Non-Asymptotic Convergence Bounds for Cross-Entropy Estimation between Neural Auto-Regressive Language Models: Theoretical Analysis

Anonymous ACL submission

Abstract

Cross-entropy (CE) represents a central metric in evaluating the performance and other characteristics of **N**eural **A**uto-**R**egressive **L**anguage **M**odels (**NARLMs**). Despite its importance, the convergence properties of its estimation remain relatively underexplored from a theoretical perspective, primarily due to the complex structure of modern language model architectures. This article aims at investigating this issue by providing a formal theoretical analysis of the covergence properties of the CE estimation between different families of NARLMs. When the test distribution is modeled by a LSTM/GRU, we will show that CE estimation exhibits a non-vacuous convergence rate, which depends linearly on the norm of the output matrix of the test model and logarithmically on the alphabet size. Additionaly, we provide a variance-based convergence bound applicable to large families of NARLM, including Decoder-only Transformer-based models and LSTMs/GRUs.

1 Introduction

Language modeling has become a foundational task in modern NLP. As such, having reliable tools to analyze the performance and properties of language models is of critical importance. The traditionally used metric for training and performance evaluation of LMs is the cross-entropy (CE). Beyond these applications, CE, and other closely related information-theoretic measures such the KL divereegence and the entropy rate, were instrumental for other LM-related purposes, including Grammatical Inference (Clark and Thollard, 2004), analyzing the calibration properties of language models (Braverman et al., 2020; Wei et al., 2024), Knowledge Distillation from larger "teacher" models to smaller ones (Liu et al., 2023, 2024), and distinguishing between human-generated and machine-generated text (Varshney et al., 2020).

However, despite its importance, convergence properties of CE estimators remains relatively underexplored for Neural LMs. This gap can be partly attributed to the highly complex structure of modern Neural LMs' architectures, not easily amenable to theoretical analysis through the lens of formal statistical theory. As a result, current practices for CE estimation remain predominantly empirical, lacking theoretical insights into the qualitative properties of its approximation, and the sample size required to obtain a reliable estimate of its exact value.

In contrast to Neural models, complexity-theoretic studies for comparing language models (LMs) has been thoroughly conducted for various classes of probabilistic automata, the predecessors of neural LMs. Carrasco (1997) introduced an iterative procedure to compute exactly the CE between two deterministic probabilistic finite automata (DPFA). This work was extended by Cortes et al. (2006), who provided a detailed complexity analysis for computing this metric between unambiguous probabilistic automata, a class of models that strictly includes DPFAs. Other related works have explored alternative metrics for comparing probabilistic automata such as the L_2 distance (Murgue and de La Higuera, 2004), the total variation distance (Lyngsø and Pedersen, 2002), and the general family of L_p distances (Cortes et al., 2007).

This article aims to address a gap in the literature concerning the complexity analysis of CE when applied to Neural Auto-Regressive LMs (NARLMs). Due to the imtractability of exact CE computation between stochastic languages generated by these models, our investigation centers around two principal inquiries, one theoretical and the other empirical. Firstly, we aim to explore the feasibility of establishing non-vacuous theoretical bounds on the token sample size complexity required for CE estimation in NARLMs. Secondly, should these

theoretical bounds be established, we seek to assess their practical applicability, specifically evaluating whether the token sample size complexity resulting from the theoretical bounds is reasonably small for practical use.

We shall focus on two widely used configurations of NARLMs: One that includes LMs equipped with the end-of-sequence (*eos*) token and one that doesn't. The former (resp. the latter) generates probability distributions over finite (resp. infinite) sequences. Section 2.1 will provide a formal definition of these families of stochastic languages, referred to as finite (resp. infinite) stochastic languages respectively.

The two computational problems under study in this article are formally defined as follows:

1. The *end-of-sequence* case: For two probability distributions P and Q with the *eos* token. Compute:

$$CE(P,Q) = \mathbb{E}_{w\sim P}\left[\log(\frac{1}{Q(w)}\right] \quad (1)$$

2. The *without end-of-sequence case*: For two probability distributions $P,\ Q$ without the *eos* token, and an integer $L > 0$. Compute:

$$CE_L(P,Q) = \frac{1}{L}\mathbb{E}_{w\sim P^{(L)}} \log\left[\frac{1}{Q^{(L)}(w)}\right] \quad (2)$$

where $P^{(L)}$ is a mapping that assigns to each sequence w of length L the probability of generating w as a prefix using P. Note that when $P = Q$, $CE_L(P,Q)$ is reduced to the entropy rate of P (Braverman et al., 2020).

In the sequel, we shall refer to languages P (resp. Q) as the target (resp. the test) distributions.

In both settings, the target distribution may be the empirical distribution that represents the underlying (unknown) distribution, which is often the case in practical applications. The (informal) question that this article aims at addressing can be framed as follows:

Question: *Given a target approximation error ϵ and a confidence interval $\delta \in (0,1)$, what is the number of tokens that needs to be generated from the target P in order to obtain an ϵ-approximate of the $CE(P,Q)$ (or, $CE_L(P,Q)$) with probability greater than $1-\delta$?*

We answer this question by providing two theoretically-backed convergence bounds: a Norm-dependent (section 3), and a Variance-dependent (section 4) bounds. While the former offers a non-vacuous, efficiently computable bound when the test distribution is generated by LSTMs/GRUs, the latter has the advantage to be general enough to cover also Decoder-only Transformer-based models (Radford et al., 2019).

2 Background

The symbol Σ is used to refer to a finite alphabet (also known as a vocabulary for readers acquainted with the NLP terminology), and the symbol \$ denotes a special symbol that marks the end of sequence. We shall denote the set $\Sigma \bigcup \{\$\}$ by the symbol $\Sigma_\$$. Σ^* (resp. Σ^∞) is the set of all finite (resp. infinite) sequences formed by Σ. For a given sequence $w \in \Sigma^*$, $|w|$ refers to its length. For an integer $n \in \mathbb{N}$, the symbol Σ^n [1] refers to the subset of sequences in Σ^* such that $|w| = n$. For an integer $N \geq 1$, $[N]$ refers to the set of integers $\{1,\ldots,N\}$, and $\mathcal{B}_N$ refers to the hypercube $[-1,1]^N \subset \mathbb{R}^N$. For $\epsilon \in (0,1)$, a quantity $\tilde{Q}$ is said to be ϵ-approximate of Q if $|\tilde{Q} - Q| \leq \epsilon$.

For a vector $v \in \mathbb{R}^n$ and an integer $p \in [1,\infty]$, the p-norm of v, denoted $||v||_p$, is given as: $||v||_p = (\sum_{i=1}^n |v_i|^p)^{\frac{1}{p}}$. For $p = \infty$, $||v||_\infty$ is equal to $\max_{i\in[n]} |v_i|$. A generalization of the p-norm of vectors to matrices is the (p,q) norm: For a matrix $A \in \mathbb{R}^{n\times m}$, and a pair of integers $(p,q) \in [1,\infty]$, the (p,q)-norm of A, denoted $||A||_{p,q}$, is given as: $\|A\|_{p,q} = \left(\sum_{i=1}^n \left(\sum_{j=1}^m |A_{ij}|^p\right)^{\frac{q}{p}}\right)^{\frac{1}{q}}$. Analogous to vectors, when $q = \infty$, we have $||A||_{p,\infty} = \max_{i\in[n]} ||A_{i:}||_p$.

2.1 Stochastic Languages (SLs).

In general, a finite (resp. infinite) language refers to any mapping from Σ^* (resp. Σ^∞) to $\mathbb{R}$. However, the main focus on this article shall be on languages describing probability distributions over sequences, which we'll refer as *stochastic languages* (SLs). One can distinguish between two variants of SLs in the literature:

1. Finite stochastic languages (FSLs). Also known as SLs with end-of-sequence (Radford et al., 2019): This class of stochastic languages includes the set of languages that describes valid probability distributions over Σ^*. In practice, the implementation of FSLs by NARLMs is performed by intro-

[1] By convention, Σ^0 refers to the singleton set comprising the empty string.

ducing a special symbol, denoted *eos*, which marks the end of sequence generation.

