A Continuum of Generation Tasks for Investigating Length Bias and Degenerate Repetition

Darcey Riley and **David Chiang**
University of Notre Dame
{darcey.riley,dchiang}@nd.edu

Abstract

Language models suffer from various degenerate behaviors. These differ between tasks: machine translation (MT) exhibits length bias, while tasks like story generation exhibit excessive repetition. Recent work has attributed the difference to *task constrainedness*, but evidence for this claim has always involved many confounding variables. To study this question directly, we introduce a new experimental framework that allows us to smoothly vary task constrainedness, from MT at one end to fully open-ended generation at the other, while keeping all other aspects fixed. We find that: (1) repetition decreases smoothly with constrainedness, explaining the difference in repetition across tasks; (2) length bias surprisingly also decreases with constrainedness, suggesting some other cause for the difference in length bias; (3) across the board, these problems affect the mode, not the whole distribution; (4) the differences cannot be attributed to a change in the entropy of the distribution, since another method of changing the entropy, label smoothing, does not produce the same effect.

1 Introduction

Neural language models serve as the core of modern NLP technologies, but they suffer from "inadequacy of the mode" (Eikema and Aziz, 2020; Zhang et al., 2021), in which the sentences with the very highest probability under the model exhibit various pathological behaviors. Specifically, machine translation suffers from *length bias*, where the generated translations are too long or (more often) too short (Murray and Chiang, 2018; Stahlberg and Byrne, 2019), while story generation suffers from *degenerate repetition*, where the generated text repeats words or phrases unnecessarily (Holtzman et al., 2020).

It has frequently been assumed that length bias and degenerate repetition are both aspects of a single phenomenon; for instance, it is very common for papers studying MT to reference issues observed in story generation. However, MT and story generation exhibit very different problems, and have been addressed using very different solutions. So it is worth pausing for a moment to ask how they relate. Are they truly two symptoms of the same problem? Why do different tasks exhibit different degenerate behaviors?

Stahlberg et al. (2022) and Wiher et al. (2022) attribute the differences to *task constrainedness*: given a particular input, how many different possible correct answers might there be? For example, grammatical error correction (GEC) and speech recognition are more constrained; image captioning and MT are in the middle; and story generation, dialogue, and pure unconditioned generation from the language model (UCG) are least constrained. However, constrainedness is only one of many differences among these tasks. They also differ in the length of the inputs and outputs, the size of the models, and so on. So although these papers provide compelling circumstantial evidence that the differences can be explained in terms of constrainedness, they do not rule out alternative hypotheses.

In this paper, we introduce a new experimental framework which lets us directly adjust constrainedness while keeping everything else (architecture, number of parameters, type of output data) fixed. We expect that, if task constrainedness really is responsible for the differences in degenerate behaviors seen across tasks, then these behaviors should vary smoothly as we adjust constrainedness. We find this to be true for degenerate repetition: we see basically none for pure MT, and an increasing amount as we lower the constrainedness down to UCG. This is consistent with the literature, which reports repetition as a problem in UCG, but not MT.

On the other hand, for length bias, we discover, to our knowledge for the first time, that length bias actually *increases* for less constrained tasks. This is inconsistent with the literature, where length bias is commonly reported for MT but very rarely re-

Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pages 426 - 440
December 8, 2022 ©2022 Association for Computational Linguistics

ported for UCG. We conclude that the difference, then, is either due to some other factor besides constrainedness influencing the model's probability distribution, or that it can be attributed to the different decoding strategies commonly used for the different tasks.

In addition, we present results showing that both length bias and degenerate repetition are problems exclusive to the mode; they do not in general affect random samples from the distribution. Lastly, we explore one possible explanation for why length bias and repetition differ across constrainedness levels: that it is because less constrained tasks have higher entropy. We find that this cannot be the explanation, as another method of increasing the entropy, label smoothing, has very little effect on these phenomena.

2 Related Work

Closely related to our work are two recent papers by Stahlberg et al. (2022) and Wiher et al. (2022), which also explore how degenerate phenomena differ across tasks. Stahlberg et al. (2022) study two more-constrained tasks, MT and GEC. Using exact search and beam search, they find that, for GEC, the distribution is peaked around a few very high-probability outputs, and that, unlike MT, it does not suffer from inadequacy of the mode.

Wiher et al. (2022) study tasks in the same constrainedness range as we do, from MT to UCG. Although their main focus is on evaluating different decoding strategies (where they confirm the trend seen in the literature, that more constrained tasks favor mode-seeking strategies, while less constrained tasks favor sampling-based methods), they look, as we do, at how degenerate repetition and length bias differ across tasks, finding that these phenomena vary across tasks and decoding methods.

Our contribution here is to provide a more rigorous empirical analysis of why these behaviors differ across tasks. Both Stahlberg et al. (2022) and Wiher et al. (2022) attribute the differences they observe to task constrainedness, and Stahlberg et al. (2022) quantify task constrainedness by looking at how much the references differ across a multi-reference test set, but neither is able to directly control task constrainedness while keeping all else fixed. Tor our knowledge, our method is the first to study the effect of task constrainedness on degeneration in a completely controlled way.

3 An Experimental Framework for Controlling Task Constrainedness

The tasks which have been compared before (GEC, MT, story generation, and others) all differ along multiple dimensions besides constrainedness: they use different architectures, different numbers of parameters and amounts of training data, and they produce different length outputs (one sentence for MT, many sentences for story generation), among other distinctions. This makes it difficult to study whether task constrainedness is actually responsible for the differences observed between these tasks. We therefore seek a way of controlling the constrainedness directly, via some sort of "knob" that we could adjust. In this section, we introduce an experimental framework that allows us to do so.

3.1 Truncation

We begin with an ordinary MT dataset and a desired constrainedness level s, which can be 0 (UCG, the least constrained task) or 100 (MT, the most constrained task in our setup) or anything in between. In our experiments, we choose $s = 0, 10, \dots, 100$. For each value of s, we truncate each source sentence in the dataset to s% of its original length. To be precise, if $x = x_1 \cdots x_n \cdot \texttt{EOS}$ is the original sentence (after separating punctuation, but before BPE), we let $n' = \lceil n \cdot s\% \rceil$ and truncate the sentence to $x_1 \cdots x_{n'} \cdot \texttt{EOS}$. See Table 1 for an example German source sentence and all of its truncations.

We apply this truncation to all of the source sentences in the train, dev, and test data, leaving the target sentences intact. This way, as s decreases, the model has to predict the target side given less and less information about what it might contain. Or, to think of it another way, as s decreases, there become more and more possible "correct" answers, since the truncated source sentence could be the prefix of many possible full source sentences, and a translation of any one of them can be considered a valid solution to the task.

