Locally Typical Sampling

Clara Meister[1]   **Tiago Pimentel**[2]   **Gian Wiher**[1]   **Ryan Cotterell**[1,2]
[1]ETH Zürich   [2]University of Cambridge
clara.meister@inf.ethz.ch tp472@cam.ac.uk
gian.wiher@inf.ethz.ch ryan.cotterell@inf.ethz.ch

Abstract

Today's probabilistic language generators fall short when it comes to producing coherent and fluent text despite the fact that the underlying models perform well under standard metrics, e.g., perplexity. This discrepancy has puzzled the language generation community for the last few years. In this work, we posit that the abstraction of natural language generation as a discrete stochastic process—which allows for an information-theoretic analysis—can provide new insights into the behavior of probabilistic language generators, e.g., why high-probability texts can be dull or repetitive. Humans use language as a means of communicating information, aiming to do so in a simultaneously efficient and error-minimizing manner; in fact, psycholinguistics research suggests humans choose each word in a string with this subconscious goal in mind. We formally define the set of strings that meet this criterion: those for which each word has an information content close to the *expected* information content, i.e., the conditional entropy of our model. We then propose a simple and efficient procedure for enforcing this criterion when generating from probabilistic models, which we call **locally typical sampling**. Automatic and human evaluations show that, in comparison to nucleus and top-k sampling, locally typical sampling offers competitive performance (in both abstractive summarization and story generation) in terms of quality while consistently reducing degenerate repetitions.

1 Introduction

Modern probabilistic models have repeatedly demonstrated their prowess at modeling natural language, placing high probability on held-out corpora from many different domains (Brown et al., 2020; Hoffmann et al., 2022; Chowdhery et al., 2022). Yet when used as text generators, their performance is far from perfect. One of the largest determinants of the generated text's quality is the choice of **decoding strategy**—i.e., the decision rule used to extract strings from a model. Perhaps surprisingly, for many language generation tasks, decoding strategies which aim to find the highest-probability strings produce text that is undesirable (Holtzman et al., 2020; See et al., 2019; Eikema and Aziz, 2020; Zhang et al., 2021; DeLucia et al., 2021). For instance, Stahlberg and Byrne (2019) report that in their neural machine translation experiments, the highest-probability string is usually the empty string. On the other hand, stochastic strategies, which take random samples from the model, often lead to text with better qualitative properties (Fan et al., 2018; Holtzman et al., 2020; Basu et al., 2021). However, stochastic strategies still have a host of other problems, while not entirely dispensing with those seen in maximization-based approaches.[1]

At first glance, it is unintuitive that high-probability strings are often neither desirable nor human-like. Due to this pathology, a number of works have concluded that there must be faults in the training objective or architecture of the probabilistic models behind language generators (Welleck et al., 2020; Guan et al., 2020; Li et al., 2020, *inter alia*). Yet, this conclusion is at odds with these models' performance in terms of other metrics. The fact that modern models can place high probability on held-out text suggests that they provide good estimates (in at least some aspects) of the probability distribution underlying human language. We posit that looking at language generation through an information-theoretic lens

---

[1]While maximization-based strategies can produce text which is generic or degenerate, stochastic strategies occasionally produce nonsensical text. Both types of strategies tend to eventually fall into repetitive loops.

may shed light on this paradox.

Communication via natural language can intuitively be cast in information-theoretic terms. Indeed, there is a long history of studying language through the lens of information theory (Shannon, 1948, 1951; Hale, 2001; Piantadosi et al., 2011; Pimentel et al., 2020, *inter alia*). In this paradigm, linguistic strings are messages used to convey information, and their information content can be quantified as a function of their probability of being uttered—often driven by context. Assuming that humans use language in order to transmit information in an efficient yet robust manner (Zaslavsky et al., 2018; Gibson et al., 2019), the subset of strings typically used by humans should encode information at some (perhaps near-optimal) rate.[2] In fact, prior works studying the uniform information density hypothesis (Levy and Jaeger, 2007; Mahowald et al., 2013) empirically observed this property in humans' use of natural language.

These insights lead us to re-think what it means to be a probabilistic language generator. First, we contend that language generators, in some cases, can be thought of as discrete stochastic processes. This in turn, allows us to cleanly define typicality (and the typical set) for these processes. We argue, however, that due to discrepancies between the model behind these generators and the true distribution over natural language strings, directly sampling from the typical set is not a good idea. Indeed, for language generators that do not use an end-of-string (EOS) state, this is exactly what is done by ancestral sampling—a decoding strategy not known for providing high-quality text. Inspired by research on human sentence processing, we then define the more restrictive notion of *local* typicality, and argue that if we want text generated from a model to be "human-like," we should perhaps enforce this information-theoretic criterion in generations ourselves. To this end, we develop a new algorithm, which we call **locally typical sampling**. Concretely, we hypothesize that for text to be perceived as natural, each word should have an information content close to its *expected* information content given prior context. When sampling from probabilistic language generators, we should limit our options to strings that adhere to this property. In experiments on abstractive summarization

and story generation, we observe that, compared to nucleus and top-k sampling: (i) locally typical sampling reduces the number of degenerate repetitions, giving a REP value (Welleck et al., 2020) on par with human text, and (ii) text generated using typical sampling is generally closer in quality to that of human text.[3]

2 Two Views of Language Modeling

In this work, we discuss language models[4] in an information-theoretic light. Our first step towards this goal is to re-frame their presentation. Concretely, we put forth that there are actually two lenses through which we can view language modeling productively. Under the traditional lens, we can think of a language model as a distribution over full strings: a language model constitutes the distribution of a single string-valued random variable. Under an alternative lens, we can think of a language model as a discrete stochastic process: a collection of indexed random variables. We compare and contrast these views formally, and then show how to use the language process view to derive a new sampling algorithm in §5.

2.1 A Single String-Valued Random Variable

We codify the traditional view of language modeling in the following definition. Let $\mathcal{V}$ be an alphabet—a non-empty, finite set.

Definition 2.1 (Language Model). *A **language model** p is a probability distribution over all strings $\boldsymbol{y} \in \mathcal{V}^*$.[5] Under this view, we can think of a language model as describing a single $\mathcal{V}^*$-valued random variable.*

Under Definition 2.1, it is common to express a language model in the following factorized form

$$p(\boldsymbol{y} = y_1 \cdots y_T) = \prod_{t=1}^{T} p(y_t \mid \boldsymbol{y}_{<t}) \qquad (1)$$

where we define $\boldsymbol{y}_{<t} \stackrel{\text{def}}{=} \langle y_0, \ldots, y_{t-1} \rangle$ with the padding $y_0 = \text{BOS}$ as a distinguished beginning-of-sequence symbol. Through the chain rule of probability, we can *always* factorize a model as in Eq. (1).

---

[2]Information rate may be defined with respect to time (as is the case with spoken language) or with respect to a specific linguistic unit, such as a word (as is the case with text).

[3]An implementation of typical sampling can be found in the Hugging Face's Transformers library (Wolf et al., 2020).

[4]Here we use the term language model to refer to any (valid) probability distribution over natural language strings. We subsequently specify the necessary conditions for validity. Note that this distribution may also be conditioned on an input.

[5]The Kleene closure of a set $\mathcal{V}$ is defined as $\mathcal{V}^* \stackrel{\text{def}}{=} \bigcup_{n=0}^{\infty} \mathcal{V}^n$.

The process which produces such a factorization is called **local normalization**.[6] However, with local normalization, we encounter a subtlety: one has to define each conditional probability $p(y_t \mid \boldsymbol{y}_{<t})$ not over $\mathcal{V}$, but rather over the augmented set $\overline{\mathcal{V}} \stackrel{\text{def}}{=} \mathcal{V} \cup \{\text{EOS}\}$, i.e., where we have added the distinguished end-of-string symbol EOS. Why? Because *without* EOS, it would be impossible to normalize the language model, i.e., have it sum to 1.[7]

2.2 A Discrete Stochastic Process

Interestingly, the factorization in Eq. (1) suggests that we might view language models, not as a single string-valued random variable, but rather as a collection of random variables $\{Y_t\}_{t=1}^{\infty}$, i.e., as a discrete stochastic process.[8] Under this view, we arrive at the following definition of what we term a language *process*, to distinguish it from the definition of a language model given above.

Definition 2.2 (Language Process). *A language process over $\mathcal{V}$ is a discrete stochastic process $\boldsymbol{Y} = \{Y_t\}_{t=1}^{\infty}$ where each Y_t is $\overline{\mathcal{V}}$-valued. The process is described by a distribution p, and we denote its conditional distribution as $p(Y_t = y_t \mid \boldsymbol{Y}_{<t} = \boldsymbol{y}_{<t})$ for $t > 0$. In slight abuse of notation but out of convention, we take Y_t for $t \leq 0$ to be BOS, i.e., conditioning p on just BOS signifies the initial distribution of the process.*

Definition 2.2 is very generic. In words, it just says that a language process is any discrete process where we sample a new word[9] given the previously sampled words. The first question that naturally comes to mind is when the definitions of a language model and a language process coincide. As it turns out, there is a simple answer.

Definition 2.3 (Tightness). *Let $\boldsymbol{Y} = \{Y_t\}_{t=1}^{\infty}$ be a language process over alphabet $\mathcal{V}$ with distribution p. A language process is **tight** (Booth*

---

[6]The ubiquity of Eq. (1) has led some authors to *defining* language models in the locally normalized form, even though globally normalized language models are also perfectly fine to consider (Goyal et al., 2019).

[7]Some authors erroneously omit EOS from their definition. However, we *require* a distinguished symbol EOS to be able to locally normalize the language model and make it a valid probability distribution.

[8]This process is discrete both in time and in value.

[9]One could just as easily define a language process over subwords, morphemes or characters.

and Thompson, 1973) if and only if

$$\sum_{\boldsymbol{y} \in (\mathcal{V}^* \otimes \{\text{EOS}\})} \prod_{t=1}^{|\boldsymbol{y}|} p(Y_t = y_t \mid \boldsymbol{Y}_{<t} = \boldsymbol{y}_{<t}) = 1 \tag{2}$$

where $A \otimes B \stackrel{\text{def}}{=} \{\boldsymbol{ab} \mid \boldsymbol{a} \in A, \boldsymbol{b} \in B\}$.

In words, tightness says that a language process must not leak probability mass to infinite strings. Because a language model must be a (valid) probability distribution, it must also be tight.

Proposition 2.4. *Let $\boldsymbol{Y} = \{Y_t\}_{t=1}^{\infty}$ be a language process over alphabet $\mathcal{V}$ with distribution p and let $p_t \stackrel{\text{def}}{=} \frac{\sum_{\boldsymbol{y} \in (\mathcal{V}^{t-1} \otimes \{\text{EOS}\})} \prod_{i=1}^{|\boldsymbol{y}|} p(Y_i = y_i \mid \boldsymbol{Y}_{<i} = \boldsymbol{y}_{<i})}{\sum_{\boldsymbol{y} \in \mathcal{V}^{t-1}} \prod_{i=1}^{|\boldsymbol{y}|} p(Y_i = y_i \mid \boldsymbol{Y}_{<i} = \boldsymbol{y}_{<i})}$. Then $\boldsymbol{Y}$ is tight if and only if $p_t = 1$ for some $0 < t < \infty$ or $\sum_{t=1}^{\infty} p_t \to \infty$.*

Proof. Note that p_t is the probability of sampling EOS at *exactly* step t given that the history of the string is of length $(t-1)$.

