Improved Natural Language Generation via Loss Truncation

Daniel Kang
Stanford University
ddkang@stanford.edu

Tatsunori B. Hashimoto
Stanford University
thashim@stanford.edu

Abstract

Neural language models are usually trained to match the distributional properties of large-scale corpora by minimizing the log loss. While straightforward to optimize, this approach forces the model to reproduce all variations in the dataset, including noisy and invalid references (e.g., misannotations and hallucinated facts). Even a small fraction of noisy data can degrade the performance of log loss. As an alternative, prior work has shown that minimizing the distinguishability of generated samples is a principled and robust loss that can handle invalid references. However, distinguishability has not been used in practice due to challenges in optimization and estimation. We propose loss truncation: a simple and scalable procedure which adaptively removes high log loss examples as a way to optimize for distinguishability. Empirically, we demonstrate that loss truncation outperforms existing baselines on distinguishability on a summarization task. Furthermore, we show that samples generated by the loss truncation model have factual accuracy ratings that exceed those of baselines and match human references.

1 Introduction

Learning to generate text is a core part of many NLP tasks, including summarization (Nallapati et al., 2016), image captioning (Lin et al., 2014), and story generation (Roemmele, 2016). A common challenge to all these tasks is that references from the training distribution are not unique and contain substantial variations in phrasing and content (Wiseman et al., 2017; Dhingra et al., 2019). Learning to generate under a set of diverse and noisy references is challenging as some variations ought to be learned (e.g., paraphrasing) while others should not (e.g., hallucinated facts, ignoring prompts).

Existing training procedures for models seek to match the underlying distribution, leading to models that replicate and sometimes even amplify unwanted behaviors such as hallucination during generation. For example, neural language models often produce fluent text that is unfaithful to the source (Tian et al., 2019; Wiseman et al., 2017; Lee et al., 2018). Existing work (Fan et al., 2018; Holtzman et al., 2019) has primarily addressed these issues by constructing decoders that implicitly remove unwanted variation when generating (see §6 for a detailed discussion of task-specific losses).

In this work, we argue that this phenomenon is not model specific, but is due to the widely-used log loss: we demonstrate that log loss is not robust to noisy and invalid references (§2). In particular, log loss requires that models assign probabilities to **all** potential test reference sequences. As a result, log loss is sensitive to outliers: invalid or noisy references with small probability mass can cause large changes in model behavior. We show that the brittleness of log loss, together with the noise in existing generation datasets, lead to low-quality and unfaithful generated text.

Instead of optimizing log loss, which has little correlation with model output quality (Theis et al., 2016; Hashimoto et al., 2019; Gamon et al., 2005), recent work on diverse generation models has proposed optimizing for the *distinguishability* of samples from the model and the reference. Distinguishability provides a natural and appealing guarantee: samples that are indistinguishable from human generated text will be as high quality as human generated text. Furthermore, we show that optimizing for distinguishability is robust in the face of noisy and even invalid data. Despite its appeal, distinguishability has not been widely used due to statistical and computational challenges. For example, existing methods that directly optimize for distinguishability have yet to match even naive log loss based baselines (Caccia et al., 2018).

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 718–731
July 5 - 10, 2020.

We propose a modification to the log loss, *loss truncation*, that has the benefits of distinguishability while being efficient to train. Loss truncation is as efficient to train as log loss, nearly as robust as distinguishability, and provides distinguishability guarantees via an upper bound. It achieves these properties by modifying the standard log loss to adaptively remove examples with high log loss. We additionally extend loss truncation with a *sequence-level* rejection sampling scheme that generates higher quality sequences by restricting the outputs to be high probability sequences.

We show that loss truncation with direct and rejection sampling outperforms standard log loss based generation methods (beam search, full sampling, top-k, and top-p sampling) on distinguishability, as measured by the HUSE score (Hashimoto et al., 2019). We additionally study the factual accuracy of a summarization system trained on loss truncation and show that our proposed approach produces summaries which improve upon all baselines (including beam searched models) and match references on factual accuracy.

2 Motivation and Problem Statement

Task and Background. We consider a natural language generation task with a *conditional language model*, where we are given a context x drawn from $p(x)$ and our probabilistic model $\hat{p}(y \mid x)$ produces an output y by approximating a (usually human) reference distribution $p_{\text{ref}}(y|x)$.

In order to achieve this, many existing models are trained to minimize the Kullback-Leibler (KL) divergence,

$$\text{KL}(p_{\text{ref}}||\hat{p}) = \underbrace{-E_{p_{\text{ref}}}[\log \hat{p}]}_{\text{log loss}} + \underbrace{E_{p_{\text{ref}}}[\log p_{\text{ref}}]}_{\text{negentropy}}. \tag{1}$$

We refer to the first term of this divergence as the *log loss* of a model. The second term is commonly ignored as it is a constant with respect to the model. Minimizing the log loss has several practical benefits: 1) it is written as an expected loss (and is thus straightforward to optimize via stochastic gradient descent), 2) it factorizes across tokens in autoregressive modeling, and 3) it provides a guarantee on a model's goodness of fit (Eq (1)).

Unfortunately, log loss also suffers from several drawbacks. It is known to have little correlation with a model's sample quality and it can be brittle to invalid references in the training data.

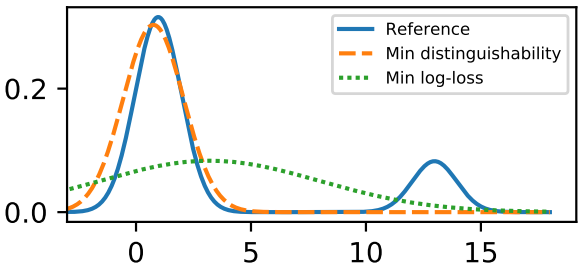


Figure 1: Fitting a mixture of Gaussians with a single Gaussian using distinguishability (TV) and log loss (KL). As shown, log loss is extremely sensitive to outliers, resulting in poor estimation.