3.2 Experimental details

We use the German-to-English (de-en) and Chinese-to-English (zh-en) datasets from IWSLT 2017 (Cettolo et al., 2012), consisting of transcribed TED talks. We use the standard dataset for training, the 2010 development set for development, and the 2010–2015 test sets for testing, following the split by Kulikov et al. (2021). Table 2 shows the size of each of these sets, after removing copy noise (pairs

s (%)	length	tokens
0	0	EOS
10	3	Sch@@ on heute EOS
20	7	Sch@@ on heute spru@@ d@@ elt in EOS
30	8	Sch@@ on heute spru@@ d@@ elt in einigen EOS
40	13	Sch@@ on heute spru@@ d@@ elt in einigen f@@ la@@ chen Se@@ en EOS
50	14	Sch@@ on heute spru@@ d@@ elt in einigen f@@ la@@ chen Se@@ en in EOS
60	19	Sch@@ on heute spru@@ d@@ elt in einigen f@@ la@@ chen Se@@ en in Al@@ as@@ ka Me@@ than EOS
70	21	Sch@@ on heute spru@@ d@@ elt in einigen f@@ la@@ chen Se@@ en in Al@@ as@@ ka Me@@ than von selbst EOS
80	22	Sch@@ on heute spru@@ d@@ elt in einigen f@@ la@@ chen Se@@ en in Al@@ as@@ ka Me@@ than von selbst aus EOS
90	24	Sch@@ on heute spru@@ d@@ elt in einigen f@@ la@@ chen Se@@ en in Al@@ as@@ ka Me@@ than von selbst aus dem Wasser EOS
100	25	Sch@@ on heute spru@@ d@@ elt in einigen f@@ la@@ chen Se@@ en in Al@@ as@@ ka Me@@ than von selbst aus dem Wasser . EOS

Table 1: Prefixes of an example German source sentence, for all values of s. Lengths do not include EOS.

	de-en	zh-en
train	205,898	231,259
dev	888	879
test	8,079	8,549

Table 2: Sizes of our training, development, and test datasets.

where the source and target are identical) from the data (Ott et al., 2018).

We preprocess the data using BPE tokenization (Sennrich et al., 2016). To ensure that the experimental setup is as similar as possible for all values of s, we learn BPE on the full, untruncated dataset. Then, once BPE has been learned, we apply it to the truncated data. Initially, we experimented with both joint and separate BPE, but found very little difference between them, so we present results for joint BPE only.

For our MT system, we use the Transformer model (Vaswani et al., 2017); specifically, we use a fork of the Transformers without Tears library (Nguyen and Salazar, 2019).[1]. We use identical hyperparameter settings for both language pairs and all values of s; these are the same as the Transformers without Tears base configuration, except that we use 6 layers and 4 heads.

We trained our systems both with and without label smoothing (Szegedy et al., 2016), thinking that, because label smoothing changes the shape of the distribution, it might impact the results. We discuss the effect of label smoothing in §6; in all other sections we look only at systems trained without it. All of our results are averaged across three random restarts.

Fearing that BLEU scores might not provide a meaningful enough signal for $s < 100$, we tried using both dev BLEU and dev perplexity to lower the learning rate and control early stopping; these gave very similar results, so we only present results for the systems tuned using dev BLEU.

To better view the natural properties of the distribution, we do not use any length normalization during decoding. We decode up to a maximum length of 300 tokens.

We make our full experimental setup publicly available on GitHub.[2]

3.3 Sanity checks

We verify via BLEU score that we have trained our systems successfully. Although, for the purposes of our experiment, it is not necessary to use a state-of-the-art MT system, we nonetheless achieve reasonable BLEU scores of 34.7 and 17.5 for de-en and zh-en respectively for the $s = 100$ systems, using the standard beam size of 4 for decoding. Predictably, lowering s also decreases the BLEU score, as can be seen in Figure 2.

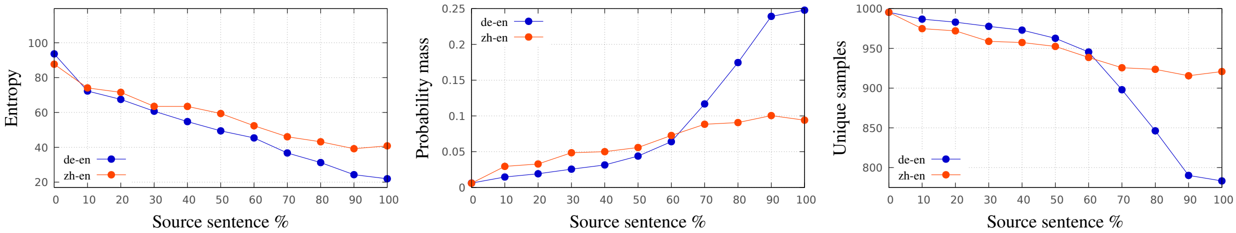
As an additional sanity check, we confirm that varying s does indeed change the spread of the distribution in the expected way. Using 1000 samples for each sentence in the test set, we estimate the entropy (Figure 1a), and find that it decreases as s increases. In addition, following Ott et al. (2018), we look at the portion of the total probability mass covered by all of the unique samples, and find that, although the number of unique samples decreases as s increases (Figure 1c), the total probability mass covered increases (Figure 1b).

4 Degeneracy in the Mode

In this section, we look at how length bias and repetition vary as we vary the constrainedness parameter s.

[1] https://github.com/darcey/transformers_without_tears/tree/mt-interpolation-paper

[2] https://github.com/darcey/mt-interpolation



(a) Per-sentence entropy (nats)

(b) Total probability mass

(c) Number of unique samples

Figure 1: Increasing constrainedness increases the peakedness of the predictive distribution as expected. Every data point is based on 1000 random samples for each sentence in the test data.

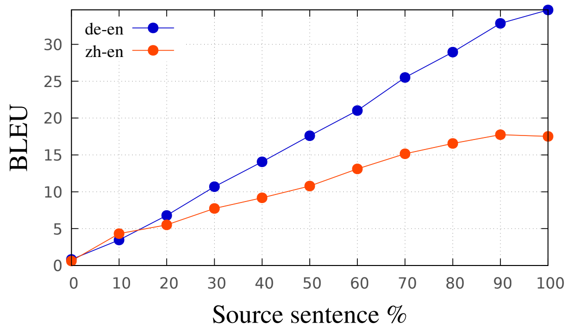
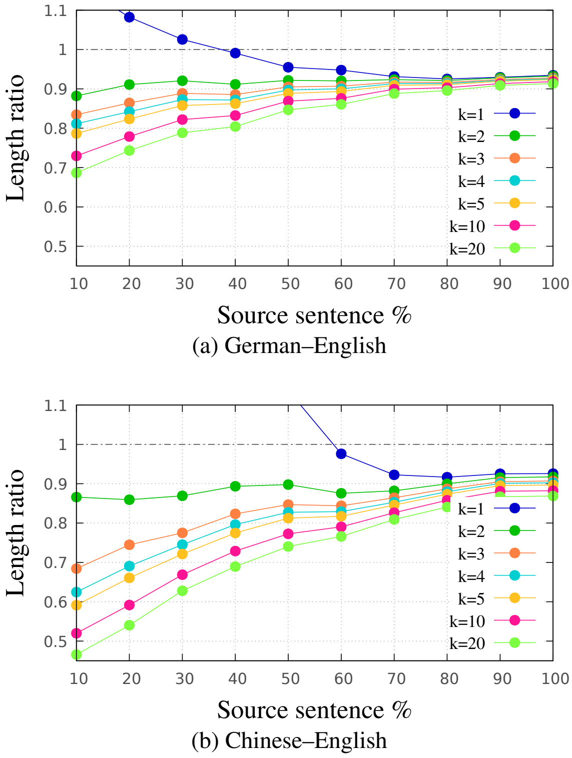


Figure 2: Predictably, BLEU score decreases smoothly as we decrease s.