2. Infinite stochastic languages (ISLs). An ISL is a stochastic language that describes a probability distributions over Σ^∞. In the context of Neural Auto-Regressive models, ISLs can be obtained by ruling out the *eos* token from the generative process. In the sequel, we shall favor an alternative formalization of this class of languages as provided in the following definition:

Definition 1. *An ISL P is represented by a sequence $\{P^{(m)}\}_{m\geq 1}$, where $P^{(m)}$ is a probability distribution over Σ^m for an integer $m \geq 1$.*

For an integer $m \geq 1$, the probability distribution $P^{(m)}$ over Σ^m in this definition can be interpreted as the probability of generating a prefix of length m during a forward run on a LM P.

In the remainder of this article, stochastic languages (SLs) will implicitly refer to the union of both FSLs and ISLs. The symbol Δ will denote either $\Sigma_\$$ or Σ, depending on the context.

The next two subsections are dedicated to provide two sub-families of SLs, namely smooth SLs and prefixial-bounded FSLs, whose pertinence in the context of this work will be highlighted in the next section.

2.2 Smooth SLs.

A class of SLs that will hold a particular importance in the theoretical analysis conducted in the next section is the class of *smooth stochastic languages* (smooth SLs). Formally, for $\gamma \in (0, 1)$, a stochastic language P is said to be γ-smooth if it satisfies the following condition:

$$\forall w \in \Sigma^* : \inf_{\sigma\in\Delta} P(\sigma|w) \geq \gamma \qquad (3)$$

A stochastic language P will be called *smooth* if it is γ-smooth for some $\gamma \in (0, 1)$.

Informally, a smooth language is a stochastic language for which there exists a lower bound on the next symbol probability distribution given that an arbitrary prefix has been generated. In the context of Neural language modeling, the softmax function plays a role in generating this smoothing effect for Neural LMs. However, as demonstrated in (Chen et al., 2018), its incorporation in the output layer is not a sufficient condition for the smoothness of the stochastic language generated by a neural language model. Later in this section, we shall introduce a sufficient technical condition on the neural architectures, satisfied by both LSTMs/GRUs and Decoder-only Transformer-based models, which imply the smoothness of the generated stochastic language.

2.3 Prefixial-bounded FSLs.

For a finite stochastic language P, we define the prefixial norm of P, denoted $||P||_p$, as:

$$||P||_p \stackrel{\text{def}}{=} \sum_{w\in\Sigma^*} P(w\Sigma^*)$$

FSLs with a finite prefixial norm will be referred to as prefixial-bounded. Prefixial-bounded FSLs admit an interesting characterization in terms of the properties of the random variable corresponding to the length of generated sequence, as highlighted by the following proposition:

Proposition 1. *Let P be a prefixial-bounded FSL. We have:*

$$||P||_p = \mathbb{E}_{w\sim P}[|w|] - 1$$

An immediate corollary of Proposition 1 is that the set of prefixial-norm finite state languages (FSLs) coincides with the set of FSLs for which the length of generated sequences admits a finite first-order moment.

We wrap up this discussion about families of stochastic languages by providing a lemma which establishes the inclusion relationship between these classes of languages. Specifically, smooth FSLs is stricly included within prefixial-bounded FSLs:

Lemma 1. *1. Inclusion: Any smooth FSL P is also prefixial-bounded.*

2. Strict Inclusion: There exists prefixial-bounded FSLs that are not smooth.

The proof can be found in appendix B.

2.4 Neural Auto-regressive Language Models (NARLMs).

In the following, we introduce a formal abstraction of Neural Auto-Regressive Language Models (NARLMs) that is both sufficiently comprehensive to encompass a wide range of NARLM variants, including LSTMs/GRUs and Transformer-based language models, and conducive to the theoretical analyses sought in this article:

Definition 2. *(NARLMs) A neural auto-regressive language model (NARLM) is defined by a tuple $M =< n, F, T, W >$ where $n \in \mathbb{N}, T > 0$, $F : \Sigma^* \to \mathbb{R}^n$, and $W \in \mathbb{R}^{|\Delta|\times n}$. For a given sequence*

$w \in \Sigma^*$, *the next token probability distribution* $P(.|w)$ *is computed as follows:*

$$P(.|w) = \mathit{softmax}_T\left(W^T \cdot F(w)\right)$$

where, for a vector $z = \begin{bmatrix} z_1 & \dots & z_n \end{bmatrix}^T \in \mathbb{R}^n$, *the softmax with temperature* $T0$ *is given by the formula:* $\mathit{softmax}_T(v) = \frac{\exp(\frac{z_i}{T})}{\sum_{i=1}^{n} \exp(\frac{z_i}{T})}$

By abstracting away the architectural specifics inherent to various families of NARLMs, the proposed abstraction in Definition 2 is agnostic to the internal mechanisms governing the model's operation. This renders it sufficiently generic to encompass a wide array of classical neural auto-regressive language models, including LSTMs/GRUs and Transformer LMs, as illustrated in the following examples:

• **LSTMs/GRUs in NARLM format:** As a Language Model, an LSTM/GRU M can be conceptualized by a tuple $< n, m, z_{init}, \{F_\sigma\}_{\sigma\in\Sigma}, T, W >$, where n and m represent the hidden and the cell state space dimensions respectively, $z_{init} = \begin{bmatrix} h_{init} & c_{init} \end{bmatrix}^T \in \mathbb{R}^{n+m}$ the initial state vector (formed by the concatenation of the initial hidden state vector h_{init} and the cell state vector c_{init}), $F_\sigma : \mathbb{R}^{n+m} \times \mathbb{R}^{n+m}$ is the transition state function associated to the symbol $\sigma \in \Delta$, T is the temperature parameter and $W \in \mathbb{R}^{n\times\Delta}$ is the output matrix.

A forward run of a LSTM/GRU $M =< n, m, z_{init}, \{F_\sigma\}_{\sigma\in\Sigma}, T, W >$ starting from a state $z = \begin{bmatrix} h & c \end{bmatrix}^T$ on the input $\sigma \in \Sigma$ is given as:

$$\begin{cases} \begin{bmatrix} h' & c' \end{bmatrix}^T & = F_\sigma\left(\begin{bmatrix} h & c \end{bmatrix}^T\right) \\ h & = \begin{bmatrix} \mathbf{I}_{n\times n} & \mathbf{O}_{n\times m} \end{bmatrix} \begin{bmatrix} h \\ c \end{bmatrix} \end{cases}$$

The next token probability distribution is then computed by applying the linear transformation W on the resulting vector h, followed by the Softmax with temperature whose value is equal to T.

A reparametrization of the LSTM/GRU M into a NARLM format is given by the tuple $< n, F, T, W >$, such that for a sequence $w \in \Sigma^*$, we have:

$$F(w) = \begin{bmatrix} \mathbf{I}_{n\times n} & \mathbf{O}_{n\times m} \end{bmatrix} \cdot (F_{w_{|w|}} \circ \ldots \circ F_{w_1})(z_{init})$$

• **Decoder-only Transformer LMs in NARLM format.** Unlike RNNs, Transformer-based language models process the input in a vertical manner through a series of Transformer blocks. This way of input processing imposes two architectural constraints on Transformer-based models: First, Transformer models can only process a bounded context to produce the next token probability distribution. As such, despite their apparent complexity, their expressiveness power is strictly limited to the class of n-gram models. Second, by contrast with RNNs, the order of the tokens in the sequence have to be encoded explicity through a position encoding scheme.

Next, we shall conduct an analogous treatment for Decoder-only Transformer models to LSTMs/GRUs developed earlier by reparametrizing them into a NARLM format. To this aim, we first introduce some notation: Denote by $K \geq 1$ the context width of a Transformer LM. Also, Denote the map $E_K : \Sigma^{\geq K} \to \Sigma^K \times \mathbb{N}$ as follows:

$$E_K(w) = (w_{|w|-K}..w_{|w|}, |w| - K)$$

for $w \in \Sigma^{\geq K}$. And the embedding map, Φ, given as

$$\begin{aligned} \Phi :& \Sigma^K \times \mathbb{N} \to \mathbb{R}^{d\times K} \\ & (w, l) \ \to U_w \cdot W_e + W_p \end{aligned}$$

where U_w is the one-hot representation of the context sequence w, W_e is the embedding matrix, W_p is the positional matrix that encodes the positions from l to $l+K$. The map Φ encodes implicitly both the input tokens and the positional information to inject in the Transformer to produce the next token probability distribution.

