Characterizing and addressing the issue of oversmoothing in neural autoregressive sequence modeling

Ilia Kulikov*
New York University
kulikov@cs.nyu.edu

Maksim Eremeev*
New York University
eremeev@nyu.edu

Kyunghyun Cho
New York University
Genentech
CIFAR Fellow in LMB

Abstract

Neural autoregressive sequence models smear the probability among many possible sequences including degenerate ones, such as empty or repetitive sequences. In this work, we tackle one specific case where the model assigns a high probability to unreasonably short sequences. We define the oversmoothing rate to quantify this issue. After confirming the high degree of oversmoothing in neural machine translation, we propose to explicitly minimize the oversmoothing rate during training. We conduct a set of experiments to study the effect of the proposed regularization on both model distribution and decoding performance. We use a neural machine translation task as the testbed and consider three different datasets of varying size. Our experiments reveal three major findings. First, we can control the oversmoothing rate of the model by tuning the strength of the regularization. Second, by enhancing the oversmoothing loss contribution, the probability and the rank of $\langle \text{eos} \rangle$ token decrease heavily at positions where it is not supposed to be. Third, the proposed regularization impacts the outcome of beam search especially when a large beam is used. The degradation of translation quality (measured in BLEU) with a large beam significantly lessens with lower oversmoothing rate, but the degradation compared to smaller beam sizes remains to exist. From these observations, we conclude that the high degree of oversmoothing is the main reason behind the degenerate case of overly probable short sequences in a neural autoregressive model.

1 Introduction

Neural autoregressive sequence modeling is a widely used scheme for conditional text generation. It is applied to many NLP tasks, including machine translation, language modeling, and conversation modeling (Cho et al., 2014; Sutskever et al., 2014; Brown et al., 2020; Roller et al., 2021). Despite the substantial success, major issues still exist, and it is still an active area of research. Here we highlight two major issues which have been discussed extensively.

The first issue is the model assigning too high a probability to a sequence which is unreasonably shorter than a ground-truth sequence. Stahlberg and Byrne (2019) report evidence of an extreme case where the model frequently assigns the highest probability to an empty sequence given a source sequence in machine translation. In addition, Koehn and Knowles (2017) demonstrate that the length of generated translation gets shorter with better decoding (i.e., beam search with a larger beam.)

In the second issue, which is more often observed in open-ended sequence generation tasks, such as sequence completion, generated sequences often contain unreasonably many repetitions (Holtzman et al., 2019; Welleck et al., 2020b). This phenomenon was partly explained in a recent year by Welleck et al. (2020a), as approximate decoding resulting in an infinitely long, zero-probability sequence.

In this work, we tackle the first issue where the model prefers overly short sequences compared to longer, often more correct ones. We assume that any prefix substring of a ground-truth sequence is an unreasonably short sequence and call such a prefix as a premature sequence. This definition allows us to calculate how often an unreasonably short sequence receives a higher probability than the original, full sequence does. This value quantifies the degree to which the probability mass is oversmoothed toward shorter sequences. We call this quantity an *oversmoothing rate*. We empirically verify that publicly available, well-trained translation models exhibit high oversmoothing rates.

We propose to minimize the oversmoothing rate during training together with the negative log-likelihood objective. Since the oversmoothing rate

*Equal contribution.

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1115–1124
November 20–23, 2022.

is difficult to minimize directly due to its construction as the average of indicator functions, we design its convex relaxation, to which we refer as an *oversmoothing loss*. This loss is easier to use with gradient-based learning.

We apply the proposed regularization to neural machine translation using IWSLT'17 and WMT tasks and observe promising findings. We effectively reduce the oversmoothing rate by minimizing the proposed oversmoothing loss across all tasks we consider. We see the narrowing gap between the length distribution of generated sequences and that of the reference sequences, even when we increase the beam size, with a lower oversmoothing rate. Finally, by choosing the strength of the proposed regularization appropriately, we improve the translation quality when decoding with large beam sizes. We could not, however, observe a similar improvement with a small beam size.

