entropy

MDPI

Article

Estimating Predictive Rate–Distortion Curves via Neural Variational Inference

Michael Hahn [1,*] and Richard Futrell [2]

[1] Department of Linguistics, Stanford University, Stanford, CA 94305, USA
[2] Department of Language Science, University of California, Irvine, CA 92697, USA
* Correspondence: mhahn2@stanford.edu

Received: 13 May 2019; Accepted: 26 June 2019; Published: 28 June 2019

Abstract: The Predictive Rate–Distortion curve quantifies the trade-off between compressing information about the past of a stochastic process and predicting its future accurately. Existing estimation methods for this curve work by clustering finite sequences of observations or by utilizing analytically known causal states. Neither type of approach scales to processes such as natural languages, which have large alphabets and long dependencies, and where the causal states are not known analytically. We describe Neural Predictive Rate–Distortion (NPRD), an estimation method that scales to such processes, leveraging the universal approximation capabilities of neural networks. Taking only time series data as input, the method computes a variational bound on the Predictive Rate–Distortion curve. We validate the method on processes where Predictive Rate–Distortion is analytically known. As an application, we provide bounds on the Predictive Rate–Distortion of natural language, improving on bounds provided by clustering sequences. Based on the results, we argue that the Predictive Rate–Distortion curve is more useful than the usual notion of statistical complexity for characterizing highly complex processes such as natural language.

Keywords: Predictive Rate–Distortion; natural language; information bottleneck; neural variational inference

1. Introduction

Predicting the future from past observations is a key problem across science. Constructing models that predict future observations is a fundamental method of making and testing scientific discoveries. Understanding and predicting dynamics has been a fundamental goal of physics for centuries. In engineering, devices often have to predict the future of their environment to perform efficiently. Living organisms need to make inferences about their environment and its future for survival.

Real-world systems often generate complex dynamics that can be hard to compute. A biological organism typically will not have the computational capacity to perfectly represent the environment. In science, measurements have limited precision, limiting the precision with which one can hope to make predictions. In these settings, the main goal will typically be to get good prediction at low computational cost.

This motivates the study of models that try to extract those key features of past observations that are most relevant to predicting the future. A general information-theoretic framework for this problem is provided by Predictive Rate–Distortion [1,2], also known as the past-future information bottleneck [3]. The Predictive Rate–Distortion trade-off seeks to find an encoding of past observations that is maximally informative about future observations, while satisfying a bound on the amount of information that has to be stored. More formally, this framework trades off prediction loss in the future, formalized as cross-entropy, with the cost of representing the past, formalized as the mutual information between the past observations and the compressed representations of the past. Due to its information-theoretic nature, this framework is extremely general and applies to processes across vastly different domains. It has been applied to linear dynamical systems [3,4], but is equally meaningful for discrete dynamical systems [2]. For biological systems that make predictions about their environment, this corresponds to placing an information-theoretic constraint on the computational cost used for conditioning actions on past observations [5].

The problem of determining encodings that optimize the Predictive Rate–Distortion trade-off has been solved for certain specific kinds of dynamics, namely for linear dynamic systems [3] and for processes whose predictive dynamic is represented exactly by a known, finite Hidden Markov Model [2]. However, real-world processes are often more complicated. When the dynamics are known, a representing Hidden Markov Model may still be extremely large or even infinite, making general-purpose automated computation difficult. Even more importantly, the underlying dynamics are often not known exactly. An organism typically does not have access to the exact dynamics of its surroundings. Similarly, the exact distribution of sentences in a natural language is not known exactly, precluding the application of methods that require an exact model. Such processes are typically only available implicitly, through a finite sample of trajectories.

Optimal Causal Filtering (OCF, Still et al. [6]) addresses the problem of estimating Predictive Rate–Distortion from a finite sample of observation trajectories. It does so by constructing a matrix of observed frequencies of different pairs of past and future observations. However, such a method faces a series of challenges [2]. One is the curse of dimensionality: Modeling long dependencies requires storing an exponential number of observations, which quickly becomes intractable for current computation methods. This exponential growth is particularly problematic when dealing with processes with a large state space. For instance, the number of distinct words in a human language as found in large-scale text data easily exceeds 1×10^5, making storing and manipulating counts of longer word sequences very challenging. A second challenge is that of overfitting: When deploying a predictive model constructed via OCF on new data to predict upcoming observations, such a model can only succeed when the past sequences occurred in the sample to which OCF was applied. This is because OCF relies on counts of full past and future observation sequences; it does not generalize to unseen past sequences.

Extrapolating to unseen past sequences is possible in traditional time series models representing processes that take continuous values; however, such methods are less easily applied to discrete sequences such as natural language. Recent research has seen a flurry of interest in using flexible nonlinear function approximators, and in particular recurrent neural networks, which can handle sequences with discrete observations. Such machine learning methods provide generic models of sequence data. They are the basis of the state of the art by a clear and significant margin for prediction in natural language [7–10]. They also have been successfully applied to modeling many other kinds of time series found across disciplines [11–18].

We propose Neural Predictive Rate–Distortion (NPRD) to estimate Predictive Rate–Distortion when only a finite set of sample trajectories is given. We use neural networks both to encode past observations into a summary code, and to predict future observations from it. The universal function approximation capabilities of neural networks enable such networks to capture complex dynamics, with computational cost scaling only linearly with the length of observed trajectories, compared to the exponential cost of OCF. When deploying on new data, such a neural model can generalize seamlessly to unseen sequences, and generate appropriate novel summary encodings on the fly. Recent advances in neural variational inference [19,20] allow us to construct predictive models that provide almost optimal predictive performance at a given rate, and to estimate the Predictive Rate–Distortion trade-off from such networks. Our method can be applied to sample trajectories in an off-the-shelf manner, without prior knowledge of the underlying process.

In Section 2, we formally introduce Predictive Rate–Distortion, and discuss related notions of predictive complexity. In Section 3, we describe the prior method of Optimal Causal Filtering (OCF).

In Section 4, we describe our method NPRD. In Section 5, we validate NPRD on processes whose Predictive Rate–Distortion is analytically known, showing that it finds essentially optimal predictive models. In Section 6, we apply NPRD to data from five natural languages, providing the first estimates of Predictive Rate–Distortion of natural language.

2. Predictive Rate–Distortion

We consider stationary discrete-time stochastic processes $(X_t)_{t\in\mathbb{Z}}$, taking values in a state space S. Given a reference point in time, say $T = 0$, we are interested in the problem of predicting the future of $X_{t\geq 0} = (X_0, X_1, X_2, ...)$ from the past $X_{t<0} = (..., X_{-2}, X_{-1})$. In general—unless the observations X_t are independent—predicting the future of the process accurately will require taking into account the past observations. There is a trade-off between the accuracy of prediction, and how much information about the past is being taken into account. On one extreme, not taking the past into account at all, one will not be able to take advantage of the dependencies between past and future observations. On the other extreme, considering the entirety of the past observations $X_{t\leq 0}$ can require storing large and potentially unbounded amounts of information. This trade-off between information storage and prediction accuracy is referred to as Predictive Rate–Distortion (PRD) [2]. The term rate refers to the amount of past information being taken into account, while distortion refers to the degradation in prediction compared to optimal prediction from the full past.

The problem of Predictive Rate–Distortion has been formalized by a range of studies. A principled and general formalization is provided by applying the Information Bottleneck idea [2,6,21]: We will write $\overleftarrow{X}$ for the past $X_{<0}$, and $\overrightarrow{X}$ for the future $X_{\geq 0}$, following [2]. We consider random variables Z, called codes, that summarize the past and are used by the observer to predict the future. Formally, Z needs to be independent from the future $\overrightarrow{X}$ conditional on the past $\overleftarrow{X}$: in other words, Z does not provide any information about the future except what is contained in the past. Symbolically:

$$Z \perp \overrightarrow{X} \,|\, \overleftarrow{X}. \tag{1}$$

This is equivalent to the requirement that $Z \leftrightarrow \overleftarrow{X} \leftrightarrow \overrightarrow{X}$ be a Markov chain. This formalizes the idea that the code is computed by an observer from the past, without having access to the future. Predictions are then made based only on Z, without additional access to the past $\overleftarrow{X}$.

The rate of the code Z is the mutual information between Z and the past: $\mathrm{I}[Z, \overleftarrow{X}]$. By the Channel Coding Theorem, this describes the channel capacity that the observer requires in order to transform past observations into the code Z.

The distortion is the loss in predictive accuracy when predicting from Z, relative to optimal prediction from the full past $\overleftarrow{X}$. In the Information Bottleneck formalization, this is equivalent to the amount of mutual information between past and future that is *not* captured by Z [22]:

$$\mathrm{I}[\overleftarrow{X}, \overrightarrow{X} | Z]. \tag{2}$$

Due to the Markov condition, the distortion measure satisfies the relation

$$\mathrm{I}[\overleftarrow{X}, \overrightarrow{X} | Z] = \mathrm{I}[\overleftarrow{X}, \overrightarrow{X}] - \mathrm{I}[Z, \overrightarrow{X}], \tag{3}$$

i.e., it captures how much less information Z carries about the future $\overrightarrow{X}$ compared to the full past $\overleftarrow{X}$. For a fixed process $(X_t)_t$, choosing Z to minimize the distortion is equivalent to maximizing the mutual information between the code and the future:

$$\mathrm{I}[Z, \overrightarrow{X}]. \tag{4}$$

We will refer to (4) as the predictiveness of the code Z.

The rate–distortion trade-off then chooses Z to minimize distortion at bounded rate:

$$\min_{Z:\mathrm{I}[\overleftarrow{X},Z]\leq d} \mathrm{I}[\overleftarrow{X},\overrightarrow{X}|Z] \tag{5}$$

or—equivalently—maximize predictiveness at bounded rate:

$$\max_{Z:\mathrm{I}[\overleftarrow{X},Z]\leq d} \mathrm{I}[Z,\overrightarrow{X}]. \tag{6}$$

Equivalently, for each $\lambda \geq 0$, we study the problem

$$\max_{Z} \left(\mathrm{I}[Z,\overrightarrow{X}] - \lambda \cdot \mathrm{I}[\overleftarrow{X},Z]\right), \tag{7}$$

where the scope of the maximization is the class of all random variables Z such that $Z \rightarrow \overleftarrow{X} \rightarrow \overrightarrow{X}$ is a Markov chain.

The objective (7) is equivalent to the Information Bottleneck [21], applied to the past and future of a stochastic process. The coefficient λ indicates how strongly a high rate $\mathrm{I}[\overleftarrow{X},Z]$ is penalized; higher values of λ result in lower rates and thus lower values of predictiveness.

The largest achievable predictiveness $\mathrm{I}[Z,\overrightarrow{X}]$ is equal to $\mathrm{I}[\overleftarrow{X},\overrightarrow{X}]$, which is known as the excess entropy of the process [23]. Due to the Markov condition (1) and the Data Processing Inequality, predictiveness of a code Z is always upper-bounded by the rate:

$$\mathrm{I}[Z,\overrightarrow{X}] \leq \mathrm{I}[\overleftarrow{X},Z]. \tag{8}$$

As a consequence, when $\lambda \geq 1$, then (7) is always optimized by a trivial Z with zero rate and zero predictiveness. When $\lambda = 0$, any lossless code optimizes the problem. Therefore, we will be concerned with the situation where $\lambda \in (0,1)$.

2.1. Relation to Statistical Complexity

Predictive Rate–Distortion is closely related to Statistical Complexity and the ϵ-machine [24,25]. Given a stationary process X_t, its causal states are the equivalence classes of semi-infinite pasts $\overleftarrow{X}$ that induce the same conditional probability over semi-infinite futures $\overrightarrow{X}$: Two pasts $\overleftarrow{X}, \overleftarrow{X}'$ belong to the same causal state if and only if $P(x_{1\ldots k}|\overleftarrow{X}) = P(x_{1\ldots k}|\overleftarrow{X}')$ holds for all finite sequences $x_{0\ldots k}$ ($k \in \mathbb{N}$). Note that this definition is not measure-theoretically rigorous; such a treatment is provided by Löhr [26].

The causal states constitute the state set of a a Hidden Markov Model (HMM) for the process, referred to as the ϵ-machine [24]. The statistical complexity of a process is the state entropy of the ϵ-machine. Statistical complexity can be computed easily if the ϵ-machine is analytically known, but estimating statistical complexity empirically from time series data are very challenging and seems to at least require additional assumptions about the process [27].

Marzen and Crutchfield [2] show that Predictive Rate–Distortion can be computed when the ϵ-machine is analytically known, by proving that it is equivalent to the problem of compressing causal states, i.e., equivalence classes of pasts, to predict causal states of the backwards process, i.e., equivalence classes of futures. Furthermore, [6] show that, in the limit of $\lambda \rightarrow 0$, the code Z that optimizes Predictive Rate–Distortion (7) turns into the causal states.