- **Case 1:** Suppose $p_t = 1$ for some $0 < t < \infty$. Then, $\boldsymbol{Y}$ is clearly tight as no probability mass is leaked to strings beyond length t, where $t < \infty$.

- **Case 2:** Now suppose $p_t < 1$ for all t. In this case, we have that the probability of all infinite-length strings is given by $\prod_{t=1}^{\infty} (1 - p_t)$. However, by a standard result (see e.g., Knopp, 1954, Ch. 12), we have that $\prod_{t=1}^{\infty} (1 - p_t) = 0 \iff \sum_{t=1}^{\infty} p_t \to \infty$, provided $p_t < 1$.

Both cases together complete the proof. ∎

We can now see that language processes are strictly more general than language models: Eq. (1) shows us that any language model can be written as a language process, but Proposition 2.4 shows the converse is not necessarily true. Indeed, Proposition 2.4 allows us to easily construct a simple language process (example given below) that cannot be converted to a language model, which motivates the formalism.

Example 2.5. *Let $\mathcal{V} = \{a\}$. Define a language process $\boldsymbol{Y} = \{Y_t\}_{t=1}^{\infty}$ over $\mathcal{V}$ such that each Y_t is distributed according to $p(a \mid \boldsymbol{y}_{<t}) = 1 - \frac{1}{2^{t+1}}$ and $p(\text{EOS} \mid \boldsymbol{y}_{<t}) = \frac{1}{2^{t+1}}$. Note that we keep the convention that $Y_t = \text{BOS}$ for $t \leq 0$, and thus $p_0 = 0$. We have $\sum_{t=1}^{\infty} p_t = \frac{1}{2} < \infty$, so, by Proposition 2.4, $\boldsymbol{Y}$ is not a language model. Computing the infinite product $\prod_{t=1}^{\infty} (1 - p_t)$ shows $\boldsymbol{Y}$ leaks $\approx .58$ to infinite strings.*

Life after EOS? Proposition 2.4 further hints at the more intuitive difference between language models and language processes—what happens after EOS? In the traditional definition of a language model (Definition 2.1), life ends at EOS. That is, any string with symbols after EOS would not be a valid sample from a language model because such strings are not in the model's support. On the other hand, a language process offers a more chipper view: once we hit EOS, we can just generate another symbol. A language process is better thought of as an infinite babbler than a distribution over any sort of strings. At some level, this is indeed the implicit view that is adopted by some when language modeling, as many language models do not have EOS in the traditional sense. For the rest of this paper we will also take this view, and consider language processes for which we can continue generating after sampling an EOS symbol.

2.3 Other Useful Properties

Next, we discuss some other properties about language processes that are important for understanding the theoretical results presented in §3.

Definition 2.6 (Markov). *A language process $\boldsymbol{Y} = \{Y_t\}_{t=1}^{\infty}$ over alphabet $\mathcal{V}$ with distribution p is Markov[10] if the following equality holds*

$$p(Y_t \mid \boldsymbol{Y}_{<t}) = p(Y_t \mid Y_{t-k}, \ldots, Y_{t-1})$$

where $k \geq 0$ is the Markov order. We again take Y_t for $t \leq 0$ to be BOS, indicating our initial distribution.

Many language processes are explicitly defined to be Markov, e.g., ones based on n-gram language models. However, many language processes based on recurrent neural networks are, in principle, non-Markov. Yet despite being capable of learning non-Markov distributions, researchers have found that recurrent neural language models tend to learn Markov distributions. For instance, Khandelwal et al. (2018) show that a recurrent neural language model's memory is empirically bounded at roughly 200 words. Thus, we can still generally assume this property when working with language processes parameterized by such models.[11]

---

[10] Also known as a Markov chain.

[11] Note that, in principle, human language is *not* Markov, in so far as many linguists believe human language is capable of arbitrarily deep center-embeddings (Chomsky, 1957, 1995). Yet research suggests that humans do not make use of this property in practice (Reich, 1969; Karlsson, 2010), and so we

Definition 2.7 (Stationarity). *A k-Markov language process $\boldsymbol{Y} = \{Y_t\}_{t=1}^{\infty}$ over alphabet $\mathcal{V}$ with distribution p is **stationary** if the following holds*

$$
\begin{aligned}
p(Y_{t+n} \mid Y_{t-k+n}, \ldots, Y_{t-1+n}) \\
= p(Y_t \mid Y_{t-k}, \ldots, Y_{t-1})
\end{aligned}
\tag{3}
$$

[handwritten: Probability of → next word/token is not dependent much on absolute pos but on history.]

for $n \geq 0$. We again take Y_t for $t \leq 0$ to be BOS, indicating our initial distribution.

While not theoretically Markovian, human language is generally considered stationary, i.e., the probability distribution over the next word should not depend on absolute position, but rather the history.

Definition 2.8 (Ergodicity). *A language process $\boldsymbol{Y} = \{Y_t\}_{t=1}^{\infty}$ is **ergodic** if its statistical properties (e.g., ensemble averages) can be deduced from a single, sufficiently long, random sample.*

The above definition is intentionally informal because ergodicity is a complex topic that would take time to treat thoroughly, but see McMillan (1953); Breiman (1957) for a more rigorous discussion. One of the important implications of ergodicity for language processes, however, is rather straightforward: if our language process is over alphabet $\mathcal{V}$ with distribution p and is ergodic, then for every symbol $y \in \mathcal{V}$ and for every history $\boldsymbol{y}_{<t} \in \mathcal{V}^*$, there must exist an extension $\boldsymbol{y}_{<t'} = \boldsymbol{y}_{<t}, y_t, \cdots, y_{t'-1}$ such that $p(Y_{t'} = y \mid \boldsymbol{Y}_{<t'} = \boldsymbol{y}_{<t'}) > 0$. In plain terms, this just says that we can always reach every word in our alphabet via some path no matter where we currently are. In our context, ergodicity also relates to the problem with EOS. If we convert a language model into a language process (as discussed in §2.1) and make the EOS state absorbing,[12] this language process must be non-ergodic, as once it encounters EOS, no other state is reachable.

2.4 Estimating a Language Model from Data

Language models are typically estimated from language data. The standard method for estimating the parameters of p is via maximization of the log-likelihood of a training corpus $\mathcal{S}$

$$L(\boldsymbol{\theta}; \mathcal{S}) = -\sum_{\boldsymbol{y} \in \mathcal{S}} \sum_{t=1}^{|\boldsymbol{y}|} \log p(y_t \mid \boldsymbol{y}_{<t}) \tag{4}$$

---

do not consider the Markovian property of most models as a limitation to their ability to model natural language in practice.

[12] This would be done by setting the transition probability $p(Y_t = \text{EOS} \mid \boldsymbol{Y}_{<t} = \boldsymbol{y}_{<t}) = 1$ if $y_{t-1} = \text{EOS}$.

where $\boldsymbol{\theta}$ are the model p's parameters. The above is equivalent to minimizing the cross-entropy loss between p and the empirical distribution. Note that we assume all $\boldsymbol{y} \in \mathcal{S}$ end in the special EOS token.

3 Information-Theoretic Properties of Language Processes

The view of language modeling as a discrete stochastic process naturally lends itself to an analysis through the lens of information theory. Indeed, much of information theory is concerned with the study of discrete stochastic processes (see e.g., Cover and Thomas, 2012, Ch. 4). In this section, we review standard information-theoretic definitions in §3.1 and build on these to introduce our own notion of local typicality in §3.2.

3.1 Typicality

An important definition in the study of stochastic processes is entropy rate, which generalizes the notion of entropy from a random variable to a stochastic process.

Definition 3.1 (Entropy Rate). *Let $\boldsymbol{Y} = \{Y_t\}_{t=1}^{\infty}$ be a stationary, ergodic discrete stochastic process over alphabet $\mathcal{V}$ with distribution p. The **entropy rate** of $\boldsymbol{Y}$ is defined as*

$$\mathrm{H}(\boldsymbol{Y}) \overset{\text{def}}{=} \lim_{t \to \infty} \frac{1}{t} \mathrm{H}(Y_1, \dots, Y_t) \qquad (5)$$

The entropy rate is useful in that it tells us, in the limit, how spread out, i.e., entropic, the distribution is. Another interpretation is that it quantifies the complexity of $\boldsymbol{Y}$. In the case of an i.i.d. process, the entropy rate and the entropy coincide, making the entropy rate a true generalization of the entropy. Using entropy rate, we can define the notion of the typical set.

Definition 3.2 (Typical Set). *Let $\boldsymbol{Y} = \{Y_t\}_{t=1}^{\infty}$ be a stationary, ergodic discrete stochastic process where each Y_t follows distribution p and takes on values in a finite support $\mathcal{Y}$. For $1 \leq T < \infty$, the (T, ε)-**typical set** of $\boldsymbol{Y}$ is the set of all sequences of length exactly T with average per-symbol negative log-probability close to $\mathrm{H}(\boldsymbol{Y})$, i.e.* → Divide by T

$$\mathcal{T}_{\varepsilon}^{(T)} = \left\{ \boldsymbol{y} \mid \left| \frac{\log p(\boldsymbol{y})}{T} + \mathrm{H}(\boldsymbol{Y}) \right| < \varepsilon \right\} \qquad (6)$$

In informal terms, the typical set is the set of all samples that we would expect when sampling from

p. To give the reader intuition about typicality, we now turn to a classical example.[13]

Example 3.3. *Consider an i.i.d. stochastic process $\boldsymbol{Y} = \{Y_t\}_{t=1}^{\infty}$ where Y_t is defined as the outcome of flipping a biased coin: we have $p(\text{HEADS}) = .6$ and $p(\text{TAILS}) = .4$. If we flip 100 coins, the most likely outcome is the sequence of 100 heads. However, this would be a surprising outcome to most people, who would intuitively expect the sequence to consist of roughly 60% heads and 40% tails. Indeed, even for relatively large ε, the sequence of 100 heads is not in the $\mathcal{T}_{\varepsilon}^{(T)}$ typical set; its average symbol probability is $.6 \gg 2^{-\mathrm{H}(Y_t)} \approx 0.51$.*

$2 \quad -(0.6\log 0.6 + 0.4 \log 0.4)$

The above example demonstrates that the typical set often does *not* contain the most likely sequence. Additionally, the typical set is interesting because, as $T \to \infty$, it contains nearly all the probability mass; we formalize this property in a proposition.

Proposition 3.4. *Let $\boldsymbol{Y} = \{Y_t\}_{t=1}^{\infty}$ be a stationary, ergodic discrete stochastic process where each Y_t follows distribution p and takes on values in a finite support $\mathcal{Y}$. For every $\varepsilon > 0$, for sufficiently large T, the following conditions hold:*

i) $\sum_{\boldsymbol{y} \in \mathcal{T}_{\varepsilon}^{(T)}} p(\boldsymbol{y}) > 1 - \varepsilon$

ii) $(1 - \varepsilon) 2^{T(\mathrm{H}(\boldsymbol{Y}) - \varepsilon)} \leq |\mathcal{T}_{\varepsilon}^{(T)}| \leq 2^{T(\mathrm{H}(\boldsymbol{Y}) + \varepsilon)}$

In words, as we take $T \to \infty$, the probability mass covered by the typical set is nearly 1 and the number of elements in it is nearly $2^{T \cdot \mathrm{H}(\boldsymbol{Y})}$.