2 Background: Neural autoregressive sequence modeling

We study how a neural sequence model assigns too high probability to unreasonably short sequences due to its design and training objective. We do so in the context of machine translation in which the goal is to model a conditional distribution over a target language given a source sentence. More specifically, we consider a standard approach of autoregressive neural sequence modeling for this task of neural machine translation, where the conditional probability of a target sentence given a source sentence is written down as:[1]

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{|\mathbf{y}|} p(y_t|y_{<t}, \mathbf{x}; \theta), \tag{1}$$

where $y_{<t}$ is a sequence of tokens up to (and not including) step t. θ refers to the parameters of an underlying neural network that computes the conditional probability. Each of the source and target sentences ends with a special $\langle\text{eos}\rangle$ token indicating the end of the sequence. As was demonstrated by Newman et al. (2020), this $\langle\text{eos}\rangle$ token is used by an autoregressive neural network to model the length of a sequence.

Given this parameterization, we assume a standard practice of maximum likelihood learning which estimates the parameters θ that maximizes the following objective function:

$$L(\theta) = \frac{1}{|D|} \sum_{n=1}^{N} \log p(\mathbf{y}^n|\mathbf{x}^n; \theta) + \mathcal{R}(\theta).$$

$\mathcal{R}$ is a regularization term that prevents overfitting, such as weight decay.

Once training is done, we use this autoregressive model as a translation system by approximately solving the following optimization problem:

$$\hat{\mathbf{y}}_{\text{map}} = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}; \theta).$$

We often resort to greedy decoding or beam search, both of which belong to a family of incomplete decoding algorithms (Welleck et al., 2020a).

3 Oversmoothing: the issue of premature sequences

In this section, we carefully describe the issue of premature translation or premature sequence in autoregressive modeling, which has more often been referred to casually as the issue of oversmoothing in earlier studies (see, e.g., Shi et al., 2020). To do so, we first define formally what we mean by a 'premature sequence'. A premature sequence is a length-t prefix of an original sequence, where t is smaller than the length of the original sequence. In other words, length-t prefix is defined as:

Definition 3.1 (Length-t prefix)**.** Given an original sequence $\mathbf{y} = (y_1, y_2, \ldots, y_{|\mathbf{y}|} = \langle\text{eos}\rangle)$, the length-$t$ prefix is $\mathbf{y}_{\leq t} = (y_1, y_2, \ldots, y_{t-1}, \langle\text{eos}\rangle)$, where $1 \leq t < |\mathbf{y}|$.

With this definition, we make a reasonable assumption that most of such premature sequences are not valid sequences on their own. In the case of natural language processing, for instance, these premature sequences correspond to sentences that suddenly terminate in the middle. Only a few of these premature sequences may be a coherent, well-formed text.

A good autoregressive language model should then assign a lower probability to such an ill-formed premature sequence than that assigned to a well-formed original sequence. That is, it must satisfy:

$$\underbrace{\prod_{t'=1}^{|\mathbf{y}|} p(y_{t'}|y_{<t'})}_{=p(\mathbf{y})} > \underbrace{p(\langle\text{eos}\rangle\,|y_{<t}) \prod_{t'=1}^{t-1} p(y_{t'}|y_{<t'})}_{=p(\mathbf{y}_{\leq t})} \tag{2}$$

[1] In the rest of the paper, we often omit X for brevity.

which is equivalent to

$$\prod_{t'=t}^{|\mathbf{y}|} p(y_{t'}|y_{<t'}) > p(\langle \text{eos} \rangle | y_{<t}),$$

because of the autoregressive formulation.

In order for this inequality to hold, the probability assigned to the $\langle \text{eos} \rangle$ must be extremely small, as the left-hand side of the inequality is the product of many probabilities. In other words, the dynamic range of the $\langle \text{eos} \rangle$ token probability must be significantly greater than that of any other token probability, in order for the autoregressive language model to properly capture the ill-formed nature of premature sequences.

It is, however, a usual practice to treat the $\langle \text{eos} \rangle$ token just like any other token in the vocabulary, which is evident from Eq. (1). This leads to the difficulty in having a dramatically larger dynamic range for the $\langle \text{eos} \rangle$ probability than for other token probabilities. In other words, this limited dynamic range due to the lack of special treatment of $\langle \text{eos} \rangle$ is what previous studies (Shi et al., 2020) have referred to as "oversmoothing", and this leads to the degeneracy in length modeling.

