THE CURIOUS CASE OF NEURAL TEXT *De*GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite considerable advances in neural language modeling, it remains an open question what the best *decoding strategy* is for text generation from a language model (e.g. to generate a story). The counter-intuitive empirical observation is that even though the use of likelihood as training objective leads to high quality models for a broad range of language understanding tasks, maximization-based decoding methods such as beam search lead to *degeneration* — output text that is bland, incoherent, or gets stuck in repetitive loops.

To address this we propose **Nucleus Sampling**, a simple but effective method to draw considerably higher quality text out of neural language models than previous decoding strategies. Our approach avoids text *de*generation by truncating the unreliable tail of the probability distribution, sampling from the dynamic nucleus of tokens containing the vast majority of the probability mass.

To properly examine current maximization-based and stochastic decoding methods, we compare generations from each of these methods to the distribution of human text along several axes such as likelihood, diversity, and repetition. Our results show that (1) maximization is an inappropriate decoding objective for open-ended text generation, (2) the probability distributions of the best current language models have an unreliable tail which needs to be truncated during generation and (3) Nucleus Sampling is the best available decoding strategy for generating long-form text that is both high-quality — as measured by human evaluation — and as diverse as human-written text.

Context: In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Beam Search, *b*=100:
"The unicorns were very intelligent, and they were very intelligent," said Dr. David S. Siegel, a professor of anthropology at the University of California, Berkeley."They were very intelligent, and they were very intelligent, and they were very intelligent."

Pure Sampling:
The researchers credited these Mages with building Fabian leather armor. Representing an increasing global problem for consumers, the Fabians encouraged different types of leather to be made by various models of the m animal. A bearded, 350-pound maurice was engaged in strenuous natural exercise, which was covered by waterproof clothing.

Figure 1: Even with significant human context and the powerful GPT-2-117M model, BeamSearch leads to degenerate repetition (highlighted in blue) and pure sampling (sampling directly from the softmax distribution) leads to incoherent gibberish (highlighted in red). This stands in stark contrast with the admirable quality of the text decoded using *top-k* sampling in Radford et al. (2019).

1 INTRODUCTION

On February 14th 2019, OpenAI surprised the scientific community with an impressively high-quality article about Ovid's Unicorn, written by GPT-2.[1] Notably, the top-quality generations obtained from the model rely on *randomness* in the decoding method, in particular through *top-k* sampling that samples the next word from the top k most probable choices (Fan et al., 2018a; Holtzman et al., 2018; Radford et al., 2019), instead of aiming to decode text that *maximizes* likelihood.

[1] https://openai.com/blog/better-language-models/

In fact, decoding strategies that optimize for output with high probability, such as beam search, lead to text that is incredibly degenerate, even when using state-of-the-art models such as GPT-2-117M, as shown in Figure 1. This may seem counter-intuitive, as one would expect that good models would assign higher probability to more human-like, grammatical text. Indeed, language models do generally assign high scores to well-formed text, yet the *highest* scores for longer texts are often generic, repetitive, and awkward.

Perhaps equally surprising is the right side of Figure 1, which shows that pure sampling — sampling directly from the probabilities predicted by the model — results in text that is incoherent and almost unrelated to the context. Why is text produced by pure sampling so degenerate? In this work we show that the "unreliable tail" is to blame. This unreliable tail is composed of tens of thousands of candidate tokens with relatively low probability that are over-represented in the aggregate.

To overcome these shortcomings we introduce *Nucleus Sampling* (§3.1). The key intuition of Nucleus Sampling is that the vast majority of probability mass at each time step is concentrated in the *nucleus*, a small subset of the vocabulary that tends to range between one and a thousand candidates. Instead of relying on a fixed top-k, or using a temperature parameter to control the shape of the distribution without sufficiently suppressing the unreliable tail distribution, we propose sampling from the top-p portion of the probability mass, expanding and contracting the candidate pool dynamically.

In order to compare current methods to Nucleus Sampling, we compare various distributional properties of generated text to the reference distribution, such as the likelihood of veering into repetition and the perplexity of *generated* text. The latter reveals that text generated by maximization or top-k sampling is *too* probable, indicating a lack of diversity and divergence in vocabulary usage from the human distribution. On the other hand, pure sampling produces text that is significantly *less* likely than the gold, corresponding to lower generation quality.

Vocabulary usage and Self-BLEU (Zhu et al., 2018) statistics reveal that high values of k are needed to make top-k sampling match human statistics. Yet, generations based on high values of k are also found to have incredibly high variance in likelihood, hinting at qualitatively observable incoherency issues. Nucleus Sampling can easily match reference perplexity through a proper value of p, without facing the resulting incoherence caused by setting k high enough to match distributional statistics.

Finally, we perform Human Unified with Statistical Evaluation (HUSE; Hashimoto et al., 2019) to jointly assess the overall quality and diversity of the decoding strategies, which cannot be captured using either human or automatics evaluation alone. The HUSE evaluation demonstrates that Nucleus sampling is the best overall decoding strategy. We include generated examples for qualitative analysis – see Figure 9 for a representative example, and further examples in the appendix.

