An Analysis of Tokenization: Transformers under Markov Data

Nived Rajaraman, Jiantao Jiao, Kannan Ramchandran

NeurIPS 2024

Presenter: Sangyoon Lee, Minjae Park

Introduction

“Tokenizer from a theoretical point of view”

Contributions.

1. There are very simple **k th-order Markov processes** such that in the **absence of any tokenization**, transformers empirically predict characters according to a **unigram model**.

2. In the presence of tokenization, transformers appear to achieve near-optimal cross-entropy loss.

3. Analyze a toy tokenizer and show that as dictionary size grows, unigram models get better at modeling the probabilities of sequences drawn from Markov sources.
   (+ LZW & BPE tokenizers)

Overview

1. Data generating process

2. Importance of tokenization in transformer
    a. Theorem & Empirical results

3. Unigram model under tokenization

4. Bound of CE loss under greedy encoder

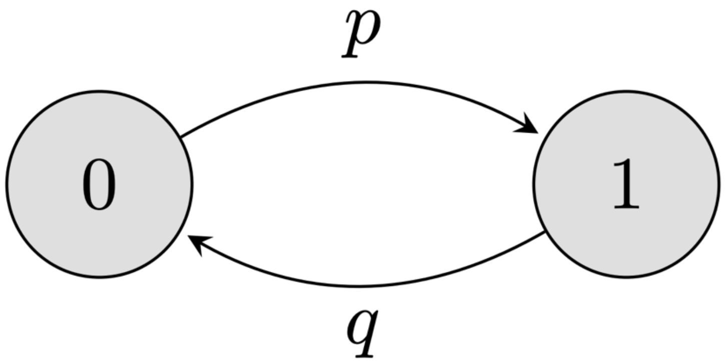
5. Expansion to LZW tokenizer

Data generating process

consider a simplification of real-world data generating process.

: **k-th order Markov process** over characters.

2-state switching process.

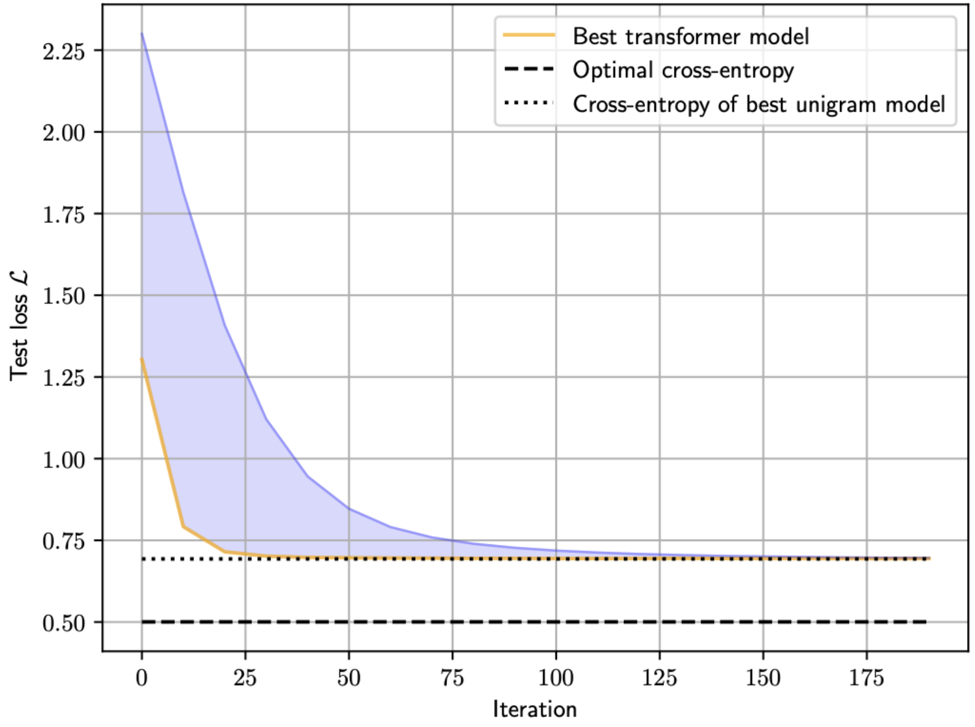


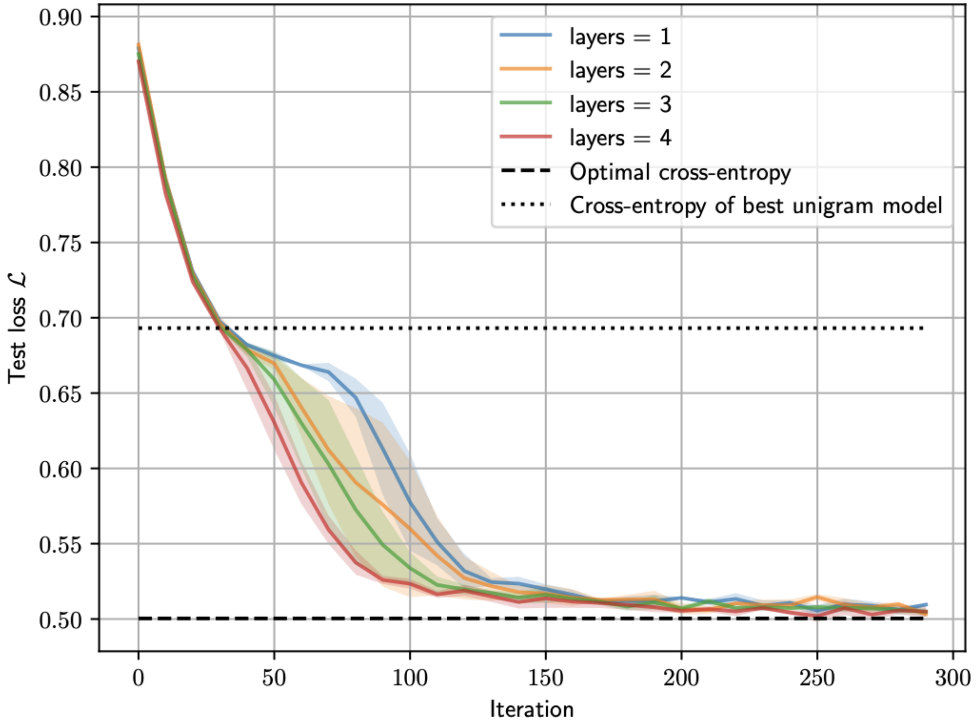
Ergodicity.

probability distribution of **Markov chain converges to stationary distribution**.

Importance of Tokenization in Transformer

Importance of tokenization in transformers





Data from simple Markov process (as previous slide).

Left : transformers **without tokenization** **fail to converge to optimal CE loss**.

Right : **with tokenization**, the test loss **achieves the optimal bound**.

Barrier of unigram model

Theorem 2.1 *Consider any ergodic data source with stationary distribution over characters x. The unconstrained optimal likelihood model achieves cross-entropy loss,* $\min_{Q} \mathcal{L}_m(Q) = H(P)$*. In contrast, the cross-entropy loss under any unigram model*

$Q \in \mathcal{Q}_{1-\mathrm{gram}}$ *satisfies* $\mathcal{L}_m(Q) \geq mH(\pi)$.