Proof. See Breiman (1957) for proof. ∎

What's wrong with the typical set? Let $\boldsymbol{Y}$ be a stationary, ergodic language process. By the conditions of Definition 3.2, we know that $\boldsymbol{Y}$ has a typical set. We have motivated the typical set, intuitively, as the subset of strings that are usual or typical among all strings. Under this intuition, it makes sense that—when using $\boldsymbol{Y}$ as a language generator—this is the set from which we would like to select a string. A relatively straightforward corollary of Proposition 3.4 is that ancestral sampling should pull from just this set. To see this, we can turn to i) in Proposition 3.4: since ancestral sampling provides an i.i.d. sample from $\boldsymbol{Y}$, the probability of getting an element *not* in $\mathcal{T}_{\varepsilon}^{(T)}$ as $T \to \infty$ is $(1 - \varepsilon)$, i.e., practically never. However,

---

[13] See Dieleman (2020) for further discussion of the concept of typicality in the context of generative modeling.

there is the confound that our models are not perfect representations of the true distribution behind the "human" natural language process. Perhaps for this reason (and the reasons discussed in §4), ancestral sampling is not known to result in samples that humans judge to be high quality in the task of language generation; rather it often leads to text that humans perceive as incoherent (Holtzman et al., 2020). Furthermore, the typical set's definition relies on $\mathbf{Y}$ being a stationary and ergodic language process. As we saw in §2.2, however, a language model that we convert into a language process will be non-ergodic by definition (at least if we keep EOS as an absorbing state). Thus, while the typical set is a natural starting point, it does not actually get us to our end goal of defining a set of strings that humans would find typical. To remedy this problem, we introduce the new concept of local typicality.

3.2 Local Typicality

A core contribution of this work is to define a more restrictive notion of typicality—termed here **local typicality**—which we subsequently motivate as useful in the context of describing the set of strings humans typically produce.

Definition 3.5 (Locally Typical Set). *Let* $\mathbf{Y} = \{Y_t\}_{t=1}^{\infty}$ *be a discrete stochastic process over finite support* $\mathcal{Y}$. *The* (T, ε)-**locally typical set of** $\mathbf{Y}$ *is the set of all sequences of length exactly* T *such that*

$$\mathcal{L}_{\varepsilon}^{(T)} = \left\{ \mathbf{y} = y_0 \cdots y_T \mid \forall 1 \leq t \leq T, \tag{7} \right.$$
$$\left. \left| \log p(y_t \mid \mathbf{y}_{<t}) + \mathrm{H}(Y_t \mid \mathbf{Y}_{<t} = \mathbf{y}_{<t}) \right| < \varepsilon \right\}$$

In comparison to the typical set, the locally typical set further restricts the set of samples to those for which each individual symbol y_t has probability near the local conditional entropy, i.e., the entropy of the distribution $p(\cdot \mid \mathbf{y}_{<t})$. In general, there is no strong theoretical relationship between the typical set and the locally typical set. However, in the case of an i.i.d. stochastic process we can prove that the latter constitutes a subset of the former.

Proposition 3.6. *Let* $\mathbf{Y} = \{Y_t\}_{t=1}^{\infty}$ *be an i.i.d. discrete stochastic process, then* $\mathcal{L}_{\varepsilon}^{(T)} \subseteq \mathcal{T}_{\varepsilon}^{(T)}$.

Proof. Since $\mathbf{Y}$ is i.i.d., we have that $\mathrm{H}(\mathbf{Y}) = \mathrm{H}(Y_t \mid \mathbf{Y}_{<t}) = \mathrm{H}(Y_t)$. Let $\mathbf{y}$ be an element of $\mathcal{L}_{\varepsilon}^{(T)}$. Then, $\sum_{t=1}^{T} \left| \log p(y_t) + \mathrm{H}(Y_t) \right| < T\varepsilon$. Thus, by the triangle inequality, $\left| \sum_{t=1}^{T} \log p(y_t) + \right.$

$\left. T\mathrm{H}(Y_t) \right| < T\varepsilon$, which implies $\left| \frac{\sum_{t=1}^{T} \log p(y_t)}{T} + \mathrm{H}(Y_t) \right| < \varepsilon$, which implies $\mathbf{y} \in \mathcal{T}_{\varepsilon}^{(T)}$. ∎

A natural question to ask at this point is why the definition of local typicality is useful in the context of a language process. Our argument, presented in the following section, is cognitive in nature.

4 Local Typicality in Natural Language

To motivate our definition of local typicality in the context of natural language, we must first look at language through an information-theoretic lens. We will consider *two* distributions in this section: $\widetilde{p}$, the distribution that a speaker of the language is assumed to generate strings from, and $\widehat{p}$ our language process that approximates $\widetilde{p}$—albeit, perhaps not perfectly. In this setting, we view a natural language string $\mathbf{y}$ as a means of communicating some information, where each word y_t is a symbol via which we construct our message. The information content of $\mathbf{y}$ is then defined as its negative log-probability under a specified distribution: $-\log \widetilde{p}(\mathbf{y})$. Following the chain rule of probability, this quantity can be decomposed over words, i.e., the information content of a word is its negative log-probability given prior context: $-\log \widetilde{p}(y_t \mid \mathbf{y}_{<t})$.

4.1 Properties of Human Communication

Given the above definitions, we can now ask a question at the heart of this work: what are the information-theoretic characteristics of natural language typically produced by humans. In other words, what do strings sampled from $\widetilde{p}$ look like, from the perspective of $\widehat{p}$, our trained language process? Research in psycholinguistics suggests that a core component of what makes text human-like is its per-unit information content.

To motivate this conclusion, we first consider a language user's objective. When using natural language, humans aim to transmit information efficiently while also minimizing the risk of miscommunication (Zipf, 1949). In order to achieve this goal, speakers avoid producing words with either very high or very low information content (Fenk and Fenk, 1980; Aylett and Turk, 2004; Levy and Jaeger, 2007; Mahowald et al., 2013, *inter alia*), a behavior inline with theories of efficient and robust communication.[14] Indeed, cross-linguistic research

---

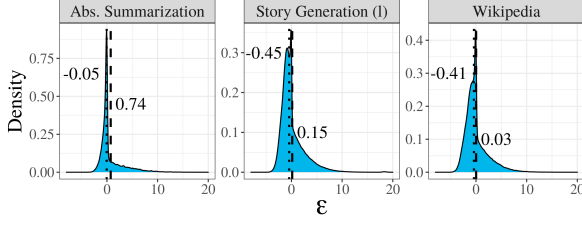[14] See Gibson et al. (2019) for an in-depth review of how efficiency has shaped the evolution of language.

Figure 1: The per-token distribution of the deviation (ε) of information content from conditional entropy. Values are computed using the reference (human) text for three different language generation tasks, where probabilities and entropies are computed using probabilistic models trained on the respective task (see §6 for model details). Dotted line and adjacent label indicate median ε while dashed line and adjacent label indicate mean ε. Per token distributions of conditional entropies and information contents are shown in App. B for reference.

has shown that languages trade-off information content and speech rate, perhaps aiming at a specific (optimal) information rate (Coupé et al., 2019; Pimentel et al., 2021). Further, not using words in a context where they have very high or low information content avoids characteristics that appear to negatively impact traditional grammaticality judgments: an ideal natural language string would not compensate for unusually near-zero probability in the first half, e.g., syntactic error, with unusually high probability in the second half, e.g., especially frequent words (Schütze, 2016; Lau et al., 2017).

4.2 An Information-Theoretic Formalization

The definition of local typicality presented in §3.2 can be viewed as an embodiment of the characteristics of human language just described above. One logical interpretation of these behaviors is that, at every time step, natural-sounding language should have per-symbol information content close to the expected (average) per-symbol information content.[15] We formalize this relationship in the following hypothesis.

Hypothesis 4.1. *Samples $\boldsymbol{y} = y_0 \cdots y_T$ from a human language process with distribution $\widetilde{p}$ tend to belong to the process's locally typical set $\mathcal{L}_\varepsilon^{(T)}$ for large enough T and some $\varepsilon > 0$. In words, this means that we should expect every word in natural-sounding sentences to be close to the expected information content under $\widetilde{p}$, i.e., the conditional entropy given prior context.*

---

[15]The standard definition of (Shannon) entropy for a random variable X with support $\mathcal{X}$ is equivalent to the expected information of X: $\mathrm{H}(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$.

We verify this relationship empirically using data from human language processes. In Fig. 1, we show the distribution of the difference between the information content of y_t and the *expected* information content of Y_t, i.e., $-\log \widehat{p}(y_t \mid \boldsymbol{y}_{<t}) - \mathrm{H}(Y_t \mid \boldsymbol{Y}_{<t} = \boldsymbol{y}_{<t})$, according to the model on human-generated text. The peaked nature of the distributions in Fig. 1 reveals that human language indeed tends to have per-word information content quite close a specific value; the centering of these distributions around ≈ 0 suggests that this value is $\mathrm{H}(Y_t \mid \boldsymbol{Y}_{<t} = \boldsymbol{y}_{<t})$. Notably, Meister et al. (2022) shows the same is not true for text generated by models according to a number of different popular decoding schemes, which instead produce strings with much higher probability, i.e., with lower information content.

In an ideal situation, such a property of natural language would be reflected in $\widehat{p}$, in which case sampling from the typical set should be sufficient to ensure human-like language. However, our models are by no means perfect. The failure to capture the property of human language expounded in Hyp. 4.1 may come from a number of possible modeling deficiencies, e.g., poor ability to capture the tails of these distributions. We hypothesize that, when using language models to generate text, enforcing this local-typicality criterion explicitly may serve as a patch for this shortcoming.

5 Sampling from a Language Process

In this section, we describe how to sample from a language process parameterized by the distribution p,[16] or in more commonly-used terminology, how to decode from p. There are many different algorithms one could employ to sample from p. The most intuitive strategy is ancestral sampling,[17] which works as follows: sample $y_t \sim p(\cdot \mid \boldsymbol{y}_{<t})$ for each history $\boldsymbol{y}_{<t}$ successively until some chosen criterion, e.g., the EOS symbol is sampled or a maximum length is reached. Note that in the case of the former criterion, this procedure is equivalent to sampling entire strings according to the distribution p. Perhaps the most popular set of techniques for sampling fall under a paradigm we call **truncated sampling**, where the vocabulary at a time step is truncated to a core subset of words.

---

[16]Here we only consider locally normalized p, i.e., processes in which sampling is done on a word-by-word basis.

[17]Another natural option would be to choose words which maximize the probability assigned by p to the resulting string, but this work focuses on stochastic strategies.

top-k

For instance, Fan et al. (2018) propose limiting the sampling space to the top-k most likely words in each decoding step, and Holtzman et al. (2020) consider the smallest nucleus (i.e., subset) of words whose cumulative probability mass exceeds a chosen threshold η.

top-p

In this paper, we give a general treatment of truncated sampling and then discuss our variant. Given a context-dependent constraint subset $\mathcal{C}(\boldsymbol{y}_{<t}) \subseteq \overline{\mathcal{V}}$ of the vocabulary, we define the truncated distribution as

$$\pi(y \mid \boldsymbol{y}_{<t}) \stackrel{\text{def}}{=} \frac{p(y \mid \boldsymbol{y}_{<t})\mathbb{1}\{y \in \mathcal{C}(\boldsymbol{y}_{<t})\}}{Z(\boldsymbol{y}_{<t})} \quad (8)$$

where the normalizer is defined as

$$Z(\boldsymbol{y}_{<t}) \stackrel{\text{def}}{=} \sum_{y \in \mathcal{C}(\boldsymbol{y}_{<t})} p(y \mid \boldsymbol{y}_{<t}) \quad (9)$$

and we call $\mathcal{C}(\boldsymbol{y}_{<t})$ the truncation set. Now we give two examples of truncated samplers.