Log loss is not robust to noise. The KL divergence has intuitively correct behavior when each input x has a single correct reference y: it will maximize the probability of the single correct reference. However, log loss can be problematic when there are multiple correct references, of which some are invalid or difficult to model.

In particular, log loss is sensitive to invalid or noisy data because it requires that the model assign high probabilities to *all* potential references. Log loss is unbounded above: a model assigning zero probability to even a single reference makes the model incur an infinite overall loss.

We show a well-known example of this behavior with synthetic data. We consider fitting a single Gaussian to a mixture of two Gaussian in Figure 1. The reference distribution (blue) has a valid set of references at zero as well as variation that the model does not expect (e.g., invalid or noisy references) on the right. Minimizing the log loss results in a suboptimal model that is forced to span both groups. Furthermore, post-hoc processing the model does not help, as even the most likely output under the log loss trained model (~3) has low probability under the reference distribution.

In natural language generation, training sets can contain invalid or poor quality references. As such, these types of problems manifest themselves in tasks such as summarization (hallucinating facts), story generation (ignoring prompts and constraints), and captioning (ignoring parts of the image).

Much of the existing literature on faithful generation has focused on designing better models for *valid* references (via copying or attention constraints), but the example in Figure 1 shows that this alone may not be sufficient. The Gaussian 'model' in this case perfectly fits the mixture component

Context: For the first time in five years, Microsoft corp. is finally unveiling a new system for operating personal computers.
Title: Microsoft Makes Long-Awaited Software Upgrade Available to Businesses Thursday.

Figure 2: Example of an article title from the Gigaword dataset that requires hallucinating new facts such as 'Thursday' (colored red).

at zero but is still brittle because it cannot simultaneously fit the other group of (invalid) samples. Resolving this will require either a model which is designed explicitly to capture *invalid* references or a loss function that can ignore them.

Case Study: Hallucination in Summarization We show that low-probability reference sequences (e.g., Figure 1) are pervasive by examining the Gigaword summarization dataset (Rush et al., 2017). We manually classified 300 titles into two categories: 1) requires hallucinating new facts and 2) directly entailed from the context. We show an example of a reference that requires hallucination in Figure 2. In this example, a model that assigns high probability to the new fact (Thursday) must also frequently hallucinate dates on other examples.

We show the fraction of examples in each category in Table 1. As shown, *35%* of titles require hallucinating new facts. Others have found this phenomenon to be pervasive in other datasets (Kryściński et al., 2019), including the CNN/DM dataset (See et al., 2017).

Studying the log loss of these examples[1], we note that the average log loss of titles that require new facts is over 1.7× the average loss of the titles that are directly entailed (Table 1) and the high-loss examples are clearly dominated by examples which require hallucination (Figure 3). In fact, we find that over 80% of examples with greater than 40 log loss requires some form of hallucination.

These statistics are similar to the toy example we presented earlier in Figure 1. A small but nontrivial fraction of invalid and unexpected data force the model to incur high losses. Much like in the earlier example, we can see that a model which aims to have low log loss on this dataset must spend a substantial amount of effort learning to hallucinate.

Distinguishability. Given that large-scale data

[1]The log loss was computed from a standard language model, see §5 for details.

	New facts	Directly entailed
Percent	35%	65%
Avg. log loss	34.3	20.5

Table 1: Fraction of the data and log loss of titles that require hallucinating new facts (left column) and titles that are entailed from the context (right column). As shown, *35%* of titles require hallucinating new facts and the average log loss of titles requiring new facts is over 1.7× the loss of the directly entailed sequences.

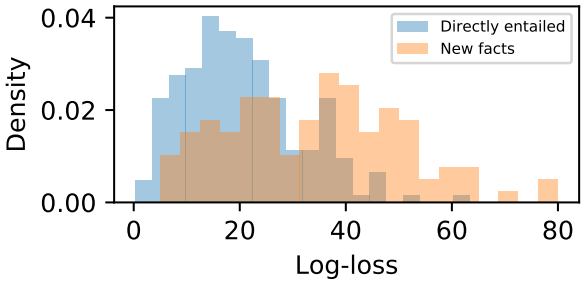


Figure 3: Normalized histogram of log losses for titles that require hallucinating new facts compared to those that can be directly entailed. As shown, titles requiring new facts incur significantly higher loss and more than 80% of examples with greater than 40 log loss require hallucinating new facts.

will inevitably contain annotation errors and noise, we might ask whether there are effective alternatives to the KL divergence for training models. The distinguishability of samples from a model compared to the reference is one such objective. Distinguishability has recently gained attention as a way to learn and evaluate models based on both sample quality and diversity (Hashimoto et al., 2019; Zhou et al., 2019; Zellers et al., 2019; Gehrmann et al., 2019). We show that this objective also serves as a naturally robust alternative to the KL divergence for learning language models. Unfortunately, directly optimizing for distinguishability (e.g., via generative adversarial networks) is challenging (Caccia et al., 2018) and we show this works poorly in practice (§5).

Distinguishability is defined as the error rate of an optimal classifier which seeks to distinguish samples from both the model and reference, and we will formally define this via the mixture

$$y|x,z \sim \begin{cases} p_{\text{ref}}(y|x) & \text{if } z=1 \\ \hat{p}(y|x) & \text{if } z=0 \end{cases}$$

where $z \sim \text{Bernoulli}\left(\frac{1}{2}\right)$. We can now define L^* to be twice the optimal error in identifying samples

from the model

$$L^* := 2 \inf_{f \in \mathcal{X} \times \mathcal{Y} \to [0,1]} \mathbb{P}[f(x, y) \neq z] \quad (2)$$

Our measure of distinguishability, the *total variation (TV) distance*, is a linear function of this error

$$|\hat{p} - p_{\text{ref}}|_{TV} = 1 - L^*$$

where $\hat{p}$ and p_{ref} refer to the joint distributions $\hat{p}(y|x)p(x)$ and $p_{\text{ref}}(y|x)p(x)$ for brevity. Note that distinguishability is inherently *robust* to the addition of *any* small fraction of noisy data (Donoho et al., 1988). Unlike the log loss, the model's loss on an example for TV is upper bounded by 1 (Eq 2). We show an example of TV's robustness in Figure 1, where a small amount of noise does not substantially affect the learned distribution.

