitting ratio γ

Pseudo random function

Hash of previous token as seed to partition vocabulary into red-green list

Prior Methods: KGW

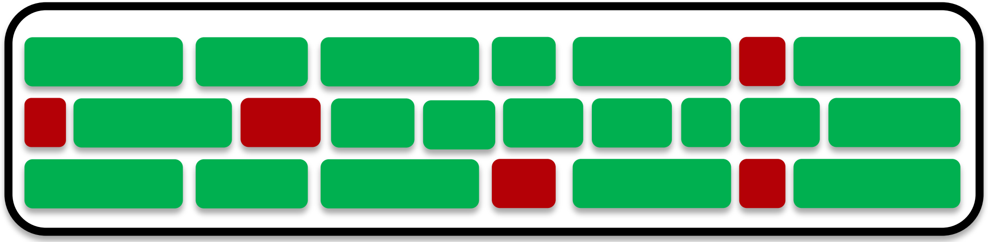


Add δ to all the green tokens to bias the distribution towards green-list

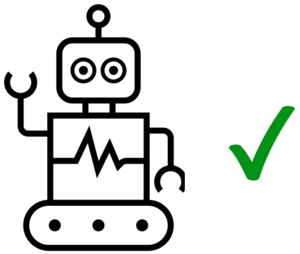
Prior Methods: KGW

Detection

- Null hypothesis that the next token is selected without the knowledge of green-red list rule, i.e., without addition of δ
- Given hash function, count the number of green tokens in the generation
- Calculate the z-score, $z = \frac{(|s|_G - \gamma T)}{\sqrt{T\gamma(1-\gamma)}}$



Z-score = $\frac{(|s|_G - \gamma T)}{\sqrt{T\gamma(1-\gamma)}}$ = 4



Z-score > τ (say 3)

Limitations

Face challenges in improving the semantics and detectability at the same time

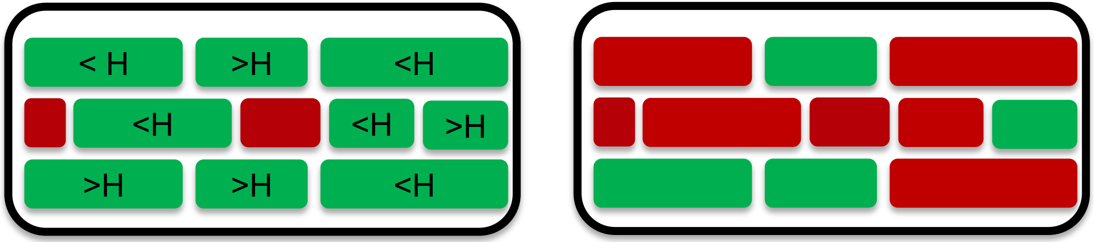
- Improving one compromises the other

Lack adaptive mechanism to adjust γ and δ appropriately

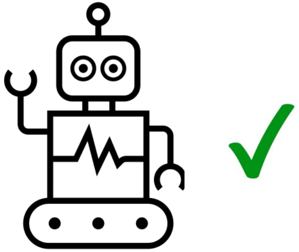
- Ex: Sun rises in the __. It is 'east' with certainty. High δ and low γ might not select 'east'.

Prior Methods: SWEET

- Modification to KGW; Watermark only high-entropy tokens, i.e., tokens whose entropy $(-\sum_{w_t \in V} p(w_t|w_{1:t-1}) \log p(w_t|w_{1:t-1}))$ is greater than a threshold, H
- The entropy is set to the average entropy of all the tokens in the training set
- Calculate the z-score, $z = \frac{(|s|^H{}_G - \gamma T^H)}{\sqrt{T^H \gamma(1-\gamma)}}$; where $|s|^H{}_G$ are the number of high entropy green tokens and T^H are the total number of high-entropy entropy tokens in the generation



Z-score = $\frac{(|s|^H{}_G - \gamma T^H)}{\sqrt{T^H \gamma(1-\gamma)}}$ = 4



Z-score > τ (say 3)

Limitations

Restrictive on the choice of entropy threshold H which is fixed; sub-optimal

Lack adaptive mechanism

- Adjust γ and δ appropriately based on the semantics of the previous token
- A smarter alternative to entropy thresholding

Proposed Method

Propose learning token-specific splitting ratio and watermark logit, i.e., γ_t and δ_t