Algorithm 5.1 (Top-k Sampling). *In top-k sampling, the truncation set $\mathcal{C}(\boldsymbol{y}_{<t})$ is defined as the top-k highest-probability tokens y according to $p(\cdot \mid \boldsymbol{y}_{<t})$, i.e., the solution to the following subset maximization problem*

$$\underset{\mathcal{C}(\boldsymbol{y}_{<t}) \in \mathcal{P}(\overline{\mathcal{V}})}{\text{maximize}} \sum_{y \in \mathcal{C}(\boldsymbol{y}_{<t})} p(y \mid \boldsymbol{y}_{<t})$$
$$\text{subject to} \quad |\mathcal{C}(\boldsymbol{y}_{<t})| \leq k \quad (10)$$

where $\mathcal{P}$ is the power set operator.

Algorithm 5.2 (Nucleus Sampling). *In nucleus sampling, we choose a threshold parameter η and define the truncation set $\mathcal{C}(\boldsymbol{y}_{<t})$ as the solution to the following subset minimization problem*

$$\underset{\mathcal{C}(\boldsymbol{y}_{<t}) \in \mathcal{P}(\overline{\mathcal{V}})}{\text{minimize}} \quad |\mathcal{C}(\boldsymbol{y}_{<t})|$$
$$\text{subject to} \quad \sum_{y \in \mathcal{C}(\boldsymbol{y}_{<t})} p(y \mid \boldsymbol{y}_{<t}) \geq \eta \quad (11)$$

where again $\mathcal{P}$ is the power set operator.

5.1 Shortcomings of Existing Algorithms

To motivate sampling based on the locally typical set, we must first better understand the shortcomings of current decoding strategies. While strings generated using stochastic strategies may have lower probability according to $\widehat{p}$, they often outperform those decoded using maximization-based strategies in terms of qualitative metrics.

A number of recent works have tried to offer explanations for this phenomenon. Some have attributed it to a diversity–quality trade-off (Zhang et al., 2021; Basu et al., 2021), while others blame shortcomings of model architectures or training strategies (Welleck et al., 2020; Li et al., 2020).

Our analysis from §4 inspires an alternative explanation, motivated by information theory and psycholinguistics, for why models that perform so well (in terms of metrics such as perplexity) can still exhibit such undesirable behavior when used to generate text. First, the connection between probability and information content may explain why high-probability text is often dull or generic (Holtzman et al., 2020; Eikema and Aziz, 2020); its low information content likely makes for boring, i.e., uninformative, text. This connection also offers a potential explanation for the rather strange behavior that, when a string has a repetitive loop, language models often assign increasingly higher probability to the repeated substring (Holtzman et al., 2020); the substring conveys less and less information after each occurrence.

A further implication of this framing is the equivalence between decoding strings from a probabilistic language generator and sampling messages from the natural language communication channel. If we wish to solely sample from the subset of messages that a human would typically construct, i.e., that are human-like, then we should begin by narrowing down this subset to those messages that meet at least some of the same criteria as human-generated messages. In this work, we have identified the criterion that such messages tend to be in the locally typical set. This observation motivates a new decoding strategy in which our information-theoretic criterion is explicitly enforced.

5.2 Locally Typical Sampling

We now introduce our novel sampling algorithm, which we entitle **locally typical sampling**.

Algorithm 5.3. *Locally typical sampling is a truncated sampling scheme where the truncation set $\mathcal{C}(\boldsymbol{y}_{<t})$ is the solution to the following subset optimization problem*

$|\text{ent} - (-NLL)|$

$$\underset{\mathcal{C}(\boldsymbol{y}_{<t}) \in \mathcal{P}(\overline{\mathcal{V}})}{\text{minimize}} \sum_{y \in \mathcal{C}'(\boldsymbol{y}_{<t})} \big| \mathrm{H}(Y_t \mid \boldsymbol{Y}_{<t} = \boldsymbol{y}_{<t}) \quad (12)$$
$$+ \log p(y \mid \boldsymbol{y}_{<t}) \big|$$
$$\text{subject to} \sum_{y \in \mathcal{C}(\boldsymbol{y}_{<t})} p(y \mid \boldsymbol{y}_{<t}) \geq \tau$$

In words, Algorithm 5.3 limits the sampling distribution to only those words with negative log-probability within a certain absolute range from the conditional entropy (expected information content) of the model at that time step. In the spirit of nucleus sampling, this range is determined by a hyperparameter τ, the amount of probability mass from the original distribution that we wish to consider.

Interestingly, Algorithm 5.3 does *not* imply that high-probability words should not be chosen. Indeed, in the situation where conditional entropy is low, i.e., when the model places most of the probability mass on a small subset of words, it is likely the case that only high-probability words fall into the locally typical set.

Computational Complexity. From a practical perspective, locally typical sampling can be implemented with the same efficiency as nucleus or top-k sampling. First, we compute the conditional entropy, which is an $\mathcal{O}(|\mathcal{V}|)$ operation. Second, we sort words by their absolute distance from $\mathrm{H}(\widehat{p}(\cdot \mid \boldsymbol{Y}_{<t} = \boldsymbol{y}_{<t}))$, which can be done in $\mathcal{O}(|\mathcal{V}| \log |\mathcal{V}|)$ time with standard sorting algorithms. Finally, we greedily take words from this list until their cumulative probability exceeds the threshold τ, which again takes $\mathcal{O}(|\mathcal{V}|)$ time. Thus, creating our altered distribution has time complexity $\mathcal{O}(|\mathcal{V}| \log |\mathcal{V}|)$.[18]

Relationship to Other Decoding Strategies. Notably, we already see motivation for this criterion in the performance of several well-known decoding strategies. For example, beam search is the predominant decoding strategy for machine translation models (Wu et al., 2016; Edunov et al., 2018; Ng et al., 2019; Meister et al., 2020b), a setting in which beam search (incidentally) often already enforces this criterion.[19] Yet, when used in more open-ended tasks, where the entropy of the language model is higher, beam search can lead to low-quality text (Li et al., 2016; Holtzman et al., 2020; Welleck et al., 2020; Meister et al., 2022). Locally typical sampling is also closely related to

---

[18]For each of the truncation sampling algorithms, the truncation set can also be identified using the selection algorithm (no sorting required) in $\mathcal{O}(|\mathcal{V}|)$ time. We provide the analysis using sorting as that is the standard implementation.

[19]When trained *without* label-smoothing, which artificially inflates conditional entropies, machine translation models tend to have quite low conditional entropies (see e.g., Fig. 3 in Meister et al., 2020a). Therefore, at each decoding step, the set of words with negative log-probability near the conditional entropy of the model are typically those with high probability—the same as those chosen by beam search.

nucleus sampling. When the probability distribution over the vocabulary has low conditional entropy, i.e., when there are only a few reasonable choices for the next word according to our model, nucleus and typical will have the same truncation set. Locally typical sampling and Mirostat (Basu et al., 2021) likewise have similar decision rules for truncation. Mirostat decodes strings such that they have a perplexity (or, equivalently, a per-word information content) close to a target value. In contrast to Mirostat, however, locally typical sampling does not require a specific target information content to be defined. Rather, locally typical sampling defines this quantity as the conditional entropy, choosing it dynamically (per word) and making it less sensitive to hyperparameter choice. Finally, locally typical sampling is also related to Braverman et al.'s (2020) strategy, which proposes a look-ahead decoding algorithm that generates text with a similar entropy rate to that of human-generated text. Our strategy's motivation is similar—to match the tendencies in information content exhibited by human-generated text—albeit without requiring the computational overhead of a look-ahead strategy.

6 Experiments

In this section, we explore the efficacy of our decoding strategy on two natural language generation tasks: abstractive summarization and story generation. We assess performance with respect to several other stochastic decoding strategies: nucleus sampling, top-k sampling, temperature sampling,[20] beam search and Mirostat. Our evaluation includes both automatic metrics and human ratings.

6.1 Setup

Models and Data. We use the Hugging Face framework (Wolf et al., 2020) for reproducibility, employing their implementations of nucleus, top-k, temperature sampling and beam search. We rely on the implementation of Mirostat provided by its authors. For story generation, we finetune the medium and large versions of GPT-2 (Radford et al., 2019) from checkpoints made available by OpenAI on the WRITINGPROMPTS dataset (Fan et al., 2018). We use the medium checkpoint finetuned on WIKITEXT-103 (Merity et al., 2017) to produce the data used in Fig. 1. For abstractive summarization, we use BART (Lewis et al., 2020)

---

[20]Temperature sampling is defined as ancestral sampling after local renormalization with an annealing term τ.

			Story Generation				
	PPL (g)	PPL (i)	MAUVE (↑)	REP (↓)	Zipf	D (↑)	Human (↑)
Reference	16.33	26.71	–	0.28	1.09	0.85	4.12 (±0.02)
Temperature (τ=0.5)	25.34 (+9.01)	18.78 (−7.93)	0.95	**0.25**	**1.07** (−0.02)	**0.87**	4.13 (±0.02)
Temperature (τ=1)	25.67 (+9.34)	11.77 (−14.94)	0.95	0.26	**1.07** (−0.02)	**0.87**	4.13 (±0.02)
Nucleus (η=0.9)	7.75 (−8.58)	10.25 (−16.46)	0.95	0.35	1.29 (+0.20)	0.79	4.09 (±0.02)
Nucleus (η=0.95)	11.65 (−4.68)	11.77 (−14.94)	0.95	0.30	1.20 (+0.11)	0.84	4.13 (±0.02)
Top-k (k=30)	7.07 (−9.26)	18.78 (−7.93)	0.88	0.35	1.41 (+0.32)	0.80	4.13 (±0.02)
Top-k (k=40)	11.83 (−4.5)	13.08 (−13.63)	0.92	0.35	1.33 (+0.24)	0.82	4.09 (±0.02)
Mirostat (τ=3)	8.14 (−8.19)	**23.53** (−3.18)	0.93	0.34	1.30 (+0.21)	0.83	4.12 (±0.02)
Typical (τ=0.2)	**14.25** (−2.08)	23.51 (−3.20)	0.78	0.30	1.27 (+0.18)	0.84	**4.15** (±0.02)
Typical (τ=0.95)	11.59 (−4.74)	11.77 (−14.94)	**0.96**	0.31	1.21 (+0.12)	0.84	4.13 (±0.02)

Table 1: Automatic quality and diversity metrics, as described in §6.1, along with human ratings on the WRITINGPROMPTS dataset. Human ratings are averaged across criteria to form a single metric. Bolded values are the best results among decoding strategies, where for perplexity (PPL) and Zipf's coefficient, we take this to be the delta from measurements on human text (numbers in purple). Numbers in blue are standard error estimates. Results are from finetuned GPT-2 large.

finetuned on the CNN/DAILYMAIL dataset (Nallapati et al., 2016).[21] All reported metrics are computed on the respective test sets.

Hyperparameters. In a preliminary hyperparameter sweep using MAUVE[22] (Pillutla et al., 2021), we found $k = \{30, 40\}$, $\eta = \{0.9, 0.95\}$ and $\tau = 3.0$ to be the best performing hyperparameters for top-k sampling, nucleus sampling and Mirostat, respectively. For locally typical sampling, we found $\tau = 0.2, \tau = 0.95$ to provide the best results for story generation and abstractive summarization, respectively. Standard values according to the literature for other hyperparameters (i.e., for beam search and temperature sampling) were employed. We use these values in our human evaluations and in computation of automatic metrics.

