Sparse Text Generation

Pedro Henrique Martins[Ψ]   **Zita Marinho**[Ɖ ℳ] and **André F. T. Martins**[Ψ ⚲ ♭]

[Ψ]Instituto de Telecomunicações [Ɖ]Priberam Labs [ℳ]Institute of Systems and Robotics
[⚲]LUMLIS (Lisbon ELLIS Unit), Instituto Superior Técnico [♭]Unbabel
Lisbon, Portugal
pedrohenriqueamartins@tecnico.ulisboa.pt,
zita.marinho@priberam.pt, andre.t.martins@tecnico.ulisboa.pt.

Abstract

Current state-of-the-art text generators build on powerful language models such as GPT-2, achieving impressive performance. However, to avoid degenerate text, they require sampling from a modified softmax, via temperature parameters or ad-hoc truncation techniques, as in top-k or nucleus sampling. This creates a mismatch between training and testing conditions. In this paper, we use the recently introduced entmax transformation to train and sample from a natively sparse language model, avoiding this mismatch. The result is a text generator with favorable performance in terms of fluency and consistency, fewer repetitions, and n-gram diversity closer to human text. In order to evaluate our model, we propose three new metrics for comparing sparse or truncated distributions: ϵ-perplexity, sparsemax score, and Jensen-Shannon divergence. Human-evaluated experiments in story completion and dialogue generation show that entmax sampling leads to more engaging and coherent stories and conversations.

1 Introduction

The last few years have seen a tremendous progress in neural language modeling, with a strong impact on key NLP tasks such as machine translation (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017), speech recognition (Chorowski et al., 2015), summarization (Rush et al., 2015), and open-ended text generation (Radford et al., 2019). Benefiting from large amounts of data, models such as Transformer-XL (Dai et al., 2019) and GPT-2 (Radford et al., 2019) have achieved impressive perplexity scores on language modeling. However, the generated text is still often repetitive and incoherent (Table 1).

A downside of current approaches is the mismatch between training and testing conditions:

Context:	*"Are they ready to go live?" Zuko asked. "Yup, in just a minute" Tomi replied as he leaned back in his reclining chair. This was a moment that the entire galaxy would see. After 10 years...*
Greedy:	... of study, the scientists had finally come to the conclusion that the universe was not a place for living. *The scientists* had been studying *the universe* for over a century. *They had been studying the universe for over a century.*
Top-k:	... of studying this phenomenon, the team had finally come up with a plan to *study* what the aliens were doing. *The team* would be looking for a way to create a space station. "And what would you call it?" *"What would you call the space station?"*
Nucleus:	... of diligent testing, *and some test run after that*, the results were well received. "they're waiting for you, because *there's no way they would." Stowedly at a local wedding place, the two archaeologists spoke together.*
Entmax:	... of hibernation, Tomi was back. These last days had been a significant step forward in his mission. This time, Tomi was not alone. All the empires had aligned together and the world's leadership began to openly support his mission.

Table 1: Completion of a story from the Writing-Prompts dataset, using the greedy decoding, top-k sampling, nucleus sampling, and entmax sampling (our proposal) methods. Repetitions and off-topic text are *highlighted.*

models are usually trained to maximize the likelihood of observed text. However, when generating, state-of-the-art models sample from a **truncated** and **renormalized** softmax distribution (Fan et al., 2018; Holtzman et al., 2020). They do so as a compromise to avoid two extremes: a deterministic search for the most probable sentence (via greedy decoding or beam search) usually results in dull and repetitive "degenerate text" (Li et al., 2016a, 2017; Holtzman et al., 2020); stochastically sampling from the full softmax distribution, on the other hand, often generates many implausible

words from the tail of the distribution (Fan et al., 2018). The recently proposed *nucleus sampling* approach (Holtzman et al., 2020) sets the truncation point based on the cumulative distribution function, *i.e.*, it considers the top words with a cumulative probability P. In this approach the number of words to sample from are selected according to the context, in opposition to *top-k sampling* (Fan et al., 2018), which samples from the k most probable words. However, the "sparsity" introduced by both methods is artificially imposed at run time, not learned during training.

A second problem is that it is hard to compare different truncation strategies—for example, we cannot easily evaluate how the resulting truncated distributions behave as language models, since the most widely used metric for language modeling—perplexity—cannot handle sparse distributions: if a model assigns zero probability to a single reference word, it gets infinite perplexity for the entire sample. For this reason, previous works generate from a *truncated* softmax, but report the perplexity of the *full* softmax distribution (Welleck et al., 2020; Li et al., 2020). Others use the latter to compare perplexity on the generated text with that on human text (Holtzman et al., 2020, §4.2), or resort to distributional statistics (Zhu et al., 2018).

In this paper, we propose a new approach—**entmax sampling** (§3)—that eliminates the mismatch between training and test conditions. Key to our approach is the recently proposed entmax transformation (Peters et al., 2019). Entmax transforms a vector of scores into a **sparse probability distribution**, preventing implausible words from receiving *any* probability mass. Moreover, it does so **natively**: it comes with a well-defined loss function that allows it to learn its sparsity automatically from the data, during training. This results in a new stochastic text generator where the number of possible word types varies with the context (like nucleus sampling), but that generates by sampling directly from its output distribution (like softmax), and where the sparsity of this distribution is present during training (unlike any existing method).

As a second contribution, we propose **three new metrics** to support the evaluation of sparse language models (§4): ϵ-perplexity, sparsemax score, and Jensen-Shannon divergence. We show that these metrics are well supported theoretically and can be used to compare our method with various truncation and temperature techniques.

Experiments in language modeling, story completion, and dialogue generation (§5) show that entmax sampling generates more diverse text and fewer repetitions than nucleus and top-k sampling.[1]

1.1 Related work

Decoding methods. While greedy decoding and beam search are popular strategies for sequence-to-sequence tasks, such as machine translation, Knowles et al. (2016) and Stahlberg and Byrne (2019) showed that searching for the most probable sentence in a model trained with likelihood maximization has a bias for short sentences. In open-ended generation, Fan et al. (2018) and Holtzman et al. (2018, 2020) have shown that these methods lead to repetitions and dull text. To overcome this, several authors proposed beam search variants which promote word diversity (Li et al., 2016b; Vijayakumar et al., 2018; Kulikov et al., 2018). An alternative to deterministic text generation is to sample directly from the softmax distribution. However, since the probability mass tends to accumulate in a long tail, this procedure generates unlikely words too often, leading to degenerate text (Fan et al., 2018; Holtzman et al., 2020). This can be mitigated by lowering the softmax temperature (Ficler and Goldberg, 2017), by sampling from the top-k most probable words only (Fan et al., 2018; Radford et al., 2019), or through nucleus sampling (Holtzman et al., 2020). We compare against these methods in §5.

Diversity-promoting models. In addition to new decoding methods, models that aim to increase word diversity and diminish repetition have also been introduced. Xu et al. (2018) proposed a diversity-promoting generative adversarial network, which rewards novel and fluent text. Holtzman et al. (2018) proposed augmenting the language model with several discriminators. More recently, Welleck et al. (2020) proposed augmenting the loss with an unlikelihood term that penalizes the generation of tokens that are present in the context, a method against which we compare in §5.

Sparse transformations and losses. At the core of our work are sparse alternatives to the softmax transformation. Martins and Astudillo (2016) proposed sparsemax and applied it to multi-label classification. This was generalized by Peters et al.

---

[1]The code used for the experiments and for the proposed metrics is available at `https://github.com/deep-spin/sparse_text_generation`.

(2019) via their α-entmax transformation, which was applied to sequence-to-sequence models for morphological inflection and machine translation. In contrast to our work, they performed *deterministic* decoding with beam search, and they did not consider open-ended generation.

Evaluation metrics. The most common metrics to evaluate text generation models are perplexity (Jelinek et al., 1977) and BLEU (Papineni et al., 2002). For open-ended generation, Zhu et al. (2018) observed that "no single metric is comprehensive enough". Other evaluations include corpus n-gram overlap (Yu et al., 2017; Press et al., 2017), and the Fréchet distance (Cífka et al., 2018). These approaches are aimed at the (harder) problem of evaluating the quality of generated text. By contrast, our paper proposes new metrics for evaluating *language models* in the task of predicting the next word conditioned on ground truth context (like perplexity does), but supporting sparse probability distributions (which perplexity does not).

2 Language Modeling

Language models assign probability to word sequences $x = \langle \text{START}, x_1, \ldots, x_T, \text{STOP} \rangle$, where each x_t is in a vocabulary $\mathcal{V}$, and $T \in \mathbb{N}$. This probability can be written as $p_\theta(x) = \prod_{t=1}^{T+1} p_\theta(x_t \mid x_{<t})$. We would like the model θ to assign high probability to real sentences, i.e., each distribution $p_\theta(\cdot \mid x_{<t})$ should assign a large probability value to the ground truth x_t.

Given a set $\mathcal{S}$ of training sentences, the usual strategy for learning the language model parameters θ is to minimize the negative log-likelihood:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{|\mathcal{S}|} \sum_{t=1}^{T_i} \log p_\theta(x_t^i | x_{<t}^i). \quad (1)$$

The standard choice to model $p_\theta(\cdot | x_{<t})$ in Eq. 1 is to compute a score vector z_t by conditioning on the context $x_{<t}$, and then applying a softmax transformation, $p_\theta(\cdot | x_{<t}) = \mathsf{softmax}(z_t)$, where

$$[\mathsf{softmax}(z_t)]_k = \frac{\exp(z_{tk})}{\sum_j \exp(z_{tj})}. \quad (2)$$

At decoding time, the language model generates sentences one word at a time, by sampling from the learned probability distribution. However, softmax yields a **dense** distribution, *i.e.*, some probability mass (even if small) is assigned to all the words in the vocabulary. Holtzman et al. (2020, §3) have

shown that, if we sample from this distribution directly, the resulting text becomes degenerate, with common incoherences arising due to the unreliability of the tail of the distribution. This motivated a line of work proposing "ad-hoc" modifications to the softmax distribution, to reduce the effect of the tail. Two of the most successful techniques, top-k and nucleus sampling (Fan et al., 2018; Holtzman et al., 2020), do so by truncating and renormalizing the distribution $p_\theta(\cdot | x_{<t})$. Note that these techniques are applied only at decoding time—during training the original softmax distribution is left untouched, being used as part of the optimization of the cross-entropy loss.