2 BACKGROUND

2.1 TEXT GENERATION DECODING STRATEGIES

A number of recent works have alluded to the disadvantages of generation by maximization, which tend to generate output with high grammaticality but low diversity (Kulikov et al., 2018; Holtzman et al., 2018; Fan et al., 2018a). Generative Adversarial Networks (GANs) have been a prominent research direction (Yu et al., 2017; Xu et al., 2018), but recent work has shown that when quality and diversity are considered jointly, GAN-generated text fails to outperform generations from language models (Caccia et al., 2018; Tevet et al., 2018; Semeniuta et al., 2018). Work on neural dialog systems have proposed methods for diverse beam search, using a task-specific diversity scoring function or constraining beam hypotheses to be sufficiently different (Li et al., 2016a; Vijayakumar et al., 2018; Kulikov et al., 2018; Pal et al., 2006). While such utility functions encourage desirable properties in generations, they do not remove the need to choose an appropriate decoding strategy, and we believe that Nucleus Sampling will have complementary advantages in such approaches.

2.2 OPEN-ENDED VS DIRECTED GENERATION

Many text generation tasks are defined through (input, output) pairs, such that the output is a constrained *transformation* of the input. Example applications include machine translation (Bahdanau et al., 2015), data-to-text generation (Wiseman et al., 2017), and summarization (Nallapati et al.,

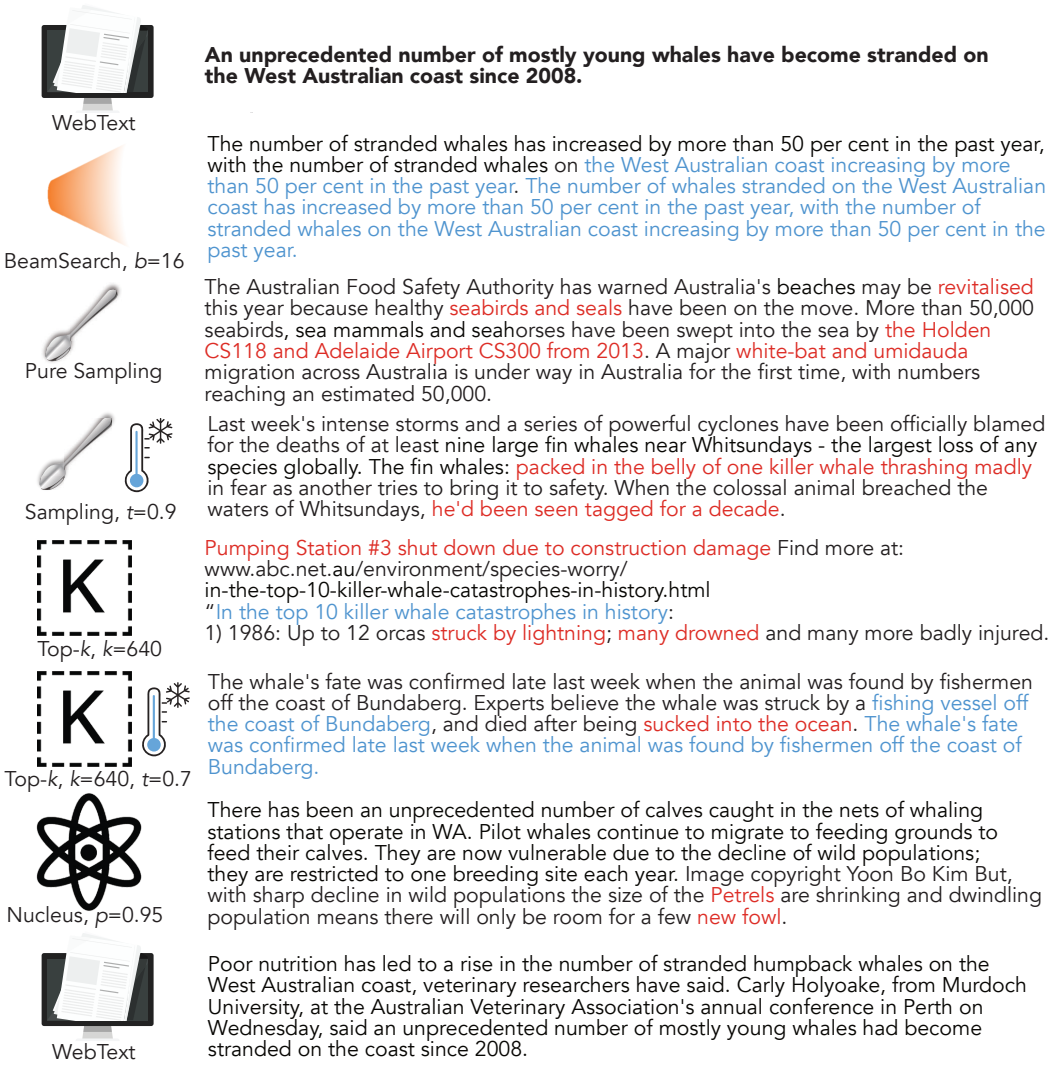

Figure 2: Example generations from an initial sentence. Maximization and top-k truncation methods lead to copious repetition (highlighted in blue), while sampling with and without temperature tends to lead to incoherence (highlighted in red). Nucleus Sampling largely avoids both issues. All generations for all hyperparameters will be made publicly available, along with our codebase.