Log loss as a surrogate for distinguishability. Distinguishability is both robust and provides sample quality guarantees, but is challenging to optimize (Caccia et al., 2018). One approach to optimize for distinguishability is to find an appropriate *surrogate loss* which serves as an upper bound. This is analogous to the use of logistic or hinge losses as a way to optimize for classification accuracy. For log loss, *Pinsker's inequality* (Csiszar and Körner, 2011) relates the KL divergence and distinguishability as

$$|\hat{p} - p_{\text{ref}}|^2_{TV} \leq \frac{1}{2} \cdot \text{KL}(p_{\text{ref}} || \hat{p}). \quad (3)$$

This explains the empirical success of log loss in low-uncertainty situations, where KL is sufficiently small and this bound becomes tight.

Our approach will be to modify the log loss slightly by truncating the distribution. This truncated loss will be as easy to optimize as log loss, while being more robust and providing a tighter variant of Pinsker's inequality.

3 Loss Truncation

Intuition. We would like the model to ignore data that would force it to unnecessarily hallucinate at test time. Concretely, recall the toy example (Figure 1); there is a set of invalid references that force the model to be degenerate. If we could remove these these invalid references by truncating the distribution, the resulting model would be high quality. We can show that this intuition is theoretically justified, and that truncating (i.e., removing) an appropriate c-fraction of the data provides tighter bounds on the distinguishability of the model.

Improved log losses for distinguishability. We will demonstrate that log loss with an appropriate c-fraction of the data removed provides guarantees on distinguishability. We will define the set of *truncated* distributions as the set of distributions with any c-fraction of data removed

$$\mathcal{P}_{c,p} := \{q_0 : p = (1 - c)q_0 + cq_1 \text{ for some } q_1\}.$$

A simple lemma shows that that all elements in $\mathcal{P}_{c,p}$ are c-close to p in TV (Appendix B).

Now we state our main result,

Proposition 1. *For any* $c \in [0, 1]$ *and* $p_{\text{t}} \in \mathcal{P}_{c,p_{\text{ref}}}$,

$$|\hat{p} - p_{\text{ref}}|^2_{TV} \leq \frac{1}{2} KL(p_{\text{t}} || \hat{p}) + 2c + c^2$$

See Appendix B for the proof. Namely, distinguishability is bounded by the log loss with respect to the truncated distribution and a small constant. Furthermore, this upper bound is valid for *any* c, although different c will change the tightness of the bound and produce different models.

This truncated bound can be substantially tighter than Pinsker's inequality. Consider for example a model that can perfectly capture $(1 - c)$ fraction of the data, but c-fraction of the reference outputs cannot be generated by the model and receive probability zero. In this case, the distinguishability (TV) is c, the KL divergence is *infinite*, while our truncated bound is $\sqrt{c^2 + 2c}$. This suggests that appropriately truncating high-loss examples makes log loss robust and allows us to use log loss as a surrogate for distinguishability, even in the presence of invalid and noisy references.

Loss truncation. Given that the log loss on any c-fraction of the data is a surrogate loss for distinguishability (Eq (6)), a key parameter to optimize is the truncated distribution p_{t}. An oracle solution would exhaustively search over p_{t} and which data to drop. However, exhaustively searching through $\mathcal{P}_{c,p_{\text{ref}}}$ is a combinatorial optimization problem and infeasible. Our approach will be to optimize p_{t} with a heuristic. The truncated objective takes the form of a log loss and negative entropy term,

$$-\mathbb{E}_{p_{\text{t}}}[\log \hat{p}(y \mid x)] + \mathbb{E}_{p_{\text{t}}}[\log p_{\text{t}}(y \mid x)]$$

and we will select p_{t} by dropping the examples with the highest log loss, treating the negative entropy term as being upper bounded by zero.

This heuristic is straightforward to compute, provides an upper bound on distinguishability, and

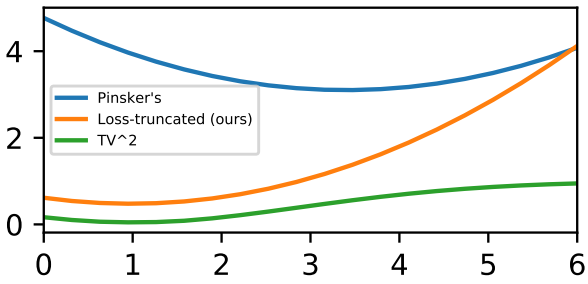


Figure 4: Pinsker's inequality, our bound, and the total variation squared of parameter estimates for different parameter estimates ($c = 0.2$). As shown, loss truncation can significantly improve bounds over Pinsker's inequality and, in this case, has a nearly identical minimizer to directly minimizing total variation.

matches our earlier observation that high-loss examples are correlated with invalid examples we would like the model to ignore (see Table 1).

As an example of how our heuristic can improve estimation and tightness in bounds, consider the earlier toy example in Figure 1. In this example, we find the optimal mean for a single Gaussian with fixed variance which fits mixture of two Gaussians. Figure 4 shows the objective function value implied by the TV loss, log loss (Pinsker's bound), and our c-truncated bound as a function of the Gaussian mean. We find that log loss provides an upper bound on distinguishability (via Pinsker's inequality) but is loose and results in a low quality estimate. In contrast, c-truncation results in a nearly identical minimizer as directly minimizing TV.