A conceptual representation in NARLM format of the processing of a sequence by a Decoder-only Transformer LMs can be given as follows []. Let w be a sequence in $\Sigma^{\geq K}$ [2]:

$$\begin{cases} F(w) & = T_{W_L} \circ .. \circ T_{W_1} \circ \Phi \circ E_K(w) \\ P(.|w) & = \text{softmax}_T(W^T \cdot F(w)) \end{cases}$$

where $\{T_{W_l}\}_{l\in[1,L]}$ is a collection of parametrized maps encompassing the Transformer's block operations.

2.4.1 Bounded NARLMs.

In this section, we build upon the NARLM abstraction by introducing one of its sub-families,

[2] To simplify, we only consider the case where the context width is equal to the full width, that is the first K tokens are assumed to be already generated, where K refers to the context window size.

namely bounded NARLMs. This family of models is defined by enforcing an additional constraint on NARLMs which will be crucial to establish the smoothness of stochastic languages generated by this latter. Formally, bounded NARLMs are defined as follows:

Definition 3. *(Bounded NARLMs) A NARLM* $M =< n, F, T, W >$ *is said to be bounded if there exists* $B_M > 0$ *such that:* $\sup_{w \in \Sigma^*} ||F(w)|| < B_M$

Informally, a bounded NARLM is a NARLM whose embedding space is uniformly bounded. We note that due to the equivalence of norms in finite-dimensional vector spaces (Kreyszig, 1989), the boundedness of a NARLM is independent of the choice of the norm.

The pertinence of this family of NARLMs in the context of our work is due to two facts.

First, LSTMs/GRUs and Decoder-only Transformer LMs models (under mild assumptions) are bounded NARLMs:

Proposition 2. *The following statements are true:*

1. *LSTMs/GRUs are bounded NARLMs,*
2. *Decoder-only Transfomer-based LMs such that:*
   (a) *There exists a constant* C_M *such that for any positional matrix, we have* $||W_p|| < C_M$*,*
   (b) *The Transformer's block mappings* $\{T_{W_l}\}_{l=1}^{l=L}$ *are continuous functions,*

   are bounded NARLMs.

Second, stochastic languages generated by NARLMs are smooth as shown in the following proposition:

Proposition 3. *Bounded NARLMs generate smooth SLs.*

As an immediate corollary of Propositions 2 and 3, it follows that SLs generated by LSTMs/GRUs and Decoder-only Transformer-based language models are smooth. This fact will be crucial in subsequent analysis. Additionally, these propositions reveal an interesting property of finite stochastic languages generated by LSTMs/GRUs and Transformer LMs concerning the characteristics of the lengths of sequences produced by these models.

Corollary 1. *The length of drawn sequences from FSLs generated by bounded NARLMs admits a finite first-order moment.*

Result of corollary 1 extends a finding established in (Welleck et al., 2020), which settles for demonstrating the consistency of FSLs generated by LSTMs/GRUs.

3 Non-asymptotic convergence bounds of CE approximation between NARLMs.

This section is dedicated to presenting the main theoretical results of the article, namely non-asymptotic convergence bounds for CE estimation for both FSLs and ISLs cases. The formal examination of both these cases share a common theoretical framework, which we shall examine simultaneously according to the following structure:

1. A model-agnostic bound assisted with an oracle: The initial phase consists at establishing model-agnostic convergence bounds of the CE between two arbitrary SLs, under the assumption of a smooth test distribution. These bounds will exhibit a dependency to the smoothness parameter γ of this latter(Lemma 2). The theoretical estimators of CE discussed in this section differ slightly from commonly used practical estimators and rely on two fundamental oracles: the `GEN` and the `POS` oracles, formally defined later in this segment.

2. A norm-dependent bound for CE estimation. This phase extends upon the theoretical findings established previously, focusing on deriving a non-asymptotic convergence bound specifically tailored to language models of interest. It particularly addresses the scenario where LSTMs/GRUs serve as the test distribution in the computation of the Cross-Entropy (CE). The token sample size complexity for CE estimation is given explicitly in terms of norms of the output matrix of the model and the temperature parameter of its softmax layer.

3.1 A Model-Agnostic bound assisted with an oracle.

Let $P,\ Q$ be two finite or infinite stochastic languages, where Q is assumed to be γ-smooth for some $\gamma \in (0, 1)$. The objective of this section is to design an estimator of the CE between P and Q that exhibits non-asymptotic convergence bounds which depend on the smoothness parameter of Q. We divide this section into two distinct parts: one addressing ISLs, and the other examining FSLs.

3.1.1 Infinite Stochastic Languages.

In the remainder of this segment, we fix $\gamma \in (0, 1)$, two arbitrary ISLs P and Q where Q is γ-smooth

(e.g. generated by a LSTM/GRUs or Decoder-only Transformer-based LMs), and an integer $L > 0$.

As mentioned previously, a key building block for the theoretical analysis of estimating $\text{CE}_L(P, Q)$ is an oracle GEN to which the designed approximation schema makes calls in order to obtain an approximation of $\text{CE}_L(P, Q)$. Formally the GEN(.,.) is defined as follows:

Definition 4. *The* GEN *oracle takes as input an ISL P, an integer $L > 0$, and outputs a pair $(w, \sigma) \in \Sigma^* \times \Sigma$ drawn from the following generative procedure:*

- *Draw uniformly at random an integer i in $[L-1]$,*
- *Sample a string $w = w' \cdot \sigma$ according to the probability distribution $P^{(i+1)}$ where $w' \in \Sigma^i$ and $\sigma \in \Sigma$,*

In the sequel, the notation $\text{GEN}(P, L)$ will be used to refer to the probability distribution over $\Sigma^* \times \Sigma$ induced by the generative procedure outlined in Definition 4.

The pertinence of this oracle in our context is highlighted by the following reformulation of the cross-entropy between two ISLs. Let P, Q be two such languages and an integer $L > 0$, we have:

$$\begin{aligned} CE_L(P, Q) &= \frac{1}{L}\mathbb{E}_{w\sim P^{(L)}}[\log\left(\frac{1}{Q^{(L)}(w)}\right)] \\ &= \sum_{i=0}^{L-1}\frac{1}{L}\sum_{w\in\Sigma^i}\sum_{\sigma\in\Sigma}P^{(m)}(w\cdot\sigma)\cdot\log\left(\frac{1}{Q(\sigma|w)}\right) \\ &= \mathbb{E}_{(w,\sigma)\sim\text{GEN}(P,L)}\left[\log\left(\frac{1}{Q(\sigma|w)}\right)\right] \end{aligned} \tag{4}$$

where the second equality is a result of an adapted version of Carrasco's decomposition lemma (Carrasco, 1997) provided in Appendix B.1.

Define the random function Z_I [3] such that:

$$Z_I(w, \sigma) \stackrel{\text{def}}{=} \log\left(\frac{1}{Q(\sigma|w)}\right) \tag{5}$$

where (w, σ) are drawn according to $\text{GEN}(P, L)$.

The reformulation of the CE in the expression (4) suggests the following empirical estimate of $\text{CE}_L(P, Q)$ given as:

$$CE_L(P, Q) \approx \frac{1}{N}\sum_{i=1}^{N} Z_I(w_i, \sigma_i) \tag{6}$$

where $\{(w_i, \sigma_i)\}_{i\in[N]}$ is a sample drawn i.i.d from $\text{GEN}(P, L)$. The remaining question pertains to the number of samples drawn from this oracle to achieve a good approximation of $\text{CE}_L(P, Q)$. The next lemma provides an answer to this question:

Lemma 2. *For any $(\epsilon, \delta) \in (0, 1)^2$, we have, for $N = \tilde{O}\left(\log(\frac{1}{\gamma}) \cdot \frac{1}{\epsilon^2}\right)$ [4], the estimator (6) is an ϵ-approximate of $CE_L(P, Q)$ with probability greater than $1 - \delta$.*

The proof of Lemma 2 can be found in appendix B.6.