2.2. Related Work

There are related models that represent past observations by extracting those features that are relevant for prediction. Predictive State Representations [28,29] and Observable Operator Models [30] encode past observations as sets of predictions about future observations. Rubin et al. [31] study agents that trade the cost of representing information about the environment against the reward they can obtain by choosing actions based on the representation. Relatedly, Still [1] introduces a Recursive Past Future Information Bottleneck where past information is compressed repeatedly, not just at one reference point in time.

As discussed in Section 2.1, estimating Predictive Rate–Distortion is related to the problem of estimating statistical complexity. Clarke et al. [27] and Still et al. [6] consider the problem of estimating statistical complexity from finite data. While statistical complexity is hard to identify from finite data in general, Clarke et al. [27] introduces certain restrictions on the underlying process that make this more feasible.

3. Prior Work: Optimal Causal Filtering

The main prior method for estimating Predictive Rate–Distortion from data are Optimal Causal Filtering (OCF, Still et al. [6]). This method approximates Predictive Rate–Distortion using two approximations: first, it replaces semi-infinite pasts and futures with bounded-length contexts, i.e., pairs of finite past contexts ($\overset{M\leftarrow}{X} := X_{-M} \ldots X_{-1}$) and future contexts ($\overset{\rightarrow M}{X} := X_0 \ldots X_{M-1}$) of some finite length M.(It is not crucial that past and future contexts have the same lengths, and indeed Still et al. [6] do not assume this). (We do assume equal length throughout this paper for simplicity of our experiments, though nothing depends on this). The PRD objective (7) then becomes (9), aiming to predict length-M finite futures from summary codes Z of length-M finite pasts:

$$\max_{Z : Z \perp \overset{\rightarrow M}{X} | \overset{M\leftarrow}{X}} \left(\mathrm{I}[Z, \overset{\rightarrow M}{X}] - \lambda \cdot \mathrm{I}[\overset{M\leftarrow}{X}, Z] \right). \tag{9}$$

Second, OCF estimates information measures directly from the observed counts of ($\overset{M\leftarrow}{X}$), ($\overset{\rightarrow M}{X}$) using the plug-in estimator of mutual information. With such an estimator, the problem in (9) can be solved using a variant of the Blahut–Arimoto algorithm [21], obtaining an encoder $P(Z|\overset{M\leftarrow}{X})$ that maps each observed past sequence $\overset{M\leftarrow}{X}$ to a distribution over a (finite) set of code words Z.

Two main challenges have been noted in prior work: first, solving the problem for a finite empirical sample leads to overfitting, overestimating the amount of structure in the process. Still et al. [6] address this by subtracting an asymptotic correction term that becomes valid in the limit of large M and $\lambda \to 0$, when the codebook $P(Z|\overleftarrow{X})$ becomes deterministic, and which allows them to select a deterministic codebook of an appropriate complexity. This leaves open how to obtain estimates outside of this regime, when the codebook can be far from deterministic.

The second challenge is that OCF requires the construction of a matrix whose rows and columns are indexed by the observed past and future sequences [2]. Depending on the topological entropy of the process, the number of such sequences can grow as $|A|^M$, where A is the set of observed symbols, and processes of interest often do show this exponential growth [2]. Drastically, in the case of natural language, A contains thousands of words.

A further challenge is that OCF is infeasible if the number of required codewords is too large, again because it requires constructing a matrix whose rows and columns are indexed by the codewords and observed sequences. Given that storing and manipulating matrices greater than $10^5 \times 10^5$ is currently not feasible, a setting where $\mathrm{I}[\overleftarrow{X}, Z] > \log 10^5 \approx 11.5$ cannot be captured with OCF.

4. Neural Estimation via Variational Upper Bound

We now introduce our method, Neural Predictive Rate–Distortion (NPRD), to address the limitations of OCF, by using parametric function approximation: whereas OCF constructs a codebook mapping between observed sequences and codes, we use general-purpose function approximation estimation methods to compute the representation Z from the past and to estimate a distribution over future sequences from Z. In particular, we will use recurrent neural networks, which are known to provide good models of sequences from a wide range of domains; our method will also be applicable to other types of function approximators.

This will have two main advantages, addressing the limitations of OCF: first, unlike OCF, function approximators can discover generalizations across similar sequences, enabling the method to calculate good codes Z even for past sequences that were not seen previously. This is of paramount importance in settings where the state space is large, such as the set of words of a natural language. Second, the cost of storing and evaluating the function approximators will scale *linearly* with the length of observed sequences both in space and in time, as opposed to the exponential memory demand of OCF. This is crucial for modeling long dependencies.

4.1. Main Result: Variational Bound on Predictive Rate–Distortion

We will first describe the general method, without committing to a specific framework for function approximation yet. We will construct a bound on Predictive Rate–Distortion and optimize this bound in a parametric family of function approximators to obtain an encoding Z that is close to optimal for the nonparametric objective (7).

As in OCF (Section 3), we assume that a set of finite sample trajectories $x_{-M} \dots x_{M-1}$ is available, and we aim to compress pasts of length M to predict futures of length M. To carry this out, we restrict the PRD objective (7) to such finite-length pasts and futures:

$$\max_{Z : Z \perp \overset{\rightarrow M}{X} | \overset{M\leftarrow}{X}} \left(\mathrm{I}[Z, \overset{\rightarrow M}{X}] - \lambda \cdot \mathrm{I}[\overset{M\leftarrow}{X}, Z] \right). \tag{10}$$

It will be convenient to equivalently rewrite (10) as

$$\min_{Z : Z \perp \overset{\rightarrow M}{X} | \overset{M\leftarrow}{X}} \left[\mathrm{H}[\overset{\rightarrow M}{X} | Z] + \lambda \cdot \mathrm{I}[\overset{M\leftarrow}{X}, Z] \right], \tag{11}$$

where $\mathrm{H}[\overset{\rightarrow M}{X} | Z]$ is the prediction loss. Note that minimizing prediction loss is equivalent to maximizing predictiveness $\mathrm{I}[\overset{\rightarrow M}{X}, Z]$.

When deploying such a predictive code Z, two components have to be computed: a distribution $P(Z| \overset{M\leftarrow}{X})$ that encodes past observations into a code Z, and a distribution $P(\overset{\rightarrow M}{X} | Z)$ that decodes the code Z into predictions about the future.

Let us assume that we already have some encoding distribution

$$Z \sim P_\phi(Z| \overset{M\leftarrow}{X}), \tag{12}$$

where ϕ is the encoder, expressed in some family of function approximators. The encoder transduces an observation sequence $\overset{M\leftarrow}{X}$ into the parameters of the distribution $P_\phi(\cdot| \overset{M\leftarrow}{X})$. From this encoding distribution, one can obtain the optimal decoding distribution over future observations via Bayes' rule:

$$P(\overset{\rightarrow M}{X} | Z) = \frac{P(\overset{\rightarrow M}{X}, Z)}{P(Z)} = \frac{\mathbb{E}_{\overset{M\leftarrow}{X}} P(\overset{\rightarrow M}{X}, Z| \overset{M\leftarrow}{X})}{\mathbb{E}_{\overset{M\leftarrow}{X}} P_\phi(Z| \overset{M\leftarrow}{X})} =^{(*)} \frac{\mathbb{E}_{\overset{M\leftarrow}{X}} P(\overset{\rightarrow M}{X} | \overset{M\leftarrow}{X}) P_\phi(Z| \overset{M\leftarrow}{X})}{\mathbb{E}_{\overset{M\leftarrow}{X}} P_\phi(Z| \overset{M\leftarrow}{X})}, \tag{13}$$

where $(*)$ uses the Markov condition $Z \perp \overset{\rightarrow M}{X} | \overset{M\leftarrow}{X}$. However, neither of the two expectations in the last expression of (13) is tractable, as they require summation over exponentially many sequences, and algorithms (e.g., dynamic programming) to compute this sum efficiently are not available in general. For a similar reason, the rate $\mathrm{I}[\overset{M\leftarrow}{X}, Z]$ of the code $Z \sim P_\phi(Z| \overset{M\leftarrow}{X})$ is also generally intractable.

Our method will be to introduce additional functions, also expressed using function approximators that approximate some of these intractable quantities: first, we will use a parameterized probability distribution q as an approximation to the intractable marginal $P(Z) = \mathbb{E}_{\overset{M\leftarrow}{X}} P_\phi(Z|\overset{M\leftarrow}{X})$:

$$q(Z) \text{ approximates } P(Z) = \mathbb{E}_{\overset{M\leftarrow}{X}} P_\phi(Z|\overset{M\leftarrow}{X}). \tag{14}$$

Second, to approximate the decoding distribution $P(\overset{\rightarrow M}{X}|Z)$, we introduce a parameterized decoder ψ that maps points $Z \in \mathbb{R}^N$ into probability distributions $P_\psi(\overset{\rightarrow M}{X}|Z)$ over future observations $\overset{\rightarrow M}{X}$:

$$P_\psi(\overset{\rightarrow M}{X}|Z) \text{ approximates } P(\overset{\rightarrow M}{X}|Z) \tag{15}$$

for each code $Z \in \mathbb{R}^N$. Crucially, $P_\psi(\overset{\rightarrow M}{X}|Z)$ will be easy to compute efficiently, unlike the exact decoding distribution $P(\overset{\rightarrow M}{X}|Z)$.

If we fix a stochastic process $(X_t)_{t\in\mathbb{Z}}$ and an encoder ϕ, then the following two bounds hold for *any* choice of the decoder ψ and the distribution q:

Proposition 1. *The loss incurred when predicting the future from Z via ψ upper-bounds the true conditional entropy of the future given Z, when predicting using the exact decoding distribution (13):*

$$-\mathbb{E}_X\mathbb{E}_{Z\sim\phi(X)}\left[\log P_\psi(\overset{\rightarrow M}{X}|Z)\right] \geq \mathrm{H}[\overset{\rightarrow M}{X}|Z]. \tag{16}$$

Furthermore, equality is attained if and only if $P_\psi(\overset{\rightarrow M}{X}|Z) = P(\overset{\rightarrow M}{X}|Z)$.

Proof. By Gibbs' inequality:

$$\begin{aligned}-\mathbb{E}_X\mathbb{E}_{Z\sim\phi(X)}\left[\log P_\psi(\overset{\rightarrow M}{X}|Z)\right] &\geq -\mathbb{E}_X\mathbb{E}_{Z\sim\phi(X)}\left[\log P(\overset{\rightarrow M}{X}|Z)\right]\\ &= \mathrm{H}[\overset{\rightarrow M}{X}|Z].\end{aligned}$$

□

Proposition 2. *The KL Divergence between $P_\phi(Z|\overset{M\leftarrow}{X})$ and $q(Z)$, averaged over pasts $\overset{M\leftarrow}{X}$, upper-bounds the rate of Z:*

$$\begin{aligned}\mathbb{E}_{\overset{M\leftarrow}{X}}\left[\mathrm{D}_{\mathrm{KL}}(P_\phi(Z|\overset{M\leftarrow}{X}) \,\|\, q(Z))\right] &= \mathbb{E}_{\overset{M\leftarrow}{X}}\mathbb{E}_{Z|\overset{M\leftarrow}{X}} \log\frac{P_\phi(Z|\overset{M\leftarrow}{X})}{q(Z)}\\ &\geq \mathbb{E}_{\overset{M\leftarrow}{X}}\mathbb{E}_{Z|\overset{M\leftarrow}{X}} \log\frac{P_\phi(Z|\overset{M\leftarrow}{X})}{P(Z)}\\ &= \mathrm{I}[\overset{M\leftarrow}{X}, Z].\end{aligned} \tag{17}$$

Equality is attained if and only if $q(Z)$ is equal to the true marginal $P(Z) = \mathbb{E}_{\overset{M\leftarrow}{X}} P_\phi(Z|\overset{M\leftarrow}{X})$.

Proof. The two equalities follow from the definition of KL Divergence and Mutual Information. To show the inequality, we again use Gibbs' inequality:

$$-\mathbb{E}_{\overset{M\leftarrow}{X}}\mathbb{E}_{Z|\overset{M\leftarrow}{X}} \log q(Z) = -\mathbb{E}_Z \log q(Z) \geq -\mathbb{E}_Z \log P(Z) = -\mathbb{E}_{\overset{M\leftarrow}{X}}\mathbb{E}_{Z|\overset{M\leftarrow}{X}} \log P(Z).$$

Here, equality holds if and only if $q(Z) = P(Z)$, proving the second assertion. □

We now use the two propositions to rewrite the Predictive Rate–Distortion objective (11) in a way amenable to using function approximators, which is our main theoretical result, and the foundation of our proposed estimation method:

Corollary 3 (Main Result). *The Predictive Rate–Distortion objective (11)*

$$\min_{Z:Z\perp \overset{\rightarrow M}{X} \mid \overset{M\leftarrow}{X}} \left[\mathrm{H}[\overset{\rightarrow M}{X} |Z] + \lambda\, \mathrm{I}[\overset{M\leftarrow}{X}, Z] \right] \tag{11}$$

is equivalent to

$$\min_{\phi,\psi,q} \left[\mathbb{E}_{\overset{M\leftarrow}{X}, \overset{\rightarrow M}{X}} \left[-\mathbb{E}_{Z\sim\phi(\overset{M\leftarrow}{X})} \left[\log P_\psi(\overset{\rightarrow M}{X} |Z) \right] + \lambda \cdot \mathrm{D}_{\mathrm{KL}} \left[(P_\phi(Z| \overset{M\leftarrow}{X}) \parallel q(Z)) \right] \right] \right], \tag{18}$$

where ϕ, ψ, q range over all triples of the appropriate types described above.