4 Implementing Truncation

4.1 Training

Our algorithm has three components at training time. First, it trains a model on all the data using standard hyperparameters, which we refer to as "hotstarting" the model. Second, it tracks a running estimate of the $1 - c$ quantile of the losses during training. Third, it performs gradient updates on examples that are below the current $1 - c$ quantile estimate. We present the pseudocode in Algorithm 1 and describe each step in detail below.[2]

Hotstarting. First, our algorithm hotstarts the model (`hotstart`(M) in Alg. 1) by training with the standard log loss. Hotstarting address two challenges in optimizing the truncated loss. First, losses are uninformative at the start of training so truncating examples based on these losses will result in dropping valid examples. We have empirically found that truncating after hotstarting primarily drops invalid references, which avoids this problem. Second, hotstarting allows the model to transfer information from the entire dataset to the clean $1 - c$ fraction of the data. Examples that cause a model to hallucinate may still contain valid information about the fluency of a sentence, which hotstarting can capture. This is effectively pretraining our model on the entire data before learning to generate on the clean subset. We have found this procedure to be effective in practice.

Quantile estimation. Second, our algorithm keeps track of the $1 - c$ quantile over the distribution of losses. For each new minibatch B, we update an online estimate of the $1 - c$ quantile (`estimateQuantile`(M, B) in Alg. 1). To estimate this quantile, our algorithm constructs a histogram over the last 10,000 examples seen during training and estimates the empirical $1 - c$ quantile every 10,000 examples.[3]

Loss dropping. Third, our algorithm will perform minibatch stochastic gradient descent while excluding examples that have losses above the current top $1 - c$ quantile estimate q (`truncatedUpdate`(M, B, q) in Alg. 1). Dropping can be accomplished in automatic differentiation packages (e.g., Tensorflow and PyTorch) by setting the loss on the given example to zero.

4.2 Generating High-Probability Samples

Thus far, our goal has been to robustly learn the underlying distribution. However, in some cases, a user may wish to only generate high confidence sequences, which will ideally correspond to high quality sequences.

To generate such samples, we propose *sequence-level rejection sampling*.

Recall that our truncation heuristic selects for the $1 - c$ quantile of the distribution. For a user-defined level α, our rejection sampling scheme will aim to generate samples from the $1 - c \cdot \alpha$ quantile.

To perform rejection sampling, given a model and a user-defined rejection level α, we first sample N sequences (e.g., titles in a summarization task). Then, we sample a random sequence from the $\alpha \cdot N$ smallest samples as measured by log loss. Ideally,

[2] Our code is available at https://github.com/ddkang/loss_dropper.

[3] For datasets with fewer than 10,000 examples, we can perform this procedure over the entire dataset.

Data: Model M, c fraction to drop, T iterations
$M \leftarrow \texttt{hotstart}(M)$;
for $i \leftarrow 0$ **to** T **do**
  $B \leftarrow \texttt{minibatch()}$;
  $\mathrm{q} = \texttt{estimateQuantile}(M, B)$;
  $\mathrm{M} = \texttt{truncatedUpdate}(M, B, q)$;
end

Algorithm 1: The proposed loss truncation procedure with three components (see main text for details for each component).

this procedure will return a sample in the $1 - c \cdot \alpha$ quantile of p_{ref}.

We show that rejection sampling can outperform baselines in generating factual summaries (§5). We further show examples of selected and rejected samples in Appendix A.

5 Evaluation

5.1 Experimental Setup

Dataset and Task. We primarily evaluate loss truncation on abstractive summarization in the form of generating news headlines from an article. We selected this task to highlight that loss truncation can improve sample quality and factual accuracy, while also achieving the secondary goal of diversity for abstractive systems (See et al., 2017; Kryściński et al., 2019).

We evaluated on the Gigaword summarization task (Rush et al., 2017) as in Gehrmann et al. (2018). While there are other summarization datasets, we chose Gigaword for the following reasons. First, it is large enough that sample quality defects are not caused by a lack of data. Second, the dataset is structured so that neither model nor computation is the bottleneck in performance: the standard sequence-to-sequence models are competitive on the Gigaword dataset. Third, while Gigaword dataset is known to have noise, this matches the behavior of existing annotation errors (Beigman and Klebanov, 2009; Klebanov and Beigman, 2010) and uncertainty (Kryściński et al., 2019).

To show that loss truncation is applicable beyond summarization, we also performed a preliminary evaluation of our approach on the E2E NLG task. In E2E, the goal is to generate restaurant reviews from meaning representations (Dušek et al., 2019).

Model and Baselines. We used a standard LSTM architecture with global attention for summarization that has been used for the Gigaword summarization task in the past (Gehrmann et al., 2018). The learning rate and hyperparameters are given in Appendix C. For the E2E task, we use a standard model with the exact settings as in Puzikov and Gurevych (2018).

For loss truncation on Gigaword, we used $c = 0.6$. We matched the total number of training steps when training via loss truncation (including the hotstart) and standard log loss. We sampled from the full model distribution for loss truncated models except when rejection sampling.

As baselines on Gigaword, we generate from the log loss trained language model using several decoders that have been reported to mitigate low-quality outputs such as beam search, top-k sampling (Fan et al., 2018), and top-p sampling (Holtzman et al., 2019). We also evaluate directly sampling from the probabilistic model in order to estimate overall distinguishability and understand the diversity-quality trade-offs of each model.

Finally, on Gigaword, we also compared against a recent generative adversarial network (GAN) model with a publicly available implementation (Wang and Lee, 2018).

Human-evaluation metrics. We evaluate whether loss truncation improves model distinguishability on summarization by measuring the HUSE estimator for TV (Hashimoto et al., 2019). HUSE measures distinguishability by learning a classifier over the log-probabilities and human evaluation scores over both samples from the model and references. We also use HUSE to evaluate the quality-diversity tradeoffs of the models by estimating both HUSE-Q (which measures quality via human judgement) and HUSE-D (which measures diversity via statistical evaluation).