4.1 Length bias



(a) German–English

(b) Chinese–English

Figure 3: Length ratio versus source sentence percentage (s), for various beam sizes (k). For high s, there is a slight bias towards shorter outputs that increases mildly with k, whereas for low s, we see extreme bias, towards longer or shorter outputs depending on k.

Length bias is a problem where the length of the output consistently differs from the length of the reference; the term typically refers to sentences being too short. In NMT, length bias is such a major and well-known problem that nearly all systems correct for it using some kind of length normalization during decoding (Wu et al., 2016; Koehn and Knowles, 2017; Murray and Chiang, 2018).

In NMT, length bias gets worse the closer one approaches the mode of the distribution. It has been repeatedly shown that, as beam size increases, bringing the output translation closer to the mode, the length bias becomes more extreme. In fact, the mode of the distribution is often simply the empty string itself (Stahlberg and Byrne, 2019).

On the other hand, length bias has been understudied in less constrained tasks such as story generation or UCG. We know of just two reports of this problem: for story generation, Holtzman et al. (2020) report worsening length bias as beam size increases, with immediate stopping when using beam sizes ≥ 64, and Wiher et al. (2022) found length bias for beam sizes $k = 5, 10$; however, neither of these is the main result of their respective papers.

This difference in emphasis seen in the literature would seem to suggest that length bias only affects MT, and does not affect less constrained tasks like story generation. Thus, if constrainedness were fully responsible for the difference seen in the literature, then we would expect to see length bias decrease with constrainedness, becoming less of a problem for less constrained tasks.

To test this, we measure how length bias changes as we vary s. To quantify length bias, we compute the (micro-averaged) *length ratio*,

$$\ell(T) = \frac{\sum_{(h,r)\in T} |h|}{\sum_{(h,r)\in T} |r|}$$

where T is a test set consisting of pairs (h, r), where h is a hypothesis (output) sentence and r is a reference sentence.

Figure 3 shows how length ratio changes as we vary both s and the beam size k.[3] Consistent with previous findings, we find that standard NMT suffers from considerable length bias, with the problem worsening as beam size k increases. But to our surprise, as s decreases, we find that not only does length bias worsen, but that the dependence on beam size grows stronger and stronger. This is surprising given the lack of concern with length bias in the literature on less constrained tasks. To our knowledge, we are the first to report a result like this, where length bias actually worsens as task constrainedness decreases.

We can think of two explanations for this result. The first is that there are other factors besides constrainedness affecting the length bias seen across tasks. We suspect that the length of the reference outputs might be a major part of this; models like GPT-2 are trained to produce much larger chunks of text than our systems, which typically just output one sentence at a time. The second is that this is an artifact of the decoding processes used. Most of the literature on NMT uses mode-seeking decoding strategies such as beam search, while literature on less constrained generation favors sampling-based approaches. So it could in fact be that all unconstrained systems also suffer from length bias, but it simply doesn't show up because beam search is not used with those systems.We also note that it may be more difficult to study length bias in less constrained tasks, since there is not necessarily a roughly "correct" length the way there is in MT.

A last interesting result is that, for $k = 1$ (greedy search), the length ratio actually increases for decreasing s, ending up well above 1. This agrees with a recent result reported by Wiher et al. (2022), who found that, for the relatively unconstrained task of story generation, beam sizes $k = 5, 10$ returned texts that were too short, while greedy search returned texts which were far too long.

4.2 Repetition

Degenerate repetition is a well-known problem where the model gets stuck in a loop, repeating the same n-grams over and over again. It so strongly affects less constrained tasks like story generation (Holtzman et al., 2020) that these tasks avoid mode-seeking strategies altogether, preferring sampling-based approaches. Since pure random sampling also produces low-quality output, most work on this topic has focused on finding a balance between mode-seeking and pure sampling, either by truncating the distribution during sampling (Fan et al., 2018; Holtzman et al., 2020; Basu et al., 2021; Zhang et al., 2021; Nadeem et al., 2020; DeLucia et al., 2021), or by using some combination of sampling and search (Massarelli et al., 2020), though Welleck et al. (2020) address the issue via training rather than search, by modifying the objective function to discourage repetition.

In contrast, to our knowledge, degenerate repetition has not been reported in the literature on NMT. (Our anecdotal experience is that degenerate repetition is a familiar sight in MT, but not a serious problem in well-trained systems.) If the difference between story generation and MT can be explained by their constrainedness, as previous work has suggested, then we should expect to see repetition increase smoothly as we decrease s.

This is, in fact, exactly what we find. Figure 4 shows the amount of repetition for German-to-English, measured as the percentage of unique n-grams which appear in each search result (that is, for each search result, the number of n-gram types divided by tokens), as compared to the reference. (The Chinese-to-English results are similar, and can be found in Appendix B.) We find that, as s decreases, repetition increases considerably. Consistent with the literature, we see basically no evidence of repetition in pure MT, where the amount of repetition almost perfectly matches that seen in the references. But as s decreases, so does the percent of unique n-grams, until for $s = 10$ there is very clear evidence of repetition. We therefore feel confident in concluding that task constrainedness adequately explains the difference in the level of concern paid to repetition in the literature for different tasks.

One interesting thing to observe is that, as beam size increases, repetition actually decreases. We suspect that this might be due to the effect of length bias: as shown in the previous section, higher beam sizes tend to return shorter sentences, and these seem less likely to experience degenerate repetition (though it is certainly possible to have both problems at once).

[3] These graphs (and all of our beam search results) exclude the $s = 0$ case, which turn out to be nearly meaningless: since the source sentence is identical (namely, the empty string) for each sentence pair in the test set, the beam search results will also be identical, meaning that the decoder will simply generate $|T|$ copies of the same sentence. So any properties of that one sentence will be magnified. The results under beam search for $s = 0$ are therefore little better than noise.