From a solution to (18), one obtains a solution to (11) by setting $Z \sim P_\phi(\cdot| \overset{M\leftarrow}{X})$. The rate of this code is given as follows:

$$\mathrm{I}[Z, \overset{M\leftarrow}{X}] = \mathbb{E}_{\overset{M\leftarrow}{X}} \left[\mathrm{D}_{\mathrm{KL}}(P_\phi(Z| \overset{M\leftarrow}{X}) \parallel q(Z)) \right] \tag{19}$$

and the prediction loss is given by

$$\mathrm{H}[\overset{\rightarrow M}{X} |Z] = -\mathbb{E}_{X_{\overset{M\leftarrow}{X}, \overset{\rightarrow M}{X}}} \mathbb{E}_{Z\sim P_\phi(\overset{M\leftarrow}{X})} \left[\log P_\psi(\overset{\rightarrow M}{X} |Z) \right]. \tag{20}$$

Proof. By the two propositions, the term inside the minimization in (18) is an upper bound on (11), and takes on that value if and only if $P_\phi(\cdot| \overset{M\leftarrow}{X})$ equals the distribution of the Z optimizing (11), and ψ, q are as described in those propositions. □

Note that the right-hand sides of (19–20) can both be estimated efficiently using Monte Carlo samples from $Z \sim P_\phi(\overset{M\leftarrow}{X})$.

If ϕ, ψ, q are not exact solutions to (18), the two propositions guarantee that we still have bounds on rate and prediction loss for the code Z generated by ϕ:

$$\mathrm{I}[Z, \overset{M\leftarrow}{X}] \le \mathbb{E}_{\overset{M\leftarrow}{X}} \left[\mathrm{D}_{\mathrm{KL}}(P_\phi(Z| \overset{M\leftarrow}{X}) \parallel q(Z)) \right], \tag{21}$$

$$\mathrm{H}[\overset{\rightarrow M}{X} |Z] \le -\mathbb{E}_{X_{\overset{M\leftarrow}{X}, \overset{\rightarrow M}{X}}} \mathbb{E}_{Z\sim P_\phi(\overset{M\leftarrow}{X})} \left[\log P_\psi(\overset{\rightarrow M}{X} |Z) \right]. \tag{22}$$

To carry out the optimization (18), we will restrict ϕ, ψ, q to a powerful family of parametric families of function approximators, within which we will optimize the objective with gradient descent. While the solutions may not be exact solutions to the nonparametric objective (18), they will still satisfy the bounds (21) and (22)), and—if the family of approximators is sufficiently rich—can come close to turning these into the equalities (19) and (20).

4.2. Choosing Approximating Families

For our method of Neural Predictive Rate–Distortion (NPRD), we choose the approximating families for the encoder ϕ, the decoder ψ, and the distribution q to be certain types of neural networks that are known to provide strong and general models of sequences and distributions.

For ϕ and ψ, we use recurrent neural networks with Long Short Term Memory (LSTM) cells [32], widely used for modeling sequential data across different domains. We parameterize the distribution

$P_\phi(Z|\overset{M\leftarrow}{X})$ as a Gaussian whose mean and variance are computed from the past $\overset{M\leftarrow}{X}$: We use an LSTM network to compute a vector $h \in \mathbb{R}^k$ from the past observations $\overset{M\leftarrow}{X}$, and then compute

$$Z \sim \mathcal{N}(W_\mu h, (W_\sigma h)^2 I_{k\times k}), \tag{23}$$

where $W_\mu, W_\sigma \in \mathbb{R}^{k\times k}$ are parameters. While we found Gaussians sufficiently flexible for ϕ, more powerful encoders could be constructed using more flexible parametric families, such as normalizing flows [19,33].

For the decoder ψ, we use a second LSTM network to compute a sequence of vector representations $g_t = \psi(Z, X_{0\ldots t-1})$ ($g_t \in \mathbb{R}^k$) for $t = 0, \ldots M-1$. We compute predictions using the softmax rule

$$P_\psi(X_t = s_i|X_{1\ldots t-1}, Z) \propto \exp((W_o g_t)_i) \tag{24}$$

for each element s_i of the state space $S = \{s_1, ..., s_{|S|}\}$, and $W_o \in \mathbb{R}^{|S|\times k}$ is a parameter matrix to be optimized together with the parameters of the LSTM networks.

For q, we choose the family of Neural Autoregressive Flows [20]. This is a parametric family of distributions that allows efficient estimation of the probability density and its gradients. This method widely generalizes a family of prior methods [19,33,34], offering efficient estimation while surpassing prior methods in expressivity.

4.3. Parameter Estimation and Evaluation

We optimize the parameters of the neural networks expressing ϕ, ψ, q for the objective (18) using Backpropagation and Adam [35], a standard and widely used gradient descent-based method for optimizing neural networks. During optimization, we approximate the gradient by taking a single sample from Z (23) per sample trajectory $X_{-M}, \ldots, X_{M-1}$ and use the reparametrized gradient estimator introduced by Kingma and Welling [36]. This results in an unbiased estimator of the gradient of (18) w.r.t. the parameters of ϕ, ψ, q.

Following standard practice in machine learning, we split the data set of sample time series into three partitions (training set, validation set, and test set). We use the training set for optimizing the parameters as described above. After every pass through the training set, the objective (18) is evaluated on the validation set using a Monte Carlo estimate with one sample Z per trajectory; optimization terminates once the value on the validation set does not decrease any more.

After optimizing the parameters on a set of observed trajectories, we estimate rate and prediction loss on the test set. Given parameters for ϕ, ψ, q, we evaluate the PRD objective (18), rate (21), and the prediction loss (22) on the test set by taking, for each time series $X_{-M}...X_{M-1} = \overset{M\leftarrow}{X}\overset{\rightarrow M}{X}$ from the test set, a single sample $z \sim \mathcal{N}(\mu, \sigma^2)$ and computing Monte Carlo estimates for rate

$$\mathbb{E}_X\left[\mathrm{D}_{\mathrm{KL}}(P_\phi(Z|\overset{M\leftarrow}{X}) \,\|\, q(Z))\right] \approx \frac{1}{N}\sum_{X_{-M\ldots M}\in TestData} \log\frac{p_{\mathcal{N}(\mu,\sigma^2)}(z)}{q(z)}, \tag{25}$$

where $p_{\mathcal{N}(\mu,\sigma^2)}(z)$ is the Gaussian density with μ, σ^2 computed from $\overset{M\leftarrow}{X}$ as in (23), and prediction loss

$$-\mathbb{E}_{Z\sim\phi(\overset{M\leftarrow}{X})}\left[\log P_\psi(\overset{\rightarrow M}{X}|Z)\right] \approx -\frac{1}{N}\sum_{X_{-M\ldots M}\in TestData} \log P_\psi(\overset{\rightarrow M}{X}|Z). \tag{26}$$

Thanks to (21) and (22), these estimates are guaranteed to be *upper bounds* on the true rate and prediction loss achieved by the code $Z \sim \mathcal{N}(\mu, \sigma^2)$, up to sampling error introduced into the Monte Carlo estimation by sampling z and the finite size of the test set.

It is important to note that this sampling error is different from the overfitting issue affecting OCF: Our Equations (25) and (26)) provide *unbiased* estimators of upper bounds, whereas overfitting *biases* the values obtained by OCF. Given that Neural Predictive Rate–Distortion provably provide upper bounds on rate and prediction loss (up to sampling error), one can objectively compare the quality of different estimation methods: among methods that provide upper bounds, the one that provides the lowest such bound for a given problem is the one giving results closest to the true curve.

Estimating Predictiveness

Given the estimate for prediction loss, we estimate predictiveness $\mathrm{I}[Z, \overset{\rightarrow M}{X}]$ with the following method. We use the encoder network that computes the code vector h (23) to also estimate the marginal probability of the past observation sequence, $P_\eta(\overset{M\leftarrow}{X})$. P_η has support over sequences of length M. Similar to the decoder ψ, we use the vector representations $f_t \in \mathbb{R}^k$ computed by the LSTM encoder after processing $X_{-M\ldots t}$ for $t = -M, \ldots, -1$, and then compute predictions using the softmax rule

$$P_\eta(X_t = s_i | X_{1\ldots t-1}, Z) \propto \exp((W_{o'} f_t)_i), \tag{27}$$

where $W_{o'} \in \mathbb{R}^{|S| \times k}$ is another parameter matrix.

Because we consider stationary processes, we have that the cross-entropy under P_η of $\overset{\rightarrow M}{X}$ is equal to the cross-entropy of $\overset{M\leftarrow}{X}$ under the same encoding distribution: $\mathbb{E}_{X_{\overset{M\leftarrow}{X}\overset{\rightarrow M}{X}}}\left[\log P_\eta(\overset{\rightarrow M}{X})\right] = -\mathbb{E}_{X_{\overset{M\leftarrow}{X}\overset{\rightarrow M}{X}}}\left[\log P_\eta(\overset{M\leftarrow}{X})\right]$. Using this observation, we estimate the predictiveness $\mathrm{I}[Z, \overset{\rightarrow M}{X}] = \mathrm{H}[\overset{\rightarrow M}{X}] - \mathrm{H}[\overset{\rightarrow M}{X} | Z]$ by the difference between the corresponding cross-entropies on the test set [37]:

$$-\mathbb{E}_{X_{\overset{M\leftarrow}{X}\overset{\rightarrow M}{X}}}\left[\log P_\eta(\overset{\rightarrow M}{X}) - \log P_\psi(\overset{\rightarrow M}{X} | Z)\right], \tag{28}$$

which we approximate using Monte Carlo sampling on the test set as in (25–26).

In order to optimize parameters for estimation of P_η, we add the cross-entropy term $-\mathbb{E}_{X_{\overset{M\leftarrow}{X}\overset{\rightarrow M}{X}}}\left[\log P_\eta(\overset{M\leftarrow}{X})\right]$ to the PRD objective (18) during optimization, so that the full training objective comes out to:

$$\min_{\phi,\psi,q,\eta}\left[\mathbb{E}_{\overset{M\leftarrow}{X}, \overset{\rightarrow M}{X}}\left[-\mathbb{E}_{Z\sim\phi(\overset{M\leftarrow}{X})}\left[\log P_\psi(\overset{\rightarrow M}{X} | Z)\right] + \lambda \cdot \mathrm{D}_{\mathrm{KL}}\left[(P_\phi(Z | \overset{M\leftarrow}{X}) \parallel q(Z))\right] - \log P_\eta(\overset{M\leftarrow}{X})\right]\right]. \tag{29}$$

Again, by Gibbs' inequality and Propositions 1 and 2, this is minimized when P_η represents the true distribution over length-M blocks $P(\overset{M\leftarrow}{X})$, $P_\phi(Z | \overset{M\leftarrow}{X})$ describes an optimal code for the given λ, q is its marginal distribution, and $P_\psi(\overset{\rightarrow M}{X} | Z)$ is the Bayes-optimal decoder. For approximate solutions to this augmented objective, the inequalities (21) and (22) will also remain true due to Propositions 1 and 2.

4.4. Related Work

In (18), we derived a variational formulation of Predictive Rate–Distortion. This is formally related to a variational formulation of the Information Bottleneck that was introduced by [38], who applied it to neural-network based image recognition. Unlike our approach, they used a fixed diagonal Gaussian instead of a flexible parametrized distribution for q. Some recent work has applied similar approaches to the modeling of sequences, employing models corresponding to the objective (18) with $\lambda = 1$ [39–42].

In the neural networks literature, the most commonly used method using variational bounds similar to Equation (18) is the Variational Autoencoder [36,43], which corresponds to the setting where $\lambda = 1$ and the predicted output is equal to the observed input. The β-VAE [44], a variant of the Variational Autoencoder, uses $\lambda > 1$ (whereas the Predictive Rate–Distortion objective (7) uses $\lambda \in (0, 1)$), and has been linked to the Information Bottleneck by [45].

5. Experiments

We now test the ability of our new method NPRD to estimate rate–distortion curves. Before we apply NPRD to obtain the first estimates of Predictive Rate–Distortion for natural language in Section 6, we validate the method on processes whose trade-off curves are analytically known, and compare with OCF.