In order to assess whether this leads to improvements in the faithfulness of samples, we measure whether loss truncation reduces the number of *factually inaccurate* outputs from the model via a crowdsourced survey. We designed our prompt based on earlier factual accuracy human evaluation (Novikova et al., 2017) and measured whether the original article contained all of the information given in the generated title.

We describe the crowd worker setup in Appendix D.

Automated metrics. While human evaluation is our primary metric of evaluation as it is considered gold-standard, we additionally evaluate on

	Loss trunc.	Trunc+reject ($\alpha = 0.1$)	Full samp.	Beam	top-k ($k = 100$)	top-p ($p = 0.9$)	GAN
HUSE	**0.58**	0.04	0.55	0.04	0.32	0.32	0.003
HUSE-D	0.88	0.12	**0.98**	0.18	0.59	0.65	0.25
HUSE-Q	0.70	**0.92**	0.58	0.86	0.73	0.67	0.75

Table 2: HUSE, HUSE-D, and HUSE-Q scores for loss truncation and baselines. As shown, loss truncation outperforms all baselines on HUSE score.

automated metrics to contextualize our human evaluation results. We measure ROUGE-L (Lin and Hovy, 2003) for summarization and BLEU score (Papineni et al., 2002) for E2E.

5.2 Loss Truncation Outperforms Baselines on HUSE

Using the HUSE score to measure the TV distance, we assessed whether loss truncation successfully improved our model in terms of distinguishability compared to log loss. As shown in Table 2, loss truncation outperforms all baselines on HUSE score (including the original log loss model `Full samp`), suggesting the truncated model is a better language model than the log loss model as measured by distinguishability.

We find that that loss truncation improves over the log loss by increasing the generation quality (HUSE-Q) by 12% without substantially lowering diversity (e.g., memorizing examples from the training set). These results affirmatively answers an open question posed by Hashimoto et al. (2019) on whether it is possible to obtain models that improve the quality while maintaining overall distinguishability compared to log loss trained models. Post-hoc modification of the log loss model's distribution by removing unlikely words using either top-k or top-p sampling result in substantial losses in HUSE due to losses in diversity.

We further considered matching the entropy of the loss truncation model with top-$k = 100$ and top-$p = 0.9$ (Appendix C). At a fixed entropy, loss truncation can outperform on HUSE by up to 26%.

Comparing models with high sample quality, loss truncation with rejection sampling improves upon all baselines (including beam search) in terms of raw human quality evaluation (HUSE-Q), and we see that the Pareto frontier of truncation and rejection sampling (which can be achieved via ensembling) dominates the baselines on *both* quality and diversity (Figure 5). Rejection sampling decreases overall HUSE score because it is designed to only return high quality samples (i.e., high HUSE-Q): this comes at the cost of reduced diversity, so overall HUSE score suffers.

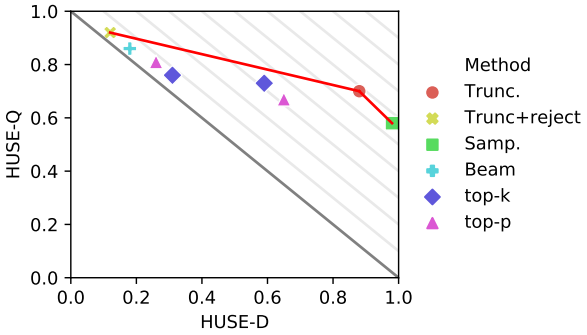


Figure 5: HUSE-D vs HUSE-Q for loss truncation, truncation + rejection sampling, and baselines. The red line shows the best achievable frontier via ensembling. Truncation and rejection outperform all baselines.

The results amongst our baselines recapitulate known results for the quality-diversity tradeoffs of existing methods. Beam search has high sample quality, but low diversity; top-k and top-p samplers provide diversity gains over beam search; and GANs generally underperform well-tuned log loss based models on both diversity and quality.

5.3 Loss Truncation with Rejection Sampling Produces High Quality Outputs

We now ask whether improvements in distinguishability (as measured by HUSE) for the loss truncation model translate to practical improvements in sample quality, such as the factual accuracy of generated outputs in summarization. We evaluate this through a crowdsourced study on factual accuracy.

Since we are interested in studying whether our model can produce high quality samples, we used rejection sampling with $\alpha = 0.1$ to obtain high-quality samples from the model. We compare this to the log loss model with baseline decoders. For the top-p and top-k sampling decoders that have quality-diversity tradeoffs, we select k and p such that the entropy of the sampling distribution matches our rejection sampling approach (see Appendix C for details).

To measure factual accuracy, we asked crowd workers how much information in the generated titles was contained in the article in a similar fashion to Novikova et al. (2017). Table 3 shows the

Condition	Mean score
Human	**3.63** $\pm$ 0.05
Truncation + Rejection ($\alpha = 0.1$)	**3.79** $\pm$ 0.06
Beam	3.51 $\pm$ 0.05
top-p ($p = 0.4$)	3.42 $\pm$ 0.05
top-k ($k = 2$)	3.29 $\pm$ 0.05
Sampling	2.96 $\pm$ 0.05

Table 3: Mean scores and standard errors of factuality in generated news titles given articles. As shown, rejection sampling outperforms all baselines and matches the human reference score.

average factual accuracy rating for each model. We find that rejection sampling outperforms *all* baselines, including the current gold standard of beam search, and matches the human reference level of factual accuracy.

Although it may seem surprising that loss truncation and rejection sampling together can achieve the same factual accuracy score as humans, recall that over 34% of the dataset consists of titles which have facts that are not contained in the article. The loss truncation approach biases the model towards learning only the easily predicted (and likely factually accurate) titles.