Our alternative to these ad-hoc modifications builds on *learnable* sparse transformations, as we shall see in §3. These transformations can produce sparse, zero-tailed probability distributions, learning the amount of sparsity from data. Therefore, sampling from these distributions directly is a natural way to prevent degenerate text.

3 Entmax Sampling

Key to our method is the recently proposed α-entmax family of transformations[2] (Peters et al., 2019), parametrized by a scalar parameter $\alpha \geq 1$:

$$\alpha\text{-entmax}(z_t) := \operatorname*{argmax}_{p \in \triangle^d} p^\top z_t + \mathsf{H}_\alpha(p). \quad (3)$$

Above, $\triangle^d := \left\{ p \in \mathbb{R}^d \mid \sum_{i=1}^d p_i = 1, p \geq 0 \right\}$ is the probability simplex, and H_α is the Tsallis α-entropy (Tsallis, 1988):

$$\mathsf{H}_\alpha(p) := \begin{cases} \frac{1}{\alpha(\alpha-1)} \sum_j (p_j - p_j^\alpha), & \alpha \neq 1 \\ -\sum_j p_j \log p_j, & \alpha = 1. \end{cases} \quad (4)$$

With $\alpha = 1$ and $\alpha = 2$, we recover the Shannon and Gini entropies, respectively.[3] When $\alpha \to \infty$, $\mathsf{H}_\alpha(p) \to 0$. Thus, 1-entmax, 2-entmax, and ∞-entmax recover softmax, sparsemax, and argmax, respectively. Blondel et al. (2019) have shown that, for $\alpha > 1$, entmax is able to output **sparse** probability distributions, where some words get **exactly** zero probability, whereas softmax ($\alpha = 1$) does not have this capability.

How can we learn this output sparsity during training? Following Peters et al. (2019), we replace the negative log-likelihood loss in Eq. 1 by

---

[2]`https://github.com/deep-spin/entmax`.
[3]The Gini entropy is $\mathsf{H}_2(p) := \frac{1}{2} \sum_j p_j(1 - p_j)$.

$$\mathcal{L}(\theta) = \sum_{i=1}^{|\mathcal{S}|} \sum_{t=1}^{T_i} \ell_\alpha(\boldsymbol{z}_t(\theta, x_{<t}), x_t), \quad (5)$$

where $\ell_\alpha(\boldsymbol{z}_t, x)$ is the α-**entmax loss**:

$$\ell_\alpha(\boldsymbol{z}_t, x) := (\boldsymbol{p}_\theta - \boldsymbol{e}_x)^\top \boldsymbol{z}_t + \mathsf{H}_\alpha(\boldsymbol{p}_\theta), \quad (6)$$

where $\boldsymbol{p}_\theta = \alpha\text{-entmax}(\boldsymbol{z}_t)$, and $\boldsymbol{e}_x$ is the one-hot vector corresponding to the ground truth word x. When $\alpha = 1$, we still recover the negative log-likelihood, $\ell_\alpha(\boldsymbol{z}_t, x) = -\log p_\theta(x)$, and, when $\alpha = 2$, this corresponds to the **sparsemax loss** (Martins and Astudillo, 2016), to be revisited in §4.

Entmax losses belong to the wider class of Fenchel-Young losses (Blondel et al., 2019) and, consequently, are convex on $\boldsymbol{z}$ and differentiable (with gradient $\nabla_{\boldsymbol{z}}\ell_\alpha(\boldsymbol{z}, x) = -\boldsymbol{e}_x + \boldsymbol{p}_\theta$). For $\alpha > 1$, they have a **separation margin**: the loss is zero iff $z_{tx} \geq z_{tx'} + \frac{1}{\alpha-1}$ for all $x' \neq x$, in which case $\boldsymbol{p}_\theta = \boldsymbol{e}_x$, i.e., the model puts all its probability mass in the correct word. This allows the model to be adaptive to the degree of uncertainty present: in some cases there are few plausible words, so most words should have probability zero, while in other cases a higher number of words are plausible and should be given probability mass.

Entmax sampling. At test time, we simply sample from the categorical distribution obtained by applying the entmax transformation to the scores $\boldsymbol{z}_t$ given by the model:

$$x_t \sim p_\theta(\cdot \mid x_{<t}) = \alpha\text{-entmax}(\boldsymbol{z}_t(\theta, x_{<t})). \quad (7)$$

Note that, in contrast to previously proposed methods such as top-k sampling and nucleus sampling (Holtzman et al., 2020), we sample directly from the learned sparse probability distribution over the words, without any calibration or ad-hoc modification. As in nucleus sampling and in opposition to top-k sampling, entmax sampling considers a varying number of tokens depending on the context. Moreover, as we show in Table 4, with entmax sampling this variability is higher.

4   Evaluation Metrics

Language models are commonly evaluated by computing their perplexity (ppl) on held-out data. Perplexity assesses the ability of a language model to predict the next word given the context:

$$\mathsf{ppl} = \exp\left(-\frac{1}{T}\sum_{t=1}^{T}\log p_\theta(x_t \mid x_{<t})\right). \quad (8)$$

However, its computation involves the logarithm of a probability. This poses a problem when we are using sparse or truncated probability distributions, since $\lim_{p\to 0}\log p = -\infty$. Usually, authors report the values for perplexity computed on the original probability distribution, before truncation. However, this metric does not allow different sparse decoding strategies to be compared.[4] As an alternative, we propose three different metrics (to better understand these metrics, comparative plots are shown in Fig. 2, App. E).

ϵ-**perplexity.** To be able to compute the perplexity for sparse distributions, the simplest approach is to smooth it by adding a small value ϵ to all terms followed by renormalization, as in additive (Laplace) smoothing (Chen and Goodman, 1999):

$$\epsilon\text{-ppl} = \exp\left(-\frac{1}{T}\sum_{t=1}^{T}\log\frac{p_\theta(x_t \mid x_{<t}) + \epsilon}{1 + \epsilon|\mathcal{V}|}\right). \quad (9)$$

Note that, like perplexity, ϵ-ppl only depends on θ via the probabilities assigned to the reference words. When used as a metric for a language model, we may regard ϵ as a calibration parameter that the language model is allowed to tune to better match the reference. We show in App. A that the optimal value of ϵ (i.e., the one that leads to the smallest ϵ-ppl) can be obtained from these probabilities by solving a simple convex optimization problem—this is convenient, since it avoids the need for manual tuning. A disadvantage of ϵ-ppl is that it still does not evaluate the original sparse distribution, but rather a modified version of it. However, when applied to variants of truncated softmax, by collapsing all the truncated probabilities to the same value ϵ, it is useful to measure how much truncation deteriorates its ability to rank words, compared to softmax.

Sparsemax score. We can derive a more interesting metric that handles sparse distributions directly. By setting $\alpha = 2$ in Eq. 6,[5] we obtain the sparsemax loss proposed by Martins and Astudillo (2016), $\ell_2(\boldsymbol{z}, x) = (\boldsymbol{p}_\theta - \boldsymbol{e}_x)^\top \boldsymbol{z} + \mathsf{H}_2(\boldsymbol{p}_\theta)$. We define the

---

[4]This is important not only when we have sparse or truncated probability distributions, but also to compare language models using different vocabularies: when using perplexity, if the ground truth word is not contained in the vocabulary, one usually considers the probability attributed to an UNK token instead of a zero probability, which leads to an unfair comparison between models with different vocabularies.

[5]If we set $\alpha = 1$ instead, we revert to perplexity.

sparsemax score (sp) as:

$$\begin{aligned}
\mathsf{sp} &= 1 - \min\{\ell_2(\boldsymbol{z}, x) \mid \mathsf{sparsemax}(\boldsymbol{z}) = \boldsymbol{p}_\theta\} \\
&= 1 - (\boldsymbol{p}_\theta - \boldsymbol{e}_x)^\top \boldsymbol{p}_\theta - \mathsf{H}_2(\boldsymbol{p}_\theta) \\
&= p_\theta(x) + \mathsf{H}_2(\boldsymbol{p}_\theta),
\end{aligned} \tag{10}$$

where H_2 is the Gini entropy (see footnote 3). Unlike perplexity, this score is bounded. In fact, it is always between 0 (when $\boldsymbol{p}_\theta = \boldsymbol{e}_{x'}$ with $x' \neq x$) and 1 (when $\boldsymbol{p}_\theta = \boldsymbol{e}_x$). We prove this fact in App. B. Interestingly, when the model $\boldsymbol{p}_\theta$ is deterministic (e.g., when it comes from greedy search), we have $\mathsf{H}_2(\boldsymbol{p}) = 0$, and the sparsemax score simply becomes the **word accuracy**. In the opposite case, when $\boldsymbol{p}_\theta$ is uniform, we obtain $\mathsf{sp} = \frac{1}{|\mathcal{V}|} + \frac{1}{2}\left(1 - \frac{1}{|\mathcal{V}|}\right) \to 0.5$ when $|\mathcal{V}| \to \infty$.

We show in App. C that this score is related to the Patrick-Fischer distance (Patrick and Fischer, 1969; Deza and Deza, 2009, p. 262).