5.1. Implementation Details

OCF

As discussed in Section 3, OCF is affected by overfitting, and will systematically overestimate the predictiveness achieved at a given rate [6]. To address this problem, we follow the evaluation method used for NPRD, evaluating rate and predictiveness on held-out test data. We partition the available time series data into a training and test set. We use the training set to create the encoder $P(Z|\overset{M\leftarrow}{X})$ using the Blahut–Arimoto algorithm as described by Still et al. [6]. We then use the held-out test set to estimate rate and prediction loss. This method not only enables fair comparison between OCF and NPRD, but also provides a more realistic evaluation, by focusing on the performance of the code Z when deployed on new data samples. For rate, we use the same variational bound that we use for NPRD, stated in Proposition 2:

$$\frac{1}{N}\sum_{\overset{M\leftarrow}{X}\in\text{Test Data}} \mathrm{D}_{\mathrm{KL}}(P(Z|\overset{M\leftarrow}{X})\,\|\,s(Z)), \tag{30}$$

where $P(Z|\overset{M\leftarrow}{X})$ is the encoder created by the Blahut–Arimoto algorithm, and $s(Z)$ is the marginal distribution of Z on the training set. N is the number of sample time series in the test data. In the limit of enough training data, when $s(Z)$ matches the actual population marginal of Z, (30) is an unbiased estimate of the rate. We estimate the prediction loss on the future observations as the empirical cross-entropy, i.e., the variational bound stated in Proposition 1:

$$\frac{1}{N}\sum_{\overset{M\leftarrow}{X},\,\overset{\rightarrow M}{X}\in\text{Test Data}} \mathbb{E}_{Z\sim P(\cdot|\overset{M\leftarrow}{X})}\log P(\overset{\rightarrow M}{X}|Z), \tag{31}$$

where $P(\overset{\rightarrow M}{X}|Z)$ is the decoder obtained from the Blahut–Arimoto algorithm on the training set. Thanks to Propositions 1 and 2, these quantities provide upper bounds, up to sampling error introduced by finiteness of the held-out data. Again, sampling error does not bias the results in either direction, unlike overfitting, which introduces a systematic bias.

Held-out test data may contain sequences that did not occur in the training data. Therefore, we add a pseudo-sequence ω and add pseudo-observations $(\omega, \overset{\rightarrow M}{X})$, $(\overset{M\leftarrow}{X}, \omega)$, (ω, ω) for all observed sequences $\overset{M\leftarrow}{X}\overset{\rightarrow M}{X}$ to the matrix of observed counts that serves as the input to the Blahut–Arimoto algorithm. These pseudo-observations were assigned pseudo-counts γ in this matrix of observed counts; we found that a wide range of values ranging from 0.0001 to 1.0 yielded essentially the same results. When evaluating the codebook on held-out data, previously unseen sequences were mapped to ω.

Neural Predictive Rate–Distortion

For all experiments, we used $M = 15$. Neural networks have hyperparameters, such as the number of units and the step size used in optimization, which affect the quality of approximation depending on properties of the dataset and the function being approximated. Given that NPRD provably provides upper bounds on the PRD objective (11), one can in principle identify the best hyperparameters for a given process by choosing the combination that leads to the lowest estimated upper bounds. As a computationally more efficient method, we defined a range of plausible hyperparameters based both on experience reported in the literature, and considerations of computational efficiency. These parameters are discussed in Appendix A. We then randomly sampled, for each of the processes that we experimented on, combinations of λ and these hyperparameters to run NPRD on. We implemented the model using PyTorch [46].

5.2. Analytically Tractable Problems

We first test NPRD on two processes where the Predictive Rate–Distortion trade-off is analytically tractable. The Even Process [2] is the process of 0/1 IID coin flips, conditioned on all blocks of consecutive ones having even length. Its complexity and excess entropy are both ≈ 0.63 nats. It has infinite Markov order, and Marzen and Crutchfield [2] find that OCF (at $M = 5$) performs poorly. The true Predictive Rate–Distortion curve was computed in [2] using the analytically known ϵ-machine. The Random Insertion Process [2] consists of sequences of uniform coin flips $X_t \in \{0, 1\}$, subject to the constraint that, if X_{t-2} was a 0, then X_t has to be a 1.

We applied NPRD to these processes by training on 3M random trajectories of length 30, and using 3000 additional trajectories for validation and test data. For each process, we ran NPRD 1000 times for random choices of $\lambda \in [0, 1]$. Due to computational constraints, when running OCF, we limited sample size to 3000 trajectories for estimation and as held-out data. Following Marzen and Crutchfield [2], we ran OCF for $M = 1, ..., 5$.

The resulting estimates are shown in Figure 1, together with the analytical rate–distortion curves computed by Marzen and Crutchfield [2]. Individual runs of NPRD show variation (red dots), but most runs lead to results close to the analytical curve (gray line), and strongly surpass the curves computed by OCF at $M = 5$. Bounding the trade-off curve using the sets of runs of NPRD results in a close fit (red line) to the analytical trade-off curves.

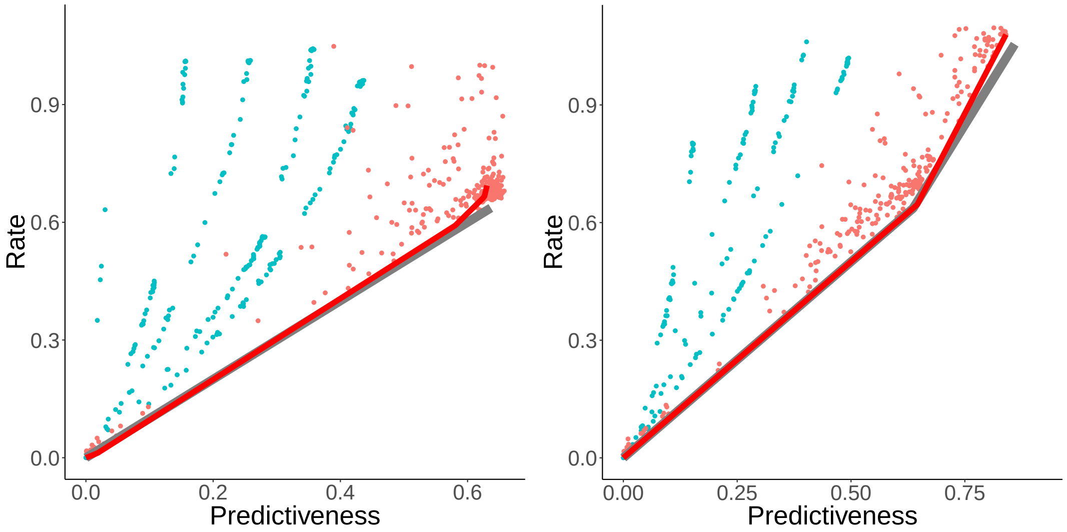


Figure 1. Rate–Distortion for the Even Process (**left**) and the Random Insertion Process (**right**). Gray lines: analytical curves; Red dots: multiple runs of NPRD; red line: trade-off curve computed from NPRD runs; blue: OCF for $M \leq 5$.

Recovering Causal States

Does NPRD lead to interpretable codes Z? To answer this, we further investigated the NPRD approximation to the Random Insertion Process (RIP), obtained in the previous paragraph.

The ϵ-machine was computed by Marzen and Crutchfield [2] and is given in Figure 2 (left). The process has three causal states: State A represents those pasts where the future starts with $1^{2k}0$ $(k = 0, 1, 2, \ldots)$—these are the pasts ending in either 001 or 10111^m $(m = 0, 1, 2, \ldots)$. State B represents those pasts ending in 10—the future has to start with 01 or 11. State C represents those pasts ending in either 00 or 01—the future has to start with $1^{2k+1}0$ $(k = 0, 1, 2, \ldots)$.

The analytical solution to the Predictive Rate–Distortion problem was computed by Marzen and Crutchfield [2]. At $\lambda > 0.5$, the optimal solution collapses A and B into a single codeword, while all three states are mapped to separate codewords for $\lambda \leq 0.5$.

Does NPRD recover this picture? We applied PCA to samples from Z computed at two different values of λ, $\lambda = 0.25$ and $\lambda = 0.6$. The first two principal components of Z are shown in Figure 3. Samples are colored by the causal states corresponding to the pasts of the trajectories that were encoded into the respective points by NPRD. On the left, obtained at $\lambda = 0.6$, the states A and B are collapsed, as expected. On the right, obtained at $\lambda = 0.25$, the three causal states are reflected as distinct modes of Z. Note that, at finite M, a fraction of pasts is ambiguous between the green and blue causal states; these are colored in black and NPRD maps them into a region between the modes corresponding to these states.

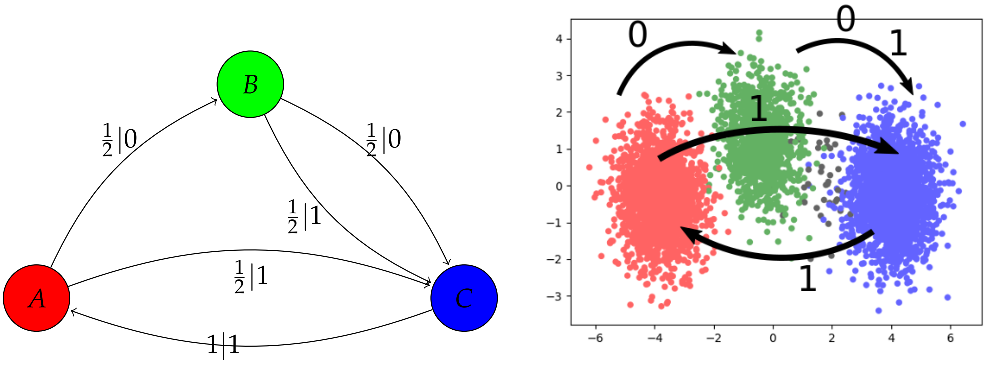


Figure 2. Recovering the ϵ-machine from NPRD. **Left**: The ϵ-machine of the Random Insertion Process, as described by [2]. **Right**: After computing a code Z from a past $x_{-15\ldots-1}$, we recorded which of the three clusters the code moves to when appending the symbol 0 or 1 to the past sequence. The resulting transitions mirror those in the ϵ-machine.

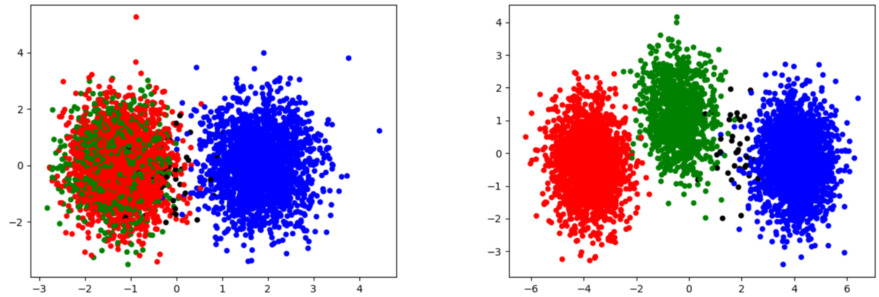


Figure 3. Applying Principal Component Analysis to 5000 sampled codes Z for the Random Insertion Process, at $\lambda = 0.6$ (**left**) and $\lambda = 0.25$ (**right**). We show the first two principal components. Samples are colored according to the states in the ϵ-machine. There is a small number of samples from sequences that, at $M = 15$, cannot be uniquely attributed to any of the states (ambiguous between A and C); these are indicated in black.

In Figure 2 (right), we record, for each of the three modes, to which cluster the distribution of the code Z shifts when a symbol is appended. We restrict to those strings that have nonzero probability for

RIP (no code will ever be needed for other strings). For comparison, we show the ϵ-machine computed by Marzen and Crutchfield [2]. Comparing the two diagrams shows that NPRD effectively recovers the ϵ-machine: the three causal states are represented by the three different modes of Z, and the effect of appending a symbol also mirrors the state transitions of the ϵ-machine.

A Process with Many Causal States

We have seen that NPRD recovers the correct trade-off, and the structure of the causal states, in processes with a small number of causal states. How does it behave when the number of causal states is very large? In particular, is it capable of extrapolating to causal states that were never seen during training?

We consider the following process, which we will call COPY3: $X_{-15}, ..., X_{-1}$, are independent uniform draws from $\{1,2,3\}$, and $X_1 = X_{-1}, ..., X_{15} = X_{-15}$. This process deviates a bit from our usual setup since we defined it only for $t \in \{-15, ..., 15\}$, but it is well-suited to investigating this question: the number of causal states is $3^{15} \approx 14$ million. With exactly the same setup as for the EVEN and RIP processes, NPRD achieved essentially zero distortion on unseen data, even though the number of training samples (3 Million) was far lower than the number of distinct causal states. However, we found that, in this setup, NPRD overestimated the rate. Increasing the number of training samples from 3M to 6M, NPRD recovered codebooks that achieved both almost zero distortion and almost optimal rate, on fresh samples (Figure 4). Even then, the number of distinct causal states is more than twice the number of training samples. These results demonstrate that, by using function approximation, NPRD is capable of extrapolating to unseen causal states, encoding and decoding appropriate codes on the fly.