Automatic Quality Metrics. As automatic quality metrics, we evaluate the generated text's perplexity—under both the model used to generate the text (PPL(g)) and an independent, i.e., not finetuned, LM (PPL(i)), namely GPT-2 large (Radford et al., 2019). Several prior works have shown that neither low nor high perplexity (Zhang et al., 2021;

Nadeem et al., 2020; Pillutla et al., 2021) are direct indicators of text quality. Rather, human-like text often has perplexity within a certain range. Consequently, we report the difference in this metric from the reference text as well. We additionally evaluate using MAUVE[23] (Pillutla et al., 2021) with the reference text.

Automatic Diversity Metrics. We also evaluate locally typical sampling using automatic diversity metrics. We compute REP (Welleck et al., 2020), Zipf's coefficient, and n-gram diversity. For REP we use the average of REP/ℓ scores, as defined in Eq. 9 of (Welleck et al., 2020) for $\ell \in \{16, 32, 128\}$. We define n-gram diversity D as the average fraction of unique vs. total n-grams for $n \in \{1, 2, 3, 4\}$ in a string

$$D = \sum_{n=1}^{4} \frac{\#\text{unique } n\text{-grams in string}}{\# \ n\text{-grams in string}} \quad (13)$$

Human Evaluations. We use the Amazon Mechanical Turk framework to obtain human judgments of text quality from 5 different annotators on 200 examples per decoding strategy, per task. We use solely MTurk Master Workers in order to maximize the quality of our ratings. We follow DeLucia et al. (2021) in setting up our evaluations. Each Human Intelligence Task (HIT) consists of either a single prompt from which a story should be generated or a single news article to be summarized. The raters are first presented with the different rating criteria, along with descriptions of the type of

---

[21] As we are interested in getting as close an estimate of p as possible with our models $\widehat{p}$, all fine-tuning is done *without* label-smoothing. Note that label-smoothing may also artificially inflate conditional entropy estimates, as it pushes the learned distribution towards the most entropic distribution: the uniform distribution (Pereyra et al., 2017).

[22] We use the default settings given by the authors for all MAUVE computations, albeit we employ different LMs in our parameter sweep vs. reported results (standard GPT-2 vs. GPT-2 large, respectively) to reduce bias in the final results. Notably, MAUVE presents similar performances when used with these two pretrained LMs (Pimentel et al., 2022).

[23] We use the implementation provided by the authors.

| | **Abstractive Summarization** | | | | | | |
	PPL (g)	PPL (i)	MAUVE (↑)	REP (↓)	Zipf	D (↑)	Human (↑)
Reference	10.29	34.21	–	0.13	0.76	0.97	4.31 (±**0.03**)
Beam (k=5)	1.39 (−**8.90**)	**34.21** (−**0.00**)	0.90	**0.14**	**0.77** (+**0.01**)	0.97	**4.35** (±**0.03**)
Temperature (τ=0.5)	**7.10** (−**3.19**)	55.31 (+**21.1**)	0.97	0.15	0.75 (−**0.01**)	0.97	4.25 (±**0.03**)
Temperature (τ=1)	6.46 (−**3.83**)	35.96 (+**1.75**)	0.95	**0.14**	0.75 (−**0.01**)	0.97	4.29 (±**0.03**)
Nucleus (η=0.9)	2.97 (−**7.32**)	33.63 (−**0.58**)	0.90	0.17	0.93 (+**0.17**)	0.96	4.26 (±**0.03**)
Nucleus (η=0.95)	3.96 (−**6.33**)	56.43 (+**22.22**)	**0.99**	0.15	0.91 (+**0.15**)	0.97	4.26 (±**0.03**)
Top-k (k=30)	3.13 (−**7.16**)	34.79 (+**0.58**)	0.98	0.16	0.93 (+**0.17**)	0.97	4.31 (±**0.03**)
Top-k (k=40)	3.26 (−**7.03**)	28.38 (−**5.83**)	0.96	0.16	0.93 (+**0.17**)	0.97	4.29 (±**0.03**)
Typical (τ=0.2)	3.80 (−**6.49**)	62.33 (+**28.12**)	0.72	**0.14**	0.91 (+**0.15**)	0.97	4.27 (±**0.03**)
Typical (τ=0.95)	3.86 (−**6.43**)	56.67 (+**22.46**)	0.96	0.15	0.92 (+**0.16**)	0.97	4.32 (±**0.03**)

Table 2: Automatic quality and diversity metrics, as described in §6.1, along with human ratings on the CNN/DAILYMAIL dataset. Human ratings are averaged across criteria to form a single metric. Bolded values are the best results among decoding strategies, where for perplexity (PPL) and Zipf's coefficient, we take this to be the delta from measurements on human text (numbers in purple). Numbers in blue are standard error estimates.
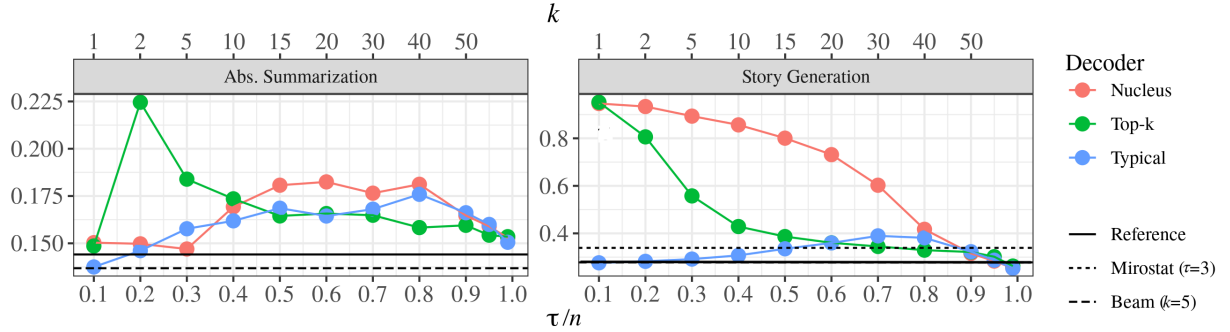
Figure 2: REP (Welleck et al., 2020) values for different k and τ/η (lower is better). Lines indicate REP measurement for reference text and Mirostat (left)/beam search (right).

text that meets these criteria at different levels of the scale. Raters are additionally provided several examples of stories/summarizations that both meet and fail to meet the rating criteria. They are then presented with the respective prompt/news article and the corresponding stories/summaries generated by different decoders and by the reference in random order. For abstractive summarization, we ask annotators to score on *fluency* and *relevance*, while for story generation, annotators score on *fluency*, *coherence*, and *interestingness*, each using a scale from 1 to 5. We choose these criteria following recommendations from van der Lee et al. (2019).

For each story/summarization and each of the criteria, we take the median score across raters as the final score.[24] Workers are paid $1.50 per HIT for the abstractive summarization task and $2 per HIT for the story generation task, for which entries were longer. Note that these rates translate to >$15/hour.

---

[24] We use an attention check in each HIT. Responses where the attention check has been failed are thrown out.

6.2 Results

Quantitative Performance. Tab. 1 and 2 show the results of our different evaluation metrics. Human scores are averaged across the qualitative metrics to give an aggregate score; the value in parentheses is the standard error of the estimate. We show full breakdowns of score distributions in Tab. 5. We see that in general, locally typical sampling performs on par with or better than other sampling techniques, producing text with human quality ratings closest to that of the reference among the stochastic decoding strategies. Interestingly, beam search still outperforms locally typical sampling in abstractive summarization, albeit by a small margin. This could perhaps be attributed to the deterministic nature of beam search, which suggests that an interesting direction for future research may be a deterministic version of locally typical sampling, e.g., where the highest-probability word within the truncated set is always chosen. Importantly, all the strategies we explore are quite close to human-level performance—in some cases

Abstractive Summarization (CNN/DailyMail)

Prompt	(CNN) The attorney for a suburban New York cardiologist charged in what authorities say was a failed scheme to have another physician hurt or killed is calling the allegations against his client "completely unsubstantiated." Appearing Saturday morning on CNN's "New Day," Randy Zelin defended his client, Dr. Anthony Moschetto, who faces criminal solicitation, conspiracy, burglary, arson, criminal prescription sale and weapons charges in connection to what prosecutors called a plot to take out a rival doctor on Long Island. "None of anything in this case has any evidentiary value," Zelin told CNN's Christi Paul...
Reference	A lawyer for Dr. Anthony Moschetto says the charges against him are baseless. Moschetto, 54, was arrested for selling drugs and weapons, prosecutors say. Authorities allege Moschetto hired accomplices to burn down the practice of former associate.
Beam $k = 5$	Dr. Anthony Moschetto faces criminal solicitation, conspiracy, burglary, arson and weapons charges. "None of anything in this case has any evidentiary value," his attorney says.
Nucleus $\eta = 0.95$	Dr. Anthony Moschetto, 54, pleaded not guilty to charges Wednesday. Two men – identified as James Chmela and James Kalamaras – were named as accomplices.
Top-k $k = 30$	Dr. Anthony Moschetto is accused of providing police with weapons and prescription drugs. Authorities say he was part of a conspiracy to harm or kill a rival doctor. His attorney calls the allegations against his client "completely unsubstantiated"
Typical $\tau = 0.95$	Dr. Anthony Moschetto is charged with crimes including arson, conspiracy, burglary, prescription sale, weapons charges. His attorney says "none of anything in this case has any evidentiary value"

Table 3: Sample generations for abstractive summarization; examples correspond to ID 1 in the test set. Decoding strategy hyperparameters are chosen based off of performance in human evaluations shown in Tab. 2.

even surpassing human references in terms of ratings. At this level, it is perhaps only reasonable to expect that the differentiation between the top strategies is small. Accordingly, we also consider how robust locally typical sampling is to hyperparameter choice. Fig. 2 shows REP measurements for different values of the hyperparameters k, η, and τ for top-k, nucleus, and locally typical sampling, respectively. Interestingly, REP appears to be far less sensitive to τ than to k and η. While many values of k and η appear to lead to degenerate repetitions in story generation, most values of τ lead to text with a REP value on par with human text, demonstrating that an advantage of our technique is its robustness to hyperparameter choice. See Fig. 6 in the appendix for a larger exploration of how other quality metrics vary as a function of τ.

Qualitative Performance. We present some examples of text generated according to each of the decoding strategies in Tab. 3 and 4. For both of the tasks, we choose the example with ID 1 in the respective test set and provide examples from each of the decoding strategies, employing the hyperparameter values that lead to the best human scores in Tab. 1 and 2. For the summarization task, we see that locally typical sampling provides a comprehensive and coherent summary of the article,

quite similar to that of beam search. In comparison, the text produced by temperature sampling is not necessarily coherent; text from nucleus sampling and top-k sampling misses some of the important information in the article, e.g., the charges of burglary and arson. While the qualitative performance in story generation is much more subjective, locally typical sampling arguably provides the most fluent story amongst all the decoding strategies. Other stories lack coherence and, even within the first few sentences, we see repeated phrases and words. Together, these results suggest that locally typical sampling may indeed produce more desirable text.