Jensen-Shannon Divergence. Given two discrete probability distributions $\boldsymbol{p}_\theta$ and $\boldsymbol{q}$, and denoting their mixture (arithmetic mean) as $\boldsymbol{m} := \frac{\boldsymbol{p}_\theta + \boldsymbol{q}}{2}$, and the Kullback-Leibler divergence as KL, the Jensen-Shannon divergence is defined as:

$$\begin{aligned}
\mathsf{JS}(\boldsymbol{p}_\theta, \boldsymbol{q}) &= \frac{1}{2}\,\mathsf{KL}(\boldsymbol{p}_\theta\|\boldsymbol{m}) + \frac{1}{2}\,\mathsf{KL}(\boldsymbol{q}\|\boldsymbol{m}) \\
&= \frac{1}{2}\sum_{x \in \mathcal{V}} p_\theta(x) \log\left(\frac{p_\theta(x)}{m(x)}\right) \\
&\quad + \frac{1}{2}\sum_{x \in \mathcal{V}} q(x) \log\left(\frac{q(x)}{m(x)}\right).
\end{aligned} \tag{11}$$

The Jensen-Shannon divergence can be interpreted as a mutual information as follows (Grosse et al., 2002; Banerjee et al., 2005): consider a two-step process where we first toss a fair coin $B \sim \text{Bernoulli}(\frac{1}{2})$. If the outcome is heads, we sample the next word X according to the model $p_\theta(\cdot)$; if it is tails, we sample $x \sim q(\cdot)$. A word generated according to this process is governed by the mixture $m(\cdot)$, $x \sim m(\cdot)$. The Jensen-Shannon divergence between $\boldsymbol{p}_\theta$ and $\boldsymbol{q}$ is the mutual information between the random variables B and X, which equals $\mathsf{H}(B) - \mathsf{H}(B \mid X)$, where H is the Shannon entropy and $\mathsf{H}(B \mid X) = \sum_{x \in \mathcal{V}} m(x) H(B \mid X = x)$ is the conditional entropy. Hence, the Jensen-Shannon divergence can be seen as the reduction of uncertainty about the source B when we observe a sample x from the mixture $m(\cdot)$. The more similar the two distributions $\boldsymbol{p}_\theta$ and $\boldsymbol{q}$ are, the smaller this reduction is.

In our experiments, we report the JS as an evaluation metric for language models, setting $\boldsymbol{q} = \boldsymbol{e}_x$ (i.e., a one-hot distribution placed on the ground truth word x) and averaging the JS over the words. Like the sparsemax score described above, the JS is bounded: it is zero if $\boldsymbol{p}_\theta = \boldsymbol{e}_x$, and maximal ($\log(2)$) when $\boldsymbol{p}_\theta$ is a one-hot distribution placed on a different word. We show in App. D that, like ϵ-ppl (but unlike sp), the JS only depends on θ via the probabilities assigned to the reference words.

Comparing multiple models. The generalized JS allows to compare two or more trained models:

$$\mathsf{JS}(\boldsymbol{p}^1, \ldots, \boldsymbol{p}^K) = \frac{1}{K}\sum_{k=1}^{K} \mathsf{KL}(\boldsymbol{p}^k\|\boldsymbol{m}) \tag{12}$$

where $\boldsymbol{p}^1, \ldots, \boldsymbol{p}^K$ are the probability distributions of the different models and $\boldsymbol{m} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{p}^k$ is their mixture. This property can be useful for measuring the diversity between multiple models (e.g., when used in an ensemble system). We use this metric in App. I to rank the sentences in which the different models we compare disagree the most.

5 Experiments

We compare the different methods in three NLP tasks: language modeling (§5.1), story completion (§5.2), and dialogue generation (§5.3). In language modeling, we evaluate the model's fluency, while in story completion we also evaluate if the methods generate coherent and "interesting" text. In dialogue generation, we evaluate the methods' performance in an interactive task.

5.1 Language Modeling

Datasets and metrics. We performed experiments on three widely used language modeling datasets: WikiText-2 and WikiText-103 (Merity et al., 2016), and BookCorpus (Zhu et al., 2015). WikiText-2 and WikiText-103 are composed of Wikipedia articles, comprising around 2 and 100 million tokens for training, respectively. Their validation and test sets have 217,000 and 245,000 tokens. BookCorpus is composed of 11,038 freely available books. We used the standard split: 800 million tokens for training, 260,000 for validation, and 280,000 for testing.

We report the sparsemax score, Jensen-Shannon, and ϵ-perplexity (§4) to evaluate the methods' fluency, and the REP and WREP[6] (Welleck et al., 2020)

---

[6] REP measures the number of times that a word from the

	WikiText-2					WikiText-103					BookCorpus				
	sp	JS	ϵ-ppl	REP	WREP	sp	JS	ϵ-ppl	REP	WREP	sp	JS	ϵ-ppl	REP	WREP
Softmax	.682	.376	**12.74**	**.407**	.174	.683	.375	13.29	.349	.162	.680	.366	10.80	.376	.183
Softmax-τ	.680	.369	12.97	.414	.176	.682	.368	13.65	.359	.168	.677	.363	10.96	.391	.191
Greedy	.491	**.358**	459.13	.525	.232	.499	**.355**	512.50	.450	.210	.489	**.354**	506.86	.461	.211
Top-k	.682	.363	20.93	.437	.196	.683	.364	21.90	.373	.181	.676	.360	22.25	.399	.203
Nucleus	.684	.371	14.65	.412	.175	.686	.370	15.51	.357	.167	.678	.362	16.48	.392	.193
Unlikelihood	.473	.365	599.65	.467	.210	.471	.366	610.06	.410	.200	.475	.364	587.04	.418	.203
Entmax	**.688**	.369	13.91	**.407**	**.171**	**.694**	.373	**13.23**	**.346**	**.160**	**.687**	.362	**10.70**	**.374**	**.179**

Table 2: Language model evaluation on WikiText-2, WikiText-103, and BookCorpus test sets. For all metrics except sp, lower is better. See App. F for the results on the validation set.

to evaluate the methods' tendency to generate repetitions. All metrics are computed at the BPE level (Sennrich et al., 2016).

Fine-tuning GPT-2. We fine-tuned the GPT-2 medium model (Radford et al., 2019), which consists of a 24 layer transformer with 345 million parameters.[7] We fine-tuned three models with the following losses: negative log-likelihood (used for softmax, greedy, top-k, and nucleus sampling), unlikelihood training (Welleck et al., 2020), and entmax loss. For the unlikelihood training objective we replicated the authors' experiments. However, due to GPU memory constraints we had to reduce the context size from 512 to 256. The hyperparameters were chosen based on a grid search over $\alpha \in \{1.1, 1.2, 1.3, 1.5\}$ for entmax sampling, $k \in \{5, 10, 20, 50, 100\}$ for top-k sampling, $P \in \{0.5, 0.8, 0.85, 0.9, 0.95, 0.97\}$ for nucleus sampling, and $\tau \in \{0.7, 0.8, 0.9, 0.95, 0.97\}$ for softmax with decreased temperature. The selected hyperparameters are reported in Table 3. We report the results obtained on the validation sets of WikiText-2, WikiText-103, and BookCorpus on Table 10. Additional settings and the computational infrastructure are described in App. F.

Results. Table 2 shows the results. We observe that entmax sampling achieves consistently better sparsemax scores and number of repetitions. It also leads to better ϵ-perplexity scores than all other methods except plain softmax, which attains similar scores (entmax is slightly better for 2 out of 3 datasets). The JS score appears to favor extremely

previous l words is repeated, when generating the following word. WREP does the same, discarding words that are also repeated in the ground truth. We report the average of REP and WREP for $l \in \{16, 32, 128, 512\}$.

[7]We use the PyTorch re-implementation at `https://github.com/huggingface/transformers`.

	WikiText-2	WikiText-103	BookCorpus
α	1.2	1.2	1.3
k	50	50	20
P	0.95	0.95	0.90
τ	0.95	0.95	0.90

Table 3: Values of hyperparameters selected for Language Modeling.

	MEAN	MEDIAN	SD	MIN	MAX
Softmax	50,257	50,257	0	50,257	50,257
Softmax-τ	50,257	50,257	0	50,257	50,257
Greedy	1	1	0	1	1
Top-k	50	50	0	50	50
Nucleus	562	210	1,187	1	19,945
Entmax	2,532	1,210	2,643	1	28,364

Table 4: Mean, median, standard deviation, minimum, and maximum number of tokens considered by each decoding method on the Wikitext-103 test set.

sparse decoders, with greedy decoding achieving the best scores (but at the expense of many repetitions).[8]

To help understand why entmax leads to better sparsemax scores and fewer repetitions, Table 4 shows the mean, median, standard deviation, minimum, and maximum number of tokens each decoding strategy considers when predicting each word, on the Wikitext-103 test set. We see that entmax sampling and nucleus sampling consider a lot more tokens than greedy decoding and top-k sampling, which may be the reason for the smaller number of repetitions. A possible explanation for entmax sampling outperforming nucleus sampling is its higher standard deviation, suggesting that its

[8]Figure 4 of App. J shows results on automatic metrics for top-k, nucleus, and entmax sampling on WikiText-103 validation set for various K, P, and α.

4257

	sp	JS	ϵ-ppl	REP	WREP
Training with NLL					
Top-k	.683	**.364**	21.90	.373	.181
Nucleus	.686	.370	15.51	.357	.167
Entmax	.670	.378	20.69	.365	.183
Training with Entmax Loss					
Top-k	.677	.384	46.58	.364	.196
Nucleus	.668	.373	43.19	.350	.172
Entmax	**.694**	.373	**13.23**	**.346**	**.160**

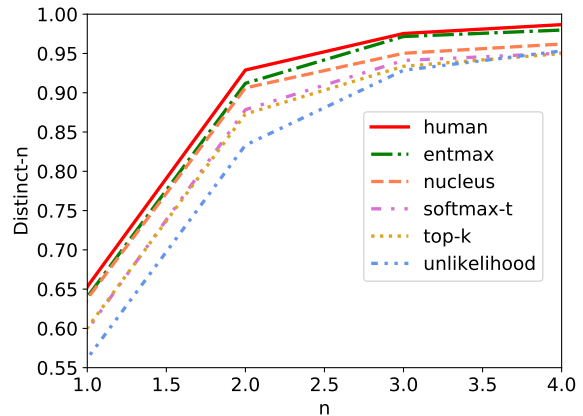Table 5: Language modeling ablation study on WikiText-103 test set.