2016). We refer to these tasks as *directed* generation. Typically encoder-decoder architectures are used, often with an attention mechanisms (Bahdanau et al., 2015; Luong et al., 2015) or using attention-based architectures such as the Transformer (Vaswani et al., 2017). Generation is usually performed using beam search; since output is tightly scoped by the input, repetition and genericness are not as problematic. Still, similar issues have been reported when using large beam sizes (Koehn & Knowles, 2017), a counter-intuitive observation since large beam sizes help maximize probability.

Open-ended generation, which includes conditional story generation and contextual text continuation (as in Figure 1), has recently become a promising research direction due to significant advances in neural language models (Clark et al., 2018; Holtzman et al., 2018; Fan et al., 2018b; Peng et al., 2018; Radford et al., 2019). While the input context restricts the space of acceptable output generations, there is a considerable degree of freedom in what can plausibly come next, unlike in directed generation settings. Our work addresses the challenges faced by neural text generation with this increased level of freedom, but we note that some tasks, such as goal-oriented dialog, may fall somewhere in between open-ended and directed generation.

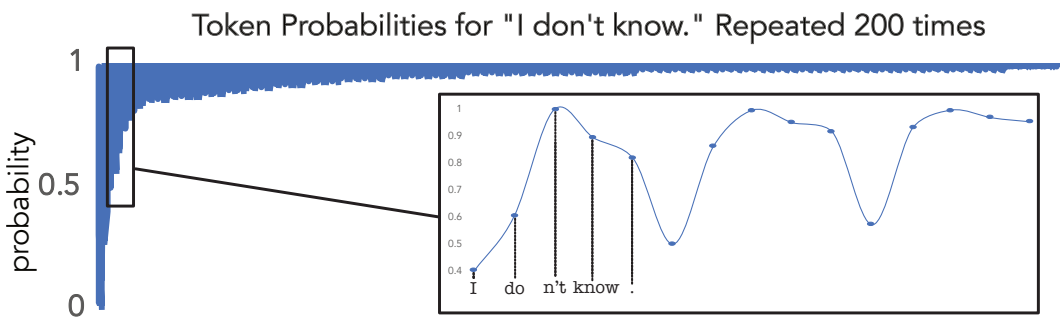

Figure 3: The probability of a repeated phrase increases with each repetition, creating a positive feedback loop. We found this effect to hold for the vast majority of phrases we tested, regardless of phrase length or if the phrases were sampled randomly rather than taken from human text.

3 Language Model Decoding

Given an input text passage as context, the task of *open-ended* generation is to generate text that forms a coherent continuation from the given context. More formally, given a sequence of m tokens $x_1 \dots x_m$ as **context**, the task is to generate the next n **continuation** tokens to obtain the completed sequence $x_1 \dots x_{m+n}$. We assume that models compute $P(x_{1:m+n})$ using the common left-to-right decomposition of the text probability,

$$P(x_{1:m+n}) = \prod_{i=1}^{m+n} P(x_i | x_1 \dots x_{i-1}), \tag{1}$$

which is used to generate the generation token-by-token using a particular *decoding strategy*.

Maximization-based decoding The most commonly used decoding objective, in particular for directed generation, is maximization-based decoding. Assuming that the model assigns higher probability to higher quality text, these decoding strategies search for the continuation with the highest likelihood. Since finding the optimum argmax sequence from recurrent neural language models or Transformers is not tractable (Chen et al., 2018), common practice is to use beam search (Li et al., 2016b; Shen et al., 2017; Wiseman et al., 2017). However, several recent studies on open-ended generation have reported that maximization-based decoding does not lead to high quality text (Fan et al., 2018a; Holtzman et al., 2018).

3.1 Nucleus Sampling

We propose a new stochastic decoding method: Nucleus Sampling. The key idea is to use the shape of the probability distribution to determine the set of tokens to be sampled from. Given a distribution $P(x|x_{1:i-1})$, we define its top-p vocabulary $V^{(p)} \subset V$ as the smallest set such that

$$\sum_{x \in V^{(p)}} P(x|x_{1:i-1}) >= p. \tag{2}$$

Let $p' = \sum_{x \in V^{(k)}} P(x|x_{1:i-1})$. The original distribution is re-scaled to a new distribution, from which the next word is sampled:

$$P'(x|x_{1:i-1}) = \begin{cases} P(x|x_{1:i-1})/p' & \text{if } x \in V^{(k)} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

In practice this means selecting the highest probability tokens whose cumulative probability mass exceeds the pre-chosen threshold p. The size of the sampling set will adjust dynamically based on the shape of the probability distribution at each time step. For high values of p, it is a small subset of vocabulary that takes up most of the probability mass — the *nucleus*.