Lemma 2 provides a convergence bound of the CE in terms of the smoothness parameter of the test distribution. The obtention of a model-dependent bound requires an estimate of a lower bound of this parameter for the considered families of NARLMs. This will be the subject of subsection 3.2.

3.1.2 Finite Stochastic Languages.

The theoretical analysis of the CE estimation for the case of FSLs will follow similar steps to the ISL case, with a slight difference on the assumption made on the target distribution and the structure of the sampling oracle to be used for this case. More precisely, a first step consists at a reformulation of the CE using Carrasco's decomposition giving rise to a sampling oracle which will play an analogous role of the GEN oracle for the case of FSLs introduced in the previous segment.

In the remainder of this segment, we fix $\gamma \in (0, 1)$, P and Q two FSLs assumed to be prefixial-bounded and γ-smooth, respectively.

The counterpart of the GEN oracle for ISLs, is the *prefixial sampling oracle*, denoted PSO, formally defined as follows:

Definition 5. *The prefixial sampling oracle* PSO*(.) takes as input a prefixial-bounded stochastic language P and draws a string $w \in \Sigma^*$ according to the FSL:*

$$P_p(w) = \frac{P(w \cdot \Sigma^*)}{||P||_p}$$

Analogous to the case of ISLs, we reformulate the CE using Carrasco's decomposition lemma:

$$\begin{aligned} CE(P, Q) &= \mathbb{E}_{w\sim P}\left[\log(\frac{1}{Q(w)})\right] \\ &= \sum_{w\in\Sigma^*}\sum_{\sigma\in\Sigma}P(w\sigma\Sigma^*)\left[\log\left(\frac{1}{Q(\sigma|w))}\right)\right] \\ &\quad + \sum_{w\in\Sigma^*}P(w)\cdot\frac{1}{\log\left(Q(\$|w)\right)} \\ &= ||P||_p \cdot \mathbb{E}_{w\sigma\sim\text{PSO}(P)}\left[\log\left(\frac{1}{Q(\sigma|w))}\right)\right] \end{aligned}$$

[3] The function Z_I depends implicitly on P and L. To ease exposition, we omit this dependency from the notation.

[4] The symbol $\tilde{O}(.)$ hides poly-logarithmic factors

$$+\mathbb{E}_{w\sim P}\left[\frac{1}{\log\left(Q(\$|w)\right)}\right] \quad (7)$$

The expression (7) suggests the following empirical estimate of $\mathrm{CE}(P,Q)$:

$$CE(P,Q) \approx \frac{||P||_p}{N_1}\sum_{i=1}^{N_1}\log(\frac{1}{Q(\sigma_i|w_i)}) + \frac{1}{N_2}\sum_{i=1}^{N_2}\log(\frac{1}{Q(\$|w_i)}) \quad (8)$$

where the samples composing the first (resp. second) term of the summation in (8) are drawn from $\mathtt{PSO}(P)$ (resp. P).

Define the random function Z_F as follows:

$$Z_F(w) = \log\left(\frac{1}{Q\left(w_{|w|}\middle|w_{1:|w|-1}\right)}\right) \quad (9)$$

where w is drawn from $\mathtt{PSO}(P)$.

By leveraging the expression (8), one can provide a comparable sample size complexity to the ISL case for the CE approximation between FSLs using the $\mathtt{PSO}(.)$ oracle:

Lemma 3. *For any $(\epsilon,\delta)\in(0,1)^2$, we have for $N_1=\tilde{O}(\frac{||P||_p^2}{\epsilon^2}\cdot\log(\frac{1}{\delta}))$ and $N_2=\tilde{O}(\frac{1}{\epsilon^2}\cdot\log(\frac{1}{\delta}))$, the estimator (8) provides an ϵ-approximate of $CE(P,Q)$ with probability greater than $1-\delta$.*

The result of lemma 3 is obtained similarily as lemma 2 (see Appendix B.6).

3.2 Norm-dependent bound for CE estimation

In the previous segment, we proposed a model-agnostic estimator of the CE between two probability distributions, where the test distribution is assumed to generate a smooth language. The convergence bound established in this segment gave rise to a logarithmic dependency on the inverse of the smoothness parameter γ of the test distribution. In this section, we shall build upon this result to derive convergence bounds tailored to the case of test distributions modeled by LSTMs/GRUs. The conversion of model-agnostic bounds to model-dependent ones shall be obtained by proving a lower bound on the smoothness parameter of LSTMs/GRUs in terms of its parameters:

Lemma 4. *Let $M =< n,T,F,W >$ be a LSTM/GRU generating a stochastic language. For any pair of integers $(p,q)\in[1,\infty]^2$ such that $\frac{1}{p}+\frac{1}{q}=1$, the stochastic language P_M is $\gamma_M^{(p,q)}$-smooth for:*

$$\gamma_M^{(p,q)} = \frac{1}{1+|\Sigma|\cdot\exp\left(\frac{2}{T}||W||_{\infty,p}\cdot||\mathcal{B}_n||_q\right)} \quad (10)$$

where:

$$||\mathcal{B}_n||_q \overset{\text{def}}{=} \sup_{\mathbf{x}\in[-1,1]^n}||\mathbf{x}||_q$$

Proof of lemma 4 can be found in appendix B.6.

The main theoretical result of the article is given in the following theorem:

Theorem 1. *The following statements are true:*

1. *Let P be an arbitrary ISL, and a LSTM/GRU $< n,T,F,W >$ generating an ISL Q. For any $(\epsilon,\delta)\in(0,1)^2$. it requires $\tilde{O}\left(\frac{1}{\epsilon^2.T}\cdot\inf_{\substack{(p,q)\in[1,\infty]^2\\ \frac{1}{p}+\frac{1}{q}=1}}||W||_{\infty,p}\cdot||\mathcal{B}_n||_q\right)$ samples from $\mathtt{GEN}(P,L)$ to obtain an ϵ-approximate of $CE_L(P,Q)$ with probability greater or equal to $1-\delta$.*

2. *Let P be a prefixial-bounded FSL, and a LSTM/GRU $< n,T,F,W >$ generating a FSL Q and any $(\epsilon,\delta)\in(0,1)^2$. It requires $\tilde{O}\left(\frac{||P||_p}{\epsilon^2\cdot T}\cdot\inf_{\substack{(p,q)\in[1,\infty]^2\\ \frac{1}{p}+\frac{1}{q}=1}}||W||_{\infty,p}\cdot||\mathcal{B}_n||_q\right)$ samples drawn from $\mathtt{PSO}(P)$ and P to obtain an ϵ-approximate of $CE(P,Q)$ with probability greater or equal to $1-\delta$,*

Theorem 1 is a direct corollary of lemmas 2 and 4.

We note that lemma 4 provides a collection of smoothness parameters for a given LSTM/GRU indexed by a pair of integers $(p,q)\in[1,\infty]^2$ such that $\frac{1}{p}+\frac{1}{q}=1$. A tight bound of the smoothness parameter can be obtained by finding the supremum of the quantity $||W||_{\infty,p}\cdot||\mathcal{B}_n||_q$ for pairs of integers satisfying this constraint. Due to the complex structure of this objective function, solving this optimization problem is unlikely to be tractable. Therefore, in our experiments, we opted to compare this quantity for the pairs $(p,q)\in\{(1,\infty),(\infty,1),(2,2)\}$. Empirically, we found that $||W||_{\infty,\infty}\cdot||\mathcal{B}_n||_1$ yields tighter estimates.