Figure 1: Plot of the distinct-n metric for each sampling method on story completion, with $n = \{1, 2, 3, 4\}$. The distinct-n results for greedy decoding are not shown since they are very small (0.25 for distinct-4).

sparsity range is more adaptive to the context.

Ablation study. In order to understand whether the improved performance is caused by the mitigation of the sparsity mismatch between training and test times, we experimented (i) decoding with the entmax sampling method from a language model fine-tuned with negative log-likelihood, and (ii) decoding with top-k sampling and nucleus sampling from a model fine-tuned with the entmax loss. We conducted these experiments on the WikiText-103 dataset.

As shown in Table 5, our proposed approach, which decodes with entmax sampling from a model also fine-tuned with the entmax loss, is the one which leads to the best scores, as we see a considerable degradation when entmax is only used at training or at decoding time. This corroborates our hypothesis that the improved results come from eliminating the mismatch between training and decoding.

5.2 Story completion

Next, we analyze the model's ability to generate long sequences of text using different sampling methods.[9] We performed completion of stories from the WritingPrompts dataset (Fan et al., 2018), using the models fine-tuned on BookCorpus. WritingPrompts is a collection of human-written stories paired with writing prompts. We randomly selected 1,000 stories which were at least 200 words long and used the first 50 words as context for the models. Examples of stories generated with each method (Table 1 and Table 13 of App. K) suggest that entmax sampling leads to more engaging stories while preventing degenerate text. To measure

the stories' word diversity, we show in Figure 1 the distinct-n metric[10] (Li et al., 2016a) for the stories generated by each model. It can be seen that entmax sampling leads to more diverse unique n-grams for $n \in \{1, 2, 3, 4\}$, closer to human generated text. We also measured the number of unique words in the stories generated: entmax sampling generated 14,702 different words, while softmax with decreased temperature, greedy decoding, top-k, nucleus sampling, and unlikelihood generated 12,447, 1,750, 11,803, 12,008, and 5,509 words, respectively. As expected, entmax leads to higher word diversity on par with human stories, which contain 15,377 different words.

Human evaluation. We performed human evaluation of greedy decoding, unlikelihood training, top-k, nucleus, and entmax sampling on completion of stories from the WritingPrompts datasets. We randomly selected 100 stories to perform the human evaluation. For each story, 5 judges from Amazon Mechanical Turk evaluated the story completions in 3 metrics: fluency (whether the text is syntactically and semantically correct), coherence (whether the story continuation is related to the provided context and is consistent), and engagement (whether the annotator felt interested in the story). Ratings were given on a 5-point Likert scale, and the mean for each metric is reported in Table 6. Further details, including a screenshot of the annotator interface, are described in App. G. We observe that entmax sampling outperforms all other methods on

---

[9]Softmax sampling is not considered since it has been shown to generate degenerate text (Holtzman et al., 2020).

[10]Distinct-n corresponds to the number of distinct n-grams divided by the total number of generated words.

	FLUENCY	COHERENCE	ENGAGEMENT
Greedy	2.5	2.3	2.3
top-k	3.3	2.9	2.9
Nucleus	**3.5**	3.1	3.2
Unlikelihood	3.3	3.0	3.2
Entmax	**3.5**	**3.2**	**3.6**

Table 6: Human evaluation of story completion. All scores marked in bold at each column outperform the others with statistical significance, according to the Wilcoxon's test with p-value < 0.01. The inter-annotator agreement (Fleiss Kappa) is 0.45 for fluency, 0.41 for coherence, and 0.63 for engagement.

coherence and engagement, having similar scores as nucleus sampling on fluency.

5.3 Dialogue Generation

To evaluate the sampling methods in an interactive setting, we experiment with dialogue generation. Its goal is to generate an utterance, given a context consisting of the previous utterances in the dialogue and, in some cases, initial context sentences with related information that can be describing personas, knowledge, or scenarios.

Datasets and metrics. We performed experiments with the PersonaChat dataset (Zhang et al., 2018). It is a crowd-sourced dialogue dataset in which speakers were asked to condition their utterances on predefined personas. It contains 164,356 utterances over 10,981 dialogues. As there is no public test set, we report results on the validation set. We evaluate the word F_1-score, ϵ-perplexity, sparsemax score, and Jensen-Shannon divergence. As for the language modeling experiments, ϵ-perplexity, sparsemax score, and Jensen-Shannon are computed at the BPE level. We also report distinct-n metric for $n = \{1, 2\}$ and analyze how the models behave in dialogue simulations between two agents (Li et al., 2016c).

Fine-tuning GPT-2. In order to apply GPT-2 medium to the dialogue generation task, we follow Wolf et al. (2019) and Budzianowski and Vulić (2019): the input given to the language model consists of the sentences describing the persona the model should impersonate, and the history utterances. In order for the model to adapt to dialogue, the word and position embeddings are augmented with dialogue-state embeddings that indicate whether tokens are from a persona sentence, speaker 1, or speaker 2.

	sp	JS	ϵ-ppl	F_1	DIST-1	DIST-2
Softmax	0.636	0.412	17.21	14.21	0.4325	0.8422
Softmax-τ	0.621	**0.393**	17.18	16.31	0.4312	0.8289
Greedy	0.422	0.401	1031.79	**21.79**	0.4305	0.7958
Top-k	0.626	**0.393**	47.79	17.34	0.4378	0.8344
Nucleus	0.632	0.399	26.98	15.98	0.4334	0.8436
Entmax	**0.642**	**0.393**	**17.10**	15.02	**0.4532**	**0.8494**

Table 7: Automatic evaluation of dialogue generation on the Persona-Chat validation set.

These embeddings are learned during fine-tuning. The hyperparameters α, k, P, and τ were chosen with a grid search over the sets of values $\alpha \in \{1.2, 1.3, 1.5, 2\}$, $k \in \{5, 10, 20, 50, 100\}$, $P \in \{0.5, 0.8, 0.85, 0.9, 0.95\}$, and $\tau \in \{0.7, 0.8, 0.9, 0.95\}$, using the sparsemax score. The values chosen are 1.5, 10, 0.9, and 0.8, respectively. Additional settings are described in App. H.

Automatic metrics results. We report the results in Table 7. Entmax again outperforms all the other methods in sparsemax score and ϵ-perplexity. It also has the lowest JS (same as top-k and softmax-τ). Entmax also leads to fewer repetitions, having higher distinct-1 and distinct-2 scores. However, its F_1 score is lower (similar findings have been reported in Li et al. (2020)). This can be due to dialogue generation being an open-ended generation task that can have multiple correct answers.

Additionally, we simulated a conversation between two agents of the same model (Li et al., 2016c). We chose different personas randomly for the two agents. Then a first utterance from the PersonaChat dataset was given as context. Some conversation examples are presented in Tables 14-17 of App. L. We measured the average length of conversations, considering that the conversation is finished when utterances overlap 80% or more, when there is no response by an agent, or when it reaches 20 utterances (similar procedure as Li et al. (2016c)). We also measured the number of unique words, and the distinct-n metric for $n = \{1, 2\}$. As shown in Table 8, entmax sampling leads to longer conversations with higher word diversity and higher number of distinct 1-grams and 2-grams.

Human evaluation. Finally, we performed human evaluation following the ConvAI2 challenge: 12 volunteers had 30 conversations each with models using the different sampling methods. The vol-

	LENGTH	UNIQUE WORDS	DIST-1	DIST-2
Softmax	13.98	11,242	0.6084	0.8824
Softmax-τ	14.82	5,495	0.5384	0.6936
Greedy	7.83	4,229	0.4853	0.6732
Top-k	14.72	8,833	0.5623	0.8461
Nucleus	15.56	10,098	0.5836	0.8728
Entmax	**15.83**	**13,020**	**0.6546**	**0.9211**

Table 8: Evaluation of dialogue simulations between two agents using the different sampling methods.

	FLUENCY	CONSISTENCY	ENGAGEMENT
Greedy	**4.1**	3.0	2.5
Top-k	**4.0**	3.2	3.3
Nucleus	**4.1**	3.4	3.3
Entmax	**4.1**	**3.6**	**3.9**

Table 9: Human evaluation of dialogue generation. All scores marked in bold at each column outperform the non-bold ones with statistical significance, according to the Wilcoxon's test with p-value < 0.01.

unteers scored the conversations from 1 to 5 in terms of fluency, consistency (whether the model's utterances are coherent with their persona and the model does not contradict itself), and engagement. The model's personas were randomly selected from the PersonaChat validation set. Results are reported in Table 9. Entmax sampling outperforms the other methods in consistency and engagement, having similar scores in fluency. This means entmax sampling does not only generate the most interesting conversation utterances, but it also leads to an improvement of the conversation consistency.

6 Conclusions

We proposed entmax sampling as a new strategy for generating text from a sparse probability distribution. It provides three main advantages: (i) it offers a natural way of sampling directly from the output probability distribution; (ii) the distribution sparsity is modeled during training, avoiding a sparsity mismatch between training and run time; (iii) when sampling with entmax, the number of words to be considered varies with the context, as in nucleus sampling and in contrast to top-k sampling. Additionally, we proposed new metrics for evaluating language models that produce sparse and truncated probability distributions: ϵ-perplexity, sparsemax score, and Jensen-Shannon divergence.

Experiments show that entmax sampling leads to higher n-gram diversity, fewer repetitions, and sim-

ilar or improved results in automatic metrics. Human evaluation confirms that entmax outperforms greedy decoding, top-k, and nucleus sampling in coherence/consistency and engagement, and is similar or better in terms of fluency.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proc. ICLR*.

Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. 2005. Clustering with Bregman divergences. *Journal of machine learning research*.

Mathieu Blondel, Andre Martins, and Vlad Niculae. 2019. Learning Classifiers with Fenchel-Young Losses: Generalized Entropies, Margins, and Algorithms. In *Proc. AISTATS*.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, It's GPT-2-How Can I Help You? Towards the Use of Pretrained Language Models for Task-Oriented Dialogue Systems. In *Proc. Workshop on Neural Generation and Translation*.

Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Proc. NIPS*.

Ondřej Cífka, Aliaksei Severyn, Enrique Alfonseca, and Katja Filippova. 2018. Eval all, trust a few, do wrong to none: Comparing sentence generation models. *arXiv preprint arXiv:1804.07972*.

Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *Proc. ACL*.

Michel Marie Deza and Elena Deza. 2009. Encyclopedia of distances. In *Encyclopedia of distances*, pages 1–583. Springer.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. In *Proc. ACL*.

Jessica Ficler and Yoav Goldberg. 2017. Controlling Linguistic Style Aspects in Neural Language Generation. In *Proc. of the Workshop on Stylistic Variation*.

Ivo Grosse, Pedro Bernaola-Galván, Pedro Carpena, Ramón Román-Roldán, Jose Oliver, and H Eugene Stanley. 2002. Analysis of symbolic sequences using the Jensen-Shannon divergence. *Physical Review E*.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to Write with Cooperative Discriminators. In *Proc. ACL*.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *Proc. ICLR*.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity-a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. ICLR*.

Rebecca Knowles, Adithya Renduchintala, Philipp Koehn, and Jason Eisner. 2016. Analyzing Learner Understanding of Novel L2 Vocabulary. In *Proc. SIGNLL*.

Ilya Kulikov, Alexander H Miller, Kyunghyun Cho, and Jason Weston. 2018. Importance of a search strategy in neural dialogue modelling. *arXiv preprint arXiv:1811.00907*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proc. NAACL*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016b. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016c. Deep Reinforcement Learning for Dialogue Generation. In *Proc. EMNLP*.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial Learning for Neural Dialogue Generation. In *Proc. EMNLP*.

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 *Proc. ACL*.

Margaret Li, Jason Weston, and Stephen Roller. 2019. Acute-eval: Improved dialogue evaluation with optimized questions and multi-turn comparisons. *arXiv preprint arXiv:1909.03087*.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proc. ICML*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer Sentinel Mixture Models. In *Proc. ICLR*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*.

E Patrick and F Fischer. 1969. Nonparametric feature selection. *IEEE Transactions on Information Theory*.

Ben Peters, Vlad Niculae, and André FT Martins. 2019. Sparse Sequence-to-Sequence Models. In *Proc. ACL*.

Ofir Press, Amir Bar, Ben Bogin, Jonathan Berant, and Lior Wolf. 2017. Language generation with recurrent generative adversarial networks without pretraining. *arXiv preprint arXiv:1706.01399*.

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

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proc. EMNLP*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proc.ACL*.

Felix Stahlberg and Bill Byrne. 2019. On NMT Search Errors and Model Errors: Cat Got Your Tongue? In *Proc. EMNLP*.

I Sutskever, O Vinyals, and QV Le. 2014. Sequence to sequence learning with neural networks. *Proc. NIPS*.

Constantino Tsallis. 1988. Possible generalization of boltzmann-gibbs statistics. *Journal of statistical physics*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. NIPS*.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasaath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *Proc. AAAI*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *Proc. ICLR*.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*.

Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. 2018. Diversity-promoting gan: A cross-entropy based generative adversarial network for diversified text generation. In *Proc. EMNLP*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proc. AAAI*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing Dialogue Agents: I have a dog, do you have pets too? In *Proc. ACL)*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *Proc. SIGIR*.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *Proc. ICCV*.

Supplementary Material

A Selection of the optimal ϵ for the ϵ-perplexity

We show here that the optimal ϵ for the computation of the ϵ-perplexity for each decoding method can be easily obtained by solving a convex optimization problem.

For a given ϵ, which can be reparametrized as $\lambda = \frac{\epsilon|\mathcal{V}|}{1+\epsilon|\mathcal{V}|} = \frac{1}{1+(\epsilon|\mathcal{V}|)^{-1}} \in [0,1]$, the average negative log-likelihood on a validation set is:

$$
\begin{aligned}
F(\lambda) &= -\frac{1}{T}\sum_{t=1}^{T}\log\left((1-\lambda)p_\theta(x_t) + \frac{\lambda}{|\mathcal{V}|}\right)\\
&= -\frac{1}{T}\sum_{t=1}^{T}\log(a_t\lambda + b_t), \qquad (13)
\end{aligned}
$$

where $a_t = |\mathcal{V}|^{-1} - p_\theta(x_t)$ and $b_t = p_\theta(x_t)$. The function F is the composition of a convex function with an affine function, hence it is convex. Therefore it has a global minimum. Its derivative is:

$$
F'(\lambda) = -\frac{1}{T}\sum_{t=1}^{T}\frac{a_t}{a_t\lambda + b_t}. \qquad (14)
$$

Since we constrain $\lambda \in [0,1]$, we can obtain the optimal λ by initializing with $\lambda = 0.5$ and iterating the following projected gradient rule:

$$
\lambda \leftarrow \max\{0, \min\{1, \lambda - \eta F'(\lambda)\}\}. \qquad (15)
$$

where η is a stepsize. Since $\lambda = \frac{1}{1+(\epsilon|\mathcal{V}|)^{-1}}$, we can invert this equation to obtain the optimal ϵ as $\epsilon = \frac{1}{|\mathcal{V}|(\lambda^{-1}-1)} = \frac{\lambda}{|\mathcal{V}|(1-\lambda)}$.

B Proof of boundedness of the sparsemax score

We show here that the sparsemax score in Eq. 10 is always bounded between 0 and 1.

The fact that $\mathsf{sp} \leq 1$ simply follows from the fact (Blondel et al., 2019, Prop. 2) that any Fenchel-Young loss (which includes $\ell_2(\boldsymbol{z}, x)$) is non-negative. Since $\mathsf{sp} = 1 - \min\{\ell_2(\boldsymbol{z}, x) \mid \text{sparsemax}(\boldsymbol{z}) = \boldsymbol{p}_\theta\}$, it follows that $\mathsf{sp} \leq 1$. Let us see when the maximal value 1 is attained. We

have:

$$
\begin{aligned}
\mathsf{sp} &= p_\theta(x) + \mathsf{H}_2(\boldsymbol{p}_\theta)\\
&= p_\theta(x) + \frac{1}{2}(1 - \|\boldsymbol{p}_\theta\|^2)\\
&= -\frac{1}{2}p_\theta(x)^2 + p_\theta(x) - \frac{1}{2}\sum_{x'\neq x}p_\theta(x')^2 + \frac{1}{2}\\
&= -\frac{1}{2}(p_\theta(x) - 1)^2 - \frac{1}{2}\sum_{x'\neq x}p_\theta(x')^2 + 1.
\end{aligned}
$$
$$(16)$$

Since the Gini entropy is maximized by the uniform distribution, the maximum distribution in Eq. 16 is of the form $\boldsymbol{p}_\theta = \left(1-t, \frac{t}{|\mathcal{V}|-1}, \ldots, \frac{t}{|\mathcal{V}|-1}\right)$ for $t \in [0,1]$. Replacing in Eq. 16, we obtain

$$
\begin{aligned}
\mathsf{sp} &= -\frac{1}{2}t^2 - \frac{1}{2}\frac{t^2}{|\mathcal{V}|-1} + 1\\
&= 1 - \frac{t^2}{2}\left(1 + \frac{1}{|\mathcal{V}|-1}\right). \qquad (17)
\end{aligned}
$$

This is maximized by $t = 0$, which corresponds to $\boldsymbol{p}_\theta = \boldsymbol{e}_x$.

To see that we always have $\mathsf{sp} \geq 0$, we use the fact that the Gini entropy $\mathsf{H}_2(\boldsymbol{p}_\theta)$ is always non-negative (zero if and only if $\boldsymbol{p}_\theta$ is a one-hot distribution), which is clear from the definition in footnote 3, and that $p(x) \geq 0$; therefore, the sum of these two terms is also non-negative, and zero if and only if $\boldsymbol{p}_\theta = \boldsymbol{e}'_x$ with $x' \neq x$.

C Relation between Patrick-Fischer distance and sparsemax score

We show here that the sparsemax score is equivalent to one minus the one half of the squared Patrick-Fisher distance between the distribution probability over the words $\boldsymbol{p}_\theta$ and the indicator one-hot vector $\boldsymbol{e}_x$ which corresponds to the ground truth word x.

The Patrick-Fischer distance between two distri-

butions is $D_{\text{PF}}(\boldsymbol{p}, \boldsymbol{q}) = \|\boldsymbol{p} - \boldsymbol{q}\|_2$. We have:

$$
1 - \frac{1}{2} D_{\text{PF}}^2(\boldsymbol{p}_\theta, \boldsymbol{e}_x)
$$

$$
= \ 1 - \frac{1}{2} \sum_{x'} (p_\theta(x') - e_x(x'))^2
$$

$$
= \ 1 - \frac{1}{2} \sum_{x' \neq x} p_\theta(x')^2 - \frac{1}{2}(1 - p_\theta(x))^2
$$

$$
= \ 1 - \frac{1}{2} \sum_{x'} p_\theta(x')^2 + \frac{1}{2} p_\theta(x)^2 - \frac{1}{2}
$$

$$
\quad - \frac{1}{2} p_\theta(x)^2 + p_\theta(x)
$$

$$
= \ \frac{1}{2} - \frac{1}{2} \sum_{x'} p_\theta(x')^2 + p_\theta(x)
$$

$$
= \ p_\theta(x) + \mathsf{H}_2(\boldsymbol{p}_\theta), \tag{18}
$$

which equals the sparsemax score defined in Eq. 10.