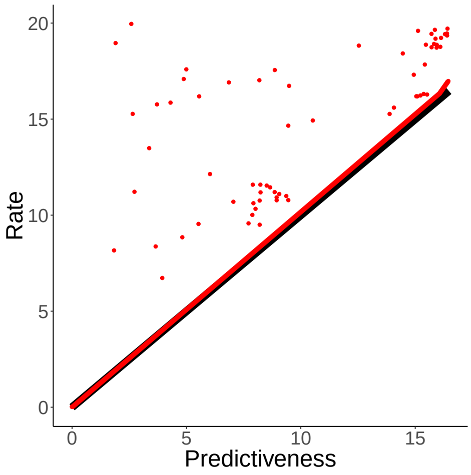


Figure 4. Rate–Distortion for the COPY3 process. We show NPRD samples, and the resulting upper bound in red. The gray line represents the anaytical curve.

Note that one could easily design an optimal decoder and encoder for COPY3 by hand—the point of this experiment is to demonstrate that NPRD is capable of inducing such a codebook purely from data, in a general-purpose, off-the-shelf manner. This contrasts with OCF: without optimizations specific to the task at hand, a direct application of OCF would require brute-force storing of all 14 million distinct pasts and futures.

6. Estimating Predictive Rate–Distortion for Natural Language

We consider the problem of estimating rate–distortion for natural language. Natural language has been a testing ground for information-theoretic ideas since Shannon's work. Much interest has been devoted to estimating the entropy rate of natural language [10,47–49]. Indeed, the information density of language has been linked to human processing effort and to language structure. The word-by-word information content has been shown to impact human processing effort as measured both by per-word reading times [50–52] and by brain signals measured through EEG [53,54]. Consequently, prediction is a fundamental component across theories of human language processing [54]. Relatedly, the Uniform Information Density and Constant Entropy Rate hypotheses [55–57] state that languages order information in sentences and discourse so that the entropy rate stays approximately constant.

The relevance of prediction to human language processing makes the *difficulty* of prediction another interesting aspect of language complexity: Predictive Rate–Distortion describes how much memory of the past humans need to maintain to predict future words accurately. Beyond the entropy rate, it thus forms another important aspect of linguistic complexity.

Understanding the complexity of prediction in language holds promise for enabling a deeper understanding of the nature of language as a stochastic process, and to human language processing. Long-range correlations in text have been a subject of study for a while [58–63]. Recently, Dębowski [64] has studied the excess entropy of language across long-range discourses, aiming to better understand the nature of the stochastic processes underlying language. Koplenig et al. [65] shows a link between traditional linguistic notions of grammatical structure and the information contained in word forms and word order. The idea that predicting future words creates a need to represent the past well also forms a cornerstone of theories of how humans process sentences [66,67].

We study prediction in the range of the words in individual sentences. As in the previous experiments, we limit our computations to sequences of length 30, already improving over OCF by an order of magnitude. One motivation is that, when directly estimating PRD, computational cost has to increase with the length of sequences considered, making the consideration of sequences of hundreds or thousands of words computationally infeasible. Another motivation for this is that we are ultimately interested in Predictive Rate–Distortion as a model of memory in human processing of grammatical structure, formalizing psycholinguistic models of how humans process individual sentences [66,67], and linking to studies of the relation between information theory and grammar [65].

6.1. Part-of-Speech-Level Language Modeling

We first consider the problem of predicting English on the level of Part-of-Speech (POS) tags, using the Universal POS tagset [68]. This is a simplified setting where the vocabulary is small (20 word types), and one can hope that OCF will produce reasonable results. We use the English portions of the Universal Dependencies Project [69] tagged with Universal POS Tags [68], consisting of approximately 586 K words. We used the training portions to estimate NPRD and OCF, and the validation portions to estimate the rate–distortion curve. We used NPRD to generate 350 codebooks for values of λ sampled from $[0, 0.4]$. We were only able to run OCF for $M \leq 3$, as the number of sequences exceeds 10^4 already at $M = 4$.

The PRD curve is shown in Figure 5 (left). In the setting of low rate and high distortion, NPRD and OCF (blue, $M = 1, 2, 3$) show essentially identical results. This holds true until $\mathrm{I}[Z, \overrightarrow{X}] \approx 0.7$, at which point the bounds provided by OCF deteriorate, showing the effects of overfitting. NPRD continues to provide estimates at greater rates.

Figure 5 (center) shows rate as a function of $\log \frac{1}{\lambda}$. Recall that λ is the trade-off-parameter from the objective function (7). In Figure 5 (right), we show rate and the mutual information with the future, as a function of $\log \frac{1}{\lambda}$. As $\lambda \rightarrow 0$, NPRD (red, $M = 15$) continues to discover structure, while OCF (blue, plotted for $M = 1, 2, 3$) exhausts its capacity.

Note that NPRD reports rates of 15 nats and more when modeling with very low distortion. A discrete codebook would need over 3 million distinct codewords for a code of such a rate, exceeding the size of the training corpus (about 500 K words), replicating what we found for the COPY3 process: Neural encoders and decoders can use the geometric structure of the code space to encode generalizations across different dimensions, supporting a very large effective number of distinct possible codes. Unlike discrete codebooks, the geometric structure makes it possible for neural encoders to construct appropriate codes 'on the fly' on new input.

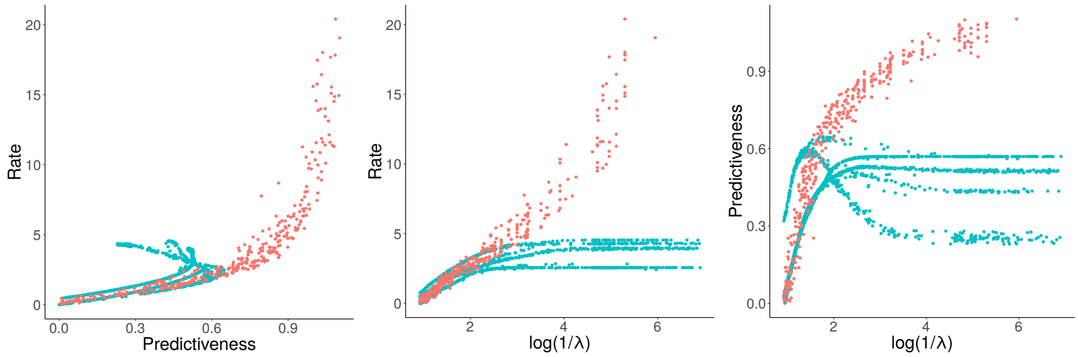


Figure 5. **Left**: Rate-Predictiveness for English POS modeling. Center and right: Rate (**Center**) and Predictiveness (**Right**) on English POS Modeling, as a function of $-\log \lambda$. As $\lambda \rightarrow 0$, NPRD (red, $M = 15$) continues to discover structure, while OCF (blue, plotted for $M = 1, 2, 3$) exhausts its capacity.

6.2. Discussion

Let us now consider the curves in Figure 5 in more detail. Fitting parametric curves to the empirical PRD curves in Figure 5, we find a surprising result that the statistical complexity of English sentences at the POS level appears to be unbounded.

The rate-predictiveness curve (left) shows that, at low rates, predictiveness is approximately proportional to the rate. At greater degrees of predictiveness, the rate grows faster and faster, whereas predictiveness seems to asymptote to ≈ 1.1 nats. The asymptote of predictiveness can be identified with the mutual information between past and future observations, $E_0 := \mathrm{I}[\overset{M\leftarrow}{X}, \overset{\rightarrow M}{X}]$, which is a lower bound on the excess entropy. The rate should asymptote to the statistical complexity. Judging by the curve, natural language—at the time scale we are measuring in this experiment—has a statistical complexity much higher than its excess entropy: at the highest rate measured by NPRD in our experiment, rate is about 20 nats, whereas predictiveness is about 1.1 nats. If these values are correct, then—due to the convexity of the rate-predictivity curve—statistical complexity exceeds the excess entropy by a factor of at least $\frac{20}{1.1}$. Note that this picture agrees qualitatively with the OCF results, which suggest a lower-bound on the ratio of at least $\frac{2.5}{0.6} > 5$.

Now, turning to the other plots in Figure 5, we observe that rate increases at least linearly with $\log \frac{1}{\lambda}$, whereas predictiveness again asymptotes. This is in qualitative agreement with the picture gained from the rate-predictiveness curve.

Let us consider this more quantitatively. Based on Figure 5 (center), we make the ansatz that the map from $\log \frac{1}{\lambda}$ to the rate $R := \mathrm{I}[\overset{M\leftarrow}{X}, Z]$ is superlinear:

$$R = \alpha \left(\log \frac{1}{\lambda} \right)^{\beta}, \tag{32}$$

with $\alpha > 0, \beta > 1$. We fitted $R \approx \left(\log \frac{1}{\lambda} \right)^{1.7}$ ($\alpha = 1, \beta = 1.7$). Equivalently,

$$\frac{1}{\lambda} = \exp\left(\frac{1}{\alpha^{1/\beta}} R^{1/\beta} \right). \tag{33}$$

From this, we can derive expressions for rate $R := \mathrm{I}[\overset{M\leftarrow}{X}, Z]$ and predictiveness $P := \mathrm{I}[Z, \overset{\rightarrow M}{X}]$ as follows. For the solution of Predictive Rate–Distortion (10), we have

$$\frac{\partial P}{\partial \theta} - \lambda \frac{\partial R}{\partial \theta} = 0, \tag{34}$$

where θ is the codebook defining the encoding distribution $P(Z|\overset{M\leftarrow}{X})$, and thus

$$\lambda = \frac{\partial P}{\partial R}. \tag{35}$$

Our ansatz therefore leads to the equation

$$\frac{\partial P}{\partial R} = \exp\left(-\frac{1}{\alpha^{1/\beta}} R^{1/\beta}\right). \tag{36}$$

Qualitatively, this says that predictiveness P asymptotes to a finite value, whereas rate R—which should asymptote to the statistical complexity—is unbounded.

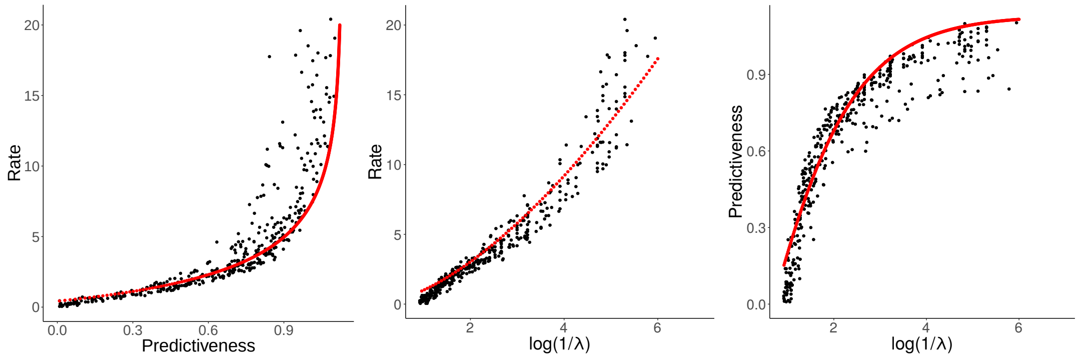


Figure 6. Interpolated values for POS-level prediction of English (compare Figure 5)

Equation (36) has the solution

$$P = C - \alpha\beta \cdot \Gamma\left(\beta, (R/\alpha)^{1/\beta}\right), \tag{37}$$

where Γ is the incomplete Gamma function. Since $\lim_{R\to\infty} P = C$, the constant C has to equal the maximally possible predictiveness $E_0 := \mathrm{I}[\overset{M\leftarrow}{X}, \overset{\rightarrow M}{X}]$.

Given the values fitted above ($\alpha = 1$, $\beta = 1.7$), we found that $E_0 = 1.13$ yielded a good fit. Using (32), this can be extended without further parameters to the third curve in Figure 5. Resulting fits are shown in Figure 6.

Note that there are other possible ways of fitting these curves; we have described a simple one that requires only two parameters $\alpha > 0$, $\beta > 1$, in addition to a guess for the maximal predictiveness E_0. In any case, the results show that natural language shows an approximately linear growth of predictiveness with a rate at small rates, and exploding rates at diminishing returns in predictiveness later.

6.3. Word-Level Language Modeling

We applied NPRD to the problem of predicting English on the level of part-of-speech tags in Section 6.1. We found that the resulting curves were described well by Equation (36). We now consider the more realistic problem of prediction at the level of words, using data from multiple languages. This problem is much closer to the task faced by a human in the process of comprehending text, having to encode prior observations so as to minimize prediction loss on the upcoming words. We will examine whether Equation (36) describes the resulting trade-off in this more realistic setting, and whether it holds across languages.

For the setup, we followed a standard setup for recurrent neural language modeling. The hyperparameters are shown in Table A1. Following standard practice in neural language modeling, we restrict the observation space to the most frequent 10^4 words; other words are replaced by their part-of-speech tag. We do this for simplicity and to stay close to standard practice in natural language processing; NPRD could deal with unbounded state spaces through a range of more sophisticated techniques such as subword modeling and character-level prediction [70,71].