4 Variance-based convergence bounds for CE estimation.

In the previous section, we proposed a theoretical bound on the sample size complexity for CE estimation when test distributions are assumed to be

	Norm Bound	VAR Bound
$T = 0.5$	1.78 M	200.6 K
$T = 1$	178.1 K	49.6 K
$T = 100$	17.8 K	3.3 K

Table 1: Evaluation of the average non-asymptotic convergence bounds (in terms of the number of calls to the GEN oracle) for LSTMs trained on corpora generated by GPT $-$ 2 for different values of the temperature. Parameters: $\epsilon = 10^{-1}$ and $\delta = 0.1$. Details of the experiment can be found in Appendix A

generated by LSTM/GRUs. This bound was formally obtained due to the non-vacuous lower bound of the smoothness parameter of stochastic SLs generated by these models (Lemma 4). However, the derived bound have two major drawbacks. First, they are highly conservative. Second, generalizing these bounds to other families of models, such as Transformer-based models, is not straightforward. Specifically, obtaining a non-vacuous upper bound for the smoothness parameter of Transformer-based language models is challenging.

In this section, we investigate the prospect of deriving non-vacuous bounds for CE estimation that can be generalized to a wider family of bounded NARLMs while maintaining theoretical guarantees. Due to space constraints, we shall focus on the case of ISLs. Analogous argument can be conducted to obtain similar results for the case of FSLs. Let $P,\ Q$ be two ISLs and an integer $L > 0$. When Q is γ-smooth, the random function $Z_I(w, \sigma)$ defined in (5) is bounded which entails that its variance is finite. Consequently, by means of the Chebychev's inequality (Vershynin, 2018), one can obtain a non-asymptotic convergence rate of the CE estimate which depends on the variance of the random variable $Z_I(w, \sigma)$:

Proposition 4. *Let $P,\ Q$ be two ISLs such that Q is a smooth SL. For any $(\epsilon, \delta) \in (0, 1)^2$, For $N = O(\frac{Var(Z_I)}{\epsilon^2 \cdot \delta})$, with probability greater than $1 - \delta$, we have:*

$$\left| CE_L(P, Q) - \frac{1}{N} Z_I(w_i, \sigma_i) \right| \leq \epsilon$$

where the set $\{(w_i, \sigma_i)\}_{i \in [N]}$ is drawn i.i.d from $\mathtt{GEN}(P, L)$.

The variance-based bound in proposition 4 doesn't make any architectural assumptions about the test distribution besides being smooth. Consequently, since Decoder-only Transformer-based models and LSTMs/GRUs generate smooth SLs , this bound is applicable to both families of models.

Table 1 provides a comparative empirical analysis of the obtained values of norm-based and variance-based bounds for LSTMs trained on corpora generated by GPT-2. It illustrates the sample efficiency of the Variance-based bound as opposed to the norm-based one (see Appendix A for more details). of However, it's worth noting that the computation of the variance-based bound requires a prior estimation of the variance via empirical approximation, which marginally increases the sample size complexity of the overall procedure of the CE estimation. Nevertheless, given that the quantity $\log(Z_I(w, \sigma))$ is bounded for bounded NARLMs, the variance can be accurately estimated with a relatively small sample size. Conducted experiments show that a good estimate of the variance can be obtained using few hundreds of samples from test distributions, which doesn't contribute substantially in reducing the wide discrepancy between norm-based and variance-based bound values. (see Appendix A).

Conclusion

This article addresses the theoretical issue of studying the convergence properties of CE estimation for NARLMs. Non-asymptotic convergence bounds for CE estimation in neural LMs are introduced, covering both widely used configurations in practice: models with and without the *eos* token.

Our findings highlight the significance of these structural properties in determining the rate of convergence for CE estimates. By identifying and leveraging these properties, we provide a theoretical framework that enhances the understanding of cross-entropy behavior in neural language models. This framework offers valuable insights into the sample sizes required for accurate cross-entropy estimation and its dependency on the temperature parameter, addressing a theoretical gap in the literature. Overall, the theoretical results presented in this article contribute to the development a more rigorous evaluation methodology of the performance and information-theoretic properties of NARLMs.

Limitations.

Limitations of our work are summarized in the following points:

a. On the implementation of the PSO oracle

for Neural Language Models. A significant limitation of the norm-dependent bound presented in Theorem 1 lies in its dependency on the existence of an efficient implementation of the PSO oracle in order to be exploited in practice. When the target distribution is the empirical distribution, this oracle can be efficiently implemented (see Appendix C). Similarly, efficient implementations are feasible for distributions generated by Stochastic Weighted Automata (Balle, 2013). However, for neural language models, the implementation of this oracle presents considerable challenges and remains unresolved at the conclusion of our study.

b. On the case of Nucleus Sampling. Another notable limitation of our work arises from the failure of our theoretical argument to capture the case of nucleus sampling (Holtzman et al., 2019). Indeed, by assigning zero probability to certain tokens during the next token generation procedure, the resulting stochastic language generated by bounded NARLMS adopting this strategy becomes inherently non-smooth (Welleck et al., 2020). Indeed, the smoothness effect enforced by the softmax function was critical in our anaylsis. The question of estimating convergence bounds for cross-entropy estimation of language models using the nucleus sampling strategy is deferred to future research.

c. On the obtention of norm-dependent bounds for Decoder-only Transformer-based LMs. Deriving a norm-based convergence bound for the CE estimator in the context of LSTMs/GRUs critically depends on establishing a lower bound for the smoothness parameter of the SLs generated by these models (Lemma 4). Extending this analysis to Decoder-Only Transformer-based models necessitates obtaining comparable lower bounds for this category of models. However, achieving this theoretical objective proves to be exceedingly challenging. Thus, this question remains open for future research.

d. On the empirical difficulty of assessing the CE estimator on FSLs. Empirical observation arising from our experimental setup (see Appendix A) resides in the exclusion of FSLs from our experiments. The bound obtained for CE estimation in section 3 for FSLs relies on the prefixial norm of the target stochastic language. Although, as formally proven in proposition 3, this quantity is finite and directly linked to the expected length of generated sequences, thus allowing for empirical estimation in principle, the occurrence of generating the *end-of-sequence* token arises quite often after the generation of extremely long sequences. Given that all the methods proposed in this work are Monte Carlo-based, generating batches of sequences from language models implementing FSLs proved to be prohibitively expensive within the constraints of our hardware capabilities.

References

B. Balle. 2013. *Learning Finite-State Machines: Algorithmic and Statistical Aspects*. Ph.D. thesis, PhD Thesis.

S. Boucheron, G. Lugosi, and P. Massart. 2013. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press.

M. Braverman, X. Chen, S. Kakade, K. Narasimhan, C. Zhang, and Y. Zhang. 2020. Calibration, entropy rates, and memory in language models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1089–1099. PMLR.

R. Carrasco. 1997. Accurate computation of the relative entropy between stochastic regular grammars. *RAIRO. Informatique Théorique et Applications*, 31.

Y. Chen, S. Gilroy, A. Maletti, J. May, and K. Knight. 2018. Recurrent neural networks as weighted language recognizers. *Preprint*, arXiv:1711.05408.

A. Clark and F. Thollard. 2004. Pac-learnability of probabilistic deterministic finite state automata. *J. Mach. Learn. Res.*, 5:473–497.

C. Cortes, M. Mohri, and A. Rastogi. 2007. Lp distance and equivalence of probabilistic automata. *International Journal of Foundations of Computer Science*, 18:761–780.

C. Cortes, M. Mohri, A. Rastogi, and M. D. Riley. 2006. Efficient computation of the relative entropy of probabilistic automata. In *LATIN 2006: Theoretical Informatics*, pages 323–336, Berlin, Heidelberg. Springer Berlin Heidelberg.

A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. 2019. The curious case of neural text degeneration. *Preprint*, arXiv:1904.09751.

E. Kreyszig. 1989. *Introductory Functional Analysis with Applications*, 1 edition. Wiley, New York.

C. Liu, F. Zhao, K. Kuang, Y. Kang, Z. Jiang, C. Sun, and F. Wu. 2024. Evolving knowledge distillation with large language models and active learning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6717–6731, Torino, Italy. ELRA and ICCL.

X. Liu, . Liu, G. Van den Broeck, and Y. Liang. 2023. Understanding the distillation process from deep generative models to tractable probabilistic circuits. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*.