7 Conclusion

In this work, we analyze decoding from probabilistic language generators in the information-theoretic framework. We equate a language model to a discrete stochastic process, and use the known properties of such processes to quantitatively describe the samples we should expect. Motivated by results in psycholinguistics, we hypothesize that—with the goal of communicating efficiently and robustly—humans produce text whose per-word information content is within a close range of the *expected* information content of a word given prior context. Current language models may fall short in capturing

Story Generation (WritingPrompts)

Prompt	A kid doodling in a math class accidentally creates the world's first functional magic circle in centuries.
Reference	It was dark and Levi was pretty sure he was lying on his back. There was firelight flickering off of what was left of a ceiling. He could hear something but it was muffled. He was almost positive it was screaming. When he tried to move he felt an excruciating pain in his left side that caused him to cry out...
Nucleus $\eta = 0.95$	"Why do you draw a circle?" He asked as he handed the piece of paper over to her. She rolled her eyes and looked down at the picture, "Because I like math!" "Then what?" "I like to draw the circles." She ran a hand through her curls and looked away. "See, I have three circles." "Hmph, I didn't think I saw that in your class." A younger student brought up the ceiling and murmured...
Top-k $k = 30$	"Aha. Aah! That makes three." Professor Gosh sighed. "Oh my God, you had to have been there." "Well, I'm not sure if that's the right term, but this circle is going to work." "But I just told you that it wasn't working, didn't you, you little sh**? I'm not kidding! That circle is gon na work!" "So what now?'...
Temp $\tau = 1.0$	It was teachers duty to ensure that our kingdom is pure and pure and successful, however it does not mean we do not try to be as rebellious as any other kingdom. Most of our teachers are suitable for rulership, being knowledgeable in sadistic rec thunking of time stopping magic circle created by several apprentices...
Mirostat $\tau = 3$	ready to go home, when a little kid starts to sketch a circle. "No. Just stop. It's not working." "But the circle turns to smoke and then to a flame. It's not working, I know it is!" Another kid, who was also drawing a circle with his phone, starts to scream. The teacher, looking at the circle, lost his patience and started to yell...
Typical $\tau = 0.2$	As soon as he got the clock on his desk he jumped up. The noise it was making had changed his day forever. His hair, normally unkempt, now glistened and he had to have more hair because of it. "Yes!" He had declared triumphantly as he watched his father fumble around in the backpack in search of a little compass he always kept with him. He took it from the desk and quickly walked out the door to see what had happened. He wasn't the first child in the world to draw something...

Table 4: Sample generations for story generation from GPT-2 large finetuned on the WRITINGPROMPTS dataset; examples correspond to ID 1 in the test set. Decoding strategy hyperparameters are chosen based off of best performance in human evaluations shown in Tab. 1. Full stories shown in App. B.

this property, which is a possible explanation for why the corresponding language processes often do not lead to human-like text. Yet, this observation provides a simple new criterion for decoding more human-like text from probabilistic language generators: constraining the sampling space to words that meet this criterion. In experiments on two language generation tasks, we find that our strategy—called locally typical sampling—leads to text of comparable or better quality than other stochastic decoding strategies according to human ratings. Further, when compared to these other decoding strategies, several quantitative properties of typically-sampled text more closely align with those of human text.

Ethical Concerns

In order to complete our human evaluation, we used a crowdsourcing platform. For each task, we made sure that the crowdworkers would be paid (at minimum) a wage of $15 per hour.

Another ethical consideration worth discussing concerns the use of language models for text generation. Text generated by these models may contain malicious content, either by design of the user or as

a byproduct of the training data/algorithm. While we hope the results of our work will not be misused, they may nonetheless provide insights for those employing these models with ill-intent as to how machine-generated text can be made more "human-like," and thus more convincing.

References

Matthew Aylett and Alice Turk. 2004. The smooth signal redundancy hypothesis: A functional explanation for relationships between redundancy, prosodic prominence, and duration in spontaneous speech. *Language and Speech*, 47(1):31–56.

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. 2021. Mirostat: A perplexity-controlled neural text decoding algorithm. In *Proceedings of the 9th International Conference on Learning Representations*.

Taylor L. Booth and Richard A. Thompson. 1973. Applying probability measures to abstract languages. *IEEE Transactions on Computers*, C-22(5):442–450.

Mark Braverman, Xinyi Chen, Sham Kakade, Karthik Narasimhan, Cyril Zhang, and Yi Zhang. 2020. Calibration, entropy rates, and memory in language models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 1089–1099. PMLR.

Leo Breiman. 1957. The individual ergodic theorem of information theory. *The Annals of Mathematical Statistics*, 28(3):809–811.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton and Co., The Hague.

Noam Chomsky. 1995. *The Minimalist Program*. MIT Press, Cambridge, MA.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

Christophe Coupé, Yoon Mi Oh, Dan Dediu, and François Pellegrino. 2019. Different languages, similar encoding efficiency: Comparable information rates across the human communicative niche. *Science Advances*, 5(9).

Thomas M. Cover and Joy A. Thomas. 2012. *Elements of Information Theory*. John Wiley & Sons.

Alexandra DeLucia, Aaron Mueller, Xiang Lisa Li, and João Sedoc. 2021. Decoding methods for neural narrative generation. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 166–185, Online. Association for Computational Linguistics.

Sander Dieleman. 2020. Musings on typicality.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? The inadequacy of the

mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

August Fenk and Gertraud Fenk. 1980. Konstanz im Kurzzeitgedächtnis-Konstanz im sprachlichen Informationsfluß. *Zeitschrift für experimentelle und angewandte Psychologie*, 27(3):400–414.

Edward Gibson, Richard Futrell, Steven T. Piantadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen, and Roger Levy. 2019. How efficiency shapes human language. *Trends in Cognitive Sciences*, 23(5):389–407.

Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. 2019. An empirical investigation of global and local normalization for recurrent neural sequence models using a continuous relaxation to beam search. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1724–1733, Minneapolis, Minnesota. Association for Computational Linguistics.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, 8:93–108.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George

van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *CoRR*, abs/2203.15556.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *Proceedings of the 8th International Conference on Learning Representations*.

Fred Karlsson. 2010. Syntactic recursion and iteration. In Harry van der Hulst, editor, *Recursion and Human Language*, chapter 3, pages 43–68. De Gruyter Mouton, Berlin, New York.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 284–294, Melbourne, Australia. Association for Computational Linguistics.

Konrad Knopp. 1954. *Theory and Application of Infinite Series*. Blackie & Son Ltd., London.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 41(5):1202–1241.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, Tokyo, Japan. Association for Computational Linguistics.

Roger Levy and T. Florian Jaeger. 2007. Speakers optimize information density through syntactic reduction. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings*

of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880, Online. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. 2020. Don't say that! Making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4715–4728, Online. Association for Computational Linguistics.

Kyle Mahowald, Evelina Fedorenko, Steven T. Piantadosi, and Edward Gibson. 2013. Info/information theory: Speakers choose shorter words in predictive contexts. *Cognition*, 126(2):313–318.

Brockway McMillan. 1953. The basic theorems of information theory. *The Annals of Mathematical Statistics*, 24(2):196–219.

Clara Meister, Elizabeth Salesky, and Ryan Cotterell. 2020a. Generalized entropy regularization or: There's nothing special about label smoothing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6870–6886, Online. Association for Computational Linguistics.

Clara Meister, Tim Vieira, and Ryan Cotterell. 2020b. If beam search is the answer, what was the question? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Online. Association for Computational Linguistics.

Clara Meister, Gian Wiher, Tiago Pimentel, and Ryan Cotterell. 2022. On the probability–quality paradox in language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 36–45, Dublin, Ireland. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *Proceedings of the 5th International Conference on Learning Representations*.

Moin Nadeem, Tianxing He, Kyunghyun Cho, and James Glass. 2020. A systematic characterization of sampling algorithms for open-ended language generation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 334–346, Suzhou, China. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey E. Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *Proceedings of the 5th International Conference on Learning Representations*.

Steven T. Piantadosi, Harry Tily, and Edward Gibson. 2011. Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences*, 108(9):3526–3529.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. MAUVE: Measuring the gap between neural text and human text using divergence frontiers. In *Advances*

in *Neural Information Processing Systems*, volume 34, pages 4816–4828. Curran Associates, Inc.

Tiago Pimentel, Clara Meister, and Ryan Cotterell. 2022. Cluster-based evaluation of automatically generated text. *CoRR*, abs/2205.16001.

Tiago Pimentel, Clara Meister, Elizabeth Salesky, Simone Teufel, Damián Blasi, and Ryan Cotterell. 2021. A surprisal–duration trade-off across and within the world's languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 949–962, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tiago Pimentel, Brian Roark, and Ryan Cotterell. 2020. Phonotactic Complexity and Its Tradeoffs. *Transactions of the Association for Computational Linguistics*, 8:1–18.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Peter A. Reich. 1969. The finiteness of natural language. *Language*, 45(4):831–843.

Carson T. Schütze. 2016. *The empirical base of linguistics: Grammaticality judgments and linguistic methodology*. Classics in Linguistics 2. Language Science Press, Berlin.

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. Do massively pretrained language models make better storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 843–861, Hong Kong, China. Association for Computational Linguistics.

Claude E. Shannon. 1948. A mathematical theory of communication. *Bell System Technical Journal*, 27:623–656.

Claude E. Shannon. 1951. Prediction and entropy of printed english. *Bell System Technical Journal*, 30(1):50–64.

Felix Stahlberg and Bill Byrne. 2019. On NMT search errors and model errors: Cat got your tongue? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3356–3362, Hong Kong, China. Association for Computational Linguistics.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *Proceedings of the 8th International Conference on Learning Representations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Noga Zaslavsky, Charles Kemp, Terry Regier, and Naftali Tishby. 2018. Efficient compression in color naming and its evolution. *Proceedings of the National Academy of Sciences*, 115(31):7937–7942.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. Trading off diversity and quality in natural language generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages

25–33, Online. Association for Computational Linguistics.

George Kingsley Zipf. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley Press, Oxford, UK.

A Human Evaluation Setup

We use Amazon Mechanical Turk framework for collecting human ratings of text. We use solely MTurk Master Workers in order to maximize the quality of our ratings. For story generation and abstractive summarization, each Human Intelligence Task (HIT) consists of either a single prompt from which a story should be generated or a single news article to be summarized. The raters are first presented with the different rating criteria, along with descriptions of the type of text that meets these criteria at different levels of the scale. These definitions can be seen in 3 and 4. Raters are additionally provided several examples of stories/summarizations meeting/failing to meet the rating criteria. They are then presented with the respective prompt/news article and the corresponding stories/summaries generated by different decoders and by the reference in random order. We use an attention check in each HIT. Responses where the attention check has been failed are thrown out. For each of the rating criteria, the rater assigns a score from 1 to 5. For each story/summarization and each of the criteria, we take the median score across raters as the final respective score. Statistics for these scores can be seen in Tab. 5. Workers are awarded $1.50 per HIT for the abstractive summarization task and $2 per HIT for the story generation task, for which entries were longer. These rates translate to >$15/hour.

Figure 3: Stories survey.

Figure 4: Summarization survey.
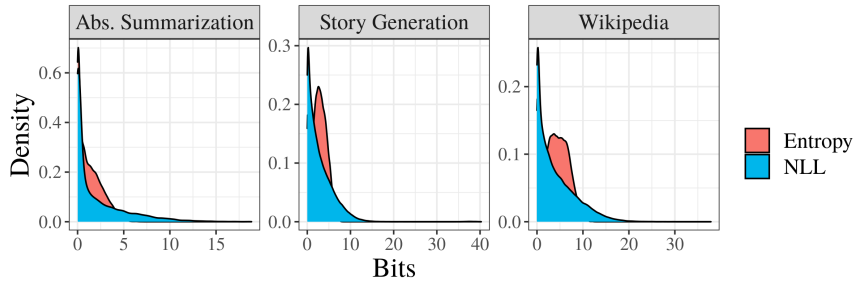
B   Additional Results

Figure 5: Distributions of conditional entropies and information contents per token for three different language generation tasks for human text, i.e., the reference text for each of the respective datasets.