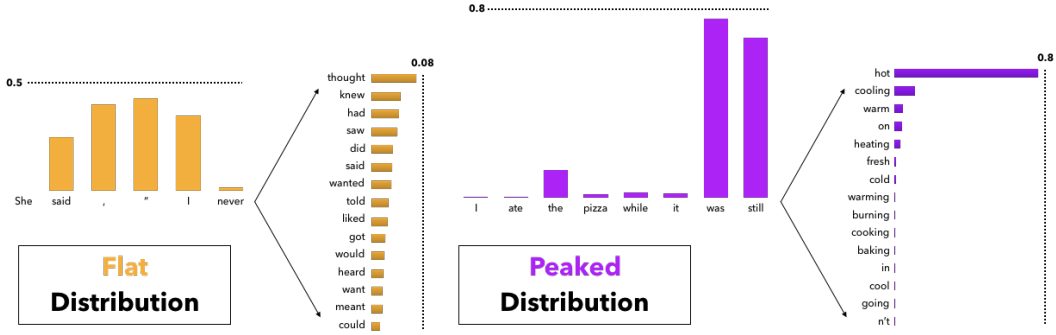

Figure 4: The probability mass assigned to partial human sentences. Flat distributions lead to many moderately probable tokens, while peaked distributions concentrate most probability mass into just a few tokens. The presence of flat distributions makes the use of a small k in top-k sampling problematic, while the presence of peaked distributions makes large k problematic.

3.2 TOP-k SAMPLING

Top-k Sampling has recently become a popular alternative sampling procedure (Fan et al., 2018a; Holtzman et al., 2018; Radford et al., 2019). Nucleus Sampling and top-k both sample from truncated Neural LM distributions, differing only in the strategy of where to truncate. Choosing where to truncate can be interpreted as determining the generative model's trustworthy prediction zone.

At each time step, the top k possible next tokens are sampled from according to their relative probabilities. Formally, given a distribution $P(x|x_{1:i-1})$, we define its top-k vocabulary $V^{(k)} \subset V$ as the set of size k which maximizes $\sum_{x \in V^{(k)}} P(x|x_{1:i-1})$. Let $p' = \sum_{x \in V^{(k)}} P(x|x_{1:i-1})$. The distribution is then re-scaled as in equation 3, and sampling is performed based on that distribution. Note that the scaling factor p' will be different at each time-step, in contrast to Nucleus Sampling.

Difficulty in choosing a suitable value of k While top-k sampling leads to considerably higher quality text than either beam search or sampling from the full distribution, the use of a constant k is sub-optimal across varying contexts. As illustrated on the left of Figure 4, in some contexts the head of the next word distribution can be flat across tens or hundreds of reasonable options (e.g. nouns or verbs in generic contexts), while in other contexts most of the probability mass is concentrated in one or a small number of tokens, as on the right of the figure. Therefore if k is small, in some contexts there is a risk of generating bland or generic text, while if k is large the top-k vocabulary will include inappropriate candidates which will have their probability of being sampled *increased* through the truncation. Under Nucleus Sampling, the number of candidates considered rises and falls dynamically, corresponding to the changes in the model's confidence region over the vocabulary which top-k sampling fails to capture for any one choice of k.

3.3 SAMPLING WITH TEMPERATURE

Another common approach to sampling-based generation is to shape a probability distribution through temperature (Ackley et al., 1985). Temperature sampling has been applied widely to text generation (Ficler & Goldberg, 2017; Fan et al., 2018a; Caccia et al., 2018). Given the logits $u_{1:|V|}$ and temperature t, the softmax is re-estimated as

$$p(x = V_l|x_{1:i-1}) = \frac{\exp(u_l/t)}{\sum_{l'} \exp(u_l/t)}. \tag{4}$$

Setting $t \in [0, 1)$ shapes the distribution to be more skewed towards high probability events, which implicitly lowers the mass in the tail distribution. Low temperature sampling has also been used to partially alleviate the issues of top-k sampling discussed above, by shaping the distribution before top-k sampling (Radford et al., 2018; Fan et al., 2018a). However, recent analysis has shown that, while lowering the temperature improves generation quality, it comes at the cost of decreasing diversity (Caccia et al., 2018; Hashimoto et al., 2019).

Method	Perplexity	Self-BLEU4	Zipf Coefficient	Repetition %	HUSE
Human	13.08	0.30	0.93	0.18	-
Greedy	1.50	0.50	1.00	73.20	-
Beam, b=16	1.49	0.50	1.00	73.26	-
Pure Sampling	22.73	**0.28**	**0.93**	0.14	0.81
Sampling, t=0.9	10.25	0.35	0.96	0.54	0.88
Top-k=640	13.82	**0.32**	0.96	0.14	0.98
Top-k=40, t=0.7	3.48	0.44	1.00	8.50	0.96
Nucleus p=0.95	**13.13**	**0.32**	0.95	**0.20**	**1.00**

Table 1: Main results for comparing all decoding methods with selected parameters of each method. The numbers *closest to human scores* are in **bold** except for HUSE (Hashimoto et al., 2019), a combined human and statistical evaluation, where the highest (best) value is **bolded** Nucleus Sampling performs the same or better (compared to the human text for distributional metrics) than all other methods on all metrics, except for pure sampling's Zipf coefficient, which is expected since truncation curtails the distribution. Note that both pure sampling and top-k *underestimate* the amount of repetition, whereas Nucleus Sampling closely matches human levels.