Barrier of unigram model

Theorem 2.1 *Consider any ergodic data source with stationary distribution over characters x. The unconstrained optimal likelihood model achieves cross-entropy loss,* $\min_{Q} \mathcal{L}_m(Q) = H(P)$*. In contrast, the cross-entropy loss under any unigram model*

$$Q \in \mathcal{Q}_{1-\mathrm{gram}} \text{ satisfies } \mathcal{L}_m(Q) \geq mH(\pi).$$

Proof)

CE loss on **unconstrained optimal likelihood model**.

$$\mathcal{L}_m(Q) = \mathbb{E}\left[\log\left(\frac{1}{Q(s)}\right)\right] = \mathbb{E}\left[\log\left(\frac{1}{P(s)}\right)\right] + \mathbb{E}\left[\log\left(\frac{P(s)}{Q(s)}\right)\right] = H(P) + D_{\mathrm{KL}}(P||Q)$$

If we choose Q same as P (the true prob. of s) then we have,

$$\mathcal{L}_m(Q) = H(P).$$

Barrier of unigram model

Proof)

CE loss on any **unigram model**.

Decompose CE loss on unigram model

$$\frac{1}{m}\mathcal{L}_m\big(Q\circ \mathrm{enc}(\cdot)\big) = -\frac{1}{m}\mathbb{E}[\log Q_{\#}(m)] - \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}[\log Q_{\mathrm{tok}}(t_i)]$$

Due to ergodicity of the source,

$$-\frac{1}{m}\sum_{i=1}^{m}\mathbb{E}[\log Q_{\mathrm{tok}}(t_i)] = -\sum_{a\in\mathcal{A}}\pi(a)\log Q_{\mathrm{tok}}(a)$$

Barrier of unigram model

Proof)

Utilizing upper bound of length prob's entropy,

$$\frac{1}{m}\mathbb{E}[\log Q_{\#}(m)] \leq 0$$

Barrier of unigram model

Proof)

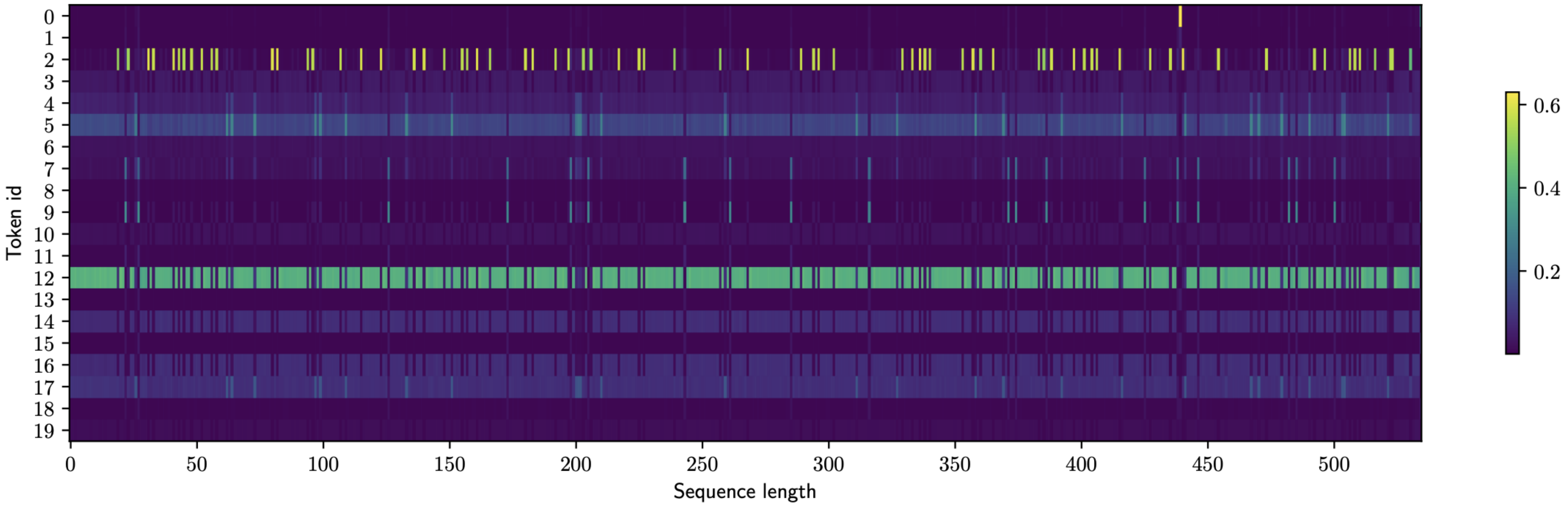
Utilizing upper bound of length prob's entropy,

$$\frac{1}{m}\mathbb{E}[\log Q_{\#}(m)] \leq 0$$

$$\begin{aligned}\frac{1}{m}\mathcal{L}_m\big(Q \circ \mathrm{enc}(\cdot)\big) &= -\frac{1}{m}\mathbb{E}[\log Q_{\#}(m)] - \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}[\log Q_{\mathrm{tok}}(t_i)] \\ &\geq -\sum_{a\in\mathcal{A}}\pi(a)\log Q_{\mathrm{tok}}(a) \\ &\geq H(\pi)\end{aligned}$$

$$\therefore Q \in \mathcal{Q}_{1-\mathrm{gram}} \textit{ satisfies } \mathcal{L}_m(Q) \geq mH(\pi).$$

Transformer learns a unigram model

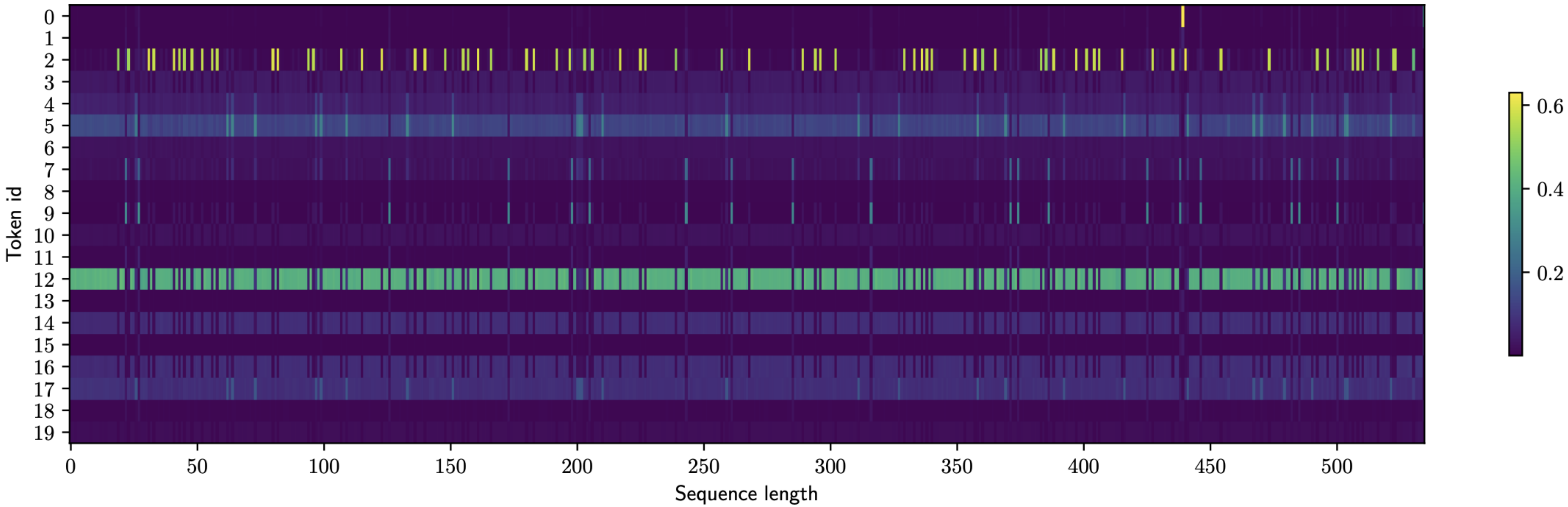


sample random sequence from **Markov chain**, feed it into a transformer **after tokenization**.

plot the **next token distribution predicted by the transformer** at every single point.