R. B. Lyngsø and C. N. S. Pedersen. 2002. The consensus string problem and the complexity of comparing hidden markov models. *Journal of Computer and System Sciences*, 65(3):545–569. Special Issue on Computational Biology 2002.

T. Murgue and C. de La Higuera. 2004. Distances between distributions: Comparing language models. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 269–277, Berlin, Heidelberg. Springer Berlin Heidelberg.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

W. Rudin. 1976. *Principles of Mathematical Analysis*, 3rd edition. McGraw-Hill, New York.

Lav R. Varshney, Nitish Shirish Keskar, and Richard Socher. 2020. Limits of detecting text generated by large-scale language models. *2020 Information Theory and Applications Workshop (ITA)*, pages 1–5.

R. Vershynin. 2018. High-dimensional probability.

L. Wei, Z. Tan, C. Li, J. Wang, and W. Huang. 2024. Large language model evaluation via matrix entropy. *Preprint*, arXiv:2401.17139.

S Welleck, I. Kulikov, J. Kim, R. Y. Pang, and K. Cho. 2020. Consistency of a recurrent language model with respect to incomplete decoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5553–5568, Online. Association for Computational Linguistics.

A Experimental Results

The stated objectives of the conducted experiment are twofold. First, to evaluate the order of magnitude of non-asymptotic convergence bounds derived from our theoretical analysis to assess their non-vacuity, i.e. their empirical exploitability in case of applications where rigorous theoretical guarantees are required. Second, to establish a comparative analysis of both norm-based bound (theorem 1) and variance-based bound (theorem 4). All experiments were conducted for the case of ISLs. Indeed, as raised in the limitations section, the event of generating the *end-of-sequence* token during the generation procedure often occurs after the generation of extremely long sequences. We note that all experiments were conducted on a Virtual Machine with 16GB in RAM memory equipped with a Tesla T4 GPU.

Description of the experimental setup. Several LSTMs with different hidden state sizes, specifically $n = \{128, 256, 1024\}$, were trained using corpora generated by `GPT-2`. To mitigate the statistical bias due to the inherent randomness of the experimental setup, we conduct five independent training runs. Each training run consists of three steps: first, the generation of a corpus from the considered LLM. Second, the training phase of the LSTM, and third, the computation of both norm-based bound (Theorem 1) and the variance-based bound (4). To estimate the variance of the log probabilities required for the computation of the variance-based bound, we observe that a sample of approximately 500 sentences were sufficient for the variance estimate to stabilize. The overall hyperparameters of the experimental setup are summarized in table 2.

Hyperparameters	Value
Number of epochs	120
Training corpus size	12.5M tokens
Corpus sentences length	500
Number of runs	5
Hidden state size	$[128, 256, 512]$
Sample size (VAR Estimation)	500

Table 2: Hyperparameters of the experimental setup

The overall output of this experimental process is a collection of 15 LSTMs, equally divided among three different hidden state sizes. We report the obtained results in figure 1.

B Technical Results.

B.1 Carrasco's decomposition lemma.

The subsequent lemma was crucial for establishing the complexity results for approximating the cross-entropy between families of SWAs and LSTMs/GRUs in section 3. Originally introduced by Carrasco (Carrasco, 1997) in the context of stochastic regular grammars, we proceed to present a restatement of this lemma applicable to the general case of arbitrary SLs: :

Lemma 5. *(Carrasco's decomposition lemma (Carrasco, 1997))*

1. *The FSL case: Let P and Q be two FSLs, we*

	Norm bound	VAR bound
$T = 0.5$	973 K	160.1 K
$T = 1$	97.3 K	39.4 K
$T = 10$	9.73 K	2.7 K

(a) $n = 128$

	Norm bound	VAR bound
$T = 0.5$	1.78 M	200.6 K
$T = 1$	178.1 K	49.6 K
$T = 10$	17.8 K	3.3 K

(b) $n = 256$

	Norm bound	VAR bound
$T = 0.5$	2.2 M	191.3 K
$T = 1$	220 K	49.9 K
$T = 10$	22 K	2.36 K

(c) $n = 512$

Figure 1: Experimental results for the average non-asymptotic convergence bounds (in terms of the number of calls to the GEN oracle on the LSTM) for CE estimation. Parameters: $\epsilon = 10^{-1}$ and $\delta = 0.95$.

have

$$CE(P,Q) = \sum_{w\in\Sigma^*}\sum_{\sigma\in\Sigma_\$} P(w\sigma\Sigma^*)\log(\frac{1}{Q(\sigma|w)})$$

By convention, for any string $w \in \Sigma^*$, we have $P(w\$\Sigma^*) = P(w)$

2. *The ISL case: Let P and Q be two ISLs. For any integer $L > 0$, we have:*

$$CE_L(P,Q) = \frac{1}{L}\sum_{i=0}^{L-1}\sum_{w\in\Sigma^i}\sum_{\sigma\in\Sigma} P^{(m)}(w\cdot\sigma) \cdot \log(\frac{1}{Q^{(m)}(\sigma|w}))$$

The key observation that enables the reformulation of the CE in the format of lemma 5 consists at noting that in the original expression of the cross-entropy, we have for any string $w \in \Sigma^*$ and a symbol $\sigma \in \Sigma_\$$ (resp. Σ) in the FSL (resp. ISL) setting, the term $\log(\frac{1}{Q(\sigma|w)})$ is multiplied by the set of quantities $\{P(w\sigma w')\}_{w'\in\Sigma^*}$. Summing over this set yields the quantity $P(w\sigma\Sigma^*)$

B.2 Proof of Proposition 1

Proof. We have:

$$\begin{aligned}||f||_p &= \sum_{w\in\Sigma^*} P(w\Sigma^*)\\ &= \sum_{m=0}^{\infty} P(\Sigma^m\Sigma^*)\\ &= \sum_{m=0}^{\infty} P(\Sigma^{\geq m})\\ &= 1 + \mathbb{E}_{w\sim f}[|w|]\end{aligned}$$

where the last equality is due to the relative $\mathbb{E}_{X\sim P}[X] = \sum_{n=1}^{\infty} P(X \geq n)$ (see lemma 1.2.1, (Vershynin, 2018)). □

B.3 Proof of Lemma 1

Recall the statement of lemma 1:

Lemma 6. *1. Inclusion: Any smooth FSL P is also prefixial-bounded.*

2. Strict inclusion: There exists prefixial-bounded FSLs that are not smooth.

The strict inclusion property can be obtained straightforwardly by considering the family of FSLs with finite support. A FSL with finite support is clearly prefixial-bounded but is not smooth. We shall focus next on the inclusion property.

Proof. Fix a smooth FSL P. We shall prove that it's also prefixial-bounded.

For an integer $n > 0$, the symbol p_n will refer refer to the probability of generating a sequence of length greater or equal to n. We have:

$$||P||_p = \sum_{n>0} p_n$$

In light of this formula, a first step towards our stated goal, we prove that p_n decreases exponentially as n increases.

$$\begin{aligned}p_{n+1} &= P(\Sigma^{n+1}\Sigma^*)\\ &= \sum_{w\in\Sigma^{n+1}} P(w\Sigma^*)\\ &= \sum_{w\in\Sigma^n}\sum_{\sigma\in\Sigma} P(w\sigma\Sigma^*)\\ &= \sum_{w\in\Sigma^n}\sum_{\sigma\in\Sigma^*} P(w\Sigma^*).P(\sigma|w)\end{aligned}$$

$$= \sum_{w \in \Sigma^*} P(w\Sigma^*) \sum_{\sigma \in \Sigma} P(\sigma|w)$$
$$= \sum_{w \in \Sigma^*} P(w\Sigma^*)[1 - P(\$|w)]$$
$$\leq (1-\gamma)p_n$$

Consequently, applying the inequality $(1-x)^n \leq e^{-n.x}$ for $x \in (0,1)$ and $p_0 = 1$, we have

$$p_n \leq (1-\gamma)^n p_0 \leq e^{-n.\gamma}$$

Consequently,

$$||P||_p = \sum_{n>0} p_n \leq \sum_n e^{-n.\gamma} < \infty$$

which completes the proof. □

B.4 Proof of proposition 2

The proposition 3 states that LSTMs/GRUs and Decoder-only Transformer-based LMs are bounded NARLMs.