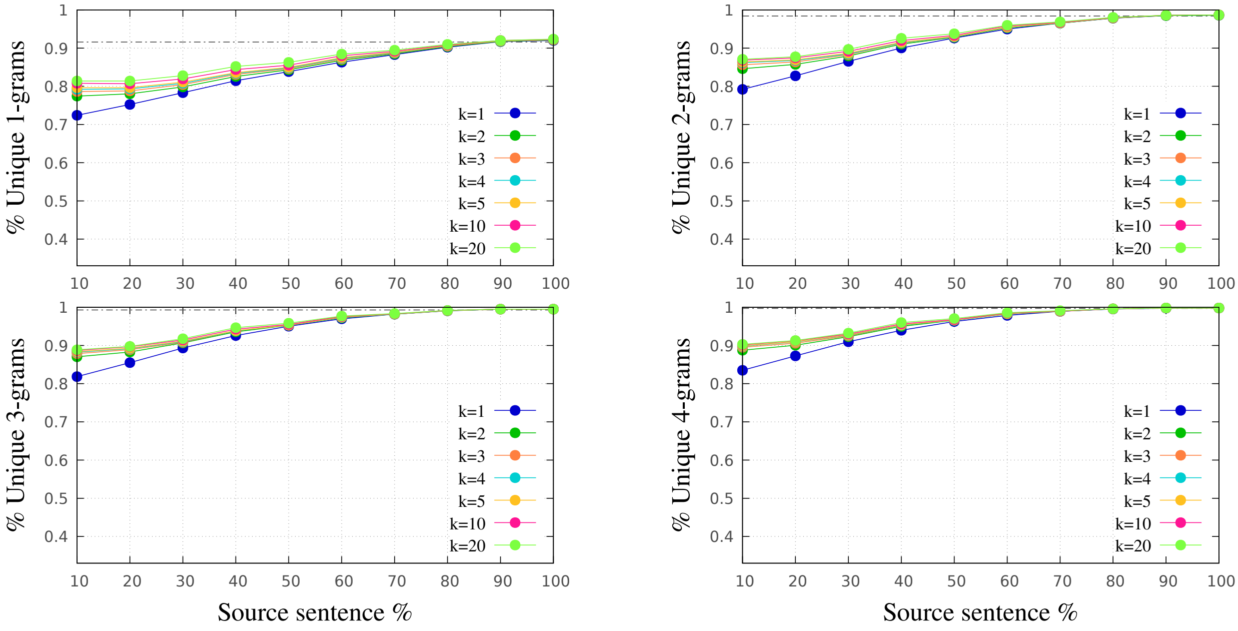


Figure 4: Amount of repetition versus source sentence percentage (s), for various beam sizes (k). Repetition is measured as the percentage of unique n-grams in a sentence; the graphs show this for different values of n. The repetition rate of the reference is plotted as a dashed grey line. Across all values of n, the percent of unique n-grams drops as s decreases. These graphs show German-to-English results only; see Appendix B for Chinese-to-English.

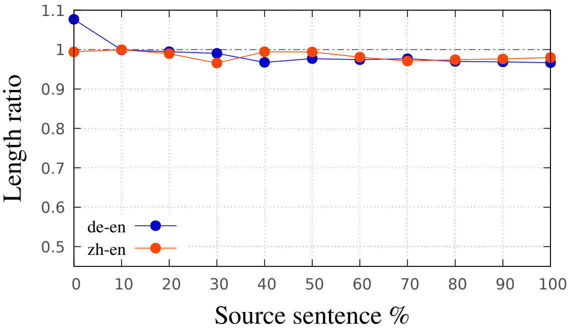


Figure 5: Length ratio of samples versus source sentence percentage (s), for both language pairs. The samples only suffer from very slight length bias, and only for higher values of s.

4.3 Discussion

It is illustrative to examine some strings generated by our systems. Table 3 shows the translations for one sentence from the test data; others can be found in Appendix B. Consistent with our results, we see length roughly decrease along with s. We also see some concrete examples of degenerate repetition for the lower s values. As is typical, the same phrase is repeated over and over, separated by commas or "and".

In addition to length bias and repetition, we can also observe that, as s decreases, the content of the generated strings diverges further and further from the reference. But we notice that, qualitatively, as it does this, the outputs get increasingly boring. This fits with what others have reported (Holtzman et al., 2020), that beam search simply produces tedious and boring output for less constrained tasks.

Another observation is that some of these sentences are simply ungrammatical. While grammatical errors are very common among random samples, it is interesting to see them even at these high probabilities.

5 No Degeneracy in Samples

In addition to looking at search results, we also look at samples from the distribution. For each system and for each sentence in the test set, we take 1000 samples, and discover that the samples do not suffer from either degenerate repetition or length bias (Figs. 5 and 6). This underscores that these problems are specific to the mode, and are not properties of the distribution as a whole.

For low values of s, this should not be particularly surprising; sampling-based decoding approaches such as top-k (Fan et al., 2018) and top-p (Holtzman et al., 2020) are favored for these tasks specifically to avoid the degenerate mode.

Yet it may be surprising to see that the pure MT (s = 100) outputs do not suffer from degeneracy either. Since MT papers rarely explore properties of the full distribution beyond the mode, one might

s (%)	Output found using beam search, k = 4
0	And I said, "Well, I'm going to show you a little bit."
10	When Steve Lopez said, "You know, I'm not going to be here."
20	When Steve Lopez, Columni, who is the first person in the world, he's the first person in the world, and he's the first person in the world.
30	When Steve Lopez, Columni, the Los Angeles Times, he said, "You know, I'm going to go to school."
40	When Steve Lopez, Columnist, the Los Angeles Times, one day, he said, "You know, we're going to have to do this."
50	When Steve Lopez, Columnist, the Los Angeles Times, one day through the Pacific Ocean Ocean, I started to think about it.
60	When Steve Lopez, Columnist in Los Angeles Times, one day through the streets in the center of the city, the city of New York.
70	When Steve Lopez, Columnist, the Los Angeles Times, one day through the streets of Los Angeles, the city of London.
80	When Steve Lopez, Columnist, the Los Angeles Times, one day went through the streets at the center of Los Angeles, I heard this story.
90	When Steve Lopez, Columnist, the Los Angeles Times, one day went through the streets at the center of Los Angeles, he heard a wonderful story.
100	When Steve Lopez, Columnist at the Los Angeles Times, walked through the streets at the center of Los Angeles, he heard a wonderful music.
ref	One day, Los Angeles Times columnist Steve Lopez was walking along the streets of downtown Los Angeles when he heard beautiful music.

Table 3: Beam search (k = 4) outputs for a sentence in the test dataset, shown across all values of s. Illustrates both length bias and degenerate repetition.