Proposed Method

Propose learning token-specific splitting ratio and watermark logit, i.e., γ_t and δ_t

$s^{(-M)}, \dots, s^{(-1)}$ $s^{(0)}, \dots, s^{(t-1)}$

Prompt Generation till now

Proposed Method

Propose learning token-specific splitting ratio and watermark logit, i.e., γ_t and δ_t

$s^{(-M)}, \ldots, s^{(-1)}$ $s^{(0)}, \ldots, s^{(t-1)}$ $\longrightarrow$ Embeddings of $s^{(t-1)}$

Prompt Generation till now

Proposed Method

Propose learning token-specific splitting ratio and watermark logit, i.e., γ_t and δ_t

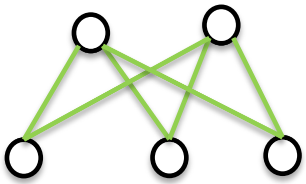
$s^{(-M)}, \dots, s^{(-1)}$

Prompt

$s^{(0)}, \dots, s^{(t-1)}$

Generation till now

Embeddings of $s^{(t-1)}$



G_γ: γ network

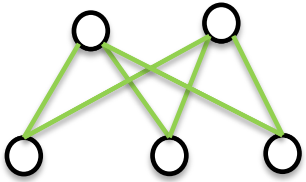
γ_t

Proposed Method

Propose learning token-specific splitting ratio and watermark logit, i.e., γ_t and δ_t

$s^{(-M)}, \ldots, s^{(-1)}$ $\quad$ $s^{(0)}, \ldots, s^{(t-1)}$ $\longrightarrow$ Embeddings of $s^{(t-1)}$

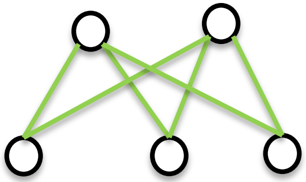
Prompt $\quad$ Generation till now



G_γ: γ network

$\downarrow$

γ_t



G_δ: δ network

$\downarrow$

δ_t

Proposed Method

Differentiable sampling for splitting the vocabulary

- For each token $v \in V$, sample $\mathrm{y}_v^{(t)} \sim B(\gamma_t)$, Bernoulli distribution parameterized by γ_t.
- If $\mathrm{y}_v^{(t)} = 1$, then the token v belongs to green list else red list
- Gumbel softmax trick makes sampling process differentiable

Proposed Method

Given original logits $l_v^{(t)}$ for token v, modified logits after biasing the green-list tokens

$$\hat{\boldsymbol{l}}_v^{(t)} = l_v^{(t)} + y_v^{(t)} * \delta_t$$

Proposed Method

Training objectives

- Detection loss
- Semantic loss

Proposed Method

Detection loss

- Since we have a token-specific γ_t and δ_t, the z-score expression has to be updated based on this distribution

Proposed Method

Theorem: Consider T independent Bernoulli random variables $X_1, \dots, X_T$, each with means $\mu_1, \dots, \mu_T, 0 < \ \mu < 1 \ \forall \ t \in 1, \dots, T$. The sum of these variables, $\sum_{t=1} X_t$, follows a Poisson binomial distribution. When T is sufficiently large, this distribution can be approximated by a Gaussian distribution with mean: $\sum_{t=1}^{T} \mu_t$ and variance: $\sum_{t=1}^{T} \mu_t (1 - \mu_t)$.

Proposed Method

Modified Z-score = $\frac{|s|_G - \sum_{t=1}^{T} \gamma_t}{\sqrt{\sum_{t=1}^{T} \gamma_t (1-\gamma_t)}}$ to account for varying γ_t

Detection loss

- Improve detectability by maximizing this objective
- However, $|s|_G$, count of green tokens, is non-differentiable w.r.t γ_t and δ_t

Proposed Method

Detection loss

- Propose differentiable surrogate $\hat{z} = \frac{\sum_{t=1}^{T} p_{gr}^{(t)} - \sum_{t=1}^{T} \gamma_t}{\sqrt{\sum_{t=1}^{T} \gamma_t (1-\gamma_t)}}$, where $p_{gr}^{(t)}$ is the probability of selecting a green token.
- Maximize $\hat{z}$ or minimize detection loss, $L_D = -\hat{z}$