Plot is **approximately homogenous** along x-axis → not depend strongly on the prefix.

Transformer learns a unigram model



If the transformer learns **unigram model in both cases (w & w/o tokenization),** why there is a such **large gap between two?**

Unigram Model under Tokenization

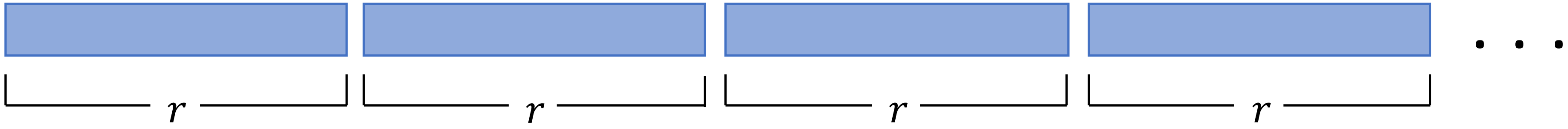
Unigram model under tokenization

Q. What happens when a unigram model is trained on the tokenized sequences?

Let's consider a toy tokenizer!

- all possible **substrings of length r** as tokens in the dictionary.
- total dictionary size $d = 2^r$. (2 state switching process)

e.g.

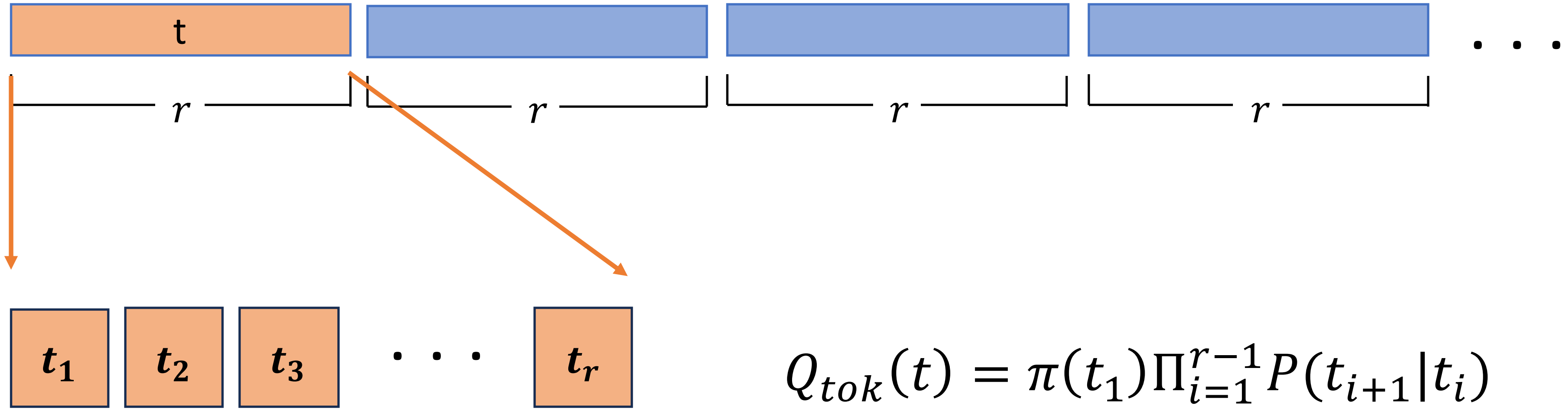


Unigram model under tokenization

$$\mathcal{L}_m\big(Q \circ \mathrm{enc}(\cdot)\big) = -\mathbb{E}\Big[Q_\#(m) \prod_{i=1}^{j} Q_{tok}(t_i)\Big] .$$

$$= \mathbb{E}\Big[\sum_{t\in\mathrm{enc}(s)} \log(1/Q_{tok}(t))\Big] + \cancel{\Theta(\log(m))}.$$

∵ We choose $Q_\# = \mathrm{Unif}([m])$ → additive $\log(m)$ term to the loss.



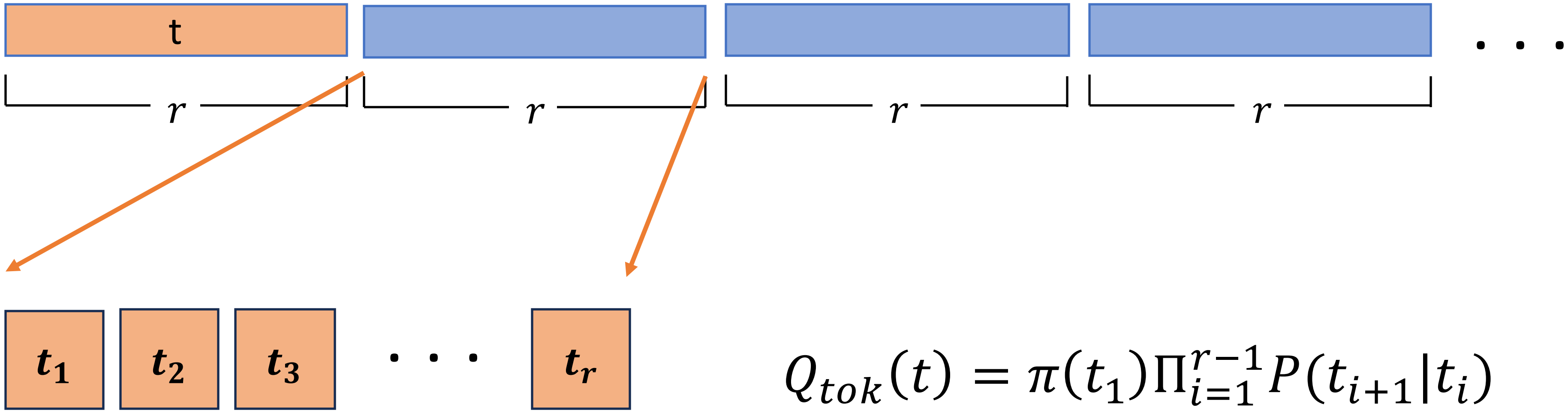
$$Q_{tok}(t) = \pi(t_1)\Pi_{i=1}^{r-1} P(t_{i+1}|t_i)$$

Unigram model under tokenization

$$\mathcal{L}_m\big(Q \circ \mathrm{enc}(\cdot)\big) = -\mathbb{E}\big[Q_\#(m) \prod_{i=1}^{j} Q_{tok}(t_i)\big]\,.$$

$$= \ \mathbb{E}\big[\textstyle\sum_{t \in \mathrm{enc}(s)} \log(1/Q_{tok}(t))\big] + \cancel{\Theta(\log(m))}.$$

∵ We choose $Q_\# = \mathrm{Unif}([m])$ → additive $\log(m)$ term to the loss.