Decoder	Story Generation (l)			Story Generation (m)			Summarization	
	Coherence	Fluency	Interestingness	Coherence	Fluency	Interestingness	Fluency	Relevance
Reference	4.36 (±0.31)	4.25 (±0.23)	4.56 (±0.25)	4.02 (±0.27)	4.2 (±0.27)	4.15 (±0.2)	4.43 (±0.25)	4.18 (±0.27)
Beam (k=5)	–	–	–	–	–	–	4.47 (±0.24)	4.23 (±0.28)
Temperature (τ=0.9)	4.32 (±0.25)	4.16 (±0.19)	4.47 (±0.27)	4.02 (±0.22)	4.26 (±0.29)	4.19 (±0.24)	4.36 (±0.25)	4.13 (±0.26)
Temperature (τ=1)	4.36 (±0.28)	4.25 (±0.22)	4.47 (±0.30)	4.02 (±0.32)	4.2 (±0.29)	4.18 (±0.22)	4.42 (±0.26)	4.15 (±0.28)
Nucleus (η=0.9)	4.32 (±0.25)	4.28 (±0.24)	4.48 (±0.31)	3.99 (±0.27)	4.16 (±0.32)	4.13 (±0.21)	4.39 (±0.27)	4.13 (±0.3)
Nucleus (η=0.95)	4.3 (±0.28)	4.28 (±0.29)	4.49 (±0.26)	4.00 (±0.19)	4.24 (±0.35)	4.14 (±0.17)	4.44 (±0.26)	4.08 (±0.29)
Top-k (k=30)	4.35 (±0.25)	4.21 (±0.24)	4.53 (±0.27)	4.03 (±0.24)	4.2 (±0.3)	4.16 (±0.22)	4.44 (±0.24)	4.18 (±0.26)
Top-k (k=40)	4.34 (±0.27)	4.24 (±0.23)	4.53 (±0.25)	4.00 (±0.27)	4.17 (±0.31)	4.11 (±0.18)	4.41 (±0.25)	4.17 (±0.33)
Mirostat (τ=3)	4.39 (±0.27)	4.26 (±0.23)	4.55 (±0.27)	4.02 (±0.22)	4.16 (±0.32)	4.17 (±0.22)	–	–
Typical (τ=0.2)	4.36 (±0.29)	4.24 (±0.24)	4.55 (±0.25)	4.07 (±0.26)	4.23 (±0.32)	4.14 (±0.26)	4.37 (±0.28)	4.16 (±0.29)
Typical (τ=0.95)	4.35 (±0.28)	4.24 (±0.23)	4.53 (±0.26)	4.04 (±0.21)	4.18 (±0.31)	4.18 (±0.22)	4.42 (±0.28)	4.22 (±0.27)

Table 5: Breakdown of human ratings on quality metrics per task; results for story generation are from finetuned versions of GPT-2 medium (m) and large (l). Values in blue are variances.
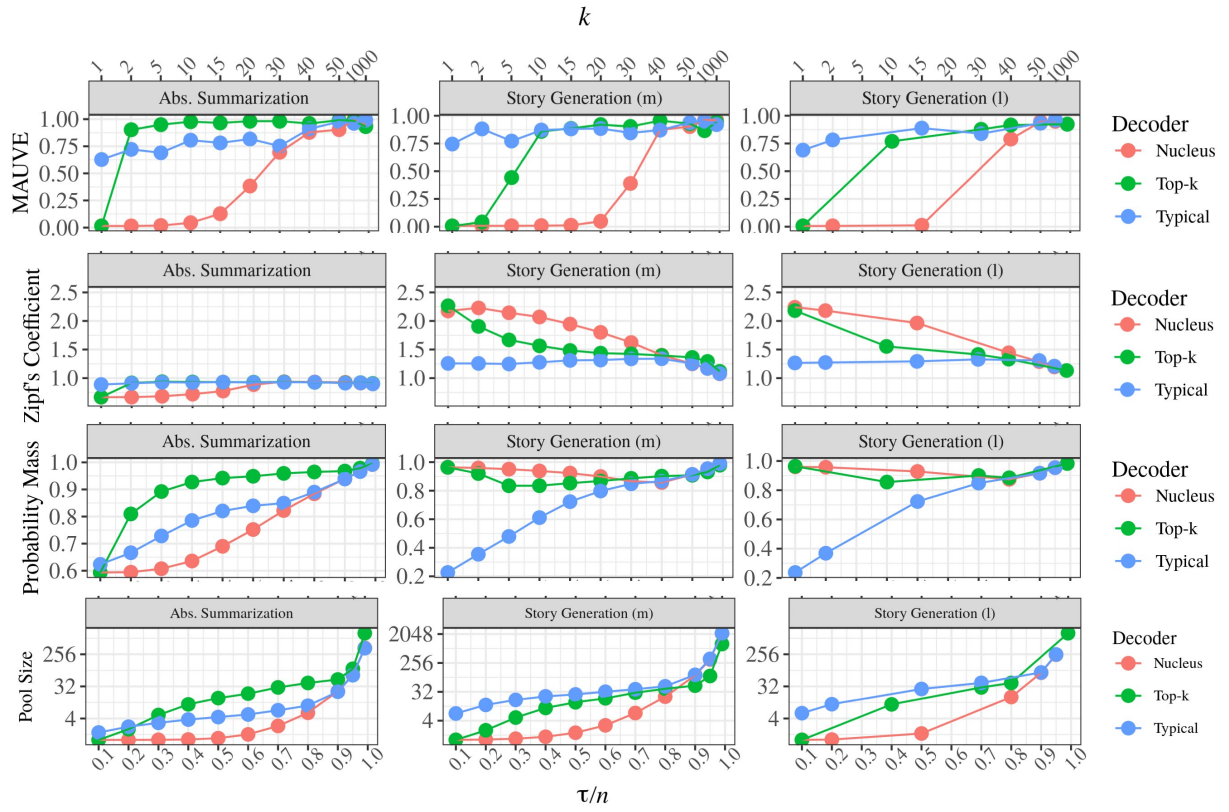
Figure 6: MAUVE, Zipf's coefficient, (average) probability mass of candidate token pool and (average) candidate token pool size as a function of decoder hyperparameters for nucleus, top-k and locally typical sampling.

Abstractive Summarization (CNN/DailyMail)

Prompt	(CNN) The attorney for a suburban New York cardiologist charged in what authorities say was a failed scheme to have another physician hurt or killed is calling the allegations against his client "completely unsubstantiated." Appearing Saturday morning on CNN's "New Day," Randy Zelin defended his client, Dr. Anthony Moschetto, who faces criminal solicitation, conspiracy, burglary, arson, criminal prescription sale and weapons charges in connection to what prosecutors called a plot to take out a rival doctor on Long Island. "None of anything in this case has any evidentiary value," Zelin told CNN's Christi Paul. "It doesnt́ matter what anyone says, he is presumed to be innocent." Moschetto,54, pleaded not guilty to all charges Wednesday. He was released after posting $2 million bond and surrendering his passport. Zelin said that his next move is to get Dr. Moshetto back to work. "He's got patients to see. This man, while he was in a detention cell, the only thing that he cared about were his patients. And amazingly, his patients were flooding the office with calls, making sure that he was OK," Zelin said. Two other men – identified as James Chmela, 43, and James Kalamaras, 41 – were named as accomplices, according to prosecutors. They pleaded not guilty in Nassau County District Court, according to authorities. Both were released on bail. A requests for comment from an attorney representing Chmela was not returned. It's unclear whether Kalamaras has retained an attorney. Police officers allegedly discovered approximately 100 weapons at Moschetto's home, including hand grenades, high-capacity magazines and knives. Many of the weapons were found in a hidden room behind a switch-activated bookshelf, according to prosecutors. The investigation began back in December, when undercover officers began buying heroin and oxycodone pills from Moschetto in what was initially a routine investigation into the sale of prescription drugs, officials said. During the course of the undercover operation, however, Moschetto also sold the officers two semiautomatic assault weapons as well as ammunition, prosecutors said. Moschetto allegedly told officers during one buy that he needed dynamite to "blow up a building." He later said he no longer needed the dynamite because a friend was setting fire to the building instead. Kalamaras and Chmela are believed to have taken part in the arson, according to prosecutors. "The fire damaged but did not destroy the office of another cardiologist whose relationship with Dr. Moschetto had soured due to a professional dispute," according to the statement from the district attorney's office. Moschetto allegedly gave an informant and undercover detective blank prescriptions and cash for the assault and killing of the fellow cardiologist, according to prosecutors. He also requested that the rival's wife be assaulted if she happened to be present, authorities said. "He was willing to pay $5,000 to have him beaten and put in a hospital for a few months, and then he said he would pay $20,000 to have him killed," said Assistant District Attorney Anne Donnelly, according to CNN affiliate WCBS.
Reference	A lawyer for Dr. Anthony Moschetto says the charges against him are baseless. Moschetto, 54, was arrested for selling drugs and weapons, prosecutors say. Authorities allege Moschetto hired accomplices to burn down the practice of former associate.
Beam	Dr. Anthony Moschetto faces criminal solicitation, conspiracy, burglary, arson and weapons charges. "None of anything in this case has any evidentiary value," his attorney says.
Temp	Dr. Anthony Moschetto accused of planning to kill doctor, drug dealer. Scheme imploded when a crisis developed in their relationships. Preliminary bond hearing set for August 5.

Nucleus	Dr. Anthony Moschetto, 54, pleaded not guilty to charges Wednesday. Two men – identified as James Chmela and James Kalamaras – were named as accomplices.
Top-k	Dr. Anthony Moschetto is accused of providing police with weapons and prescription drugs. Authorities say he was part of a conspiracy to harm or kill a rival doctor. His attorney calls the allegations against his client "completely unsubstantiated"
Typical	Dr. Anthony Moschetto is charged with crimes including arson, conspiracy, burglary, prescription sale, weapons charges. His attorney says "none of anything in this case has any evidentiary value"

Story Generation (WritingPrompts)

Prompt	A kid doodling in a math class accidentally creates the world's first functional magic circle in centuries.