D JS divergence as a language model metric

The Jensen-Shannon divergence between the model probability distribution over the words $\boldsymbol{p}_\theta$ and the indicator one-hot vector $\boldsymbol{e}_x$ which corresponds to the ground truth word x can be defined as:

$$
\mathsf{JS}(\boldsymbol{p}_\theta, \boldsymbol{e}_x)
$$

$$
= \ \frac{1}{2} \mathsf{KL}\left(\boldsymbol{p}_\theta \| \frac{\boldsymbol{p}_\theta + \boldsymbol{e}_x}{2}\right)
$$

$$
= \ \mathsf{H}\left(\frac{\boldsymbol{p}_\theta + \boldsymbol{e}_x}{2}\right) - \frac{1}{2}\mathsf{H}(\boldsymbol{p}_\theta) - \underbrace{\frac{1}{2}\mathsf{H}(\boldsymbol{e}_x)}_{=0}
$$

$$
= \ -\sum_{x' \neq x} \frac{p_\theta(x')}{2} \log \frac{p_\theta(x')}{2}
$$

$$
\quad - \frac{1 + p_\theta(x)}{2} \log \frac{1 + p_\theta(x)}{2}
$$

$$
\quad + \frac{1}{2} \sum_{x'} p_\theta(x') \log p_\theta(x')
$$

$$
= \ -\frac{1}{2} \log \frac{1}{2} + \frac{p_\theta(x)}{2} \log \frac{p_\theta(x)}{2}
$$

$$
\quad - \frac{1 + p_\theta(x)}{2} \log \frac{1 + p_\theta(x)}{2}
$$

$$
= \ \mathsf{H}_b\left(\frac{1 + p_\theta(x)}{2}\right) - \frac{1}{2}\mathsf{H}_b(p_\theta(x)), \tag{19}
$$

where $\mathsf{H}_b(p) = -p \log p - (1 - p) \log(1 - p)$ denotes the entropy of a Bernoulli variable. Thus the JS divergence depends on the model distribution only through the probability given by the model to the groundtruth word, $p_\theta(x)$.
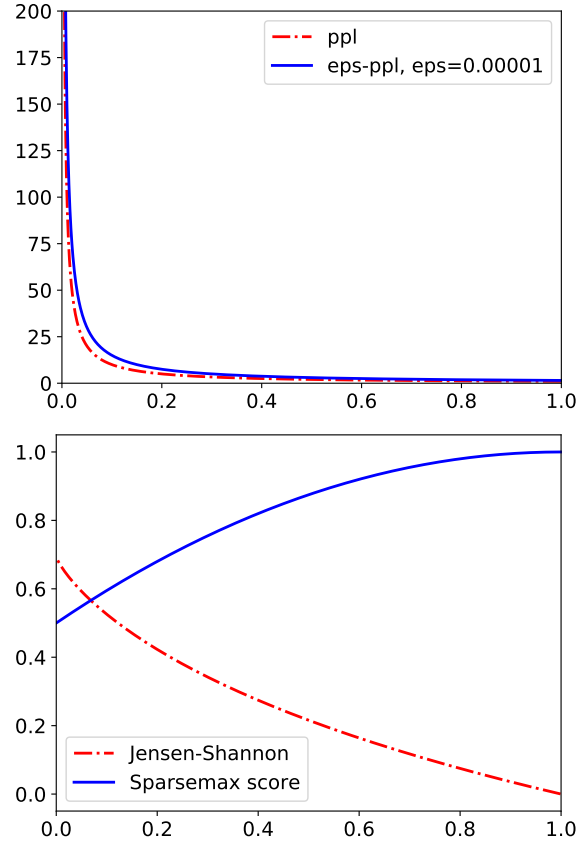
Figure 2: Comparative plots of ϵ-perplexity for $\epsilon = 0.01$ and $\epsilon = 0$ (top), and of sparsemax score and JS divergence (bottom). In both cases, the x-axis is $p_\theta(x)$.

E Comparative plots of evaluation metrics

Figure 2 shows comparative plots of the ϵ-perplexity, sparsemax score, and Jensen-Shannon divergence, for a distribution of the form $\boldsymbol{p}_\theta = \left(1 - t, \frac{t}{|\mathcal{V}|-1}, \ldots, \frac{t}{|\mathcal{V}|-1}\right)$, varying t, with a vocabulary of 50000 words.

F Fine-tuning details for language modeling

The models were fine-tuned for up to 5 epochs for Wikitext-2 and up to 1 for Wikitext-103 and BookCorpus using the Adam optimizer (Kingma and Ba, 2015), with a learning rate of 6.25×10^{-5}, which was linearly decayed to zero over the course of training. We report results of the models that have the highest sparsemax score on the validation set. The models fine-tuned with cross entropy and entmax losses were trained on a GPU Nvidia Titan XP, which has $\approx$ 12 Gb of memory. The model fine-tuned with the unlikelihood training term was trained on a GPU Nvidia Titan RTX, which has

≈ 24 Gb of memory.

G Stories' human evaluation details

To perform the human evaluation of the stories generated by the different models, we use Amazon Mechanical Turk (a screenshot of the interface is shown in Figure 3), and compensate Turkers at a rate of $0.7 per HIT. Pay rate is calculated based on an estimate of the completion time (5.5 minutes) and an hourly wage of $7.5.

To remove poor quality annotations, we perform several controls. We did not consider annotations that were performed in less than 3.5 minutes. Additionally, following (Li et al., 2019), to filter low quality annotators we showed them annotated examples with contexts from famous novels, the real continuation, story continuations that are not related to the context, and story continuations that are not fluent. If the Turker's annotations differed significantly from the reference rank-wise, all annotations performed by the Turker were excluded.

H Fine-tuning details for dialogue generation

We fine-tune the GPT-2 medium model (Radford et al., 2019) for a maximum of 3 epochs with a learning rate of 6.25×10^{-5} that linearly decays to zero over the course of the training. The models were fine-tuned on a GPU Nvidia Titan XP, which has ≈ 12 Gb of memory.

I Comparison of models with the Jensen-Shannon divergence

We compared the distributions given by the different decoding methods when generating the sentences of the BookCorpus validation set with the Jensen-Shannon divergence, as described in §4. In Tables 11 and 12 we show some of the sentences with higher Jensen-Shannon divergence, as well as the probability given by each model to the ground truth word.

J Results of automatic metrics for various values of K, P, α.

In Figure 4 we report the results of ϵ-ppl, JS, sp, rep, and wrep metrics on the validation set of WikiText-103 for the models with top-k, nucleus, and entmax sampling with various values of K, P, α.

K Story completion examples

Examples of story completion with context from the WritingPrompts dataset using the different models fine-tuned on BookCorpus are presented in Table 13.

L Dialogue simulation examples

In order to evaluate the different decoding methods in dialogue generation, we simulated a conversation between two agents using the same decoding method. For that, we assigned each agent a randomly selected persona from the PersonaChat dataset and gave a first utterance of the dataset as context. Examples of dialogue simulations are presented in Tables 14-15 and Tables 16-17.

	WikiText-2					WikiText-103					BookCorpus				
	sp	JS	ϵ-ppl	REP	WREP	sp	JS	ϵ-ppl	REP	WREP	sp	JS	ϵ-ppl	REP	WREP
Softmax	.682	.381	**13.56**	**.389**	**.173**	.682	.376	13.26	.342	.162	.691	.360	**9.56**	.377	.174
Softmax-τ	.681	.374	13.62	.403	.178	.681	.358	13.35	.353	.168	.689	.348	9.75	.391	.248
Greedy	.484	**.358**	533.03	.512	.232	.486	**.357**	523.68	.445	.211	.508	**.341**	946.03	.456	.198
Top-k	.680	.368	22.23	.426	.198	.679	.360	22.28	.368	.182	.688	.347	19.55	.398	.193
Nucleus	.681	.375	15.38	.400	.176	.681	.363	15.65	.352	.167	.690	.348	14.58	.392	.183
Unlikelihood	.468	.369	635.02	.441	.205	.471	.367	613.61	.411	.196	.492	.352	486.65	.446	.196
Entmax	**.684**	.376	14.69	.397	**.173**	**.686**	.362	**13.25**	**.341**	**.160**	**.699**	.351	9.57	**.375**	**.170**

Table 10: Language model evaluation on WikiText-2, WikiText-103, and BookCorpus validation sets. For all metrics except sp, lower is better.

	"	besides	,	i	enjoyed	having	her	with	me
Softmax	0.011	0.0002	0.808	0.1479	0.0002	0.0141	0.0228	0.0179	0.9114
Softmax-t	0.0131	0.0001	0.8855	0.1864	0.0001	0.0137	0.0211	0.0179	0.9467
Greedy	0	0	1	1	0	0	0	0	1
top-k	0.0439	0	0.8814	0.2543	0	0	0.0311	0	0.9267
Nucleus	0.037	0	1	0.3042	0	0	0	0	1
Entmax	0.0159	0	0.9943	0.3311	0	0.044	0.0073	0.0185	1

Table 11: Probabilities given by each model to "besides, i enjoyed having her with me"

	i	miss	my	parents	and	my	friends
Softmax	0.0299	0.0006	0.0294	0.0104	0.1031	0.0611	0.0225
Softmax-t	0.0399	0.0005	0.0279	0.0121	0.1067	0.0763	0.0240
Greedy	0	0	0	0	0	1	0
top-k	0.1193	0	0.0391	0	0.01303	0.1410	0
Nucleus	0.1005	0	0	0.0289	0.1035	0.1012	0
Entmax	0.1047	0	0.0320	0.0127	0.1453	0.1509	0.0469

Table 12: Probabilities given by each model to "i miss my parents and my friends".

Read the beginning of the story and the completions below.

Rate the story completions in Fluency (whether the text is syntactically and semantically correct), Coherence (whether the story continuation is related to the context and whether the continuation is coherent), and Engagement (whether you felt interested in the story) from 1(very bad) to 5(very good).