With regards to LSTMs/GRUs, this statement has already been proven in (Welleck et al., 2020). We shall prove next that it is also the case for Decoder-only Transformer-based LMs.

The proof of the proposition will depend on a classical lemma from the field of calculus theory:

Lemma 7. *Let $f : X \to Y$ be a continuous map. The image of a compact set is also compact.*

Recall that a compact set can be characterized as follows (Rudin, 1976):
X is a compact set if and only if for any sequence $\{x_n\}_{n\in\mathbb{N}}$ in X, there exists a sub-sequence $\{x_{\phi(n)}\}_{n\in\mathbb{N}}$, where ϕ is an increasing map from $\mathbb{N}$ to $\mathbb{N}$, that converges to an element in X.

Proof. (7) Let $D \subseteq X$ a compact set. We shall prove that $f(D)$ is also compact. Let $\{y_n\}_{n\in\mathbb{N}}$ be a sequence in $f(D)$. Then, there exists a sequence $\{x_n\}_{n\in\mathbb{N}}$ of elements in D such that for any $n \in \mathbb{N} : f(x_n) = y_n$. Since D is compact, then there exists a sub-sequence $\{x_{\phi(n)}\}_{n\in\mathbb{N}}$ that converges to an element x^* in D. Hence, by continuity of the function f, the sequence $\{y_{\phi(n)}\}_{n\in\mathbb{N}}$ converges to $f(x^*)$. □

Now we are ready to prove point (b) of proposition 2:

Proof. Let M be a transformer-based LM with context size $K \in \mathbb{N}$. First, we show that $\sup_{w\in\Sigma^{\geq K}} ||\Phi \circ E_K(w)|| < \infty$. Let $w \in \Sigma^{\geq K}$, we have

$$||\Phi \circ E_K(w)|| \leq ||U \cdot W_e|| + ||W_p||$$
$$\leq \sup_{p\in\Sigma^K} ||U \cdot W_e|| + C_M$$

which is finite, due to the finiteness of the set Σ^K. Define the ball $\mathcal{B}(0; \sup_{w\in\Sigma^{\geq K}} ||\Phi \circ E_K(w)||)$. By lemma 7, the image of this ball according to the continuous map $T_{W_L} \circ .. \circ T_{W_1}$ (by assumption in the proposition statement and by the closure property of continuity under the composition operator) is compact, hence bounded by Bolzano-Weierstrass theorem. Consequently, transformer-based LMs belong to the bounded NARLM family. □

B.5 Proof of proposition 3

In this subsection, we will prove that bounded NARLMs generate smooth SLs.

For a matrix $W \in \mathbb{R}^{n\times m}$, we shall use the notation $W_{i:}$ (resp. $W_{:j}$) to designate its i-th row (resp. j-th column). When the columns of W are indexed by elements of the alphabet Σ, the notation $W_{:\sigma}$ will be employed to refer to the column of W indexed by the symbol $\sigma \in \Sigma$.

Let M be a bounded NARLM that generates a SL denoted P. Fix $(w, \sigma) \in \Sigma^* \times \Delta$. We have:

$$P_M(\sigma|w) = \frac{\exp(\frac{W_{:\sigma}^T \cdot F(w)}{T})}{\sum_{\sigma' \in \Delta} \exp(\frac{W_{\sigma':}^T \cdot F(w)}{T})}$$
$$= \frac{1}{1 + \sum_{\sigma' \in \Delta\setminus\{\sigma\}} \exp\left(\frac{1}{T}(W_{:\sigma} - W_{:\sigma'})^T \cdot F(w)\right)}$$
$$\geq \frac{1}{1 + |\Sigma| \cdot \exp\left(\frac{2}{T} \max_{\sigma\in\Delta} |W_{:\sigma}^T \cdot F(w)|\right)}$$
$$\geq \frac{1}{1 + |\Sigma| \cdot \exp\left(\frac{2}{T} \max_{\sigma\in\Delta} ||W_{:\sigma}||_2 \cdot ||F(w)||_2\right)}$$
$$\geq \frac{1}{1 + |\Sigma| \cdot \exp\left(\frac{2}{T} ||W||_{\infty,2} \cdot ||\mathcal{B}_n||_2\right)}$$

where the second inequality is obtained by Hölder's inequality.

Since P is bounded, there exists a scalar $B > 0$ such that $||W||_{\infty,2} \leq B$. In addition, we have $||\mathcal{B}_n||_2 = \sqrt{n}$.

Consequently, P is $\frac{1}{1+|\Sigma|\cdot B\cdot\sqrt{n}}$-smooth which completes the proof.

B.6 Proofs of results in section 3

B.6.1 Proofs of lemma 2 and lemma 3.

We prove lemma 2 for the case of ISLs. A similar proof can be obtained for the case of FSLs (lemma 3) by mimicking the argument herein.

Fix $(\epsilon, \delta) \in (0,1)^2$. Define the random function:

$$Z_I(w,\sigma) \stackrel{\text{def}}{=} \log(\frac{1}{Q(\sigma|w)})$$

where $(w,\sigma) \in \Sigma^* \times \Sigma$ are generated according to $\texttt{GEN}(P,L)$.

Given that Q is γ-smooth and $Q(\sigma|w) \geq 1$, the random function $Z_I(w,\sigma)$ is in $[0, \log(\frac{1}{\gamma})]$ with probability equal to 1.

By the application of Höeffding's inequality (Theorem 2.8, (Boucheron et al., 2013)), for $N \geq \tilde{O}\left(\log(\frac{1}{\gamma}) \cdot \frac{1}{\epsilon^2}\right)$, the event:

$$\left| \mathbb{E}_{(w,\sigma)\in \texttt{GEN}(P,L)} \left[\log\left(\frac{1}{Q(\sigma|w)}\right)\right] - \frac{1}{N}\sum_{i=1}^{N} Z_I(w_i,\sigma_i) \right| \leq \frac{\epsilon}{L}$$

which is equivalent to

$$\left| CE_L(P,Q) - \frac{L}{N}\sum_{i=1}^{N} Z_I(w_i,\sigma_i) \right| \leq \epsilon$$

holds with probability greater or equal to $1-\delta$.

B.6.2 Proof of lemma 4

The proof of lemma 4 will follow a similar structure of the proof of proposition 3.

Let $M =< n, T, F, W >$ be a LSTM/GRU, and $p,\ q$ be two integers in $[1,\infty]$ such that $\frac{1}{p}+\frac{1}{q}=1$. For any $(w,\sigma) \in \Sigma^* \times \Delta$, we have:

$$\begin{aligned}
P_M(\sigma|w) &= \frac{\exp(\frac{W_{:\sigma}^T \cdot F(w)}{T})}{\sum_{\sigma' \in \Delta} \exp(\frac{W_{\sigma',:}^T \cdot F(w)}{T})} \\
&= \frac{1}{1 + \sum_{\sigma' \in \Delta \setminus \{\sigma\}} \exp\left(\frac{1}{T}(W_{:\sigma} - W_{:\sigma'})^T \cdot F(w)\right)} \\
&\geq \frac{1}{1 + |\Sigma| \cdot \exp\left(\frac{2}{T} \max_{\sigma\in\Delta} |W_{:\sigma}^T \cdot F(w)|\right)} \\
&\geq \frac{1}{1 + |\Sigma| \cdot \exp\left(\frac{2}{T} \max_{\sigma\in\Sigma} ||W_{:\sigma}||_p \cdot ||F(w)||_q\right)} \\
&\geq \frac{1}{1 + |\Sigma| \cdot \exp\left(\frac{2}{T} ||W||_{\infty,p} \cdot ||\mathcal{B}_n||_q\right)}
\end{aligned}$$

where the first inequality is obtained from the fact that the exponential function is a monotonic increasing function and:

$$\begin{aligned}
W_{:\sigma}^T \cdot F(w) - W_{:\sigma'}^T \cdot F(w) &\leq |W_{:\sigma}^T \cdot F(w) - W_{:\sigma'}^T \cdot F(w)| \\
&\leq 2 \max_{\sigma\in\Sigma} |W_{:\sigma}^T F(w)|
\end{aligned}$$

The second inequality is obtained using Hölder's inequality.