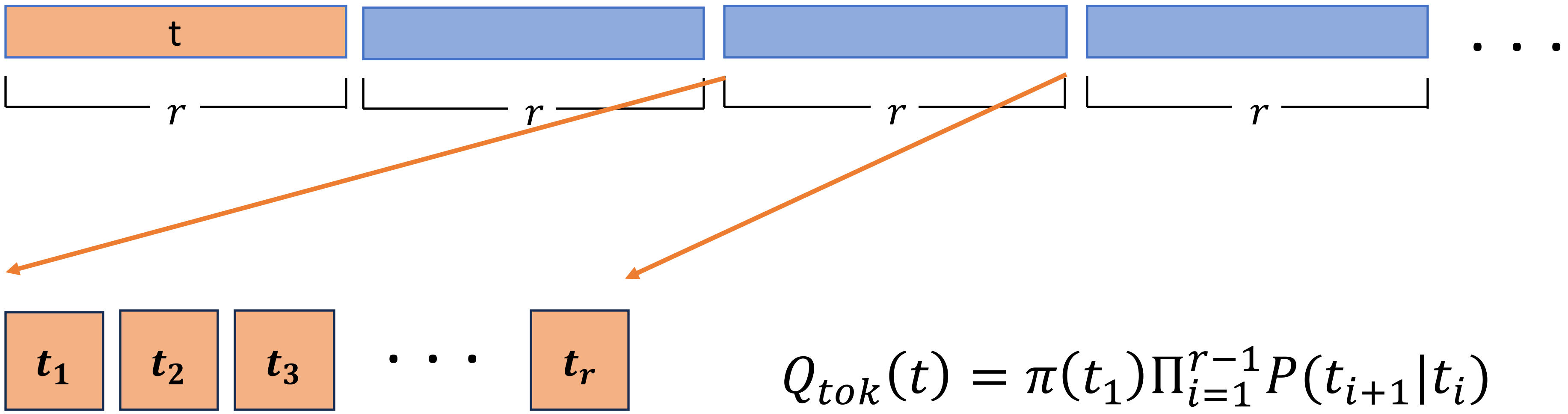
$$Q_{tok}(t) = \pi(t_1)\Pi_{i=1}^{r-1} P(t_{i+1}|t_i)$$

Unigram model under tokenization

$$\mathcal{L}_m\big(Q \circ \mathrm{enc}(\cdot)\big) = -\mathbb{E}\big[Q_{\#}(m)\prod_{i=1}^{j} Q_{tok}(t_i)\big]\,.$$

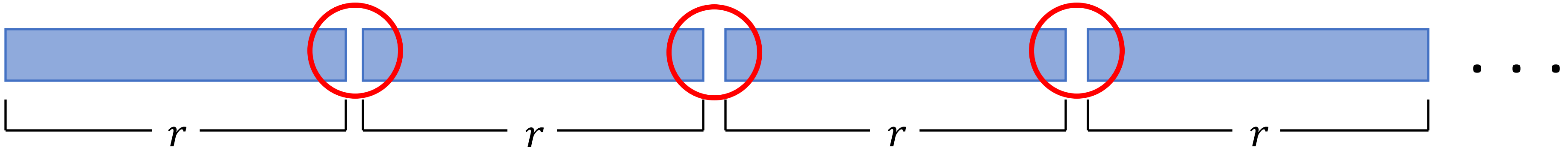
$$= \mathbb{E}\big[\textstyle\sum_{t\in\mathrm{enc}(s)} \log(1/Q_{tok}(t))\big] + \cancel{\Theta(\log(m))}.$$

∵ We choose $Q_{\#} = \mathrm{Unif}([m])$ → additive $\log(m)$ term to the loss.



$$Q_{tok}(t) = \pi(t_1)\Pi_{i=1}^{r-1}P(t_{i+1}|t_i)$$

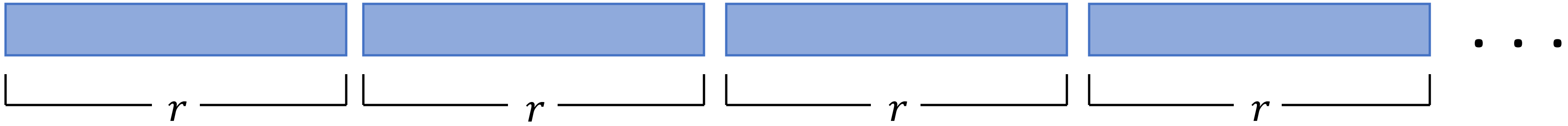
Unigram model under tokenization



$$\frac{1}{m}\mathcal{L}_m\big(Q \circ \mathrm{enc}(\cdot)\big) \approx -\frac{1}{m}\mathbb{E}\left[\sum\nolimits_{t\in\mathrm{enc}(s)} \log(1/Q_{tok}(t))\right]$$

$$= -\frac{1}{m}\mathbb{E}\left[\log P(s) + \sum\nolimits_{i=0}^{m/k-1} \log\left(\frac{\pi(s_{ki+1})}{P(s_{ki+1}|s_{ki})}\right)\right]$$

Unigram model under tokenization



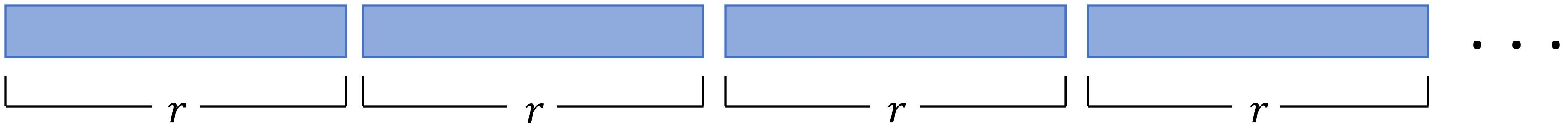
$$\frac{1}{m}\mathcal{L}_m\big(Q \circ \mathrm{enc}(\cdot)\big) \approx -\frac{1}{m}\mathbb{E}\left[\log P(s) + \Sigma_{i=0}^{m/k-1} \log\left(\frac{\pi(s_{ki+1})}{P(s_{ki+1}|s_{ki})}\right)\right]$$

As m grows large,

$\Sigma_{i=0}^{m/k-1} \log\big(\pi(s_{ki+1})\big) = \frac{m}{k} H(\pi).$

$\Sigma_{i=0}^{m/k-1} \log\big(P(s_{ki+1}|s_{ki})\big) = \frac{1}{k} H(P).$

Unigram model under tokenization



$$
\begin{aligned}
\frac{1}{m}\mathcal{L}_m\big(Q\circ \mathrm{enc}(\cdot)\big) &\approx -\frac{1}{m}\mathbb{E}\left[\log P(s) + \Sigma_{i=0}^{m/k-1}\log\left(\frac{\pi(s_{ki+1})}{P(s_{ki+1}|s_{ki})}\right)\right] \\
&\approx \frac{1}{m}H(P) + \frac{1}{mk}\big(mH(\pi) - H(P)\big) \\
&\approx \frac{H(P)}{m}\left(1 - \frac{1}{\log_2 d}\right) + \frac{H(\pi)}{\log_2 d}
\end{aligned}
$$

With d=2 (i.e. r=1), recover the performance of character tokenizer in Theorem 2.1.