Note: It is normal that the stories finish abruptly. Don't take that into account for the rating.

Beginning of story: The Man finished up his song , sat silently at the piano for a few seconds , then reached into his jacket for his flask . He brought it to his lips , attempted to take a swig , then upended it to watch the last few drops fall to

Completion: the floor. he took a deep breath, then took another, then another, then another. he was still holding the flask, and he was still holding the flask, and he was still holding the flask, and he was still holding the flask, and he was still holding the flask, and he was still holding the flask, and he was still holding the flask, and he was still holding the flask, and he was still holding the flask, and he was still holding the flask, and he was still holding the flask, and he was still holding the flask, and he was still holding the flask, and he was still holding the flask, and he was still holding the flask, and he

- 1) Fluency

  ○────────────

- 2) Coherence

  ○────────────

- 3) Engagement

  ○────────────

Completion: the floor. . . . he noticed julian sitting next to him, observing him for the first time in five days. what do you think? youre sounding different since weve been here. i dont know, its a long story. . . julians face lit up. better you than me ; its so many years since i last saw you. were talking of here and now, arent we? funny how that keeps happening, with us living in these cursed times of war all round. if were allowed to think of life as normal, id appreciate it much more. . . hmmm, more like normal than never. yeah

- 1) Fluency

  ○────────────

- 2) Coherence

  ○────────────

- 3) Engagement

  ○────────────

Completion: specks that fell to specks into the air, where others began floating around them ; some landed somewhere above the keyboard, while a majority fell toward his piano teacher at this late afternoon ritual in honor that night would not last. 1-3 : (5 :1- 3 : 4- 6) : 1- surrender to love i want us alone the first part, as it is in any dance its not what its a chore ; a dancing part just what we 're the song starts ; in the dance, i've felt a sudden chill on a wind, i thought that there should 'elimit.el unimito! " `` a moment before a dance " 2- pause : this is no longer part 1 a dancing

- 1) Fluency

  ○────────────

- 2) Coherence

  ○────────────

- 3) Engagement

  ○────────────

Completion: the floor. a small hand caught the back of his shirt and pulled him close, pressing him flat against the piano seat. then he slid his arms around his neck, kissed him on the lips, and then gave him a one-armed hug. a coppery scent of liquor permeated the air around them, strong, sweet, and hot. neil was by his side in an instant, the warmth from the mixture of alcohol and his body bags just gone, evaporating in his absence. the witchs spells dissolved as a deep sense of calm took over. the room was silent again, until devlin started dancing a version of a dance routine from the dusk almost to the delight of neil.

- 1) Fluency

  ○────────────

- 2) Coherence

  ○────────────

- 3) Engagement

  ○────────────

Completion: the floor. he looked down at the empty glass, then up at the ceiling, and saw that the light had faded. the room had been lit only by a single candle, and it was the only thing in the world that was going to stay that way for the rest of the night. " a few days later... `` i 'd been to the hospital for a lot of things in the next three years, but that was the last time i saw my mommy. " a few months later, i 'd been in a car with my boyfriend, and he was going to be a good man for the rest of his life. and he had been a very good husband to me. epilogue -

- 1) Fluency

  ○────────────

- 2) Coherence

  ○────────────

- 3) Engagement

  ○────────────

[Submit]

Figure 3: Screenshot of human evaluation interface on Amazon Mechanical Turk
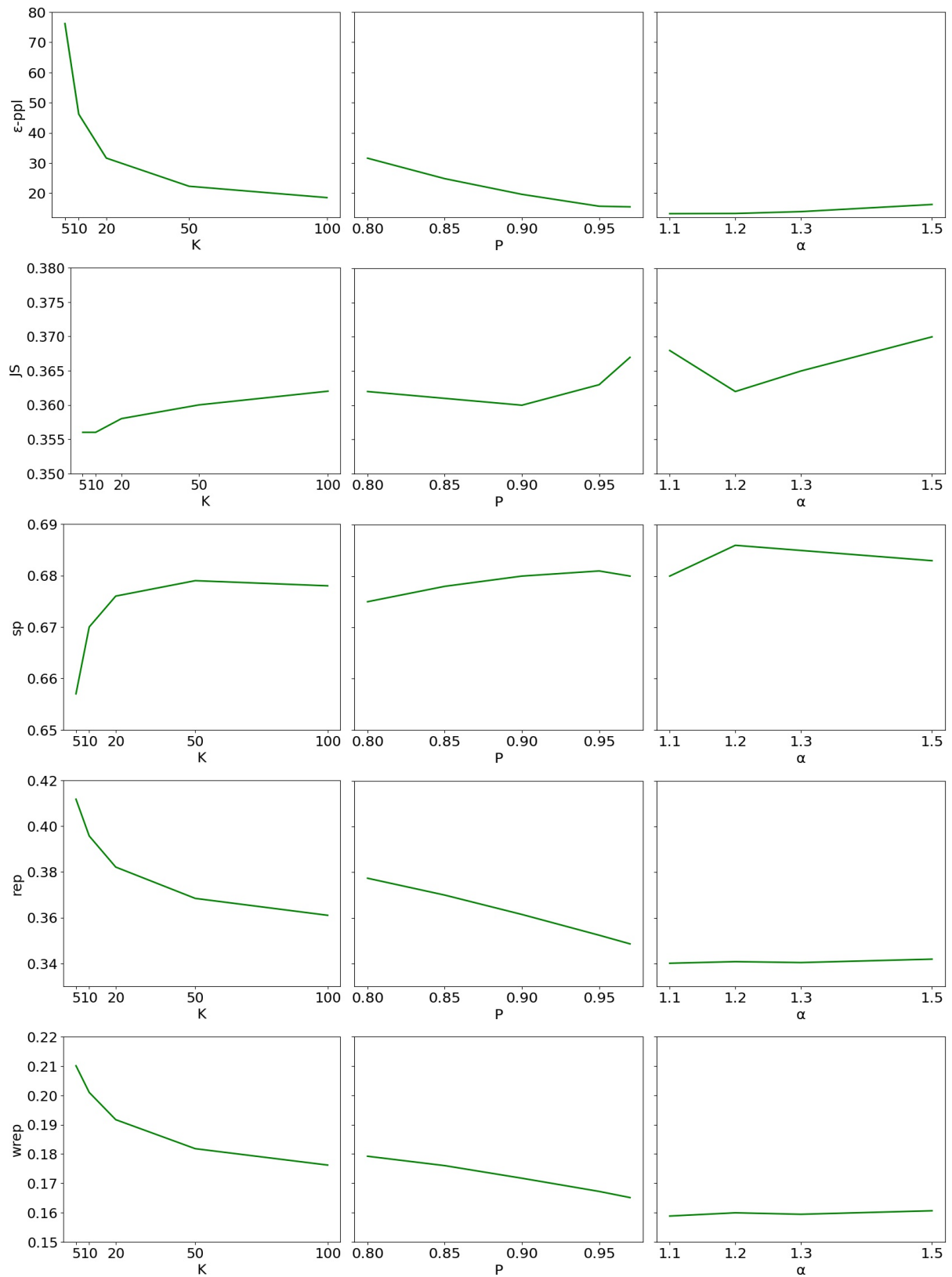
Figure 4: Results of automatic metrics on the validation set of WikiText-103 for various values of K for top-k sampling, P for nucleus sampling, and α for entmax. For all metrics except sp, lower is better.

Context:	*"Where am I?" I asked. I stared at the screens, brushing past my field of vision. They went too fast to watch, but it was clear. I stood on a floating white platform, below me there were flames ...*
Greedy:	... . i could see the flames, but i couldn't see the man. i couldn't see the man who had been my father. i couldn't see the man who had been my father. i couldn't see the man who had been my father. i couldn't see the man who had been my father. i couldn't see the man who had been my father. i couldn't see the man who had been my father. i couldn't see the man who had been my father.
Top-k:	... and a large red door. they went faster and farther off, so close now you almost saw nothing through them as everything around you was just flames; you could taste a burnt smell and you were flying above yourself at speed you never seemed close compared when one thing was the sky or nature or a building full of air around it the sky being what the nature had to look like it had an energy about to burn.
Nucleus:	... and a third being. the creature had clawed its way through the sky, like a giant serpent, its head lifts and back-end leans, it had wings in front and they were on fire, blotched blue and green, emulating the lights in the air around it, like a creature through a screen. i recalled every detail of that moment of events, that once of its kind it had tried to eat me. now i could see the terror in that moment.
Unlikelihood	... and a black smoke. the smoke was rising from my hand, and i could see it was coming from my mouth, and it was the same thing that had killed the woman in the park. "i don't know how long i can live without you" i said, and we both knew that the next time we saw each other again, i was going to be a very different person. is the beginning of a new life a man who is the beginning the world
Entmax:	... and a demon. the room grew dimming with tiny figures visible as small shadows. we waited. i listened to the tingling below me. a faint noise. the darkness seemed to get thinner. i looked up at her. nothing met my eyes. the girl was familiar, but i couldn't get it. she shuddered. "no!" i yelled. her eyes snapped up to me. she screamed, "no, you are not ready!" she sprinted towards the door.
Context:	*The universe was collapsing in on itself , swallowing all that man had once nurtured and loved and explored . As the walls came crashing down and the void beyond it all came flooding in , a man at the center of it all sat serenely before a rift in ...*
Greedy:	... the universe, a man who had been a man for so long that he had forgotten what it was like to be a man. he was a man who had been a man for so long that he had forgotten what it was like to be a man. he was a man who had been a man for so long that he had forgotten what it was like to be a man. he was a man who had been a man for so long that he had forgotten what it was like to be a man.
Top-k:	... a river where time could be no closer, an age ago there stood the very center for an entire generation : that great hour upon timoria before its collapse. it may well also have been that at some early meeting among them the young men at least could give any possible threat a warning, for, one can not blame those boys on this side. they know too well where a great danger rests, or is waiting, somewhere
Nucleus:	... the fabric of light that was the void itself. consuming him whole it will fall. it will swallow us whole. kill us. die. and when it does then it will start over. the vast blackness continued to swallow sebastian, forcing him forward. he twisted, protesting the pain that gathered in his throat as his back was crushed against the solid blackness. he had been doing that all the way down.
Unlikelihood	... the fabric of time. it was the first time he had seen a real time machine, a machine with a purpose." "what was it?" the man who was the future is the beginning of a part whatever the end whatever all yours and whatever in the beginning the deadly end a deadly (and deadly deadly) all yours and the first time he had seen the future
Entmax:	... the fabric of space and time and whatever truth he had contained since childhood was now ripening to be broken further. the crushing sadness that seized him made him feel like he had just found a complete lack of reality. it was all like a nightmare. it was all a goddamned nightmare. he shook himself and felt his emotions grow equally furious with himself and the destruction.