Under this observation, we can now quantify the degree of oversmoothing[2] by examining how often the inequality in Eq. (2) is violated:

Definition 3.2 (Oversmoothing rate)**.** The oversmoothing rate of a sequence is defined as

$$r_{\text{os}}(\mathbf{y}) = \frac{1}{|\mathbf{y}|-1} \sum_{t=1}^{|\mathbf{y}|-1} \mathbb{1}\Big(\prod_{t'=t}^{|\mathbf{y}|} p(y_{t'}|y_{<t'}) < p(\langle \text{eos} \rangle | y_{<t}) \Big), \quad (3)$$

where $\mathbb{1}$ is an indicator function returning 1 if true and otherwise 0.

With this definition, we can now quantify the degree of oversmoothing and thereby quantify any improvement in terms of the issue of oversmoothing by any future proposal, including our own in this paper.

Because premature sequences may be well-formed, it is not desirable for the oversmoothing rate to reach 0. We, however, demonstrate later empirically that this oversmoothing rate is too high for every system we considered in this work.

[2]To be strict, this should be called the degree of 'smoothing', but we stick to oversmoothing to be in line with how this phenomenon has been referred to in previous studies (Shi et al., 2020).

3.1 Minimizing the oversmoothing rate

The oversmoothing rate above is defined as the average of indicator functions, making it challenging to directly minimize. We instead propose to minimize an upper bound on the original oversmoothing rate, that is differentiable almost everywhere and admits gradient-based optimization:

Definition 3.3 (Oversmoothing loss)**.** Given a sequence $\mathbf{y}$, the oversmoothing loss is defined as

$$l_{\text{os}}(\mathbf{y}) = \frac{1}{|\mathbf{y}|} \sum_{t=1}^{|\mathbf{y}|} \max\Bigg(0, \log p(\langle \text{eos} \rangle | y_{<t}) - \sum_{t'=t}^{|\mathbf{y}|} \log p(y_{t'}|y_{<t'}) + m\Bigg),$$

which is an upper bound of $r_{\text{os}}(y)$ with $m \geq 1$.

We use this oversmoothing loss as a regularization term and augment the original objective function with it. We use $\alpha \in [0, 1)$ to balance the relative strengths of these two terms:

$$l(\mathbf{y}) = (1-\alpha) \cdot l_{\text{nll}}(\mathbf{y}) + \alpha \cdot l_{\text{os}}(\mathbf{y}),$$

where

$$l_{\text{nll}}(\mathbf{y}) = -\sum_{t=1}^{|\mathbf{y}|} \log p(y_t|y_{<t}).$$

When the inequality in Eq. (2) is satisfied at step t with the log-probability difference between the l.h.s. and r.h.s. at least as large as m, the oversmoothing loss disappears, implying that the step t does not contribute to the issue of oversmoothing. When this loss is activated at step t, we have two terms, excluding the constant margin m, the log-probability of *incorrect* $\langle \text{eos} \rangle$ given the context $y_{<t}$ and the negative log-probability of the *correct* suffix given the same context.

Minimizing the first term explicitly prevents a premature sequence $\mathbf{y}_{\leq t}$ from being a valid sequence by lowering the probability y_t being $\langle \text{eos} \rangle$ even further compared to the other tokens in the vocabulary. The second term on the other hand prevents the premature sequence by ensuring that the full sequence $\mathbf{y} = (y_{<|y|}, \langle \text{eos} \rangle)$ is more likely than the premature sequence $\mathbf{y}_{\leq t} = (y_{<t}, \langle \text{eos} \rangle)$. In short, the proposed oversmoothing loss addresses both of these scenarios which lead to oversmoothing. Finally, only when both of these factors are suppressed enough, the loss vanishes.

The second scenario above, i.e., increasing the probability of a suffix at each position t, has the effect of greatly emphasizing the latter part of the sequence during training. This can lead to a degenerate case in which the earlier part of a sequence cannot be modeled by an autoregressive sequence modeling, if the strength of the proposed oversmoothing loss is too large. We thus use this loss together with the original negative log-likelihood loss ($\alpha > 0$) only after pretraining a model with the negative log-likelihood loss only ($\alpha = 0$).

4 Related work

The issue of generating sequences that are shorter than the ground-truth one has been studied from various aspects including model parametrization, data collection, and decoding. Here we highlight some of these projects in the context of our work.