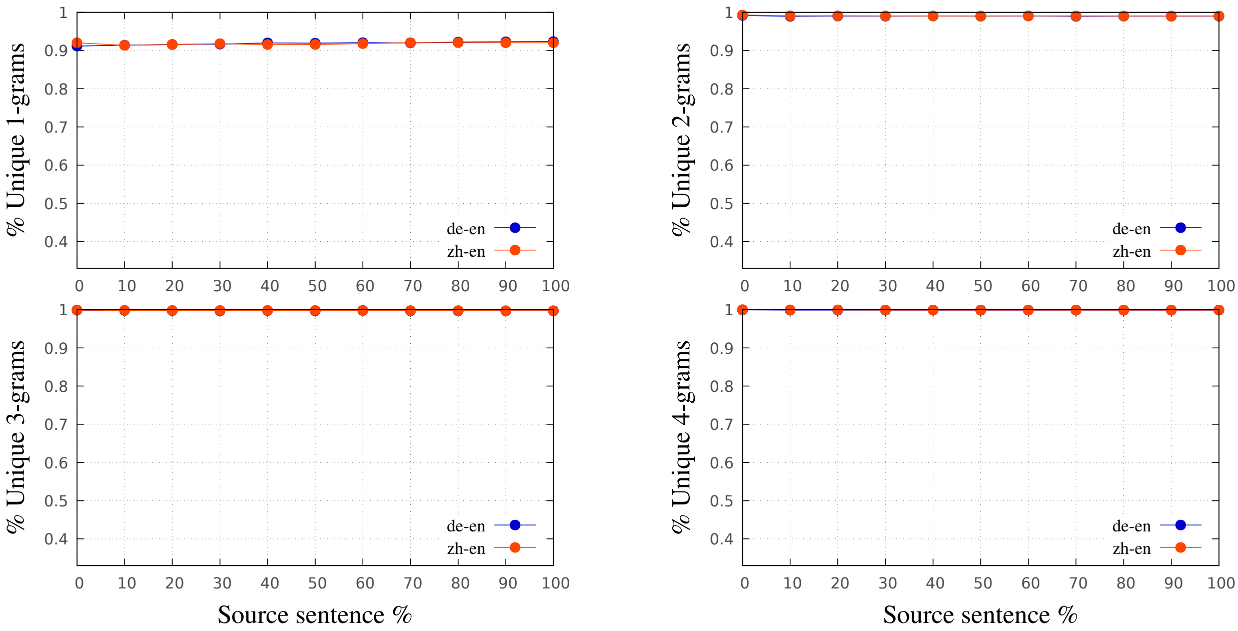


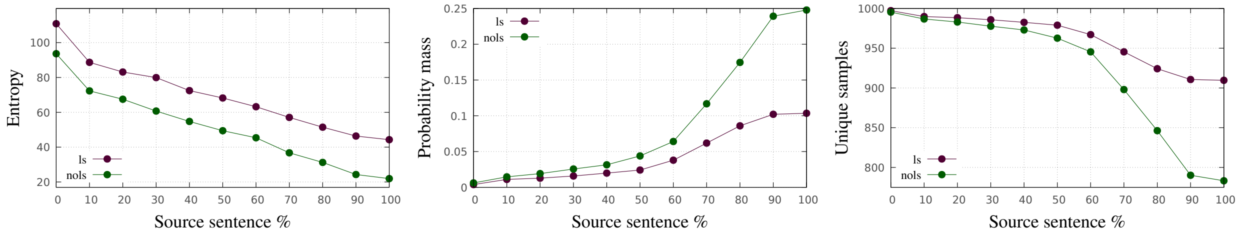
Figure 6: Amount of repetition versus source sentence percentage (s), for various values of n, computed over 1000 random samples for each sentence in the test data. The samples show no evidence of degenerate repetition whatsoever; the level of repetition matches extremely closely to the reference (shown as a dashed grey line which is completely hidden behind the sampling results).

get the false impression that length bias is a problem that affects most sentences in the distribution. Figure 5 shows that this is definitely not the case. This supports the argument by Eikema and Aziz (2020) that it is a mistake to focus too much on the mode during decoding.

6 Label Smoothing and Degeneration

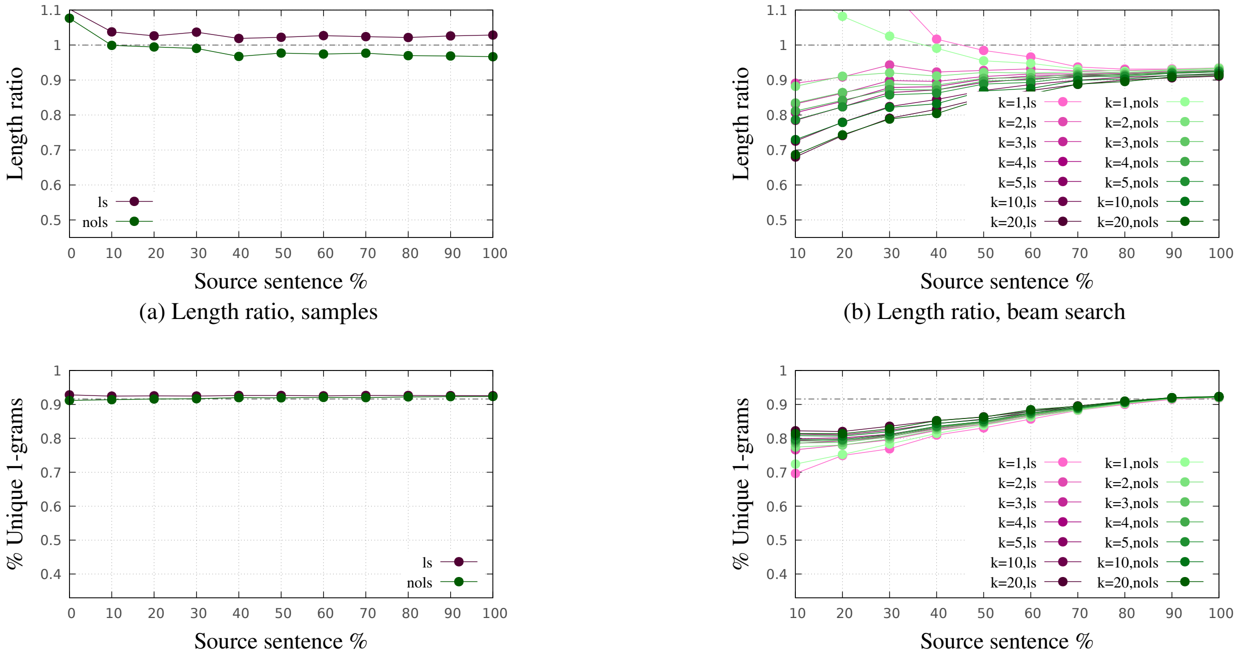
We now begin to examine exactly what it is about task constrainedness which affects the amount of degeneration. One possible explanation is that, as we vary s, we vary the distribution's peakedness: the distribution becomes much less peaked as s decreases (as shown in §3.3). To examine whether differences in peakedness fully explain the level of degeneration, we contrast with a different method of adjusting peakedness: label smoothing. Label smoothing (Szegedy et al., 2016) is an alternative to the standard cross-entropy loss function. Instead of comparing the next-word distribution against a one-hot vector, it compares against a mixture of a one-hot vector and the uniform distribution. It is commonly used in modern NMT systems, and has generally been found to be helpful, though the reasons why are still being investigated (Müller et al., 2019; Lukasik et al., 2020; Gao et al., 2020).