Proposed Method

Semantic loss

- Generate sentence embeddings of texts before and after watermarking, i.e., s and s_w using the SimCSE model f_θ
- Maximize the cosine similarity between them, $\cos_{sim}(f_\theta(s),\ f_\theta(s_w))$
- Thus, minimize semantic loss, $L_S = -\cos_{sim}(f_\theta(s),\ f_\theta(s_w))$

Proposed Method

Multi-objective Optimization

- Optimizing for two competing loss functions L_D and L_S

$$\min_{G_\gamma, G_\delta} L_D(G_\gamma, G_\delta) \text{ and } \min_{G_\gamma, G_\delta} L_S(G_\gamma, G_\delta)$$

- Estimate pareto optimal solutions using multiple-gradient descent algorithm (MGDA) [6]

Multiple-Gradient Descent Algorithm

Let g_D and g_S are the gradients of L_D and L_S w.r.t (G_γ, G_δ)

$$\lambda^* = argmin_{\lambda \in [0,1]} \left|\left|\lambda g_D + (1-\lambda) g_S\right|\right|_2$$

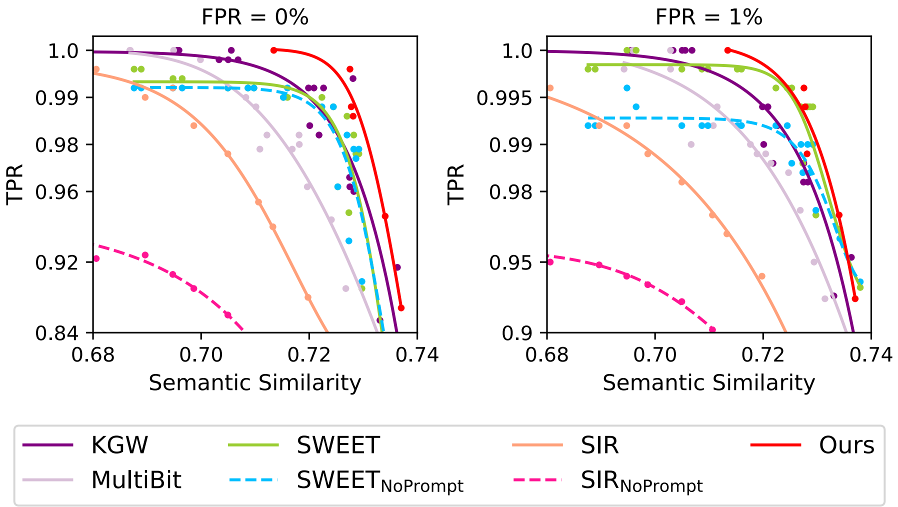
$$g = \lambda^* g_D + (1-\lambda^*) g_S$$

Update (G_γ, G_δ) using the gradient g

Experimental Setup

- Main experiments
  - C4 dataset
    - Training split 6400, Validation split 500, Test split 500
  - Generation length set to 200
- Z-score threshold is empirically determined on respective test sets
  - Set z-score threshold to maintain FPR at 0% and 1%

Results



Comparison of the trade-off for semantic integrity and detectability of different methods applied to OPT-1.3B.

Results

Method	TPR @ 0%	TPR @ 1%	SimCSE
EXP-edit	0.922	0.996	0.655
EXP-edit (Top-k=50)	0.968	0.996	0.677
Ours (Top-k=50)	**1.000**	**1.000**	**0.713**

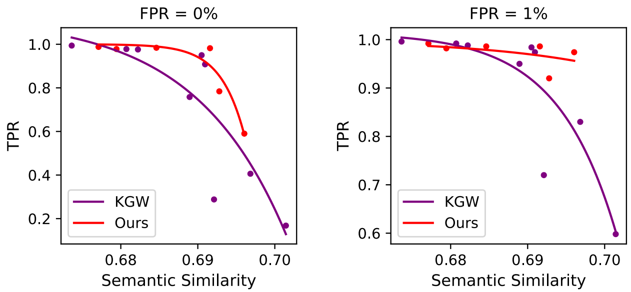
Comparison of our method with indistinguishable method - EXP-edit and its variant EXP-edit (Top-k=50) [1].