4 Likelihood Evaluation

4.1 Experimental Setup

While many neural network architectures have been proposed for language modeling, including LSTMs (Sundermeyer et al., 2012) and convolutional networks (Dauphin et al., 2017), the Transformer architecture (Vaswani et al., 2017) has been the most successful in the extremely large-scale training setups in recent literature (Radford et al., 2018; 2019). In this study we use the Generative Pre-trained Transformer, version 2 (Radford et al., 2019), which was trained on WebText, a 40GB collection of text scraped from the web. We perform experiments using the Large model (762M parameters). Our analysis is based on generating 5,000 text passages, which end upon reaching an end-of-document token or a maximum length of 200 tokens. We consider a conditional generation setting, in which texts are conditioned on the initial paragraph (restricted to 1-40 tokens) of documents in the validation portion of WebText.

4.2 Perplexity

Our first evaluation is to compute the perplexity of generated text using various decoding strategies, according to the model that is being generated from. We compare these perplexities against that of the gold text (Figure 5). Importantly, we argue that the optimal generation strategy should produce text which has a perplexity close to that of the gold text: Even though the model has the ability to generate text that has lower perplexity (higher probability), such text tends to be low-diverstiy and get stuck in repetition loops, as shown in §5 and illustrated in Figure 3.

We see that perplexity of text obtained from pure sampling (sampling with $t = 1$) is actually worse than the perplexity of the gold. This indicates that the model is actually confusing itself: sampling too many unlikely tokens and creating context that makes it difficult to recover the human distribution of text, as happens in Figure 1. Yet setting the temperature lower creates diversity repetition issues, as we shall see in §5. Interestingly, even with the relatively fine-grained parameter sweep conducted, Nucleus Sampling gets closer perplexity in both the conditional and under conditional case, than any other method, as shown in Table 1.

4.3 Natural Language Does Not Maximize Probability

One might wonder if the issue with maximization *search error* is a, i.e., there are higher quality sentences to which the model assigns higher probability than to the decoded ones, beam search has just failed to find them. Yet Figure 5 shows that the per-token probability of natural text is, on average, much *lower* than text generated by beam search. Natural language rarely remains in a high probability zone for multiple consecutive time steps, instead veering into lower-probability but

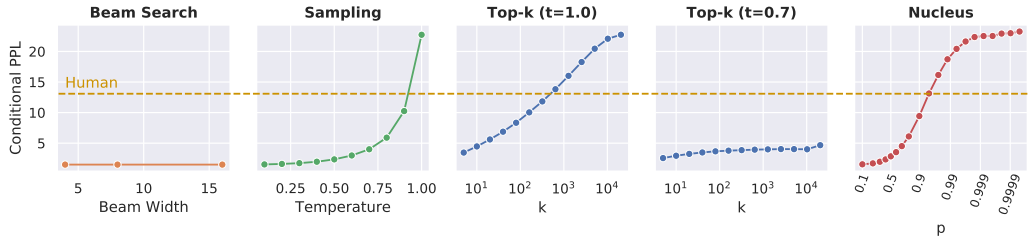

Figure 5: Perplexities of generations from various decoding methods. Note that beam search has unnaturally low perplexities. A similar effect is seen using a temperature of 0.7 with top-k as in both Radford et al. (2019) and Fan et al. (2018a). Sampling, Top-k, and Nucleus can all be calibrated to human perplexities, but we shall see the first two face coherency issues when their parameters are set this high.

more informative tokens. Nor does natural language tend to fall into repetition loops, even though the model tends to assign high probability to this, as seen in Figure 3.

Why is naturally existing human text *not* the most probable text? We conjecture that this is an intrinsic property of human language. Language models that assign probabilities one word at a time without context for the underlying document will have trouble capturing this effect. Grice's Maxims of Communication (Grice, 1975) show that people optimize against stating the obvious. Thus, it is natural that making every word as predictable as possible will be disfavored. This makes solving the problem simply by training larger models or improving neural architectures using standard per-word learning objectives unlikely: models that are blindly predicting the future will be forced to predict the lowest common denominator, rather than information rich language.

5 Distributional Statistical Evaluation

5.1 Zipf Distribution Analysis

In order to compare generations to the reference text, we begin by analyzing their use of vocabulary. Zipf's law suggests that there is an exponential relationship between the rank of a word and its frequency in text. The Zipfian coefficient s can be used to compare the distribution in a given text to a theoretically perfect exponential curve, where $s = 1$ (Piantadosi, 2014). Figure 6 shows the vocabulary distributions along with estimated Zipf coefficients for selected parameters of different decoding methods. As expected, pure sampling is the closest to the human distribution, followed by