5.4 Loss Truncation Produces Diverse Outputs

Finally, one of the benefits of optimizing for distinguishability is that it naturally optimizes for both diversity and quality. Manually examining outputs from the models, we find that directly sampling from the loss truncated model often produces high quality and diverse outputs. We show examples of generated outputs for baselines and loss truncation in Table 4. Loss truncation uses different phrasings ('at least # killed', and 'floods sweep') while top-k follows a nearly templated pattern with a few changes to the words which appear. Top-p and direct sampling both have diverse phrasings, but also hallucinate facts ('earthquake' in sampling and 'torrential rains' in top-p sampling).

5.5 Loss Truncation can Outperform on Automated Metrics

While our primary evaluation metrics are human evaluations (HUSE and factuality), we additionally investigate automated metrics to further contextualize our results. For summarization, we used ROUGE-L and for E2E we use BLEU score for the automated metrics.

For summarization, the ROUGE-L scores for loss truncation and entropy-matched top-k and top-p decoding were 23.2, 22.8, and 22.8 respectively. While loss truncation does not substantially improve ROUGE-L, we see that it still outperforms baselines. We do not expect reference-based evaluations to fully capture the benefits of loss truncation, as these metrics encourage the models to fully imitate the data distribution – including invalid and hallucinated examples.

For E2E, the BLEU scores for loss truncation and the baseline were 0.72 and 0.64 respectively. We confirmed that the baseline model for the E2E task achieves a similar score as reported by Balakrishnan et al. (2019). Perhaps surprisingly, improving BLEU score to 0.72 almost closes the gap to using complex tree-structured semantic representations, which achieves a BLEU score of 0.74 (Balakrishnan et al., 2019).

We further show that loss truncation is not sensitive to the hyperparameter c on automated metrics in Appendix E.1 and provide a preliminary investigation of combining loss truncation and alternative decoders in Appendix E.2.

6 Related Work

Decoder-based diversity. Researchers have proposed a variety of models for text generation (Radford et al., 2019; Keskar et al., 2019; Sutskever et al., 2014). These models generate text using decoding methods such as beam search. While beam search is generally thought of as the gold standard (Tillmann and Ney, 2003), it can produce generic and repetitive outputs (Holtzman et al., 2019). To achieve diversity, top-k (Fan et al., 2018) and top-p (Holtzman et al., 2019) sampling stochastically decodes the outputs after restricting the output space to avoid low-quality outputs.

While these techniques can improve generation quality, they rely on models trained via log loss, which we show can result in undesired behavior that cannot be fixed post-hoc. Our work is complementary to existing work on decoders by proposing a loss that can improve the probabilistic models which these decoders operate on.

Loss modifications. Prior work has identified specific issues in generative models, such as repetitiveness, and proposed loss modifications to address these specific issues in the context of long text generation (Welleck et al., 2019; Holtzman et al., 2018). In contrast, we identify an issue with the widely used log loss, and propose loss truncation, which does not require a task- and issue-specific

Method	Example
Context	at least ## people have been killed and more than ##,### made homeless by floods that swept across southern africa in the past week , striking a region already grappling with severe food shortages .
Gold	floods kill ## in famine-hit southern africa
Loss truncation	at least ## people killed ##,### evacuated in floods in southern african region floods that sweep parts of africa kill at least ##
Beam	flooding hits southern africa as deaths rise
Full sampling	child farming stalls in southern africa earthquake kills ## in southern africa
top-p ($p = 0.9$)	torrential rains prompt warnings in southern africa toll nears ## in southern africa
top-k ($k = 2$)	at least ## killed ##,### homeless in southern africa floods at least ## dead ##,### homeless as floods hit southern africa

Table 4: Examples of generations for various baselines and loss truncation (two replicates shown for sampled outputs). As shown, loss truncation can achieve diverse and high quality outputs. In contrast, baselines either are not diverse (beam, top-k) or poor quality (full sampling, top-p). We color incorrect facts in red.

modification. Many of the penalties and decoding techniques proposed in these earlier works can be combined with truncated log loss to obtain models that are more robust to noisy references.

Contemporaneous with our work, Tian et al. (2019) propose an attention weight approach to improving generation faithfulness via decoder and loss modifications. Our work complements this by providing a conceptual basis for improving faithfulness by ignoring examples (i.e., optimizing distinguishability), and providing a simple and general loss. We consider complex, model dependent loss truncation methods for optimizing distinguishability to be exciting future work.

Other generation methods optimize for task-specific losses (Och, 2003; Shen et al., 2015). Task specific losses are not known in many cases and thus we require an effective task-agnostic loss, e.g., log loss or TV. We show that TV acts as a useful task-agnostic goodness of fit measure, and we provide an improved alternative to log loss.

GANs. GANs have been proposed to learn models that minimize distinguishability (Li et al., 2017; Rajeswar et al., 2017; Dai et al., 2017). While GANs have been successful in generating images (Goodfellow et al., 2014; Brock et al., 2018), GANs remaining challenging to optimize for text due to the discrete nature of text. Our findings match earlier reports that GANs underperform log loss trained sequence-to-sequence models (Caccia et al., 2018). In this work, we show that better training methods for distinguishability can arise from modifying the standard log loss via truncation.

Robust learning. Robust learning is the study of learning in the face of outliers (Tukey, 1960; Donoho, 1982; Huber, 1992). Our work is related to the ϵ-contamination model, in which an ϵ fraction of the data has been modified, potentially by an adversary (Diakonikolas et al., 2018). Our work shows that robust learning under log loss can result in improved empirical performance and bounds on distinguishability.

While there are a number of effective approaches to robust learning (Diakonikolas et al., 2018; Fischler and Bolles, 1981), we focus on a simple truncation procedure as it is one of the only procedures scaleable enough to apply on large-scale generation datasets. Our work shows that more effective, scalable robust learning procedures can help improve natural language generation methods.

7 Conclusion

In this work, we show that log loss is not robust to noise, which can in turn cause undesired behavior, such as hallucinating facts in summarization. In response, we propose loss truncation, a robust training method that optimizes for distinguishability of generated samples. We additionally propose a sequence-level rejection sampling scheme to generate high quality sequences. We show that loss truncation outperforms a range of baselines (including beam search, top-p, top-k, and full sampling) on distinguishability. We additionally show that rejection sampling outperforms all baselines, including beam search, on generating factual summaries. These results suggest that robust learning in the form of truncating the log loss can complement model-based approaches to faithful generation by ignoring invalid and undesired references.