Results

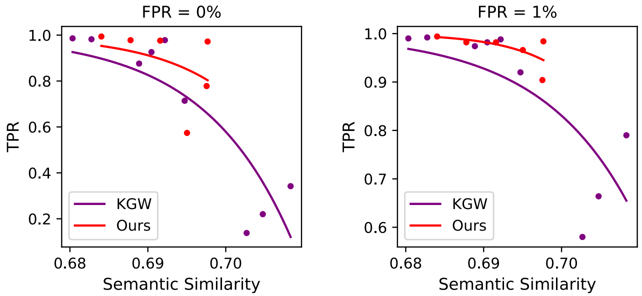
Method	Generation (s)	Detection (s)
No Watermark	3.220	-
KGW	3.827	0.067
SWEET	4.030	0.127
EXP-edit	24.693	155.045
SIR	8.420	0.337
MultiBit	6.500	0.610
Ours	3.946	0.166

Generation and detection speed on OPT-1.3B for generating 200 tokens, measured in seconds.

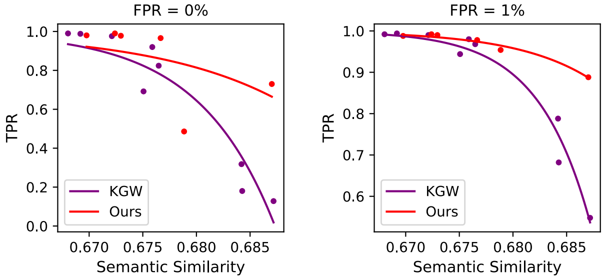
Results



a. LLAMA2 7B



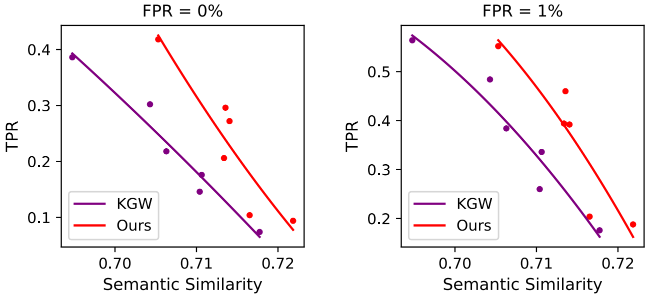
b. LLAMA2 13B



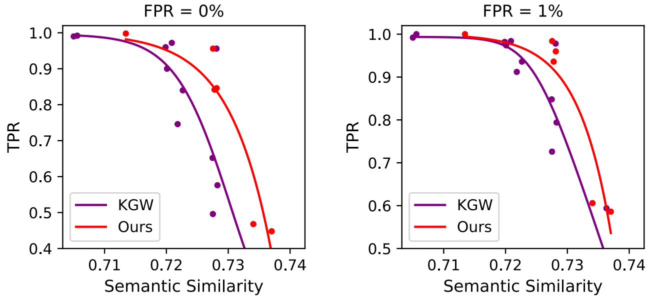
c. LLAMA2 70B

Performance of Ours (trained on OPT-1.3B) and KGW when applied to LLAMA2 7B, 13B, and 70B.

Results



a. Dipper paraphrase attack



b. Copy-Paste-3 attack

Comparison of our method with KGW under dipper paraphrase attack (left) and copy-paste-3 attack (right). Please refer to the paper for other attack results.

Conclusions

- Propose to adapt the watermark strength based on the semantics of the preceding token
- Propose a light-weight network to output token-specific γ_t and δ_t
- Propose a differentiable surrogate of z-score metric for optimization
- Optimize in a multi-objective optimization framework
- Extensive experiments on various scenarios shows the efficacy of our proposed method

References

[1] Kuditipudi, Rohith, et al. "Robust distortion-free watermarks for language models." *arXiv preprint arXiv:2307.15593* (2023).

[2] Kirchenbauer, John, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. "A watermark for large language models." In *International Conference on Machine Learning*, pp. 17061-17084. PMLR, 2023.

[3] Lee, T., Hong, S., Ahn, J., Hong, I., Lee, H., Yun, S., Shin, J., and Kim, G. Who wrote this code? watermarking for code generation. *arXiv preprint arXiv:2305.15060*, 2023.

[4] Liu, Aiwei, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. "A semantic invariant robust watermark for large language models." *arXiv preprint arXiv:2310.06356* (2023).

[5] Piet, Julien, Chawin Sitawarin, Vivian Fang, Norman Mu, and David Wagner. "Mark my words: Analyzing and evaluating language model watermarks." *arXiv preprint arXiv:2312.00273*(2023).

References

[6] Sener, Ozan, and Vladlen Koltun. "Multi-task learning as multi-objective optimization." *Advances in neural information processing systems* 31 (2018).