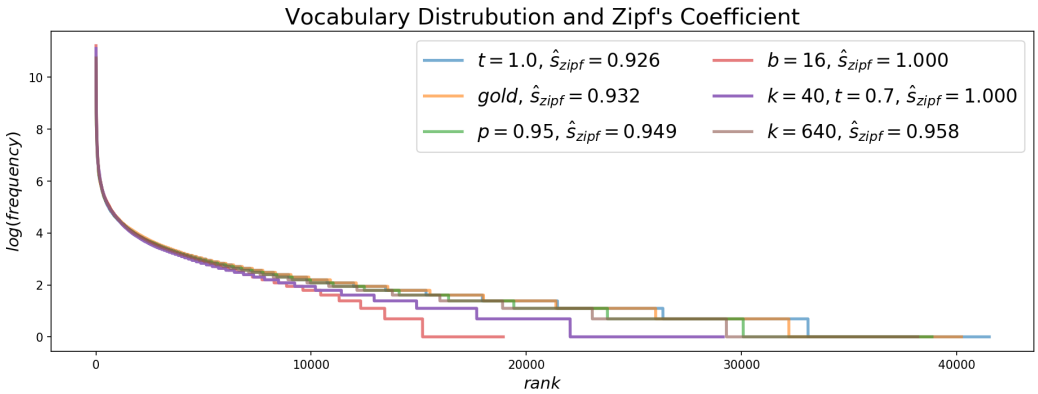

Figure 6: A chart describing the distributional differences between n-gram frequencies of human and machine text. Sampling and Nucleus Sampling are by far the closest, with maximization methods greedy and Beam Search clearly in a very different distribution than natural language.

Nucleus Sampling. The visualization of the distribution shows that pure sampling slightly *overestimates* the use of rare words, likely one reason why pure sampling also has higher perplexity than human text. Furthermore, lower temperature sampling avoids sampling these rare words from the tail, which is why it has been used in some recent work (Fan et al., 2018a; Radford et al., 2019).

5.2 SELF-BLEU

We follow previous work and compute Self-BLEU (Zhu et al., 2018) as a metric of diversity. Self-BLEU is calculated by computing the BLEU score of each generated document using *all other generations* in the evaluation set as references. Thus a lower Self-BLEU score implies higher diversity. Figure 7 shows that Self-BLEU results largely follow that of the Zipfian distribution analysis as a diversity measure. It is worth noting that very high values of k and t are needed to get close to the reference distribution, though these result in unnaturally high perplexity (§4).

5.3 REPETITION

One attribute of text quality that we can quantify is repetition. Figure 8 shows that Nucleus Sampling and top-k sampling have the least repetition for reasonable parameter ranges. Generations from temperature sampling have more repetition unless very high temperatures are used, which we have shown negatively affects coherence (as measured by high perplexity). Further, all stochastic methods face repetition issues when their tuning parameters are set too low, which tends to *over-truncate*, mimicking greedy search. Therefore we conclude that only Nucleus Sampling satisfies all the distributional criteria for desirable generations.

6 HUMAN EVALUATION

6.1 HUMAN UNIFIED WITH STATISTICAL EVALUATION (HUSE)

Statistical evaluations are unable to measure the coherence of generated text properly. While the metrics in previous sections gave us vital insights into the different decoding methods we compare, human evaluation is still required to get a full measure of the quality of the generated text. However, pure human evaluation does not take into account the diversity of the generated text; therefore we use HUSE (Hashimoto et al., 2019) to combine human and statistical evaluation. HUSE is computed by training a discriminator to distinguish between text drawn from the human and model distributions,

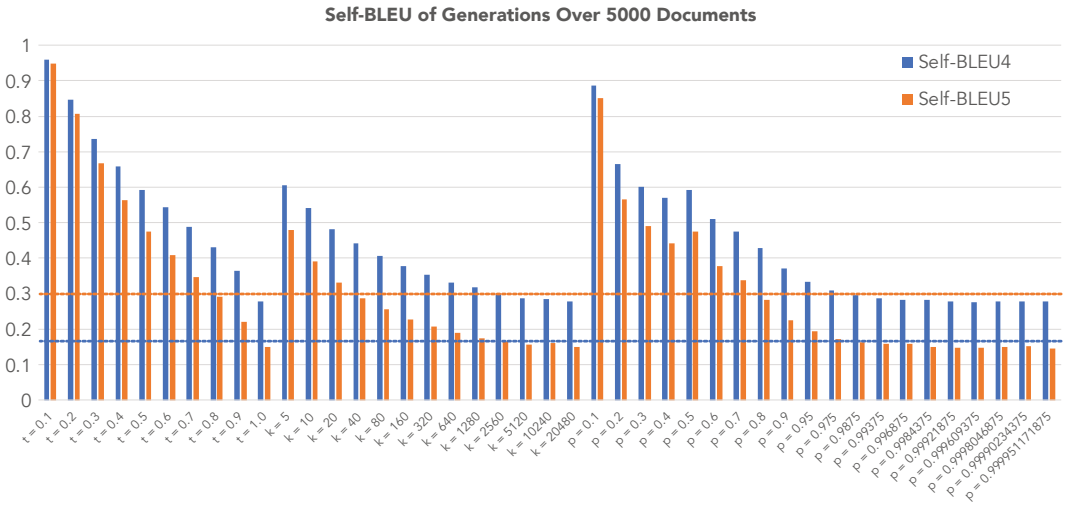

Figure 7: Self-BLEU calculated on the unconditional generations produced by stochastic decoding methods; lower Self-BLEU scores imply higher diversity. Horizontal blue and orange lines represent human self-BLEU scores. Note how common values of $t \in [0.5, 1]$ and $k \in [1, 100]$ result in high self-similarity, whereas "normal" values of $p \in [0.9, 1)$ closely match the human distribution of text.