As $m \to \infty$, there is a **unigram model which is nearly optimal** as $\mathrm{d} \to \infty$.

Unigram model under tokenization

$$\frac{1}{m}\mathcal{L}_m\big(Q \circ \mathrm{enc}(\cdot)\big) \approx \frac{H(P)}{m}\left(1 - \frac{1}{\log_2 d}\right) + \frac{H(\pi)}{\log_2 d}$$

Obvious issues.

To get a CE loss of $2\ H(P)$,

size of dictionary required by toy tokenizer = $e^{mH(P)/H(\pi)}$

For switching Markov process with $p = q = \delta \to 0$,

size of dictionary required by toy tokenizer $\propto e^{1/\delta \log(1/\delta)}$

On a much larger alphabet (e.g. English/ASCII),

Toy tokenizer results in a prohibitively **large dictionary**!

CE Loss under Greedy Encoder

Main Theorem

Theorem 3.1 *Consider a Markov data generating process which satisfies Assumption 3.2. Let* d *denote a budget on the size of the dictionary. Then, there exists a tokenizer with at most* d *tokens and encoding function* $\mathrm{enc}(\cdot)$ *such that,*

$$\min_{Q \in \mathcal{Q}_{1-\mathrm{gram}}} \mathcal{L}\big(Q \circ \mathrm{enc}(\cdot)\big) \leq \frac{1}{1-\varepsilon} \min_{Q} \mathcal{L}(Q)$$

where ε *is* $\frac{\log\left(\frac{1}{\delta}\right)}{0.99 \log d}$. *Furthermore, a tokenizer satisfying equation above with probability* $\geq 1 - d^{-\Omega_{\delta}(\log(d))}$ *can be learnt from a dataset of* $\widetilde{O_{\delta}}(d)$ *characters.*

Main Theorem

Assumption 3.2 (Data generating process). *Assume that the data source is an ergodic Markov process with transition $P(\cdot \mid \cdot)$ and stationary distribution π. Assume that* $\min_{a,a'} P(a'|a) \triangleq \delta > 0$.

Bound on CE loss under greedy encoder

Theorem 3.4 (Bound on cross-entropy loss of dictionaries under greedy encoder). *Consider a source satisfying Assumption 3.2 and any tokenizer* $\mathcal{T}$ *equipped with the greedy encoder,* $enc_{gre}(\cdot)$ *with finitely long tokens. Define,* $P(t) = \mathbb{E}_{a\sim\pi}[P(t|a)]$ *and suppose* $H(Q_{MLE}, P) \geq \frac{1}{\varepsilon}\log(\frac{1}{\delta})$ *for some* $\varepsilon < 1$. *Then,*

$$\min_{Q\in\mathcal{Q}_{1-\mathrm{gram}}} \mathcal{L}\left(Q \circ \mathrm{enc}_{\mathrm{gre}}(\cdot)\right) \leq \frac{\min_{Q}\mathcal{L}(Q)}{1-\varepsilon}.$$

Proof of Theorem 3.4

Step 1. Relate greedy encoding length to the source entropy.

Assume the token sequence follows **Assumption 3.2**, decomposing $P(s)$,

$$P(s) = P(t_1) \prod_{i=2}^{|\mathrm{enc}_{\mathrm{gre}}(s)|} P(t_i \mid t_{i-1}) \leq \prod_{i=1}^{|\mathrm{enc}_{\mathrm{gre}}(s)|} \max_{a \in \mathcal{A}} P(t_i \mid a)$$

Proof of Theorem 3.4

Step 1. Relate greedy encoding length to the source entropy.

$$P(s) \leq \prod_{i=1}^{|\mathrm{enc_{gre}}(s)|} \max_{a \in \mathcal{A}} P(t_i \mid a)$$

Then replace $\max_{a \in \mathcal{A}} P(t_i \mid a)$ term using ,

$$\max_{a} P(\boldsymbol{t} \mid a) \leq \frac{1}{\delta} \min_{a} P(\boldsymbol{t} \mid a) \leq \frac{1}{\delta} \mathbb{E}_{a \sim \pi}[P(\boldsymbol{t} \mid a)] = \frac{P(\boldsymbol{t})}{\delta}$$

Therefore,

$$P(s) \leq \prod_{i=1}^{|\mathrm{enc_{gre}}(s)|} \frac{P(t_i)}{\delta}$$

Proof of Theorem 3.4

Step 1. Relate greedy encoding length to the source entropy.

By the Asymtotic Equipartition Property (AEP)*,

$$\Pr\left(\lim_{m\to\infty} -\frac{1}{m}\log P(s) = H_\infty\right) = 1\,.$$

Therefore,

$$\lim_{m\to\infty}\frac{1}{m}\sum_{i=1}^{|\mathrm{enc}_{\mathrm{gre}}(s)|} -\log\big(P(t_i)\big) - \log\left(\frac{1}{\delta}\right) \le\; H_\infty\ (\mathrm{a.s.})\,.$$

* Looking at just one very long string, the average log probability of that string converges to the entropy rate.

Proof of Theorem 3.4

Step 1. Relate greedy encoding length to the source entropy.

From previous slide, we get,

$$\lim_{m\to\infty}\frac{1}{m}\sum_{i=1}^{|\mathrm{enc}_{\mathrm{gre}}(s)|} -\log\big(P(t_i)\big)-\log\left(\frac{1}{\delta}\right)\le\ H_\infty\ (\mathrm{a.s.})\,.$$

Greedy encoder satisfies the lemma below,

Lemma A.4 $\lim_{m\to\ \infty}\frac{n_t}{\sum_{t'} n_{t'}}\ =\lim_{m\to\infty}\ \mathbb{E}[\frac{n_t}{\sum_{t'} n_{t'}}]$ *for any tokenizer having a finite vocabulary and finitely long tokens, using the greedy encoder.*

and also satisfies,

$$t\ \in \mathrm{Dict},\qquad \lim_{m\to\infty}\frac{n_t}{|\mathrm{enc}_{\mathrm{gre}}(s)|}=Q_{\mathrm{MLE}}(t)\ (\mathrm{a.s.}).$$

Proof of Theorem 3.4

Step 1. Relate greedy encoding length to the source entropy.

Rewriting the sequence-level sum in terms of **token frequencies**,

$$\lim_{m\to\infty} \frac{\left|\mathrm{enc}_{\mathrm{gre}}(s)\right|}{m} \sum_{t\in\mathrm{Dict}} \frac{n_t}{\left|\mathrm{enc}_{\mathrm{gre}}(s)\right|} \left(-\log\bigl(P(t)\bigr) - \log\left(\frac{1}{\delta}\right)\right).$$

Where n_t is the number token t occurred in sequence s.