Label smoothing has the effect of smoothing the distribution over more output tokens at each timestep. This has a big effect on the peakedness, as shown in Fig 7. But, as we will show, it has almost no impact on either length bias or repetition. (All the graphs in this section show German-to-



(a) Per-sentence entropy (nats), de-en (b) Total probability mass, de-en (c) Number of unique samples, de-en

Figure 7: Effect of label smoothing (ls) on the peakedness of the distribution, compared with no label smoothing (nols), for German-to-English (see Appendix C for Chinese-to-English). Label smoothing consistently increases entropy and decreases total probability mass across all values of s.



(a) Length ratio, samples

(b) Length ratio, beam search

(c) Repetition, samples

(d) Repetition, beam search

Figure 8: Length ratio of translations and percentage of unique 1-grams versus source sentence percentage (s), both with label smoothing (ls) and without (nols). Results for samples are computed based on 1000 samples for each test sentence; results for beam search vary across beam sizes (k). For samples, label smoothing increases the length ratio from slightly below the reference length to slightly above it; otherwise it has no discernible effect. (These results are for German-to-English; see Appendix C for Chinese-to-English.)

English only; the Chinese-to-English results are similar and can be found in Appendix C.)

The biggest effect we see is in Figure 8a, which shows how adding label smoothing impacts the length bias when sampling. Here, label smoothing changes the length bias from just below 1 to just above 1, giving sentences which are, on average, very slightly longer than the reference.

However, although label smoothing affects length bias for the overall distribution, we see essentially no effect on length bias when using beam search (Figure 8b).[4] Similarly, Figures 8c and 8d show the effect of label smoothing on 1-gram repetition, for both search and sampling; there is essentially no effect. (We found this to be true for other values of n as well.)

From this, we can conclude that it is not merely the spread of the distribution which causes these degenerate behaviors to occur. There must be some other property of task constrainedness which is influencing them. We leave further investigation of what that property might be to future work.

7 Conclusion

We introduced a new experimental framework for directly controlling the level of task constrainedness, by truncating sentences on the source side of an MT system. Using this experimental framework, we analyzed how task constrainedness affected degenerate behaviors.

For less constrained tasks, we observe three failure modes: beam search decoding that is too short, greedy decoding that is too long and repetitive, and random samples that are disfluent. We note that the same three failure modes are also displayed by a simple unigram language model: since every sentence contains EOS, the highest-probability output must be empty (just EOS with no real words); since $P(\texttt{EOS}) < P(\text{the})$, a greedy search will choose *the* over and over; and random samples from a unigram distribution are of course disfluent. So the simplest explanation may be that the neural models used here are still insufficiently sensitive to context.

For more constrained tasks, these effects are much milder. The presence of the source sentence seems to be sufficient to all but eliminate repetition and noticeably improve fluency. Although some work on RNN models for NMT focused on adding coverage models to reduce skipping and repeating of source words (Tu et al., 2016; Mi et al., 2016; Li et al., 2018), Transformers seem to suffer from these problems far less. As Transformers were originally designed for the $s = 100$ case, one direction for future research may be to investigate modifications of the Transformer that are better-suited to less constrained tasks.

[4] We do note, however, that Peters and Martins (2021) did find that label smoothing affected length bias in the mode of the distribution.

Acknowledgements

We thank Toan Nguyen for his help with the `transformers-without-tears` library, and Ari Holtzman for useful discussion.

This material is based upon work supported by the National Science Foundation under Grant No. CCF-2019291. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. 2021. Mirostat: A neural text decoding algorithm that directly controls perplexity. In *International Conference on Learning Representations*.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy. European Association for Machine Translation.

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

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Yingbo Gao, Weiyue Wang, Christian Herold, Zijian Yang, and Hermann Ney. 2020. Towards a better understanding of label smoothing in neural machine translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 212–223, Suzhou, China. Association for Computational Linguistics.

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

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

Ilia Kulikov, Maksim Eremeev, and Kyunghyun Cho. 2021. Characterizing and addressing the issue of oversmoothing in neural autoregressive sequence modeling. arXiv:2112.08914.

Yanyang Li, Tong Xiao, Yinqiao Li, Qiang Wang, Changming Xu, and Jingbo Zhu. 2018. A simple and effective approach to coverage-aware neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 292–297, Melbourne, Australia. Association for Computational Linguistics.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. 2020. Does label smoothing mitigate label noise? In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6448–6458. PMLR.

Luca Massarelli, Fabio Petroni, Aleksandra Piktus, Myle Ott, Tim Rocktäschel, Vassilis Plachouras, Fabrizio Silvestri, and Sebastian Riedel. 2020. How decoding strategies affect the verifiability of generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 223–235, Online. Association for Computational Linguistics.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage embedding models for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 955–960, Austin, Texas. Association for Computational Linguistics.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223, Brussels, Belgium. Association for Computational Linguistics.

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

Toan Q. Nguyen and Julian Salazar. 2019. Transformers without tears: Improving the normalization of self-attention. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong. Association for Computational Linguistics.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *Proceedings of the 35th International Conference on Machine Learning*.

Ben Peters and André F. T. Martins. 2021. Smoothing and shrinking the sparse Seq2Seq search space. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2642–2654, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

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

Felix Stahlberg, Ilia Kulikov, and Shankar Kumar. 2022. Uncertainty determines the adequacy of the mode and the tractability of decoding in sequence-to-sequence models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8634–8645, Dublin, Ireland. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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

Gian Wiher, Clara Meister, and Ryan Cotterell. 2022. On Decoding Strategies for Neural Text Generators. *Transactions of the Association for Computational Linguistics*, 10:997–1012.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv:1609.08144.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. Trading off diversity and quality in natural language generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33, Online. Association for Computational Linguistics.

A Additional results on repetition

Here we show some additional outputs from our systems. Figure 10 graphs the amount of repetition for the Chinese-to-English systems; we see similar results to the German-to-English systems, but with an even more pronounced decrease in repetition for higher beam sizes k.

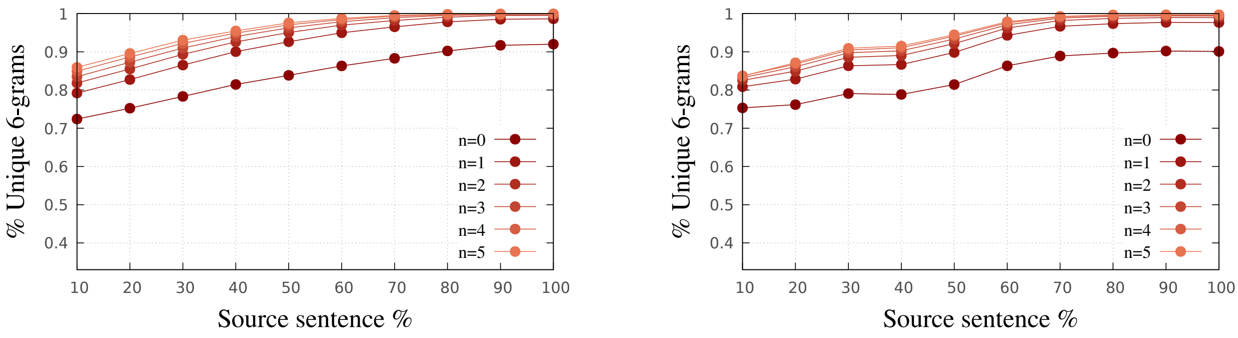
Figure 9 displays the same results, but graphed in terms of n. (We look at beam size $k = 1$ since repetition is most pronounced in that case.) This graph shows a surprising consistency across n; although the effect is most pronounced for 1-gram repetition, we still see quite a bit of degenerate repetition even up to 6-grams, suggesting that the phrases which are being repeated are quite long.