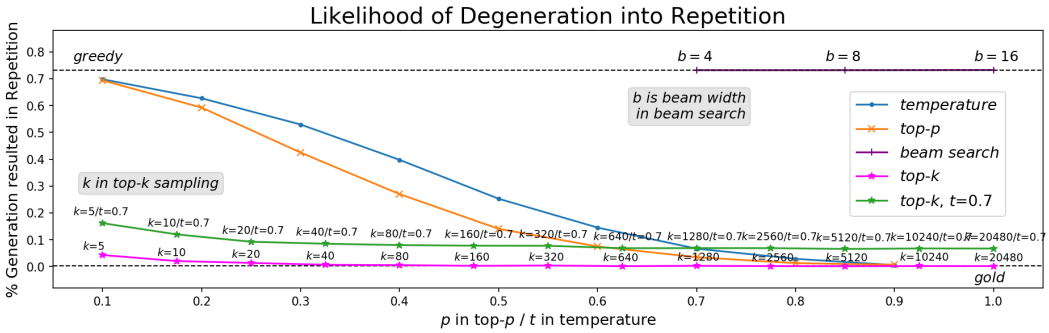

Figure 8: We visualize how often different decoding methods get "stuck" in loops within the first 200 tokens. Concretely, a phrase (sequence of words with minimum length 2) must repeat at least **three** times at the end of the generation to be considered a repetition. Since all the methods described used different parameter ranges, we label points with the appropriate parameter value except for t and p. Since values of k greater than 100 are rarely used in practice and values of p are usually in $[0.9, 1)$, it becomes clear that Nucleus sampling is far closer to the human distribution in its usual parameter range. Finally, temperatures lower than 0.9 severely effect repetition, making it problematic for use in long-form generation.

based on only two features: The probability assigned by the language model, and human judgements of typicality of generations. Text that is close to the human distribution in terms of quality and diversity should perform well on both likelihood evaluation and human judgements. The results in Table 1 shows that Nucleus Sampling obtains the highest HUSE score, with Top-k sampling performing next best.

6.2 Qualitative analysis

Figure 9 shows representative example generations. Unsurprisingly, Beam Search gets stuck in a repetition loop it cannot escape. Of the stochastic decoding schemes, the output of full sampling is clearly the hardest to understand, even inventing a new word "umidauda", apparently a species of bird. The generation produced by Nucleus Sampling isn't perfect – the model appears to get get confused that it was talking about whales, and begins talking about birds instead. Yet, Top-k Sampling immediately veers off into an unrelated event. When top-k sampling is combine with a temperature of 0.7, as is commonly done (Radford et al., 2019; Fan et al., 2018a), the output devolves into repetition.

7 Conclusion

This paper provided a deep analysis into the properties of the most common decoding methods for open-ended language generation. We have shown that likelihood maximizing decoding causes repetition and overly generic language usage, while sampling methods without truncation risk sampling from the low-confidence tail of a model's predicted distribution. Further, we proposed Nucleus Samplingas a solution that captures the region of confidence of language models effectively. In future work, we wish to dynamically characterize this region of confidence and include a more semantic utility function to guide the decoding process.

References

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *Proceedings of the 2015 International Conference on Learning Representations*, 2015.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. Language gans falling short. In *Critiquing and Correcting Trends in Machine Learning: NeurIPS 2018 Workshop*, 2018. URL http://arxiv.org/abs/1811.02549.

Yining Chen, Sorcha Gilroy, Andreas Maletti, Jonathan May, and Kevin Knight. Recurrent neural networks as weighted language recognizers. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2261–2271, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1205. URL https://www.aclweb.org/anthology/N18-1205.

Elizabeth Clark, Yangfeng Ji, and Noah A. Smith. Neural text generation in stories using entity representations as context. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2250–2260, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1204. URL https://www.aclweb.org/anthology/N18-1204.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 933–941. JMLR. org, 2017.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *ACL*, 2018a.

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

Jessica Ficler and Yoav Goldberg. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pp. 94–104, 2017.

H Paul Grice. Logic and conversation. In P Cole and J L Morgan (eds.), *Speech Acts*, volume 3 of *Syntax and Semantics*, pp. 41–58. Academic Press, 1975.

Tatsunori B. Hashimoto, Hugh Zhang, and Percy Liang. Unifying human and statistical evaluation for natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019. To appear.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. Learning to write with cooperative discriminators. In *Proceedings of the Association for Computational Linguistics*, 2018.

Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pp. 28–39, 2017.

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

Jiwei Li, Will Monroe, and Dan Jurafsky. A simple, fast diverse decoding algorithm for neural generation. *CoRR*, abs/1611.08562, 2016a.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1192–1202, 2016b.

Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, 2015.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 280–290, 2016. URL https://www.aclweb.org/anthology/K16-1028.