B.6.3 Proof of theorem 1

Let P be an arbitrary probability distribution, and $M =< n, T, F, W >$ be a LSTM/GRU. Fix $(\epsilon,\delta) \in (0,1)^2$. By lemma 2 and 4 and the fact that the $\log(.)$ function is a monotonic increasing function, it requires $\tilde{O}\left(\inf_{\substack{(p,q)\in[1,\infty]^2 \\ \frac{1}{p}+\frac{1}{q}=1}} \log\left(\frac{1}{\gamma_M^{(p,q)}}\right) \cdot \frac{1}{\epsilon^2}\right)$ samples from $\texttt{GEN}(P,L)$ (where the expression of $\gamma_M^{(p,q)}$ is given in equation (10)) to obtain an ϵ-approximate of $CE_L(P,P_M)$ with probability greater or equal to $1-\delta$. On the other hand, For any $(p,q) \in [1,\infty]$ such that $\frac{1}{p}+\frac{1}{q}=1$, we have:

$$\begin{aligned}
\log(\frac{1}{\gamma_M^{(p,q)}}) &= \log\left(1 + |\Sigma| \cdot \exp(\frac{2}{T}||W||_{\infty,p} \cdot ||\mathcal{B}_n||_q)\right) \\
&= O\left(\log(\Sigma) + \frac{2}{T} \cdot ||W||_{\infty,p} \cdot ||\mathcal{B}_n||_q\right) \\
&= \tilde{O}\left(\frac{2}{T} \cdot ||W||_{\infty,p} \cdot ||\mathcal{B}_n||_q\right)
\end{aligned}$$

C Implementation of the $\texttt{PSO}(.)$ oracle for the empirical distribution.

Fix an alphabet Σ. Let $\mathcal{C} = \{w_1, \ldots, w_N\}$ be a corpus of size $N >> 1$. We denote by P^{emp} the empirical distribution associated to $\mathcal{C}$, and by $\texttt{len}(\mathcal{C})$ the maximum length of sequences in $\mathcal{C}$, i.e. $\texttt{len}(\mathcal{C}) = \max_{i\in[N]} |w_i|$.

We assume that during the pre-processing step, the corpus $\mathcal{C}$ was arranged in the format:

$$\tilde{\mathcal{C}} = \{(i, S_i)\}_{i\in[\texttt{len}(\mathcal{C})]}$$

where for each $i \in \texttt{len}(\mathcal{C})$:

$$S_i = \{w \in \mathcal{C} :\ \ |w| = i\}$$

Note that pre-processing a corpus to convert it into this format can be done simultaneously with the tokenization step, so as it doesn't require performing an additional pass on the corpus during the pre-preprocessing phase.

For a sequence $w \in \Sigma^*$, define:

$$N_w \stackrel{\text{def}}{=} |\{w' = w \cdot p :\ \ w' \in \mathcal{C}\}|$$

The distribution generated by $\texttt{PSO}(P^{emp})$ is given as:

$$P_p^{emp}(w) = \frac{N_w}{\sum_{j=1}^{\texttt{len}(\mathcal{C})} j \cdot |S_j|}$$

We first present a sampling procedure for generating one sequence from the $\texttt{PSO}(P^{emp})$ oracle. Afterwards, we shall present a batch sampling procedure that enables an efficient generationg of a large sample.

• **Sampling a sequence.** Define the following generative procedure:

1. Generate an element from $i^* \in [\texttt{len}(\mathcal{C})]$ according to the Binomial distribution parametrized by $\left[\frac{|S_1|}{\sum_{i=1}^{\texttt{len}(\mathcal{C})} i|S_i|} \quad \frac{2|S_2|}{\sum_{i=1}^{\texttt{len}(\mathcal{C})} i|S_i|} \quad \cdots \quad \frac{|\texttt{len}(\mathcal{C})\cdot|S_{\texttt{len}(\mathcal{C})}|}{\sum_{i=1}^{\texttt{len}(\mathcal{C})} i|S_i|}\right]$
2. Generate uniformly at random a sequence w from S_{i^*},
3. Output a sequence $w_{1:k}$ for $k \in [i^*]$ where k is drawn uniformly at random from $[i^*]$,

The correctness of this procedure is given in the following proposition:

Proposition 5. *The distribution generated by the sampling procedure outlined above is equal to the prefixial distribution.*

Proof. Let $w \in \Sigma^{\leq \texttt{len}(\mathcal{C})}$. we need to show that the probability of generating w using the sampling procedure defined above is equal to $P_p(w)$. We denote by $N_{w,i}$ the number of sequences in S_i for which w constitutes a prefix. The probability of generating w is given as:

$$\begin{aligned} Q(w) &= \sum_{i=1}^{\texttt{len}(\mathcal{C})} \frac{i\cdot|S_i|}{\sum_{j=1}^{\texttt{len}(\mathcal{C}))} j|S_j|} \cdot \frac{N_{w,i}}{|S_i|} \cdot \frac{1}{i} \\ &= \sum_{i=1}^{\texttt{len}(\mathcal{C})} \frac{N_{w,i}}{\sum_{j=1}^{\texttt{len}(\mathcal{C}))} j\cdot|S_j|} \\ &= \frac{N_w}{\sum_{j=1}^{\texttt{len}(\mathcal{C})j\cdot|S_j|}} = P_p^{emp}(w) \end{aligned}$$

□

• **Batch Sampling:** We shall leverage the sampling procedure provided above, we can provide a batch version sampling procedure. The key ingredient towards this goal is the multionomial distribution:

Definition 6 (Multinomial Distribution)**.** *Let n be the number of trials, and $\mathbf{p} = [p_1, p_2, \ldots, p_k]$ be vector in $[0,1]^n$ such that $\sum_{i=1}^k p_i = 1$.*

The probability mass function of the multinomial distribution parametrized by n and $\mathbf{p}$, denoted Multinomial$(n, \mathbf{p})$, is given as:

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_k = x_k) = \frac{n!}{x_1!x_2!\cdots x_k!} p_1^{x_1} p_2^{x_2} \cdots p_k^{x_k}$$

for non-negative integers $x_1, x_2, \ldots, x_k$ such that $\sum_{i=1}^k x_i = n$.

The multinomial distribution is a generalization of the binomial distribution. It describes the probabilities of the outcomes of a multinomial experiment, which consists of n independent trials, each of which can result in one of k possible outcomes, where the probability of obtaining the outcome k is equal to p_k.

An equivalent sampling procedure to the one introduced for single sequences but more adapted to generating batches from the PSO oracle is outlined in the following. For a desired sample size $M > 0$:

1. Generate a vector $\begin{bmatrix} m_1 \\ \ldots \\ m_{\texttt{len}(\mathcal{C})} \end{bmatrix}$ according to the probability distribution:

$$\text{Multinomial}\left(M, \left[\frac{|S_1|}{\sum_{j=1}^{\texttt{len}(\mathcal{C})} j\cdot|S_j|}, \ldots, \frac{\texttt{len}(\mathcal{C})\cdot|S_{\texttt{len}(\mathcal{C})}|}{\sum_{j=1}^{\texttt{len}(\mathcal{C})} j\cdot|S_j|}\right]\right)$$

2. For each $i \in [\texttt{len}(\mathcal{C})]$:
   (a) Generate uniformly at random (without replacement) m_i sequences from S_i, denoted $S_i^{m_i} = \{s_1, \ldots, s_{m_i}\}$.
   (b) Draw uniformly at random a vector $\tilde{\mathbf{J}} = \begin{bmatrix} l_1 \\ \vdots \\ l_{m_i} \end{bmatrix}$ according to the probability distribution:

$$\text{Multinomial}(m_i, \left[\frac{1}{m_i}, \ldots, \frac{1}{m_i}\right])$$

   (c) Output the truncated prefixes of $S_i^{m_i}$ according to the vector $\tilde{\mathbf{J}}$ (regardless of the order of sequences in $S_i^{m_i}$)