B Additional outputs from our model

As a supplement to Table 3, we present some additional outputs from our system, which show similar trends.

C Additional results on label smoothing

Here we present additional results on label smoothing, for the Chinese-to-English language pair. These results are quite similar to the ones observed for German-to-English. Again, we see a substantial difference in the peakedness of the distribution. And again, we notice a slight change in length ratio for the samples, but otherwise, we observe essentially no effect of label smoothing on degenerate behavior.



(a) German–English

(b) Chinese–English

Figure 9: Amount of repetition, measured as the percentage of n-grams in the sentence which are unique, versus source sentence percentage (s). This is mostly the same information shown in Figures 4 and 10, but viewed in a different way: here, we look at just one beam size ($k = 1$, for which the repetition was most pronounced), and compare multiple n. All values of n show a similar pattern, with considerable repetition observed even for 6-grams for low s.

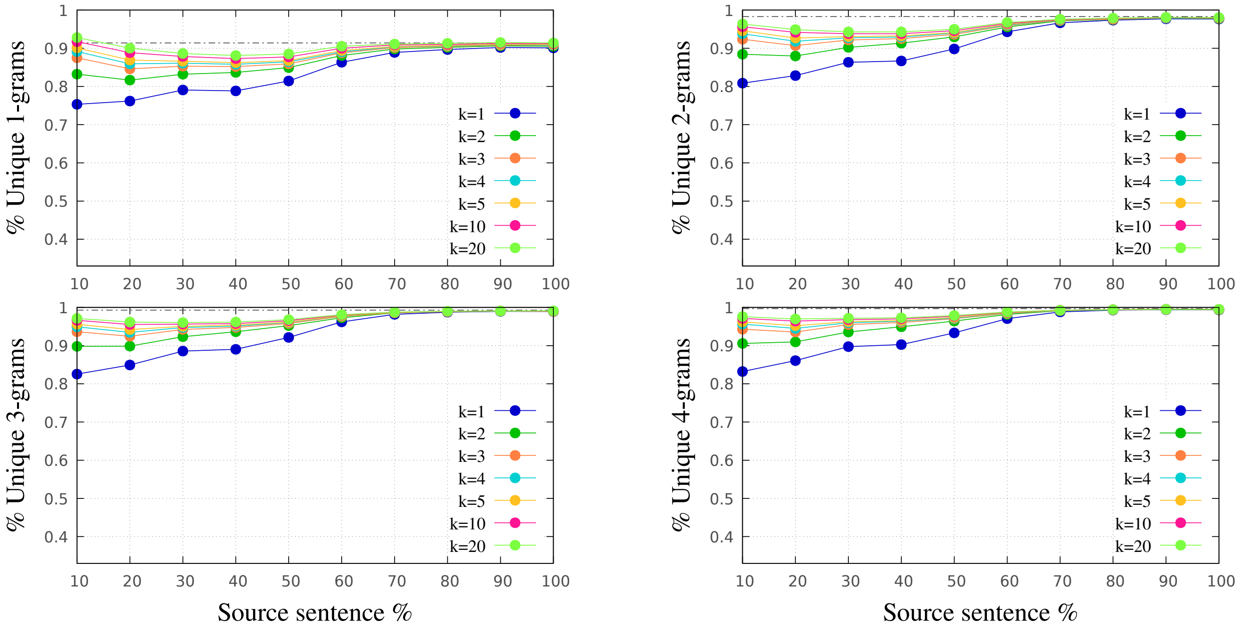


Figure 10: Amount of repetition versus source sentence percentage (s), for various beam sizes (k). (Graphs show Chinese-to-English results only.) Repetition is measured as the percentage of unique n-grams in a sentence; the graphs show this for different values of n. The repetition rate of the reference is plotted as a dashed grey line. As in German-to-English, the percent of unique n-grams drops as s decreases across all values of n, while repetition actually becomes less of a problem for higher values of k.

s (%)	Output found using beam search, $k = 4$
0	And I said, "Well, I'm going to show you a little bit."
10	A few years ago, I was in the hospital, and I was in the hospital.
20	A few years ago, when I was a kid, I was a kid.
30	A few years ago, here at TED, I'm going to tell you a little bit about this.
40	A couple of years ago, at TED, I'm going to tell you a little bit about this.
50	A couple of years ago, at TED, Peter Peter asked me, "What are you doing?"
60	A couple of years ago, here at TED, Peter Skillman introduced a book called "The Sun."
70	A couple of years ago, here at TED, Peter Skillman introduced a design competition called "The House."
80	A few years ago, here at TED, Peter Skillman introduced a design competition called "The Government."
90	A few years ago, here at TED, Peter Skillman made a design competition called "The Marshmallow Child."
100	A few years ago, here at TED, Peter Skillman introduced a design competition called "The Marshmallow Child."
ref	Several years ago here at TED, Peter Skillman introduced a design challenge called the marshmallow challenge.

Table 4: Beam search ($k = 4$) outputs for a sentence in the test dataset, shown across all values of s.

s (%)	Output found using beam search, $k = 4$
0	And I said, "Well, I'm going to show you a little bit."
10	A child, a child, a child, a child, a child, a child, a child, a child.
20	A child who is living in the world today is a child, a child, a child, a child, a child, a child, a child.
30	A child who's born in New Delhi today will be born in a new world, a new world, a new world, a new world.
40	A child born today in New Delhi can expect to be a child who has been born in the United States.
50	A child who's born in New Delhi today can expect to be as long as they're born, and that's where they are.
60	A kid who can be born in New Delhi today would expect to live as long as they were, and that's what they were doing.
70	A child born in New Delhi today will expect to live as long as the richest child in the world.
80	A child born in New Delhi today will expect to live as long as the richest man on the planet.
90	A child born today in New Delhi can expect to live as long as the richest man in the world, 100 years ago.
100	A child born today in New Delhi can expect to live as long as the richest man in the world 100 years ago.
ref	A kid born in New Delhi today can expect to live as long as the richest man in the world did 100 years ago.