On the aspect of model parametrization, Peters and Martins (2021) suggest using sparse transformation of the next-token distribution rather than the usual way of using softmax. Such a model is then able to assign zero probability to short sequences more readily and thereby reduce the oversmoothing rate. Their approach, however, does not explicitly encourage $\langle\text{eos}\rangle$ tokens to be assigned zero probability, unlike ours where $\langle\text{eos}\rangle$ is treated specially. Shi et al. (2020) embed the $\langle\text{eos}\rangle$ token with a distinct vector at each position within the sequence. This was shown to help the probability of empty sequence, although they do not report its impact on translation quality at all.

On data collection, Nguyen et al. (2021) analyze data collection and show that data augmentation techniques altering sequence length may address the issue of oversmoothing and improve translation quality. Their work is however limited to low-resource tasks. With respect to decoding, Wang et al. (2020) observe the oversmoothing while studying "look-ahead" decoding strategies. They reduce the probability of the $\langle\text{eos}\rangle$ using the auxiliary loss term, similarly to the token-level unlikelihood loss (Welleck et al., 2020b). Murray and Chiang (2018) design a decoding algorithm that learns to correct the underestimated length. Alternative decoding algorithms, such as minimum Bayes risk decoding (Eikema and Aziz, 2020; Müller and Sennrich, 2021), have been shown to alleviate the length mismatch to a certain extent when compared to beam search.

These earlier approaches do not attempt at formally characterizing the cause behind the issue of oversmoothing. This is unlike our work, where we start by formalizing the issue of oversmoothing and propose a way to alleviate this issue by directly addressing this cause.

5 Experimental Setup

We follow a standard practice to train our neural machine translation models, following (Ott et al., 2018a), using the FairSeq framework (Ott et al., 2019). We use BPE tokenization via either fastBPE (Sennrich et al., 2016) or SentencePiece (Kudo and Richardson, 2018), depending on the dataset. Although it is not required for us to use state-of-the-art models to study the issue of oversmoothing, we use models that achieve reasonable translation quality. The code implementing FairSeq task with the oversmoothing rate metric, oversmoothing loss, and experimental results is available on Github.[3]

5.1 Tasks and Models

We experiment with both smaller datasets using language pairs from IWSLT'17 and larger datasets using language pairs from WMT'19 and WMT'16. In the latter case, we use publicly available pretrained checkpoints in FairSeq. We execute five training runs with different random initialization for every system. These language pairs and checkpoints cover different combinations of languages and model sizes. This allows us to study the oversmoothing rate under a variety of different settings.

IWSLT'17 {De,Fr,Zh}→En: We adapt the data preprocessing procedure from FairSeq IWSLT recipe and use SentencePiece tokenization. The training sets consist of 209K, 236K, and 235K sentence pairs for De→En, Fr→En, and Zh→En, respectively. We use the TED talks 2010 development set for validation, and the TED talks 2010-2015 test set for testing. The development and test sets, respectively, consist of approximately 800 and 8,000 sentence pairs for all tasks.

We use the same architecture named `transformer_iwslt_de_en` in FairSeq for each language pair. It consists of 6 encoder and decoder layers with 4 self-attention heads followed by feed-forward transformations. Both encoder and decoder use embeddings of size 512 while the input and output embeddings are not shared. Both the encoder and decoder use learned positional

[3] https://github.com/uralik/oversmoothing_rate

embedding. We early-stopping training based on the validation set. Evaluation is done on the test set.

WMT'16 En→De: We prepare the data following the recipe from FairSeq Github. The training set has 4.5M sentence pairs. Following Ott et al. (2018b), we use newstest13 as the development set and newstest14 as the test set, they contain 3K sentence pairs each. We fine-tune the pretrained checkpoint which was originally released by (Ott et al., 2018b) and is available from FairSeq. The recipe uses a transformer architecture based on (Vaswani et al., 2017). Different from all other models considered in this work, this architecture shares vocabulary embeddings between the encoder and the decoder.

WMT'19 Ru→En, De↔En We closely follow Ng et al. (2019) in preparing data, except for filtering based on language identification. We use the subset of WMT'19 training set consisting of news commentary v12 and common crawl resulting in slightly more than 1M and 2M training sentence pairs for Ru→En and De↔En pairs, respectively. We fine-tuned single model checkpoints from Ng et al. (2019). We early-stop training on the official WMT'19 development set. For evaluation, we use the official WMT'19 test set.