Using

$$H(Q_{\mathrm{MLE}}, P) = \sum_{t\in\mathcal{D}} Q_{\mathrm{MLE}}(t) \log \frac{1}{P(t)} \,,$$

Therefore,

$$\lim_{m\to\infty} \frac{\left|\mathrm{enc}_{\mathrm{gre}}(s)\right|}{m} \left(H(Q_{\mathrm{MLE}}, P) - \log\left(\frac{1}{\delta}\right)\right).$$

Proof of Theorem 3.4

Step 1. Relate greedy encoding length to the source entropy.

Utilizing the assumption from theorem 3.4,

$$H(Q_{\mathrm{MLE}}, P) \geq \frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right).$$

Therefore,

$$\lim_{m\to\infty}\frac{(1-\varepsilon)\left|\mathrm{enc}_{\mathrm{gre}}(s)\right|H(Q_{\mathrm{MLE}}, P)}{m} \leq H_\infty\ (\mathrm{a.s.}).$$

$$\lim_{m\to\infty}\frac{\left|\mathrm{enc}_{\mathrm{gre}}(s)\right|H(Q_{\mathrm{MLE}}, P)}{m} \leq \frac{H_\infty}{(1-\varepsilon)}\ (\mathrm{a.s.}).$$

Proof of Theorem 3.4

Step 2. Bound the expected CE loss of the tokenizer.

Define unigram model P_π,

$$P_\pi = \underbrace{P_{\text{unif}}(j)}_{\text{Length Prob.}} \prod_{i=1}^{j} P(t_i)\,,$$

$$\min_{Q \in \mathcal{Q}_{1\text{-gram}}} \lim_{m\to\infty} \frac{1}{m}\, \mathcal{L}_m\left(Q \circ \text{enc}_{\text{gre}}(\cdot)\right) \leq \lim_{m\to\infty} \frac{1}{m}\, \mathcal{L}_m\left(P_\pi \circ \text{enc}_{\text{gre}}(\cdot)\right).$$

Focusing on RHS,

$$\mathcal{L}_m\left(P_\pi \circ \text{enc}_{\text{gre}}(\cdot)\right) = -\mathbb{E}\left[\log P_{\text{unif}}\left(\left|\text{enc}_{\text{gre}}(s)\right|\right)\right] - \mathbb{E}\left[\sum_{t \in \text{enc}_{\text{gre}}(s)} \log\left(P(t)\right)\right]$$

$$\leq \log(m) - \mathbb{E}\left[\sum_{t \in \text{enc}_{\text{gre}}(s)} \log\left(P(t)\right)\right].$$

Proof of Theorem 3.4

Step 2. Bound the expected CE loss of the tokenizer.

For uniform distribution probability term,

$$As\ m \to \infty\,, \mathrm{then}\ \frac{\log(m)}{m} \to 0.$$

Again, rewrite the sequence-level sum in terms of token frequencies,

$$\begin{aligned}
\frac{1}{m}\mathcal{L}_m\left(P_\pi \circ \mathrm{enc}_{\mathrm{gre}}(\cdot)\right) &\leq \lim_{m\to\infty}\left(\cancel{\frac{1}{\mathrm{m}}\log(m)} - \frac{1}{m}\mathbb{E}\left[\sum\nolimits_{t\in \mathrm{enc}_{\mathrm{gre}}(s)}\log\big(P(t)\big)\right]\right) \\
&= -\lim_{m\to\infty}\frac{1}{m}\mathbb{E}\left[\sum\nolimits_{t\in \mathrm{enc}_{\mathrm{gre}}(s)}\log\big(P(t)\big)\right] \\
&= \lim_{m\to\infty}\frac{1}{m}\mathbb{E}\left[\left|\mathrm{enc}_{\mathrm{gre}}(s)\right|\sum\nolimits_{t\in \mathrm{Dict}} Q_{\mathrm{MLE}}(t)\log\big(1/P(t)\big)\right] \\
&= \lim_{m\to\infty}\frac{1}{m}\mathbb{E}\left[\left|\mathrm{enc}_{\mathrm{gre}}(s)\right| H(Q_{\mathrm{MLE}}, P)\right].
\end{aligned}$$

Proof of Theorem 3.4

As a result from step 1, and step 2,

$$\min_{Q \in \mathcal{Q}_{1-\mathrm{gram}}} \lim_{m \to \infty} \frac{1}{m} \mathcal{L}_m\left(Q \circ \mathrm{enc}_{\mathrm{gre}}(\cdot)\right) \leq \lim_{m \to \infty} \frac{1}{m} \mathbb{E}\left[\left|\mathrm{enc}_{\mathrm{gre}}(s)\right| H(Q_{\mathrm{MLE}}, P)\right] \leq \frac{H_\infty}{1-\varepsilon},$$

and with ergodic source, infinite data, and optimal Q,

$$H_\infty = \min_{Q} \lim_{m \to \infty} \frac{1}{m} \mathcal{L}_m(Q \circ \mathrm{enc}(\cdot)) = \min_{Q} \mathcal{L}(Q),$$

Therefore,

$$\min_{Q \in \mathcal{Q}_{1-\mathrm{gram}}} \mathcal{L}\left(Q \circ \mathrm{enc}_{\mathrm{gre}}(\cdot)\right) \leq \frac{\min_{Q} \mathcal{L}(Q)}{1-\varepsilon}$$

Expansion to LZW Tokenizer

LZW tokenizer

Theorem 3.6 *Suppose the LZW tokenizer is trained on a dataset of length at most for* d *(thereby learning a dictionary with at most* d *tokens). For Markov sources satisfying Assumption 3.2, with probability* $\geq 1 - d^{-\Omega_\delta(\log(d))}$*, the resulting tokenizer satisfies,*

$$\min_{Q \in \mathcal{Q}_{1-\mathrm{gram}}} \mathcal{L}\left(Q \circ \mathrm{enc}_{\mathrm{gre}}(\cdot)\right) \leq \frac{\min_{Q} \mathcal{L}(Q)}{1 - \varepsilon}.$$

Where $\varepsilon = \frac{\log\left(\frac{1}{\delta}\right)}{0.99 \log d}$.

Steps for proving Theorem 3.6

1. Heavy-hitting dictionaries ensures large $H(Q_{\mathrm{MLE}}, P)$.
2. Heavy-hitting substrings have bounded length.
3. LZW learns all heavy-hitting tokens with high probability.

LZW tokenizer

What is LZW tokenizer?

> **Definition 3.5** (*LZW tokenizer*). *Iterating from left to right, the shortest prefix of the training dataset which does not already exist as a token is assigned as the next token in the dictionary. This substring is removed and the process is iterated on the remainder of the dataset. The tokenizer uses the greedy encoding algorithm to encode new strings into tokens.*

e.g. 0100111 → created dictionary : {0, 1, 00, 11}

Proof of Theorem 3.6

Step 1. Heavy-hitting dictionaries ensures large $H(Q_{\mathrm{MLE}}, P)$.

What is heavy-hitting dictionary?