We used data from five diverse languages. For English, we turn to the Wall Street Journal portion of the Penn Treebank [72], a standard benchmark for language modeling, containing about 1.2 million tokens. For Arabic, we pooled all relevant portions of the Universal Dependencies treebanks [73–75]. We obtained 1 million tokens. We applied the same method to construct a Russian corpus [76], obtaining 1.2 million tokens. For Chinese, we use the Chinese Dependency Treebank [77], containing 0.9 million tokens. For Japanese, we use the first 2 million words from a large processed corpus of Japanese business text [78]. For all these languages, we used the predefined splits into training, validation, and test sets.

For each language, we sampled about 120 values of $\log \frac{1}{\lambda}$ uniformly from $[-6, 0]$ and applied NPRD to these. The resulting curves are shown in Figures 7 and 8, together with fitted curves resulting from Equation (36). As can be seen, the curves are qualitatively very similar across languages to what we observed in Figure 6: In all languages, rate initially scales linearly with predictiveness, but diverges as the predictiveness approaches its supremum E_0. As a function of $\log \frac{1}{\lambda}$, rate grows at a slightly superlinear speed, confirming our ansatz (32).

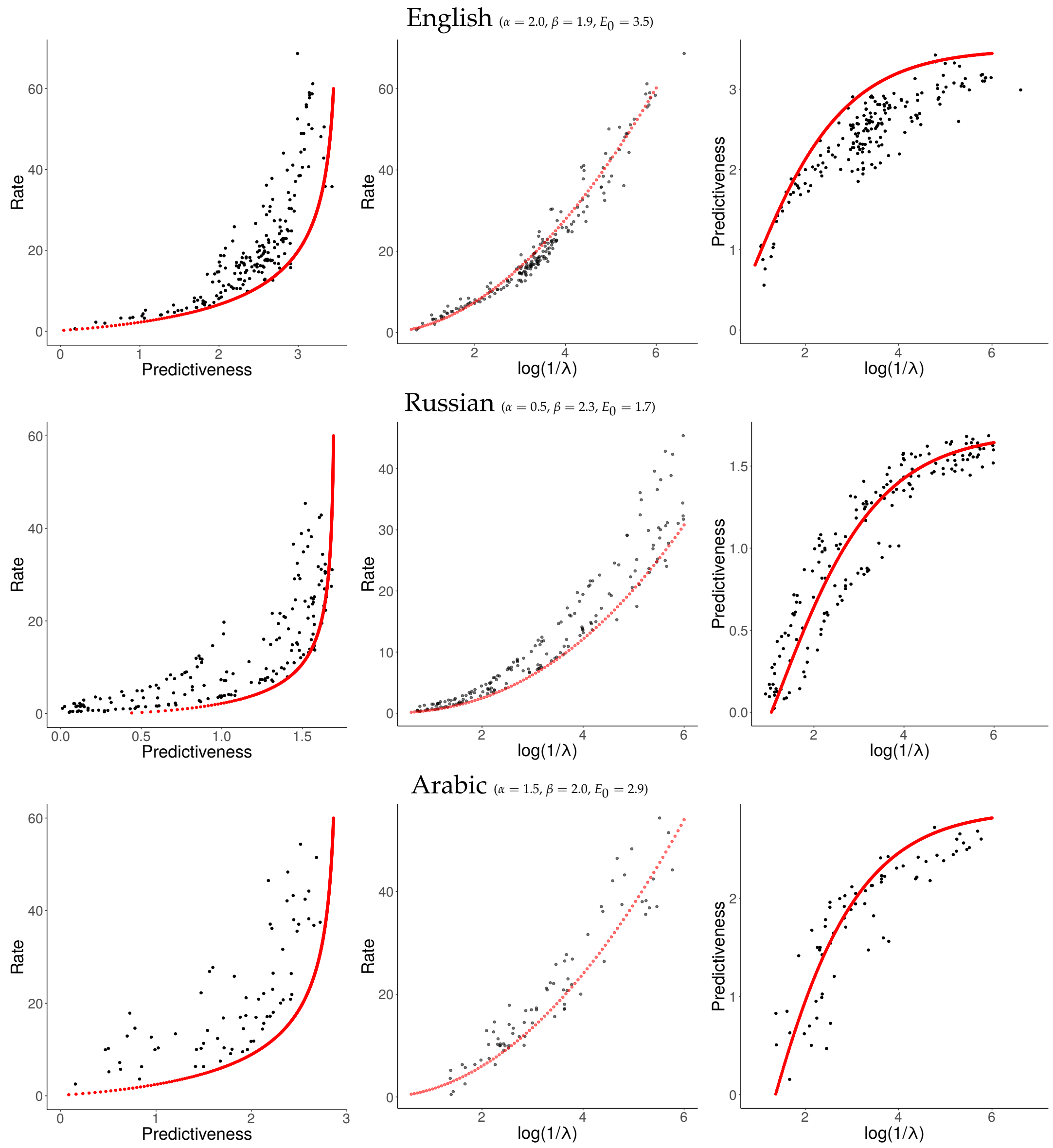
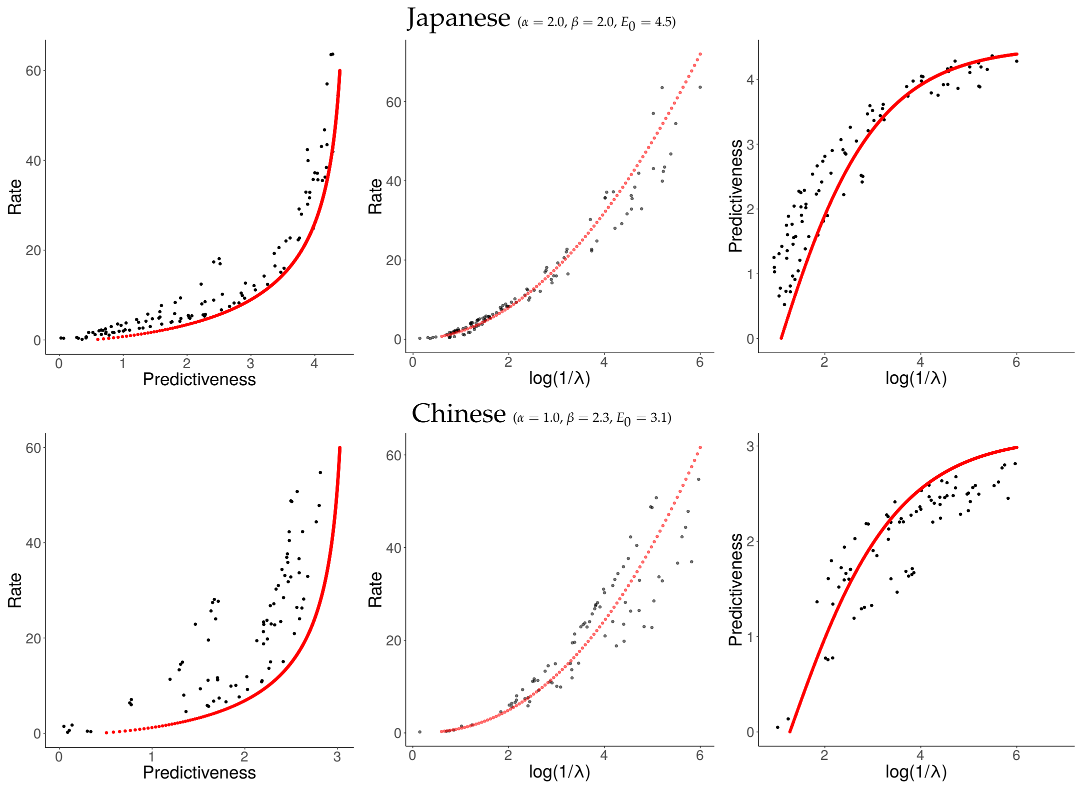


Figure 7. Word-level results.



Japanese $(\alpha = 2.0, \beta = 2.0, E_0 = 4.5)$

Chinese $(\alpha = 1.0, \beta = 2.3, E_0 = 3.1)$

Figure 8. Word-level results (cont.).

These results confirm our results from Section 6.1. At the time scale of individual sentences, Predictive Rate–Distortion of natural language appears to quantitatively follow Equation (36). NPRD reports rates up to ≈ 60 nats, more than ten times the largest values of predictiveness. On the other hand, the growth of rate with predictiveness is relatively gentle in the low-rate regime. We conclude that predicting words in natural language can be approximated with small memory capacity, but more accurate prediction requires very fine-grained memory representations.

6.4. General Discussion

Our analysis of PRD curves for natural language suggests that human language is characterized by very high and perhaps infinite statistical complexity, beyond its excess entropy. In a similar vein, Dębowski [64] has argued that the excess entropy of connected texts in natural language is infinite (in contrast, our result is for isolated sentences). If the statistical complexity of natural language is indeed infinite, then statistical complexity is not sufficiently fine-grained as a complexity metric for characterizing natural language.

We suggest that the PRD curve may form a more natural complexity metric for highly complex processes such as language. Among those processes with infinite statistical complexity, some will have a gentle PRD curve—meaning that they can be well-approximated at low rates—while others will have a steep curve, meaning they cannot be well-approximated at low rates. We conjecture that, although natural language may have infinite statistical complexity, it has a gentler PRD curve than other processes with this property, meaning that achieving a reasonable approximation of the predictive distribution does not require inordinate memory resources.

7. Conclusions

We introduce Neural Predictive Rate–Distortion (NPRD), a method for estimating Predictive Rate–Distortion when only sample trajectories are given. Unlike OCF, the most general prior method, NPRD scales to long sequences and large state spaces. On analytically tractable processes, we show that it closely fits the analytical rate–distortion curve and recovers the causal states of the process. On part-of-speech-level modeling of natural language, it agrees with OCF in the setting of low rate and short sequences; outside these settings, OCF fails due to combinatorial explosion and overfitting, while NPRD continues to provide estimates. Finally, we use NPRD to provide the first estimates of Predictive Rate–Distortion for modeling natural language in five different languages, finding qualitatively very similar curves in all languages.

All code for reproducing the results in this work is available at https://github.com/m-hahn/predictive-rate--distortion.

Author Contributions: Methodology, M.H.; Writing—original draft, M.H. and R.F.

Funding: This research received no external funding.

Acknowledgments: We thank Dan Jurafsky for helpful discussion, and the anonymous reviewers for helpful comments.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

NPRD	Neural Predictive Rate–Distortion
OCF	Optimal Causal Filtering
POS	parts of speech
PRD	Predictive Rate–Distortion
LSTM	Long Short Term Memory
VAE	Variational Autoencoder

Appendix A. Hyperparameters

All hyperparameter choices are shown in Table A1. We defined separate hyperparameter ranges for the three Sections 5.2, 6.1, and 6.3. Guided by the fact that the analytically known processes in Section 5.2 are arguably less complex than natural language, we allowed larger and more powerful models for modeling of natural language (Sections 6.1 and 6.3), in particular word-level modeling (Section 6.3).

In Table A1, hyperparameters are organized into four groups. The first group of parameters is the dimensions of the input embedding and the recurrent LSTM states. The second group of parameters is regularization parameters. We apply dropout [79] to the input and output layers. The third group of parameters is related to the optimization procedure [35]. Neural Autoregressive Flows, used to approximate the marginal q, also have a set of hyperparameters [20]: the length of the flow (1, 2, ...), the type (DSF/Deep Sigmoid Flow or DDSF/Deep Dense Sigmoid Flow), the dimension of the flow (an integer) and the number of layers in the flow (1,2, ...). Larger dimensions, more layers, and longer flows lead to more expressive models; however, they are computationally more expensive.

Training used Early Stopping using the development set, as described in Section 4.3. Models were trained on a TITAN Xp graphics card. On Even Process and Random Insertion Process, NPRD took a median of 10 minutes to process 3M training samples. OCF took less than one minute at $M \leq 5$; however, it does not scale to larger values of M. On English word-level modeling, training took a median number of nine epochs (max 467 epochs) and five minutes (max 126 minutes).

Table A1. NPRD Hyperparameters. See Appendix A for description of the parameters.

	Section 5.2	**Section 6.1**	**Section 6.3**
Embedding Dimension	50, 100	50, 100, 200, 300	150
LSTM Dimension	32	64, 128, 256, 512	256, 512
Dropout rate	0.0, 0.1, 0.2	0.0, 0.1, 0.2	0.1, 0.4
Input Dropout	0	0.0, 0.1, 0.2	0.2
Adam Learning Rate	$\{1,5\} \cdot 10^{-4}$, $\{1,2,4\} \cdot 10^{-3}$	$\{1,5\} \cdot 10^{-4}$, $\{1,2,4\} \cdot 10^{-3}$	0.00005, 0.0001, 0.0005, 0.001
Batch Size	16, 32, 64	16, 32 64	16, 32, 64
Flow Length	1, 2	1,2,3,4,5	1,2,3,4,5
Flow Type	DSF, DDSF	DSF, DDSF	DSF, DDSF
Flow Dimension	32, 64, 128, 512	512	512
Flow Layers	2	2	2

Appendix B. Alternative Modeling Choices

In this section, we investigate the trade-offs involved in alternative modeling choices. It may be possible to use simpler function-approximators for ϕ, ψ, and q, or smaller context windows sizes M, without harming accuracy too much.