5.2 Training

We use Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. We use the inverse square root learning scheduler with 4,000 warm-up steps. We use the initial learning rate of 5×10^{-4}, dropout rate of 0.3 (Srivastava et al., 2014) , and weight decay with its rate set to 10^{-4}. We use label smoothing with 0.1 of probability smoothed uniformly during pretraining with NLL loss and turn it off after starting to use the oversmoothing loss. We vary the oversmoothing loss weight α from 0.0 to 0.95 with a step size of 0.05. We use a fixed margin $m = 10^{-4}$ whenever we use the oversmoothing loss.

Early stopping We use early stopping for model selection based on the value of the objective function computed on the development set. We evaluate the model on the development set every 2K updates for IWSLT (∼2K tokens per update) and WMT (∼9K tokens per update) systems. We stop training when the objective has not improved over more 5 consecutive validation runs. We fine-tune models

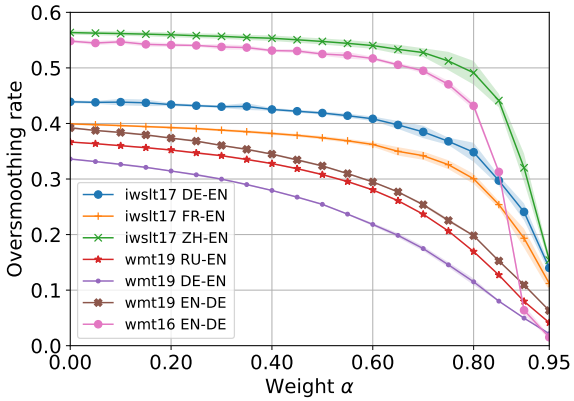


Figure 1: Average oversmoothing rate is going down as we increase contribution of the oversmoothing loss during fine-tuning. Filled regions denote the standard deviation across training runs according to Section 5.

around 5K updates for IWSLT'17 DE-EN and ZH-EN, and 7K updates for IWSLT'17 FR-EN. As for WMT'19, it takes approximately 45K updates for DE-EN and EN-DE language pairs to early-stop, and 76K updates for RU-EN model, and 12K updates for WMT'16. Alternative methods for model selection such as checkpoint averaging or moving-averaged parameter set are applicable here as well and we leave experimenting with it for future work.

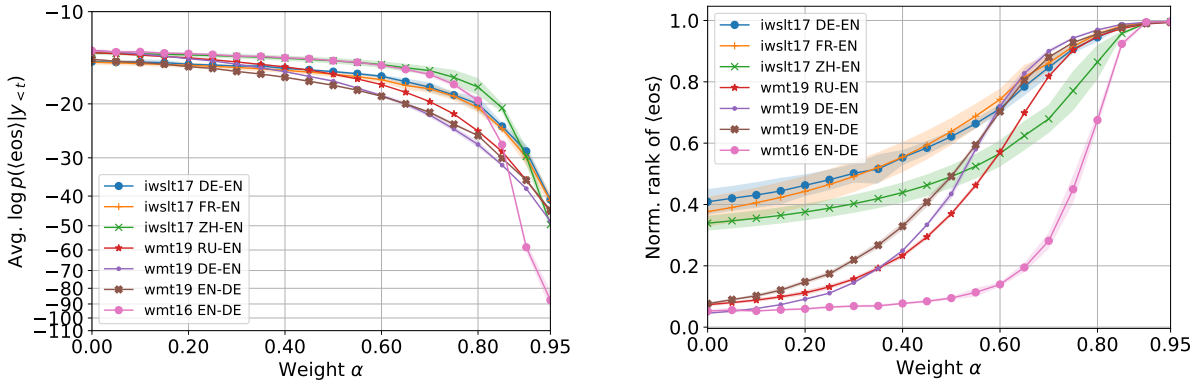
5.3 Decoding

To test translation quality, we translate a test set with beam search decoding, as implemented in FairSeq. We vary beam sizes to study their effect in-depth. The standard choice of beam size is on the smaller side, such as 10, because of the exponential complexity of the beam search w.r.t. the target sequence length. We set the lower- and upper-bound of a generated translation to be, respectively, 0 and $1.2 \cdot l_x + 10$, where l_x is the length of the source x. We do not use either length normalization nor length penalty, in order to study the impact of oversmoothing on decoding faithfully. We compute and report BLEU scores using `sacreBLEU` on detokenized predictions.