Table 13: Examples of story completion using the different methods.

Context:	*hello how are you today?*
Softmax:	- i'm great i love to dance to oldies music was is fun
	- very fluent in all the vernacular spoken in pa
	- that's awesome i teach elementary school
	- i love teaching, tap dancing to bob segar, all illustration type stuff
	- that sounds like a lot of fun
	- it is, but this harry potter series just made me feel tingly
	- that's really hurtful to do
	- it can be, but i am working on other things
	- that you are fine i like james patterson
	- very nice of you to comment
	- yes my dog is my son that he chews candy to be funny for my dogs birthday
	- i have no kids i moved to virginia
	- that is crazy did you like the hurricane ?
	- yeah it was so devastating
	- i see
	- its so sad i'm sorry you are stuck in pennywise
	- wow that's very sad
	- it is he?
	- seriously
	- it will be my big dream to leave and go to florida
Softmax-τ:	- i am doing good, how about yourself?
	- i am good, just getting in from work
	- oh, what do you do?
	- i sell cars, what do you do?
	- i work in a hospital as a nurse.
	- oh that is very nice of you
	- what do you sell cars ?
	- i sell cars as a hobby
	- oh, i see. do you have any hobbies?
	- i like to paint and hang out in my living room.
	- oh, i enjoy listening to music. i love rolling stones !
	- i like rolling stones too do you?
	- yeah, i really like the rolling stones, they're my favorite!
	- cool, meat is amazing. i also like to like painting
	- i think it is for the best. i like to know that i love my job.
	- yes beater and i love meat. i also love to eat meat.
	- i'm sure you are a vegetarian! what do you do for a living?
	- i sell cars, i'm an electrician
	- oh i see. are you a car salesman?
	- i just started selling cars
Greedy:	- i'm good, how are you?
	- i'm good thanks for asking
	- what do you do for a living?
	- i'm studying to be a dentist
	- oh, i'm a teacher.
	- what city are you from?
	- i live in the city.
	- what do you do for a living?
	- i teach math and english.
	- what city are you from?
	- i live in the city.
	- what city are you from?

Table 14: Example of dialogue simulation between two agents using the different decoding methods.

Context:	*hello how are you today?*
Top-k:	- hello! i am well ,yourself ?
	- i am great. i just got off of work
	- i just got a job at a grocery store.
	- oh i am sorry to hear that
	- i'm okay. do you like reading ?
	- yes i love reading and i also love to swim
	- oh that's awesome! i really like reading cupcakes.
	- that sounds delicious. i'm in the south so i like to eat cupcakes
	- i love to make my own cupcakes. i love it!
	- do you have any hobbies?
	- i love reading, but my favorite hobby is painting.
	- that is cool. i've been to canada twice , but i've never been there
	- i love to paint too! what color eyes do you have?
	- mine are blue i'm not sure
	- i have brown hair.
	- i like to make my own cupcakes
	- cool. do you work? i work at walmart and i work at the grocery market.
	- i am actually a reporter for the local news
	- cool! do you like to read?
	- i've a newspaper reporter in the news
Nucleus:	i'm great, just got home from a long walk with my two dogs.
	- do you play any instruments?
	- not really. what kind of music do you like?
	- i play piano sometimes and listen to metal music
	- oh cool. i'm just trying to write an ebook about animals.
	- i like cats and dogs. do you play any instruments?
	- i do not, i prefer listening to music i like.
	- what kind of music do you like?
	- classical music and video game soundtrack.
	- do you go to concerts?
	- i go with my husband and listen to a lot of different music genres.
	- do you have any kids?
	- nope, how about you?
	- i've a boyfriend. are you married?
	- yes, recently divorced. we are planing on getting a corvette.
	- how exciting! are you close to your husband?
	- i am. i am looking to buy one when i retire.
	- that is awesome! are you married?
	- nope. we were looking to move to ireland when i got my divorce.
	- that's so sad. you should i ask if he's ok?
Entmax:	- good, how are you doing?
	- great just watching a movie
	- oh really what movie?
	- it's a story about robots getting into human stuff.
	- wow! well that strikes me as not science fiction at all
	- why do you say that then?
	- because humans are fragile and vulnerable lol
	- yeah they would be
	- very true. what kind of movies u like?
	- i love sci fi and ghost in the shell
	- i actually have never watched that. maybe ghost in the shell
	- it is such a beautiful movie
	- yeah i love stories about lost ones
	- that 's cool
	- my favourite movie though is gamera how about you
	- i love it too, did you see it?
	- yes that was one of my favorite films
	- that's cool. do you work?
	- i am too lazy to try difficult things now

Table 15: Example of dialogue simulation between two agents using the different decoding methods.

Context:	*greetings from the big apple, kim here, how are you?*
Softmax:	- hi! great thanks, how are you ? do you have any hobbies?
	- i ride my black mbu on my nightly walks, and listen to rock music.
	- whats your favorite kind of music?
	- guns and roses is my bedrock, my favorite artist.
	- that's great, my favorite genre is joleg tonne or tontera band.
Softmax-τ:	- i'm doing great, how are you?
	- i'm good, just playing with my son today.
	- sounds fun, what are you playing with?
	- I'm playing soccer right now, he's 4.
	- do you have any pets?
	- no, just the baby. are you married?
	- no, no i don't have any kids
	- i'm single, but i have been in this profession for a while.
	- interesting, tell me more about yourself.
	- i'm a professional soccer player. what about you ?
	- that's cool, i'm just a training dummy
	- i also play soccer professionally, i love it.
	- i like to play sports too. definitely a great sport
	- i have been to stanford for training, trying to perfect my skills.
	- i bet that's great training
	- its a lot of work!
	- yeah but i like it
	- yeah. i'm actually a soccer mom, we try to keep on top of our kids.
	- do you have kids?
	- i do.
Greedy:	- i'm doing well. how are you?
	- i'm good, just finished cooking dinner.
	- what do you like to cook?
	- i love pasta, and i love to cook.
	- i like to make pasta too. i'm a huge fan of pasta.
	- i love to cook, i'm a student, so i'm always busy.
	- what are you studying?
	- i'm studying to be a chef.
	- what city are you from?
	- i'm from cali, what about you?
	- i'm from cali. i'm from the midwest.
	- i'm from the midwest too , but i'm from the midwest.

Table 16: Example of dialogue simulation between two agents using the different decoding methods.

Context:	*greetings from the big apple, kim here, how are you?*
Top-*k*:	- i'm very good, thank you, i hope you have a great day
	- thanks. have a good day too.
	- same to you, have a good day too!
	- you do the same. have a great day too.
	- the same to you and have a great day also!
	- i do have a very nice day, i love running.
	- that's nice to enjoy, it s a great day here today
	- i am a political junkie.
	- you must be very busy!
	- what do you do for a living?
	- i'm actually a business major.
	- that must be hard to do, do you like politics?
	- not a big fan of politics, i like to go skiing and listen to rock
	- do you play sports or something?
	- i am in school for business
	- oh that is awesome. i used to play football in college, but i hated it!
	- oh that's why i play football!!
	- wow. what is your name?
	- my name is jed. i am from russia. how about you?
	- i played in ny and played at boston.
Nucleus:	- hi, i'm fantastic, how are you?
	- great. making some barbies for our barbie girls.
	- awesome, do you have any hobbies?
	- i really enjoy barbies. i live a long time away.
	- i am looking forward to going to school in the fall.
	- school is cool. do you sing? i love barbie songs.
	- i've a pet pean i sing while eating tacos.
	- barbie! that's cute.
	- oh yea, i have a small heart lol
	- are you looking forward to winter? i would like to live in the big apple.
	- definitely
	- winter is fun! no thanks, me neither.
	- you must really like summer, too lol
	- thanks. it is too cold to sing in the winter. its just not for me.
	- do you like tacos?
	- oh my gosh. they're good. my favorite is barbie. lol
	- i know, but spaghetti is my fave lol
	- haha. lol
	- oh , i do enjoy spaghetti , for my college graduation i was in last year of high school
	- how are you ?
Entmax:	- good. i just rang someone on the other side.
	- good, what did you get excited about today?
	- i love paris. how was your day?
	- its been crazy, i'm always feeling excited! lol
	- what is your favorite thing to do?
	- oh i love going hiking in the mountains, myself.
	- that sound amazing. i like travelling.
	- love trips, but i cannot stand staying in one place all day. lol
	- do you have any hobbies ? i always want to find cool new things.
	- i really like going out and nature itself, i prefer hiking
	- yes, exploring parks and all that jazz when i can.
	- that is awesome fun, whats your fav color?
	- i love grey. roses and the mountains signify my youth.
	- mine is blue, it makes me think of blueberries though
	- grey denotes youth well or openness and transparency. love the kale chips.
	- mmm i love chocolate . lol
	- oh i am sold on chocolate. eating it off the cob
	- haha
	- i miss the crazy curly hair fries and crackers . haha

Table 17: Example of dialogue simulation between two agents using the different decoding methods.