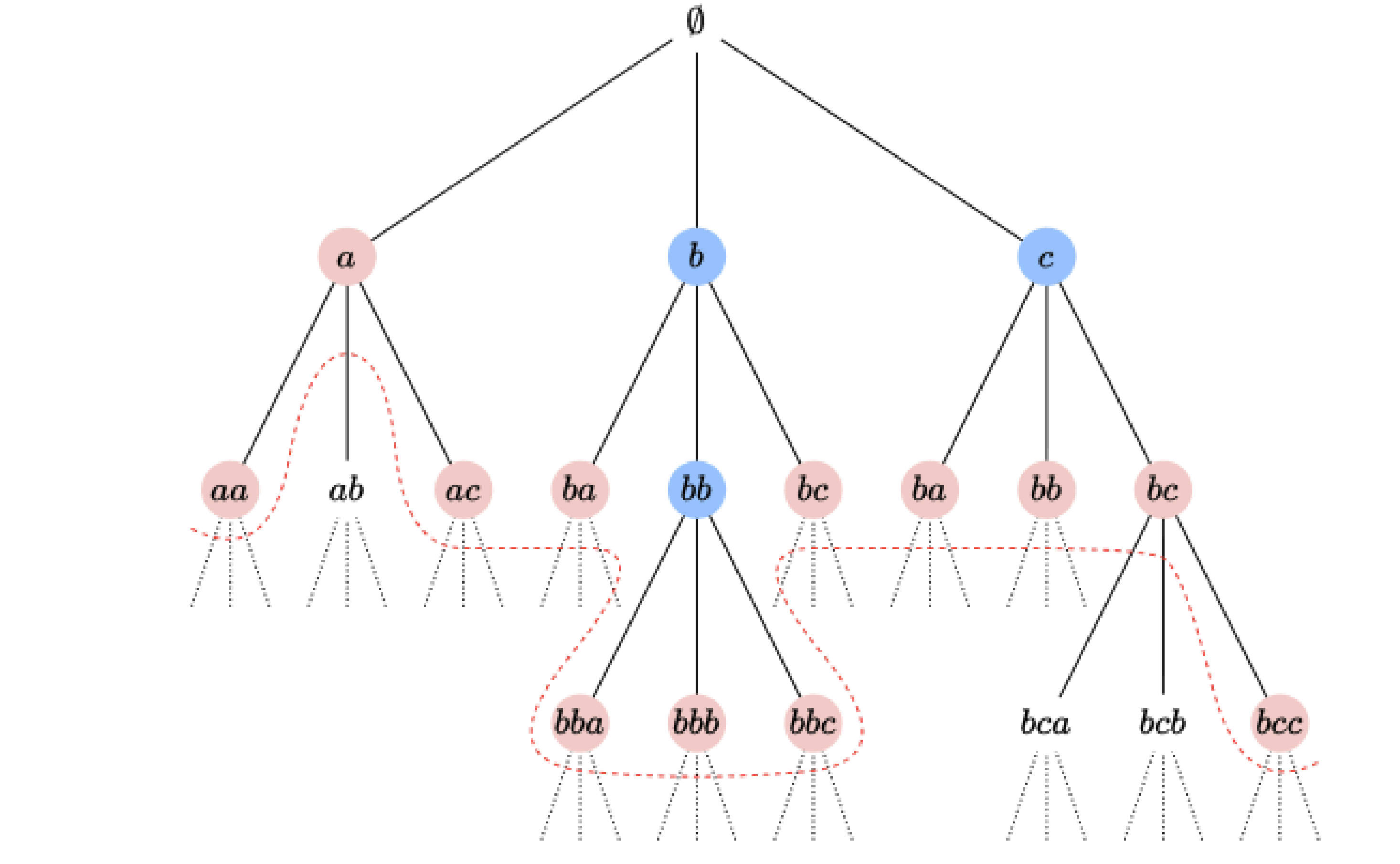
Definition A.5 (β-heavy-hitting dictionary) *A token* $\boldsymbol{t}$ *of a dictionary is said to be maximal if there exists an arbitrary substring containing* $\boldsymbol{t}$ *as a strict prefix, and in addition,* $\boldsymbol{t}$ *is also the largest prefix of the substring which is a token. A dictionary* Dict *is said to be* β*-heavy hitting if the set of maximal tokens is a subset of* $\left\{s' : \max_{a \in \mathcal{A}} P(s'|a) \leq \frac{1}{d^{\beta}}\right\}$.

➢ β-heavy-hitting dictionaries include substrings with probability $\geq \frac{1}{d^{\beta}}$.

Proof of The

Step 1. Heavy-h

What is heavy-h



Blue nodes: tokens included in the dictionary (not selected)

Red nodes: maximal tokens actually selected by the greedy encoder

Definition A.5 … *said to be maximal*
if there exists … *in addition,* ***t*** *is*
also the larges … *ct is said to be β-*
heavy hitting i … $) \le \frac{1}{d^{\beta}}\}.$

➢ β-heavy-hitting dictionaries include substrings with probability $\ge \frac{1}{d^{\beta}}$.

Proof of Theorem 3.6

Step 1. Heavy-hitting dictionaries ensures large $H(Q_{\mathrm{MLE}}, P)$.

Lemma A.6 *For a* β*-heavy-hitting dictionary, with the greedy encoder,* $H(Q_{MLE}, P) \geq \beta \log(d)$

Proof)

By the heavy-hitting property, every maximal substring has $\max_{a \in \mathcal{A}} P(s'|a) \leq \frac{1}{d^\beta}$.

$$P(\boldsymbol{t}) \leq \max_{a \in \mathcal{A}} P(s'|a) \leq \frac{1}{d^\beta}$$

$$\log\left(\frac{1}{P(\boldsymbol{t})}\right) \geq \beta \log d$$

$$H(Q_{MLE}, P) = \mathbb{E}_{t \sim Q_{MLE}}\left[\log\left(\frac{1}{P(\boldsymbol{t})}\right)\right] \geq \beta \log d$$

Proof of Theorem 3.6

Step 1. Heavy-hitting dictionaries ensures large $H(Q_{\mathrm{MLE}}, P)$.

Under a β-heavy-hitting dictionary, the induced cross-entropy satisfies

$$H(Q_{MLE}, P) \geq \beta \log d \ .$$

Therefore, to meet the (ε, δ) guarantee of Theorem 3.4 ($H(Q_{MLE}, P) \geq \frac{1}{\varepsilon} \log\left(\frac{1}{\delta}\right)$),

The lower bound of ε is

$$\varepsilon \geq \frac{\log\left(\frac{1}{\delta}\right)}{\beta \log d}.$$

Proof of Theorem 3.6

Step 2. Heavy-hitting substrings have bounded length.

Lemma A.7 *Every sub string in* $\mathcal{M}_\beta$ *has length at most* $\ell_* \triangleq \left(\beta \log(d) + \log\left(\frac{1}{\delta}\right)\right)$.

Proof)

Define $\mathcal{M}_\beta = \{\boldsymbol{t} : \max_{a \in \mathcal{A}} P(\boldsymbol{t} \mid a) \geq \frac{\delta}{d^\beta}\}$. → the set of "high-probability" substrings under the stochastic source.

From assumption 3.2, $\min_{a,a'} P(a'|a) = \delta \rightarrow \max_{a,a'} P(a'|a) \leq 1 - \delta$.