6 Experiments

As we pointed out earlier, publicly available translation systems exhibit a high degree of oversmoothing. See the left-most part of Figure 1, where $\alpha = 0$. In particular, this rate ranges from **34%** (WMT'19 DE→EN) up to **56%** (IWSLT'17 ZH→EN).

According to Section 3.1, the oversmoothing rate



(a) (b)

Figure 2: (a) Log-probabilities of ⟨eos⟩ token within length-t prefixes averaged across all positions per translation and then averaged across all translations. (b) Normalized rank of ⟨eos⟩ token within length-t prefixes averaged across all positions t per translation and then averaged across all translations. 1 means the lowest rank within the vocabulary. Filled regions denote the standard deviation across training runs according to Section 5.

should decrease as we increase the relative strength of the oversmoothing loss. To verify this, we fine-tune these models while varying the coefficient α. In Figure 1 we demonstrate the oversmoothing rate reduces all the way down to **3%** (WMT'19 DE→EN) and **17%** (IWSLT'17 ZH→EN) as we increase the strength of the regularizer. The oversmoothing rate monotonically decreases for every system we consider, as we increase α up to 0.95.

6.1 Regularization and ⟨eos⟩ token

Minimizing the proposed oversmoothing loss minimizes the log-probability of ⟨eos⟩ token at the end of every length-t prefix unless it is already low enough. We analyze how the strength of regularization affects the average log-probability of ⟨eos⟩ token measured at the end of each premature translation. As presented in Figure 2 (a), the log-probability of ⟨eos⟩ at the end of premature sequences decreases monotonically as the oversmoothing rate decreases (i.e., as the strength of the oversmoothing loss increases).

Although the log-probability of ⟨eos⟩ is an important factor in oversmoothing, Welleck et al. (2020a) claim that it is the rank of ⟨eos⟩ token that matters when using an incomplete approximate decoding strategy, such as beam search, for generation. We thus look at the average normalized rank of ⟨eos⟩ token at the end of every length-t prefix in Figure 2 (b). The rank drops rapidly and almost monotonically as we add more regularization. The effect of regularization is more visible with the rank than with the log-probability, especially when α is small.