First, we investigated the performance of a simple fixed approximation to the marginal q. We considered a diagonal unit-variance Gaussian, as is common in the literature on Variational Autoencoders [36]. We show results in Figure A1. Even with this fixed approximation to q, NPRD continues to provide estimates not far away from the analytical curve. However, comparison with the results obtained from full NPRD (Figure 1) shows that a flexible parameterized approximation still provides considerably better fit.

Second, we investigated whether the use of recurrent neural networks is necessary. As recurrent models such as LSTMs process input sequentially, they cannot be fully parallelized, posing the question of whether they can be replaced by models that can be parallelized. Specifically, we considered Quasi-Recurrent Neural Networks (QRNNs), which combine convolutions with a weak form of recurrence, and which have shown strong results on language modeling benchmarks [80]. We replaced the LSTM encoder and decoder with QRNNs and fit NPRD to the Even Process and the Random Insertion Process. We found that, when using QRNNs, NPRD consistently fitted codes with zero rate for the Even Process and the Random Insertion Process, indicating that the QRNN was failing to extract useful information from the past of these processes. We also found that the cross-entropies of the estimated marginal distribution $P_\eta(\overset{M\leftarrow}{X})$ were considerably worse than when using LSTMs or simple RNNs. We conjecture that, due to the convolutional nature of QRNNs, they cannot model such processes effectively in principle: QRNNs extract representations by pooling embeddings of words or n-grams occurring in the past. When modeling, e.g., the Even Process, the occurrence of specific n-grams is not informative about whether the length of the last block is even or odd; this requires some information about the positions in which these n-grams occur, which is not available to the QRNN, but which is generally available in a more general autoregressive model such as an LSTM.

Third, we varied M, comparing $M = 5, 10$ to the results obtained with $M = 15$. Results are shown in Figure A2. At $M = 5$, NPRD provides estimates similar to OCF (Figure 1). At $M = 10$, the estimates are close to the analytical curve; nonetheless, $M = 15$ yields clearly more accurate results.

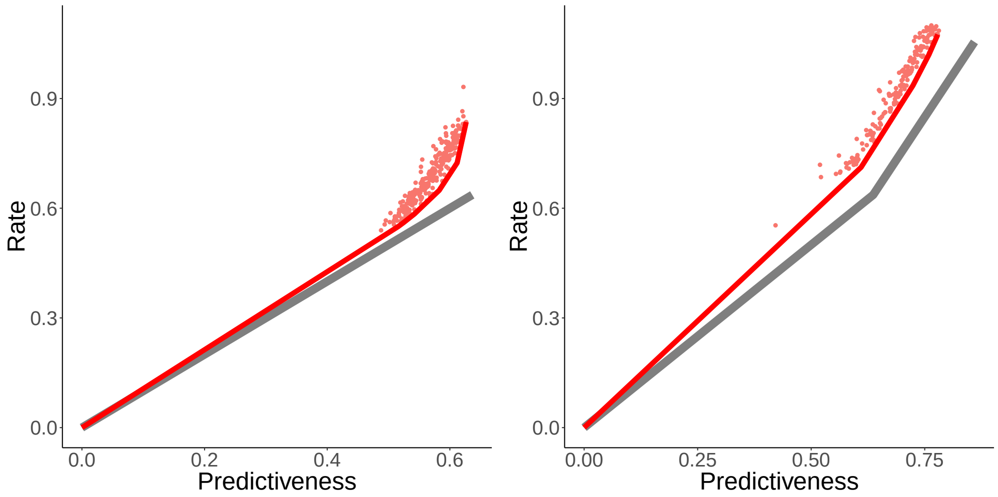


Figure A1. Rate–Distortion for the Even Process (**left**) and the Random Insertion Process (**right**), estimated using a simple diagonal unit Gaussian approximation for q. Gray lines: analytical curves; red dots: multiple runs of NPRD (>200 samples); red line: trade-off curve computed from NPRD runs. Compare Figure 1 for results from full NPRD.

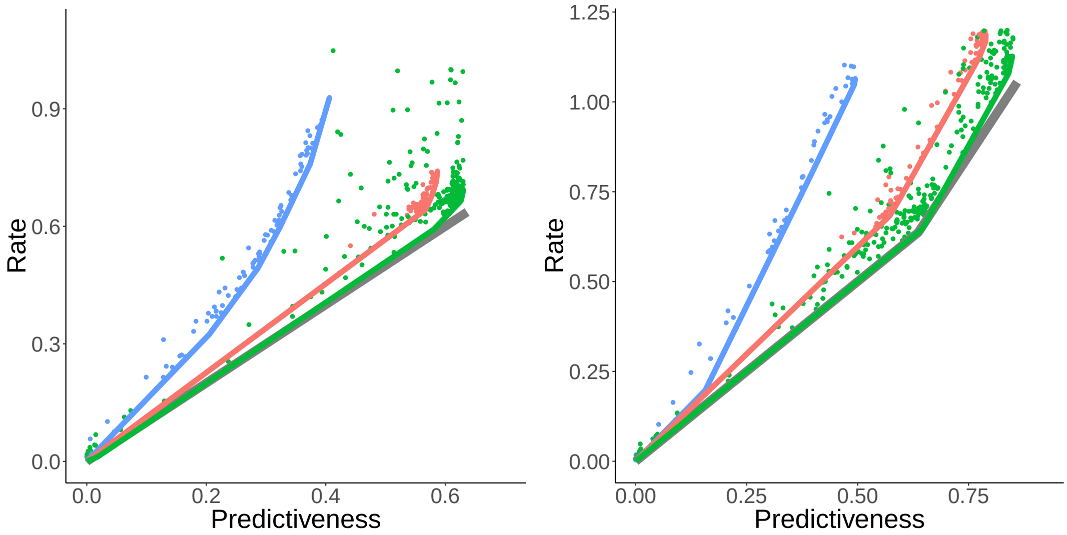
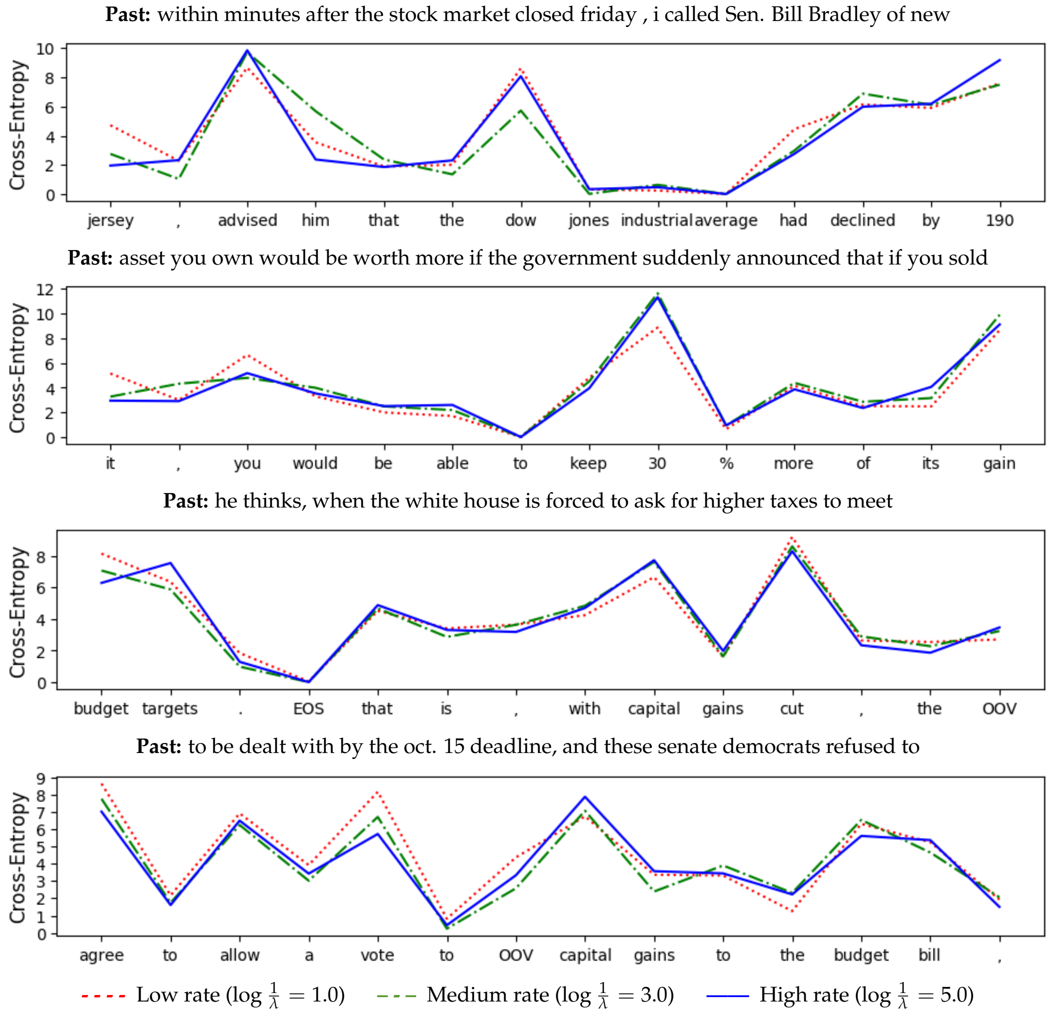


Figure A2. Rate–Distortion for the Even Process (**left**) and the Random Insertion Process (**right**), varying $M = 5$ (blue), 10 (red), 15 (green); gray lines: analytical curves; red dots: multiple runs of NPRD; red line: trade-off curve computed from NPRD runs. Compare Figure 1 for results from full NPRD.

Appendix C. Sample Runs on English Text

In Figure A3, we provide four sample outputs from English word-level modeling with three values of $\log \frac{1}{\lambda}$ (1.0, 3.0, 5.0), corresponding to low (1.0), medium (3.0), and high (5.0) rates (compare Figure 7). We obtained sample sequences by selecting the first 32 sequences $\overset{M\leftarrow}{X}\overset{\rightarrow M}{X}$ (at $M = 15$) from the Penn Treebank validation set, and selected four examples where the variation in cross-entropy values at X_0 was largest between the three models.

Across these samples, models generated at $\log \frac{1}{\lambda} = 5$ show lower cross-entropy on the first future observation X_0, as these codes have higher rates. For instance, in the first sample, the cross-entropy on the first word *Jersey* is lowest for the code with the higher rate; indeed, this word is presumably strongly predicted by the preceding sequence *...Sen. Billd Bradley of New*. Codes with higher rates are better at encoding such predictive information from the past.



Past: within minutes after the stock market closed friday , i called Sen. Bill Bradley of new

Past: asset you own would be worth more if the government suddenly announced that if you sold

Past: he thinks, when the white house is forced to ask for higher taxes to meet

Past: to be dealt with by the oct. 15 deadline, and these senate democrats refused to

Figure A3. Four example outputs from English word-level modeling, with low rate (log $\frac{1}{\lambda}$ = 1; red, dotted), medium rate (log $\frac{1}{\lambda}$ = 3; green, dashed), high rate (log $\frac{1}{\lambda}$ = 5; blue, solid). For each sample, we provide the prior context $\overset{M\leftarrow}{X}$ (**top**), and the per-word cross-entropies (in nats) on the future words $\overset{\rightarrow M}{X}$ (**bottom**).

References

1. Still, S. Information Bottleneck Approach to Predictive Inference. *Entropy* **2014**, *16*, 968–989, doi:10.3390/e16020968.
2. Marzen, S.E.; Crutchfield, J.P. Predictive Rate-Distortion for Infinite-Order Markov Processes. *J. Stat. Phys.* **2016**, *163*, 1312–1338, doi:10.1007/s10955-016-1520-1.
3. Creutzig, F.; Globerson, A.; Tishby, N. Past-future information bottleneck in dynamical systems. *Phys. Rev. E* **2009**, *79*, doi:10.1103/PhysRevE.79.041925.
4. Amir, N.; Tiomkin, S.; Tishby, N. Past-future Information Bottleneck for linear feedback systems. In Proceedings of the 54th IEEE Conference on Decision and Control (CDC), Osaka, Japan, 15–18 December 2015; pp. 5737–5742.
5. Genewein, T.; Leibfried, F.; Grau-Moya, J.; Braun, D.A. Bounded rationality, abstraction, and hierarchical decision-making: An information-theoretic optimality principle. *Front. Robot. AI* **2015**, *2*, 27.
6. Still, S.; Crutchfield, J.P.; Ellison, C.J. Optimal causal inference: Estimating stored information and approximating causal architecture. *Chaos Interdiscip. J. Nonlinear Sci.* **2010**, *20*, 037111, doi:10.1063/1.3489885.
7. Józefowicz, R.; Vinyals, O.; Schuster, M.; Shazeer, N.; Wu, Y. Exploring the Limits of Language Modeling. *arXiv* **2016**, arXiv:1602.02410.