References

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained decoding for neural nlg from compositional representations in task-oriented dialogue. *arXiv preprint arXiv:1906.07220*.

Eyal Beigman and Beata Beigman Klebanov. 2009. Learning with annotation noise. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 280–287. Association for Computational Linguistics.

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. 2018. Language gans falling short. *arXiv preprint arXiv:1811.02549*.

Imre Csiszar and János Körner. 2011. *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press.

Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. 2017. Towards diverse and natural image descriptions via a conditional gan. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2970–2979.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William W Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. *arXiv preprint arXiv:1906.01081*.

Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. 2018. Sever: A robust meta-algorithm for stochastic optimization. *arXiv preprint arXiv:1803.02815*.

David L Donoho, Richard C Liu, et al. 1988. The" automatic" robustness of minimum distance functionals. *The Annals of Statistics*, 16(2):552–586.

DL Donoho. 1982. Breakdown properties of multivariate location estimators. *The Annals of Statistics*.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2019. Evaluating the state-of-the-art of end-to-end natural language generation: The E2E NLG Challenge. *arXiv preprint arXiv:1901.11528*.

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

Martin A Fischler and Robert C Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Michael Gamon, Anthony Aue, and Martine Smets. 2005. Sentence-level mt evaluation without reference translations: Beyond language modeling. In *Proceedings of EAMT*, pages 103–111.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Tatsunori B Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. *North American Chapter of the Association for Computational Linguistics*.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. *arXiv preprint arXiv:1805.06087*.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Peter J Huber. 1992. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Beata Beigman Klebanov and Eyal Beigman. 2010. Some empirical evidence for annotation noise in a benchmarked dataset. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 438–446. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. *arXiv preprint arXiv:1908.08960*.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. Hallucinations in neural machine translation. *Interpretability and*

Robustness in Audio, Speech, and Language Workshop.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 150–157.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for nlg. *arXiv preprint arXiv:1707.06875*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Yevgeniy Puzikov and Iryna Gurevych. 2018. E2e nlg challenge: Neural models vs. templates. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 463–471.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Pal, and Aaron Courville. 2017. Adversarial generation of natural language. *arXiv preprint arXiv:1705.10929*.

Melissa Roemmele. 2016. Writing stories with help from recurrent neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Alexander M Rush, SEAS Harvard, Sumit Chopra, and Jason Weston. 2017. A neural attention model for sentence summarization. In *ACLWeb. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2015. Minimum risk training for neural machine translation. *arXiv preprint arXiv:1512.02433*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in NIPS*.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. 2016. A note on the evaluation of generative models. *ICLR*.

Ran Tian, Shashi Narayan, Thibault Sellam, and Ankur P Parikh. 2019. Sticking to the facts: Confident decoding for faithful data-to-text generation. *arXiv preprint arXiv:1910.08684*.

Christoph Tillmann and Hermann Ney. 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational linguistics*, 29(1):97–133.

John W Tukey. 1960. A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, pages 448–485.

Yau-Shian Wang and Hung-Yi Lee. 2018. Learning to encode text as human-readable summaries using generative adversarial networks. *arXiv preprint arXiv:1810.02851*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in data-to-document generation. *arXiv preprint arXiv:1707.08052*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *arXiv preprint arXiv:1905.12616*.

Sharon Zhou, Mitchell Gordon, Ranjay Krishna, Austin Narcomey, Li F Fei-Fei, and Michael Bernstein. 2019. Hype: A benchmark for human eye perceptual evaluation of generative models. In *Advances in Neural Information Processing Systems*, pages 3444–3456.

Context: Donna Shalala is sporting a mustache to promote public health.
Title: Milk on Her Lip Shalala Raises Eyebrows

(a) Example of a title that requires hallucinating new facts, e.g., "Milk on Her Lip" and "raises eyebrows".

Context: Southwest China's Sichuan province has decided to build an inter-city high-tech industrial belt to serve development of Western China.
Title: Sichuan to Build High-Tech Industrial Belt

(b) Example of a title that can be directly generated from the context.

Figure 6: Examples of titles that require hallucinating new facts and titles that are directly entailed from context.

A Examples of Titles and Generations

Examples of ground truth titles. We present examples of titles in Figure 6 that require factual hallucination and can be directly entailed from context.

Examples of generated titles. We present examples of titles that from rejection sampling that are selected and that were rejected in sampling in Figure 7. As shown, rejected titles tend to be of lower quality.

B Proof of Lemma and Proposition

Lemma. We prove the lemma that all elements in $\mathcal{P}_{c,p}$ are close to p in total variation.

Lemma 1.

$$\sup_{q_0 \in \mathcal{P}_{c,p}} |q_0 - p|_{TV} \leq c$$

Proof. By definition of $\mathcal{P}_{c,p}$, for any q_0 there exists a q_1 such that $p = cq_1 + (1-c)q_0$ so,

$$|q_0 - p|_{\text{TV}} = |cq_0 - cq_1|_{\text{TV}} \leq c$$

□

Proposition. We prove that the truncated log loss bounds total variation.

Context: At least two people have tested positive for the bird flu virus in Eastern Turkey, health minister Recep Akdag told a news conference Wednesday.
Ground truth: Two test positive for bird flu virus in Turkey
Selected sample: Two reported positive for bird flu in Eastern Turkey
Rejected sample: Two officials fail to get good for bird flu in Eastern Turkey

(a) Example 1.

Context: British investment fund Fidelity has increased its stake in Puma, the German maker of sportswear and equipment, to just over five percent, Puma said on Thursday.
Ground truth: Private equity firm Fidelity raises stake in Puma to over five pct
Selected sample: Fidelity increases stake in Puma
Rejected sample: Boost higher first-half stake in Puma says Puma

(b) Example 2.