For any substring $\boldsymbol{t}$, the upper bound of probability is,

$$\max_{a \in \mathcal{A}} P(\boldsymbol{t}|a) \leq (1-\delta)^{|t|} \leq e^{-\delta|t|}$$

$$\therefore \frac{\delta}{d^\beta} \leq \max_{a \in \mathcal{A}} P(\boldsymbol{t}|a) \leq e^{-\delta|t|} \rightarrow |\mathbf{t}| \leq \beta \log(d) + \log\left(\frac{1}{\delta}\right)$$

Proof of Theorem 3.6

Step 2. Heavy-hitting substrings have bounded length.

Lemma A.8 *With probability* $\geq 1 - d^{-\Omega\left(\frac{\log(d/\delta)}{\delta}\right)}$*, in a run of the LZW algorithm, no substring* $\boldsymbol{t}$ *added as a token to the dictionary satisfies* $|\boldsymbol{t}| \geq \ell_{\max} \triangleq 4\frac{\log(d|\mathcal{A}|)}{\delta}$.

Proof)

Choosing length s from dict size d

$$P(\boldsymbol{t}\ is\ token) \leq \binom{d}{s} \prod_{i=1}^{s} \max_{a \in \mathcal{A}} P(t_{1:i}|a)$$

Each of prefixes (e.g. $t_{1:2}$, $t_{1:s}$) must disjointly (=independently) appear at least once in the string.

Proof of Theorem 3.6

Step 2. Heavy-hitting substrings have bounded length.

Lemma A.8 *With probability* $\geq 1 - d^{-\Omega\left(\frac{\log(d/\delta)}{\delta}\right)}$*, in a run of the LZW algorithm, no substring* $\boldsymbol{t}$ *added as a token to the dictionary satisfies* $|\boldsymbol{t}| \geq \ell_{\max} \triangleq 4\frac{\log(d|\mathcal{A}|)}{\delta}$.

Proof)

$$P(\boldsymbol{t}\ is\ token) \leq \binom{d}{s}\prod_{i=1}^{s}\max_{a\in\mathcal{A}} P(t_{1:i}|a) \leq \binom{d}{s}\prod_{i=1}^{s}(1-\delta)^{i} \leq \binom{d}{s}(1-\delta)^{\frac{s(s-1)}{2}} \leq e^{s\log d - \frac{\delta s(s-1)}{2}}$$

$$P(length\ s\ string\ is\ token) \leq e^{slog(|\mathcal{A}|)\,+\,s\log d - \frac{\delta s(s-1)}{2}} = e^{slog(d|\mathcal{A}|) - \frac{\delta s(s-1)}{2}}$$

If quadratic term gets larger, the probability converges to zero.

Proof of Theorem 3.6

Step 2. Heavy-hitting substrings have bounded length.

Lemma A.8 *With probability* $\geq 1 - d^{-\Omega\left(\frac{\log(d/\delta)}{\delta}\right)}$*, in a run of the LZW algorithm, no substring* $\boldsymbol{t}$ *added as a token to the dictionary satisfies* $|\boldsymbol{t}| \geq \ell_{\max} \triangleq 4\frac{\log(d|\mathcal{A}|)}{\delta}$.

Proof)

For choosing maximum length s,

$$2slog(d|\mathcal{A}|) \leq \frac{\delta s(s-1)}{2} \rightarrow s = \frac{4\log(d|\mathcal{A}|)}{\delta} + 1 \triangleq \ell_{\max} + 1$$

Then the probability would be near zero.

$$e^{-\frac{\delta \ell_{\max}^2}{4}} \leq d^{-\Omega\left(\frac{\log\left(\frac{d}{\delta}\right)}{\delta}\right)}$$

Proof of Theorem 3.6

Step 3. LZW learns all heavy-hitting tokens with high probability.

Corollary A.9 *With probability* $1 - d^{-\Omega_\delta(\log(d))}$, *learns a dictionary with **at least*** $\mathrm{d}^* = \mathrm{d}/\ell_{\max}$ *tokens when run on a training sequence of length* n *drawn from a stochastic source satisfying Assumption 3.2.*

Lemma A.10 *For any constant* $\beta < 1$, *with prob* $\geq 1 - d^{-\Omega\left(\frac{\log(d/\delta)}{\delta}\right)} - \exp\left(-\widetilde{\Omega}_d\left(d^{1-\beta}\right)\right)$ *over the source dataset, every substring in* $\mathcal{M}_\beta$ *is added as a token to the dictionary in a run of the LZW algorithm. In other words, with the same probability, the LZW tokenizer results in a* β*-heavy hitting dictionary.*

- LZW tokenizer contains β-heavy substrings as token with high probability.

Proof of Theorem 3.6

Step 3. LZW learns all heavy-hitting tokens with high probability.

From corollary A.9 we now consider effective dictionary size d^*.

Running LZW, sampling d^* length of prefix and construct dictionary simultaneously.

Condition of substring $\boldsymbol{t}$ is not added as a token only if prefix of t appears less than $|t| - 1$.

Proof of Theorem 3.6

Step 3. LZW learns all heavy-hitting tokens with high probability.

From corollary A.9 we now consider effective dictionary size d^*.

Running LZW, sampling d^* length of prefix and construct dictionary simultaneously.

Condition of substring t is not added as a token only if prefix of t appears less than $|t| - 1$.

$$P(t\ not\ token) \leq \sum_{i=0}^{|t|-1} \binom{d^*}{i} x^i (1-x)^{d^*-i}, x = \max_{a \in \mathcal{A}} P(t|a)$$

Intuition: Expectation of substring t occurrence = $d^* x \gg |t| \rightarrow P(t\ not\ token) \approx 0$

Therefore, the probability of t not be tokenized is negligible, and LZW learns a β-heavy-hitting dictionary with high probability.

Proof of Theorem 3.6

- Wrap up.

Lemma A.6 shows heavy-hitting dictionaries imply large cross-entropy, and lower bound of $\varepsilon \geq \frac{\log\left(\frac{1}{\delta}\right)}{\beta \log d}$.

Proof of Theorem 3.6

- ➢Wrap up.

Lemma A.6 shows heavy-hitting dictionaries imply large cross-entropy, and lower bound of $\varepsilon \geq \frac{\log\left(\frac{1}{\delta}\right)}{\beta \log d}$.

Lemma A.7 & Lemma A.8: heavy-hitting substrings have maximum length to be included in token with probability $\geq\ 1 - d^{-\Omega_\delta(\log(d))}$.

Proof of Theorem 3.6

- Wrap up.

Lemma A.6 shows heavy-hitting dictionaries imply large cross-entropy, and lower bound of $\varepsilon \geq \frac{\log\left(\frac{1}{\delta}\right)}{\beta \log d}$.

Lemma A.7 & Lemma A.8: heavy-hitting substrings have maximum length to be included in token with probability $\geq\ 1 - d^{-\Omega_\delta(\log(d))}$.

Lemma A.10: LZW learns all heavy-hitting tokens with high probability.

Together, these results establish Theorem 3.6.

Conclusion

- **Theoretical framework** to analyze tokenization algorithms.

- **Unigram likelihood model with tokenization** approaches the CE loss of **the optimal likelihood model**, as the vocabulary size d grows.