8. Merity, S.; Keskar, N.S.; Socher, R. An analysis of neural language modeling at multiple scales. *arXiv* **2018**, arXiv:1803.08240.
9. Dai, Z.; Yang, Z.; Yang, Y.; Cohen, W.W.; Carbonell, J.; Le, Q.V.; Salakhutdinov, R. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *arXiv* **2019**, arXiv:1901.02860.
10. Takahashi, S.; Tanaka-Ishii, K. Cross Entropy of Neural Language Models at Infinity—A New Bound of the Entropy Rate. *Entropy* **2018**, *20*, 839.
11. Ogunmolu, O.; Gu, X.; Jiang, S.; Gans, N. Nonlinear systems identification using deep dynamic neural networks. *arXiv* **2016**, arXiv:1610.01439.
12. Laptev, N.; Yosinski, J.; Li, L.E.; Smyl, S. Time-series extreme event forecasting with neural networks at uber. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 11 August 2017; pp. 1–5.
13. Meyer, P.; Noblet, V.; Mazzara, C.; Lallement, A. Survey on deep learning for radiotherapy. *Comput. Biol. Med.* **2018**, *98*, 126–146.
14. Hundman, K.; Constantinou, V.; Laporte, C.; Colwell, I.; Soderstrom, T. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 387–395.
15. White, G.; Palade, A.; Clarke, S. Forecasting qos attributes using lstm networks. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
16. Woo, J.; Park, J.; Yu, C.; Kim, N. Dynamic model identification of unmanned surface vehicles using deep learning network. *Appl. Ocean Res.* **2018**, *78*, 123–133.
17. Sirignano, J.; Cont, R. Universal features of price formation in financial markets: perspectives from Deep Learning. *arXiv* **2018**, arXiv:1803.06917.
18. Mohajerin, N.; Waslander, S.L. Multistep Prediction of Dynamic Systems With Recurrent Neural Networks. *IEEE Transa. Neural Netw. Learn. Syst.* **2019**, doi:10.1109/TNNLS.2019.2891257.
19. Rezende, D.J.; Mohamed, S. Variational inference with normalizing flows. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 37, pp. 1530–1538.
20. Huang, C.W.; Krueger, D.; Lacoste, A.; Courville, A. Neural Autoregressive Flows. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 2083–2092.
21. Tishby, N.; Pereira, F.C.; Bialek, W. The Information Bottleneck Method. In Proceedings of the Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 22–24 September 1999; pp. 368–377.
22. Harremoës, P.; Tishby, N. The information bottleneck revisited or how to choose a good distortion measure. In Proceedings of the IEEE International Symposium on Information Theory, Nice, France, 24–29 June 2007; pp. 566–570.
23. Feldman, D.P.; Crutchfield, J.P. Synchronizing to Periodicity: The Transient Information and Synchronization Time of Periodic Sequences. *Adv. Complex Syst.* **2004**, *7*, 329–355, doi:10.1142/S0219525904000196.
24. Crutchfield, J.P.; Young, K. Inferring statistical complexity. *Phys. Rev. Lett.* **1989**, *63*, 105–108, doi:10.1103/PhysRevLett.63.105.
25. Grassberger, P. Toward a quantitative theory of self-generated complexity. *Int. J. Theor. Phys.* **1986**, *25*, 907–938, doi:10.1007/BF00668821.
26. Löhr, W. Properties of the Statistical Complexity Functional and Partially Deterministic HMMs. *Entropy* **2009**, *11*, 385–401, doi:10.3390/e110300385.
27. Clarke, R.W.; Freeman, M.P.; Watkins, N.W. Application of computational mechanics to the analysis of natural data: An example in geomagnetism. *Phys. Rev. E* **2003**, *67*, 016203.
28. Singh, S.P.; Littman, M.L.; Jong, N.K.; Pardoe, D.; Stone, P. Learning predictive state representations. In Proceedings of the 20th International Conference on Machine Learning (ICML-03), Washington, DC, USA, 21–24 August 2003; pp. 712–719.
29. Singh, S.; James, M.R.; Rudary, M.R. Predictive state representations: A new theory for modeling dynamical systems. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*; AUAI Press: Arlington, VA, USA, 2004; pp. 512–519.
30. Jaeger, H. *Discrete-Time, Discrete-Valued Observable Operator Models: A Tutorial*; GMD-Forschungszentrum Informationstechnik: Darmstadt, Germany, 1998.

31. Rubin, J.; Shamir, O.; Tishby, N. Trading value and information in MDPs. In *Decision Making with Imperfect Decision Makers*; Springer: Berlin, Germany, 2012; pp. 57–74.
32. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.
33. Kingma, D.P.; Salimans, T.; Jozefowicz, R.; Chen, X.; Sutskever, I.; Welling, M. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2016; pp. 4743–4751.
34. Papamakarios, G.; Pavlakou, T.; Murray, I. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2017; pp. 2338–2347.
35. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
36. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. In Proceedings of the International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.
37. McAllester, D.; Statos, K. Formal Limitations on the Measurement of Mutual Information. *arXiv* **2018**, arXiv:1811.04251.
38. Alemi, A.A.; Fischer, I.; Dillon, J.V.; Murphy, K. Deep Variational Information Bottleneck. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
39. Grathwohl, W.; Wilson, A. Disentangling space and time in video with hierarchical variational auto-encoders. *arXiv* **2016**, arXiv:1612.04440.
40. Walker, J.; Doersch, C.; Gupta, A.; Hebert, M. An uncertain future: Forecasting from static images using variational autoencoders. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin, Germany, 2016; pp. 835–851.
41. Fraccaro, M.; Kamronn, S.; Paquet, U.; Winther, O. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *Advances in Neural Information Processing Systems*; 2017; pp. 3601–3610.
42. Hernández, C.X.; Wayment-Steele, H.K.; Sultan, M.M.; Husic, B.E.; Pande, V.S. Variational encoding of complex dynamics. *Phys. Rev. E* **2018**, *97*, 062412.
43. Bowman, S.R.; Vilnis, L.; Vinyals, O.; Dai, A.M.; Jozefowicz, R.; Bengio, S. Generating Sentences from a Continuous Space. In Proceedings of the CoNLL, Berlin, Germany, 11–12 August 2016.
44. Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; Lerchner, A. β-VAE: Learning basic visual concepts with a constrained variational framework. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
45. Burgess, C.; Higgins, I.; Pal, A.; Matthey, L.; Watters, N.; Desjardins, G.; Lerchner, A. Understanding disentangling in β-VAE. *arXiv* **2018**, arXiv:1804.03599.
46. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic Differentiation in PyTorch 2017. Available online: https://openreview.net/forum?id=BJJsrmfCZ (accessed on 28 June 2019).
47. Shannon, C.E. Prediction and entropy of printed English. *Bell Syst. Tech. J.* **1951**, *30*, 50–64.
48. Takahira, R.; Tanaka-Ishii, K.; Dębowski, Ł. Entropy rate estimates for natural language—A new extrapolation of compressed large-scale corpora. *Entropy* **2016**, *18*, 364.
49. Bentz, C.; Alikaniotis, D.; Cysouw, M.; Ferrer-i Cancho, R. The entropy of words—Learnability and expressivity across more than 1000 languages. *Entropy* **2017**, *19*, 275.
50. Hale, J. A Probabilistic Earley Parser as a Psycholinguistic Model. In Proceedings of the NAACL, Pittsburgh, PA, USA, 1–7 June 2001; Volume 2, pp. 159–166.
51. Levy, R. Expectation-based syntactic comprehension. *Cognition* **2008**, *106*, 1126–1177, doi:10.1016/j.cognition.2007.05.006.
52. Smith, N.J.; Levy, R. The effect of word predictability on reading time is logarithmic. *Cognition* **2013**, *128*, 302–319, doi:10.1016/j.cognition.2013.02.013.
53. Frank, S.L.; Otten, L.J.; Galli, G.; Vigliocco, G. Word surprisal predicts N400 amplitude during reading. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 4–9 August 2013; Volume 2, pp. 878–883.
54. Kuperberg, G.R.; Jaeger, T.F. What do we mean by prediction in language comprehension? *Lang. Cogn. Neurosci.* **2016**, *31*, 32–59.
55. Fenk, A.; Fenk, G. Konstanz im Kurzzeitgedächtnis—Konstanz im sprachlichen Informationsfluß. *Z. Exp. Angew. Psychol.* **1980**, *27*, 400–414.

56. Genzel, D.; Charniak, E. Entropy rate constancy in text. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002.
57. Jaeger, T.F.; Levy, R.P. Speakers optimize information density through syntactic reduction. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 3–6 December 2007; pp. 849–856.
58. Schenkel, A.; Zhang, J.; Zhang, Y.C. Long range correlation in human writings. *Fractals* **1993**, *1*, 47–57.
59. Ebeling, W.; Pöschel, T. Entropy and long-range correlations in literary English. *EPL (Europhys. Lett.)* **1994**, *26*, 241.
60. Ebeling, W.; Neiman, A. Long-range correlations between letters and sentences in texts. *Phys. A Stat. Mech. Appl.* **1995**, *215*, 233–241.
61. Altmann, E.G.; Cristadoro, G.; Degli Esposti, M. On the origin of long-range correlations in texts. *Proc. Natl. Acad. Sci. USA* **2012**, *109*, 11582–11587.
62. Yang, T.; Gu, C.; Yang, H. Long-range correlations in sentence series from A Story of the Stone. *PLoS ONE* **2016**, *11*, e0162423.
63. Chen, H.; Liu, H. Quantifying evolution of short and long-range correlations in Chinese narrative texts across 2000 years. *Complexity* **2018**, *2018*, 9362468, doi:10.1155/2018/9362468.
64. Dębowski, Ł. Is natural language a perigraphic process? The theorem about facts and words revisited. *Entropy* **2018**, *20*, 85.
65. Koplenig, A.; Meyer, P.; Wolfer, S.; Mueller-Spitzer, C. The statistical trade-off between word order and word structure–Large-scale evidence for the principle of least effort. *PLoS ONE* **2017**, *12*, e0173614.
66. Gibson, E. Linguistic complexity: locality of syntactic dependencies. *Cognition* **1998**, *68*, 1–76.
67. Futrell, R.; Levy, R. Noisy-context surprisal as a human sentence processing cost model. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April 2017; pp. 688–698.
68. Petrov, S.; Das, D.; McDonald, R.T. A Universal Part-of-Speech Tagset. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), Istanbul, Turkey, 23–25 May 2012; pp. 2089–2096.
69. Nivre, J.; Agic, Z.; Ahrenberg, L.; Antonsen, L.; Aranzabe, M.J.; Asahara, M.; Ateyah, L.; Attia, M.; Atutxa, A.; Augustinus, L.; et al. Universal Dependencies 2.1 2017. Available online: https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2515 (accessed on 28 June 2019).
70. Kim, Y.; Jernite, Y.; Sontag, D.; Rush, A.M. Character-aware neural language models. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
71. Luong, M.T.; Manning, C.D. Achieving open vocabulary neural machine translation with hybrid word-character models. *arXiv* **2016**, arXiv:1604.00788.
72. Marcus, M.P.; Marcinkiewicz, M.A.; Santorini, B. Building a large annotated corpus of English: The Penn Treebank. *Comput. Linguist.* **1993**, *19*, 313–330.
73. Nivre, J.; de Marneffe, M.C.; Ginter, F.; Goldberg, Y.; Hajic, J.; Manning, C.D.; McDonald, R.T.; Petrov, S.; Pyysalo, S.; Silveira, N.; et al. Universal Dependencies v1: A Multilingual Treebank Collection. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portorož, Slovenia, 23–28 May 2016.
74. Maamouri, M.; Bies, A.; Buckwalter, T.; Mekki, W. The penn arabic treebank: Building a large-scale annotated arabic corpus. In Proceedings of the NEMLAR Conference on Arabic Language Resources and Tools, Cairo, Egypt, 27–29 March 2004; Volume 27, pp. 466–467.
75. Hajic, J.; Smrz, O.; Zemánek, P.; Šnaidauf, J.; Beška, E. Prague Arabic dependency treebank: Development in data and tools. In Proceedings of the NEMLAR Internaional Conference on Arabic Language Resources and Tools, Cairo, Egypt, 22–23 September 2004; pp. 110–117.
76. Dyachenko, P.B.; Iomdin, L.L.; Lazurskiy, A.V.; Mityushin, L.G.; Podlesskaya, O.Y.; Sizov, V.G.; Frolova, T.I.; Tsinman, L.L. Sovremennoe sostoyanie gluboko annotirovannogo korpusa tekstov russkogo yazyka (SinTagRus). *Trudy Instituta Russkogo Yazyka im. VV Vinogradova* **2015**, *10*, 272–300.
77. Che, W.; Li, Z.; Liu, T. *Chinese Dependency Treebank 1.0 LDC2012T05*; Web Download; Linguistic Data Consortium: Philadelphia, PA, USA, 2012.
78. Graff, D.; Wu, Z. *Japanese bUsiness News Text*; LDC95T8; Linguistic Data Consortium: Philadelphia, PA, USA, 1995.

79. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
80. Bradbury, J.; Merity, S.; Xiong, C.; Socher, R. Quasi-recurrent neural networks. In Proceedings of the ICLR 2017, Toulon, France, 24–26 April 2017.