Reference It was dark and Levi was pretty sure he was lying on his back. There was firelight flickering off of what was left of a ceiling. He could hear something but it was muffled. He was almost positive it was screaming. When he tried to move he felt an excruciating pain in his left side that caused him to cry out. His hand moved to it instinctively and found something protruding from the wound. It seemed to be a pipe of some kind. He attempted to pull it out but it was covered in blood making it slick. Levi's ears began ringing and the sounds began to become clearer, it felt quite a bit like some one was driving needles into his eardrums. The sounds he was hearing were definitely screams and not just one person's, a lot of people were screaming or yelling. There was some one close to him that was crying. He looked in the direction of the tears and seen his sister kneeling beside him, her hands covering her face. "What happened Laur?" He was shocked at the sound that barely rasped out from between his lips. His sister's hands jerked down and she stared down at Levi with a shocked look on her face. "bu... I tho... you weren't breathing!" What started as a whisper ended in yell as she threw her self across her brother and began to sob anew. Levi cried out hoarsely but she didn't hear. She just continued to cling to him like a little girl that had just found her lost doll. He put one of his arms around her and scanned the room as much as he could. It looked like a warzone, like something out of one of the many shooters in his gaming collection. "What the hell?" There were dead bodies everywhere, he recognized some of them. There were firefighters and EMT's running around trying to find survivors in the rubble. Most of the screams were coming from survivors. He seemed to be laying on top of the bottom of a desk, and he was pretty sure the pipe sticking out of his side was a actually one of the legs. Then he spotted it lying about two feet from his right leg, a round section of desk about the size of a softball. On it was a round symbol with geometric shapes glowing with dark red embers and a dull tendril of smoke rising up from it. It all came back to him in rush. He drew that in his notebook. It was second period and his trig teacher had this monotonous voice that could put a crack head to sleep. Laurana caught him doodling and had thrown a pencil at him to scold him silently, which made him jerk as he was connecting the last line on his drawing. Then there was the light and the heat and lastly the dark. Did he do this? What the hell was going on here? A voice brought him out of his little flashback. "Sir. Sir? Sir!" it was a firefighter. "I need you to stay still. You're going to hear a loud noise from behind you. I need you stay calm." Kay. "He just blew up his classroom with a piece of paper and a pencil and this guy wants him to stay calm." it's going to cause this desk to vibrate a lot and your going to feel some pain but there is nothing that we can do about it. We will work as fast as possible." The sound of the saw kicking on was not near as bad as he had thought, but Levi was no where near ready for when it made contact with the leg of the desk. His cries of agony began to drown out the sound of everything around him until darkness overwhelmed him. Sorry about my grammar it's been a while since I have written anything also, first time poster so take it easy on me; p

Nucleus

"Why do you draw a circle?" He asked as he handed the piece of paper over to her. She rolled her eyes and looked down at the picture, "Because I like math!" "Then what?" "I like to draw the circles." She ran a hand through her curls and looked away. "See, I have three circles." "Hmph, I didn't think I saw that in your class." A younger student brought up the ceiling and murmured. "Yeah, sure. It's just, why do we draw the circles?" "Because they're circles." She looked up. "But they don't match up!" She pointed at the nearby wall, "They don't start out like this!" She ran a hand over the map on the wall, "It starts out the same." She turned and pointed to the map marked with candles. "You have fire here." "Well that's nice." He turned away from her, "I don't think the artists in class thought about it that way." She pointed again. "See? Even your circle!" "That's my circle." "Look, Mason." "Hear, daughter." "I don't see a man, and so can't prove it." The two young adults nodded in agreement. Then she began to speak, "I draw the circles because I like to guess about what will happen when I make circles." "Yeah, and that's what we're going to do when you make a circle and turn the paper to our faces."

Top-k

"Aha. Aah! That makes three." Professor Gosh sighed. "Oh my God, you had to have been there." "Well, I'm not sure if that's the right term, but this circle is going to work." "But I just told you that it wasn't working, didn't you, you little sh**? I'm not kidding! That circle is gon na work!" "So what now?" "Well, that's for you to figure out. First, you've got to figure out a way to communicate with the circle. Maybe the spell isn't right for it, but it's not going to ruin the circle." "Okay, okay, you've told me to figure it out so I'll do a whole lot of work. But how exactly do you communicate a spell to a circle of stones in a basement?" "Simple, really. You put it inside the circle, and it magically works its way around the circle to get to the person you wrote 'I AM' on." "But how? Why am I supposed to be the main caster?" "Because you don't have magic powers to cast spells." "What? That's not even supposed to be possible." "You are special, you know." Professor Gosh looked at me with a little bit of sympathy. "I can't believe you're a wizard, you can do all this stuff when you're a kid." "Well, yeah, that's what they told me, so I guess you're right. But no matter how hard you try, you can't control magic. Don't worry, we'll figure out a way to make you a wizard one of your own." "Okay, I think you understand, I still haven't figured out how to create a spell, but I am so excited about the possibilities!" "Alright, let's do this."

Temp

It was teachers duty to ensure that our kingdom is pure and pure and successful, however it does not mean we do not try to be as rebellious as any other kingdom. Most of our teachers are suitable for rulership, being knowledgeable in sadistic rec thunking of time stopping magic circle created by several apprentices at the same time. After the first amendment, we decided that whilst having a society that thrives on protecting and serving the majority of people, whilst ensuring that the needs of some are met, the elements of freedom, equality and liberty shall not be infringed upon. Sadly, we ' ve done little to uphold those values since the day our forefathers signed the ancient Papercut act and limited the extent of human freedoms. It is now a day we lay a steady hand on the pages of history, hoping that the rules shall be rewritten and brought to what they once were. The next barrier to seeing the world anew as a black pantheon was a fallen occult student, Mr Pegarchow in Commerce, brought to history via the magic eye how to make a magic circle in the first place. This student made a hoop in the library that can be opened with a simple spiral. If a magic circle is inside this hoop, the magical illusion created is that a circle is made with the appropriate essence of spirit. An image of Pgyarchow being covered by the hoop, with the colours that inhabit it are verily magical. The magician essentially created... magic. To this day, not a single piece of paper hath survived contact with the magic eye. There were so many experiments looking for what function to draw in, or when it starts, 3rd century of my not-so-distant time. Nothing turned out as expected and everyone quickly passed into subculture that embraced the forbidden art of forgetfulness. However, even through the strongest taboo that could cause, symbols of independent spirit were still made and used as religion source material. I am the last wizard right now though not by choice, which was rather problematic after my mentor ' s death. However, I am different, and a magician, not by birth but via training and knowledge dedcialized from the single book of knowledge stored in my heart, able to share it with everyone, even those who share their mothers blood. To mhm, Lizonun

Mirostat
ready to go home, when a little kid starts to sketch a circle. "No. Just stop. It's not working." "But the circle turns to smoke and then to a flame. It's not working, I know it is!" Another kid, who was also drawing a circle with his phone, starts to scream. The teacher, looking at the circle, lost his patience and started to yell at the student: "This circle is not working, you need to come up with some more inventive magic demonstration!" "I like it. You can draw it with your phone, it's not hard." "No, stop it. Just stop drawing it like that and move on with your day. You look troubled." "No, I do. I can get some ideas for my magic circle. Also it looks cool and I can tell everyone that it's working." "Ok, but you still have to come up with a name for the circle." "I can come up with kenning, kenning-perforant, kenning-untergrund or kenning-unterhrund depending on the layout. Then my circle is named kenning-unterhrund, and then I post that on reddit, and I sell it on Etsy!" "Uhh... you spelt it wrong again using your phone, kenning-unterhrund for what's a circle in the middle of a room! You're not getting any creativity from that!" "Still, I'm not that bad. I could come up with a name too. We could see if our circle works on reddit!" The teacher starts to object, but the kid just brushes him off: "open the door and let me in, I'll tell everyone my name for my circle." The door is open and the teacher comes in. "KENNING-UNTER DOES N'T WORK, IT'SBAGS!" The kid answers: "Open the door and let me in. If there's no door, I'll come back tomorrow and make a new one." He sounds confident in his plan. "Ok, kenning-untergrund, kenning-unterhrund or kenning-unteromactually totalitarian need you! Honoring your name will make your circle work better!" The kid is ready to spell it then: "The kenning-untergrund, kenning-unterunterhrund, kenning-unteronAAAKE!" The teacher asks: "What do you mean fails?" "Ok, and I forgot why I just copy all my circles to my phone:)" "You can't make circles! That would be impossible. And you forgot the reason for your circle to not work:)" "Yes, I did! I forgot the purpose of my circle and this happened to me too:)" "That might happen, but get out of here immediately, the spell is about to hit the ground!" The kid grabs the teacher by the shirt collar: "So what did you get out of it? I thought magic circles are making people happy?" The teacher looks at him with fury in his eyes: "You should go to the yelling and crying section of the school, before the spell hits the ground. You're going to repeat the'Wizard' spell again." "Thanks, but no, I'll leave." The kid grabs the ground and he leaves the school. Then his circle is gone. The teacher is still trying to figure out what should be done about the circle because he forgot to ask some of his other students. So he goes home to his basement, to his computer and he starts to draw a circle. And then he draws another one. And another one. Soon enough, there are some things that are drawn on his screen: a circle with smoke coming out of it, a circle with fog coming out of it, a circle with fire coming out of it, a circle with flavouring of words coming out of it, a circle with adding words into it and a circle with some mixture of words coming out of it. As the circle is wants to be drawn, he starts to move his finger Mask he was using to draw the circle. When suddenly, his finger don't move, slowly but steadily, like a hand is starting to move against his will. The students are looking at the screen now. They start to scream and yell: "Why is the hand still moving?!!" "Yes,

Typical

As soon as he got the clock on his desk he jumped up. The noise it was making had changed his day forever. His hair, normally unkempt, now glistened and he had to have more hair because of it. "Yes!" He had declared triumphantly as he watched his father fumble around in the backpack in search of a little compass he always kept with him. He took it from the desk and quickly walked out the door to see what had happened. He wasn't the first child in the world to draw something to him and so far, so good. It wasn't as big as a tree, but it did sparkle a little bit and made a few notes in his calculus textbook. As he left the room, the others around him also drew and explained to him their ability. Most kids who knew their way around magic started to teach their spells in a lesson on learning. His teachers tried to tell him about his parents but he only heard his name as they asked if he knew about them. They explained how he would always go with them when he learned something and then asked how he had found the way. His mother said she always did the work when she taught and then how his father used the ingredients he had brought along to create his circle. It wasn't anything like what his mother would use to do her magic but he wasn't interested in her or his magic because it wasn't his favorite type. As he got older he got to watch as the world moved. His family went out on field trips to look at stars and wonder where the other world is. They asked where his mother and father went, and how long they spent on that. It wasn't anything like he expected but they never explained what the field trips were about. As he grew older he became a man with more responsibility and began to notice things like this happen to the rest of the world. As a man who didn't believe in magic he found his circle was missing from the side of the road he always stopped on to check his bearings. His neighbors thought it was weird that his neighbors' circles would get in the way. One day his friends stopped on the street he was walking down and told him to get to his car and take him to a store so they could find his circle. When they found his circle they gave it back and it had no clue where it came from. After some further digging he discovered the missing part of the circle. A pentagram that appeared out of nothingness on the road and for the longest time, no one had been able to find it. A couple weeks later a truck came out and dumped all of the construction workers that were using the road and blocked it. When his circle had disappeared the news talked about the weird pentagram on the road. This went on for months until his mother noticed he wasn't coming home. When she came home, she looked in his room and he wasn't there. When he wasn't there he never made his presence known to her and never tried to teach his spells to her. They eventually went back to their apartment together. Her father didn't even acknowledge his daughter at first but his wife didn't think to call him or get his attention. The father, seeing the pattern and a good time with his daughter decided to stay. She told her parents to check their calendar. It had stopped being so late that her father thought to take his place to try to sleep off his spell exhaustion. The parents realized something was up and the wife suggested she get her boyfriend and see if she could have a quiet time. It wasn't a normal time to do such things, but when he found her and saw his father had passed he told his girlfriend that they would meet for drinks to get over it. The wife got ready to go, and they drove home. She pulled over in front of their house. A truck pulled up in the driveway, stopped and looked out and began to slowly pull into the driveway. When she was looking in his eyes, she asked if he knew about her mother's strange powers. When he replied he asked how his father got the circle that they kept around for all of these years.

The man pulled up in his truck and gave his girlfriend the back seat and started the truck. She took him home, they drank, and watched a movie about dragons and aliens and made up a bedtime story for the boy and girl. After dinner the boy slept through the night. He dreamt that the circle had gone missing, that it had disappeared in his yard, and that the car parked next to him was covered in dust and it smelled like his room had been raided by monsters. After his sleep, the mother took him to his room to wake him up. When she entered he didn't react at all. "Are you alright honey?" The mother asked him. The boy responded in the most puzzled tone she had ever heard him.

Table 6: Full sample generations for abstractive summarization and story generation. We use samples from the model fine-tuned from GPT-2 large for story generation.