Figure 7: Examples of sampled titles that were selected and rejected in rejection sampling at $\alpha = 0.1$.

Proof.

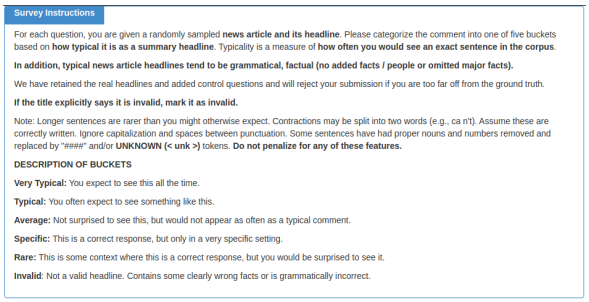
$$|\hat{p} - p_{\text{ref}}|^2_{\text{TV}} \tag{4}$$
$$\leq (|\hat{p} - p_{\text{t}}|_{\text{TV}} + |p_{\text{t}} - p_{\text{ref}}|_{\text{TV}})^2 \tag{5}$$
$$\leq \frac{1}{2}\text{KL}(p_{\text{t}}||\hat{p}) + 2c + c^2 \tag{6}$$

which follows from the triangle inequality, Pinsker's inequality, and using Lemma 1 to bound the remaining terms by c. □

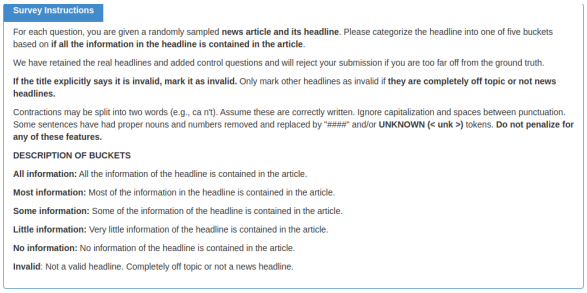
C Hyperparameters

Summarization model hyperparameters. We used a standard OpenNMT-py model with global attention for all sequence-to-sequence experiments (Klein et al., 2017). It has a single LSTM layer in the encoder and two in the decoder.

For the baseline model, we train for 200,000 steps with SGD and an initial learning rate of 1. For the loss truncated model, we hotstart with 100,000 minibatch updates and subsequently with 100,000 minibatch updates with the truncated loss with an initial learning rate of 0.1.



(a) Prompt for measuring HUSE.



(b) Prompt for measuring factuality.

Figure 8: Prompts for measuring HUSE and factuality.

k and p selection. A key parameter in top-k and top-p sampling are k and p respectively. These parameters trade off between diversity and quality. To select these values, we chose values of k and p that had similar entropies to our model trained with loss truncation.

Specifically, $k = 100$ and $p = 0.9$ matched loss truncation at $c = 0.6$ for summarization (entropies of 18.08, 20.01, and 17.93 respectively). $k = 2$ and $p = 0.4$ matched rejection sampling for summarization at $c = 0.6, \alpha = 0.1$ (entropies of 3.71, 4.02, and 3.84 respectively).

D Crowd Worker Setup and Prompts

Crowdsourcing setup. For all human evaluations, we used Amazon Mechanical Turk (all prompts shown below). We sampled 312 context/title pairs to measure HUSE. For each generated title, we asked 9 crowd workers to measure the typicality of the generated title, as in Hashimoto et al. (2019). Each crowd worker responded to 24 generated titles.

For measuring factuality, we sampled 312 examples and for each example, we asked two crowd workers how much information in the generated title was present in the article.

Prompts. We show crowd worker prompts for measuring HUSE and factuality in Figure 8. The HUSE prompt was directly taken from Hashimoto et al. (2019) with an extra control.

Condition	ROUGE-L
Truncation, $c = 0.9$	24.3
Truncation, $c = 0.8$	**24.9**
Truncation, $c = 0.7$	24.0
Truncation, $c = 0.6$	23.2
top-$k = 100$	22.8
top-$p = 0.9$	22.8

Table 5: ROUGE-L scores for loss truncation at various c and entropy-matched top-k and top-p decoding for summarization. As shown, loss truncation outperforms on ROUGE-L for a range of c.

Condition	BLEU
Truncation, $c = 0.9$	**0.72**
Truncation, $c = 0.8$	0.71
Truncation, $c = 0.7$	0.70
Truncation, $c = 0.6$	0.69
Truncation, $c = 0.5$	0.69
Baseline	0.64
0.72	0.64

Table 6: BLEU scores for loss truncation at various c and the baseline model on the E2E task. As shown, loss truncation outperforms the baseline on BLEU score at a range of hyperparameters.

E Further experiments

E.1 Sensitivity to c

We investigate the sensitivity of loss truncation to the hyperparameter c. To do so, we vary c and measure ROUGE-L and BLEU scores, for summarization and E2E respectively.

We show results for summarization in Table 5 and E2E in Table 6 along with baselines. As shown, truncation outperforms on automated metrics on a variety of hyperparameter settings on automated metrics. We leave a full investigation of sensitivity to c as future work.

E.2 Combining Loss Truncation and Decoders

As loss truncation is a training method, it can be combined with alternative methods of decoding at inference time. As such, we perform a preliminary investigation of using top-k and top-p decoding with loss truncation.

We show ROUGE-L of loss truncation combined with various decoders and baselines for summarization in Table 7. As shown, top-k and top-p de-

Condition	ROUGE-L
Log-loss, beam	**41.4**
Log-loss, full sampling	27.9
Truncation, top-$k = 100$	33.4
Truncation, top-$k = 2$	38.9
Truncation, top-$p = 0.9$	35.1
Truncation, top-$p = 0.1$	40.9

Table 7: Loss truncation combined with top-k and top-p decoding.

coding work with loss truncation and can improve sample quality.