C. Pal, C. Sutton, and A. McCallum. Sparse forward-backward using minimum divergence beams for fast training of conditional random fields. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 5, pp. V–V, May 2006. doi: 10.1109/ICASSP.2006.1661342.

Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, pp. 43–49, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-1505. URL https://www.aclweb.org/anthology/W18-1505.

Steven T Piantadosi. Zipfs word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21(5):1112–1130, 2014.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018. URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf. Unpublished manuscript.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, February 2019. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. Unpublished manuscript.

Stanislau Semeniuta, Aliaksei Severyn, and Sylvain Gelly. On accurate evaluation of gans for language generation. *arXiv preprint arXiv:1806.04936*, 2018.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*, pp. 6830–6841, 2017.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

Guy Tevet, Gavriel Habib, Vered Shwartz, and Jonathan Berant. Evaluating text gans as language models. *CoRR*, abs/1810.12686, 2018. URL http://arxiv.org/abs/1810.12686.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

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

Sam Wiseman, Stuart Shieber, and Alexander Rush. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2253–2263, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1239. URL https://www.aclweb.org/anthology/D17-1239.

Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. Diversity-promoting gan: A cross-entropy based generative adversarial network for diversified text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3940–3949, Brussels, Belgium, oct 2018. URL http://www.aclweb.org/anthology/D18-1428.

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

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

A EXAMPLE GENERATIONS

We include another set of examples for further qualitative comparison.

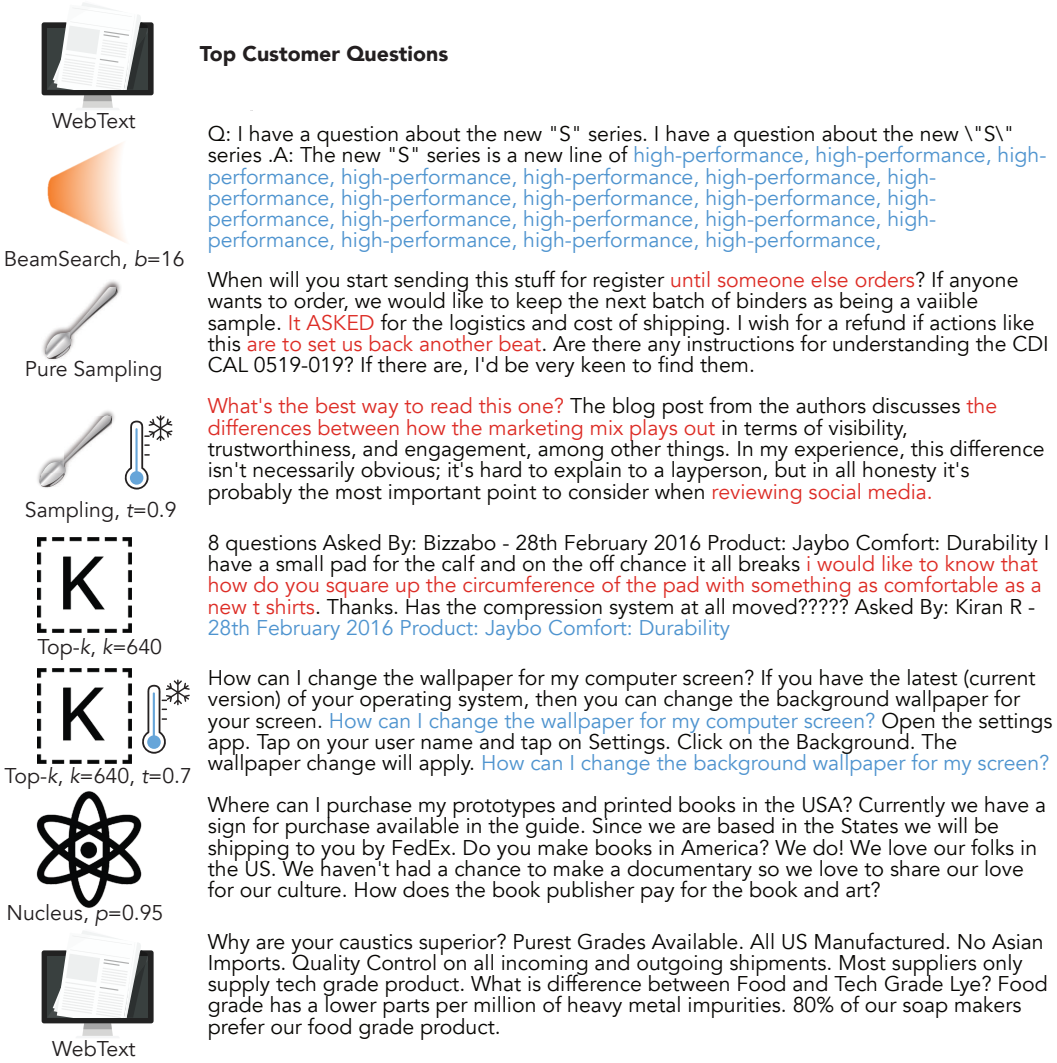

Figure 9: More example generations from an initial tag line. All generations for all hyperparameters will be made publicly available, along with our codebase.