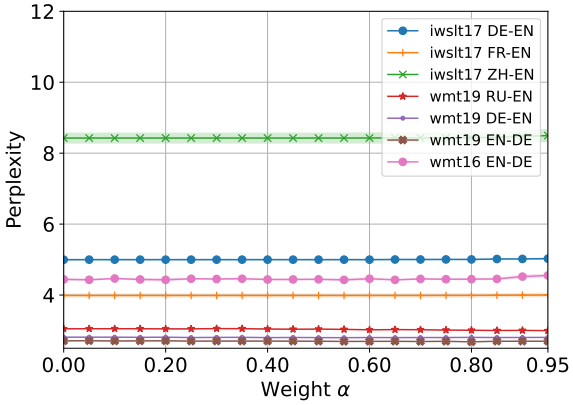
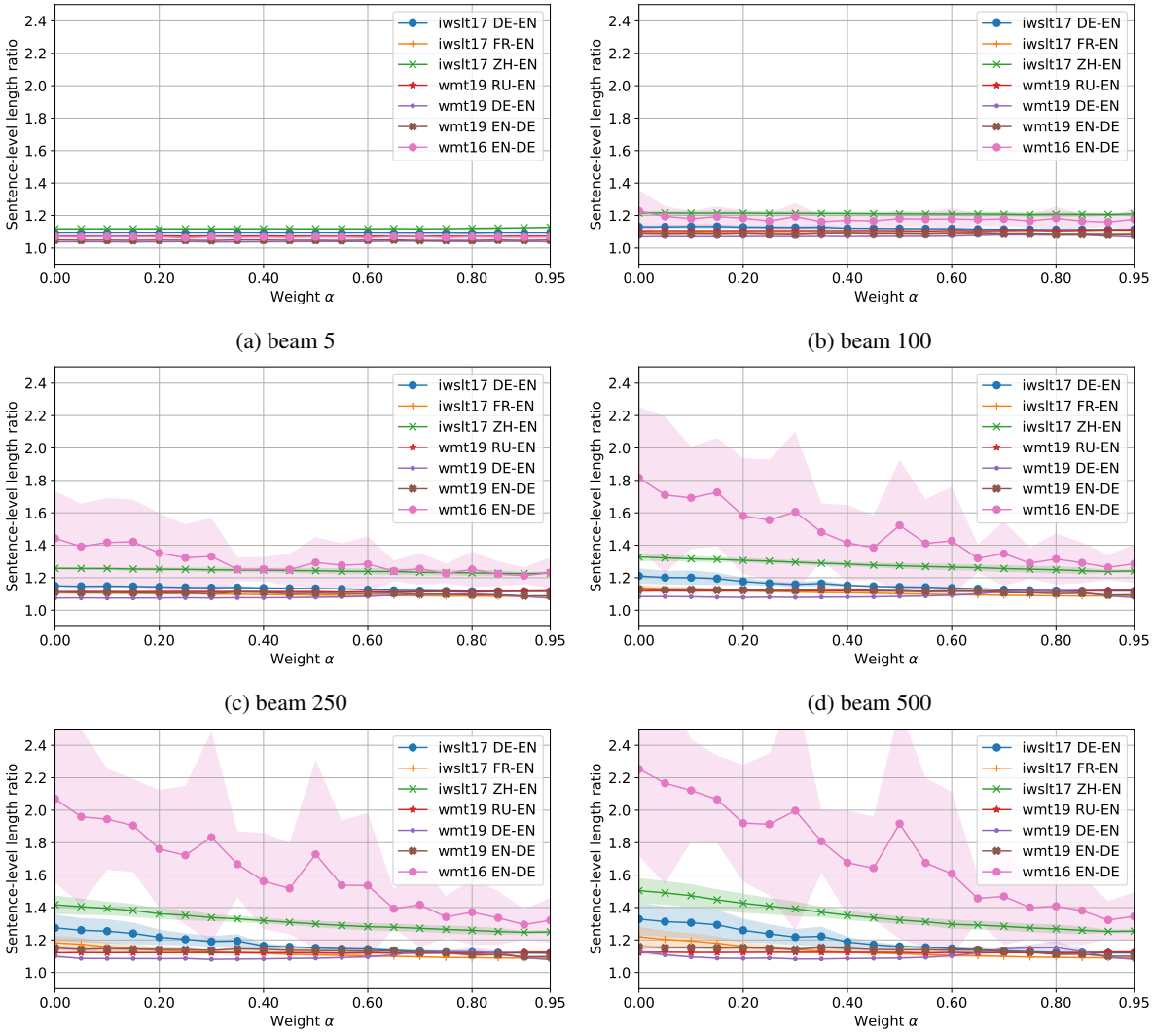


Figure 3: Perplexity measured on reference translations remains stable as we increase the strength of the regularization. Filled regions denote the standard deviation across training runs according to Section 5.

Although the proposed regularization reduces the probability of ⟨eos⟩ token where it is not supposed to be, we observe that the performance of the system as a language model does not degrade much regardless of the chosen value of α. This is evident from the flat lines in Figure 3 where we plot the perplexity of each model while varying α. This demonstrates that there are many different ways to minimize the negative log-likelihood, and some of those solutions exhibit a higher level of oversmoothing than the others. The proposed oversmoothing loss is an effective way to bias the solution toward a lower level of oversmoothing.