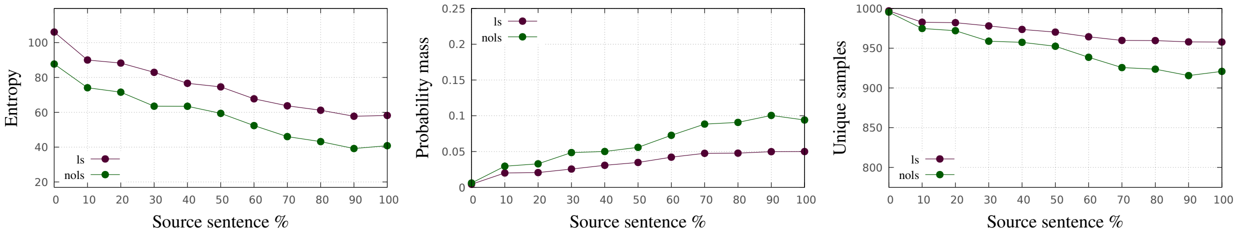
Table 5: Beam search ($k = 4$) outputs for a sentence in the test dataset, shown across all values of s.

s (%)	Output found using beam search, $k = 4$
0	And I said, "Well, I'm going to show you a little bit."
10	Today, I'm going to show you a couple of examples of how this works.
20	Today, I don't know how many of you have heard of this, but I think it's a very important thing.
30	Today, I don't know, maybe it's going to go up today, but I'm going to tell you a little bit about what I'm going to talk about today.
40	Now, I don't know, maybe someone in this room came to me and said, "You know, I'm going to go to school."
50	Today, I don't know, maybe someone in Lagos came to a place where I was born, and I didn't know what to do.
60	Today, I don't know, maybe somebody in Lagos came into an airplane this morning, and it's going to be like, "Oh, I'm going to die."
70	Now, I don't know, maybe someone in Lagos this morning went into an airplane, and it's going to go to Los Angeles, and it's going to be a big deal.
80	Today, I don't know, maybe someone in Lagos came to an airplane this morning, and it's just going to Los Angeles, in the middle of the Central Sea.
90	Today, I don't know, maybe someone in Lagos is going to go to an airplane this morning, and he's going to go to Los Angeles, right now he's gone.
100	Today, I don't know, maybe someone in Lagos climbed into an airplane this morning, and it's just going to Los Angeles right now, it's over Ohio.
ref	Today, I don't know, maybe a guy got on a plane in Lagos this morning, and he's flying to LAX, right now he's over Ohio.

Table 6: Beam search ($k = 4$) outputs for a sentence in the test dataset, shown across all values of s.

s (%)	Output found using beam search, $k = 4$
0	And I said, "Well, I'm going to show you a little bit."
10	If you look at it, you can see that it's a little bit different.
20	If you're 10 teams, you're going to have to be able to do that.
30	If you have 10 teams, you have 10 teams, and you have 10 teams, and you have them.
40	If you have 10 teams, typically, you have 10 teams, and you have 10 teams.
50	If you have 10 teams that are typically predicting, you're not going to be able to do that.
60	If you have 10 teams that are typically predicted, you get 10 teams, and you get 10 teams.
70	If you have 10 teams that typically go, you get about six teams per second.
80	If you have 10 teams that typically go ahead, you get about six, the two teams.
90	If you have 10 teams that are typical, you get about six, the stable structures.
100	If you have 10 teams that go typically, you get about six that have stable structures.
ref	If you have 10 teams that typically perform, you'll get maybe six or so that have standing structures.

Table 7: Beam search ($k = 4$) outputs for a sentence in the test dataset, shown across all values of s.

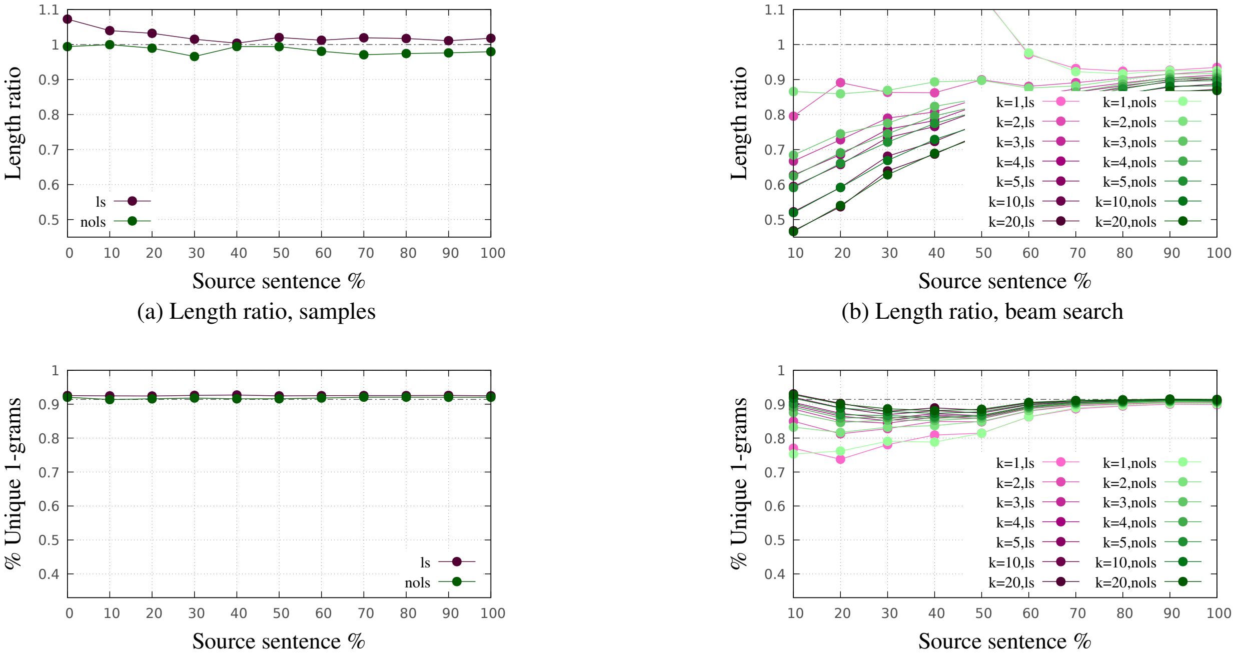


(a) Per-sentence entropy (nats), zh-en

(b) Total probability mass, zh-en

(c) Number of unique samples, zh-en

Figure 11: Effect of label smoothing (ls) on the peakedness of the distribution, compared with no label smoothing (nols), for Chinese-to-English. As with German-to-English, label smoothing consistently increases entropy and decreases total probability mass across all values of s.



(a) Length ratio, samples

(b) Length ratio, beam search

(c) Repetition, samples

(d) Repetition, beam search

Figure 12: Length ratio of translations and percentage of unique 1-grams versus source sentence percentage (s), both with label smoothing (ls) and without label smoothing (nols). Results for samples are computed based on 1000 samples for each test sentence; results for beam search vary across beam sizes (k). As with the German-to-English results, we find that, for samples, label smoothing increases the length ratio from slightly below the reference length to slightly above it; otherwise it has no discernable effect.