(a) beam 5

(b) beam 100

(c) beam 250

(d) beam 500

(e) beam 750

(f) beam 1000

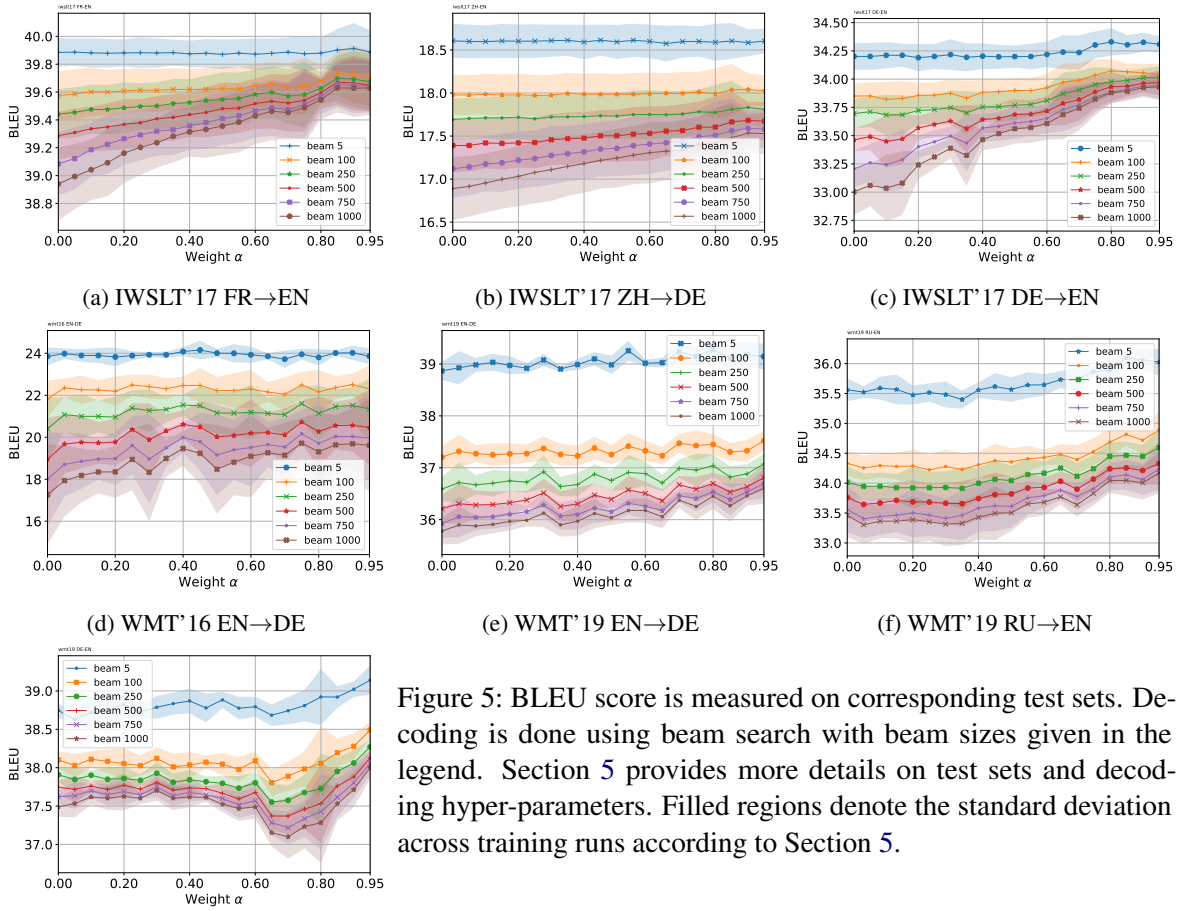
Figure 4: Sentence-level length ratio is $\frac{1}{|D_{\text{test}}|}\sum_{i=1}^{|D_{\text{test}}|}|\mathbf{y}_i^{\text{ref}}|/|\mathbf{y}_i^{\text{beam}}|$, where $\mathbf{y}_i^{\text{beam}}$ is generated translation using beam search for i-th input sentence from the test set D_{test}, and $\mathbf{y}_i^{\text{ref}}$ is the corresponding reference translation. Filled regions denote the standard deviation across training runs according to Section 5.

6.2 Oversmoothing rate and decoding

Earlier Koehn and Knowles (2017) noticed this issue of oversmoothing by observing that the length of generated sequences dramatically dropped as the beam width increased. We confirm the decreasing length of generated translation as the beam size increases in Figure 4 when $\alpha = 0$. We study the change of this length as we add more regularization and calculate the sentence-level length ratio in Figure 4.

When fine-tuned with the proposed oversmoothing loss, the length ratio degrades significantly less, as we increase the beam size during decoding, than without. For instance, with $\alpha \geq 0.8$ the length ratio remains more or less constant with respect to the size of the beam. Despite the observed robustness, decoding with a smaller beam size produces translations with lengths which match reference lengths better regardless of the strength of regularization.

Translation quality The quality of the produced translation is directly related to its length, because this length needs to closely match the length of the reference translation. However, the length information is not sufficient to make a conclusion about the translation quality. We quantify the quality of the translation by calculating the corpus-level BLEU score. The scores in Section 6.2 indicate that the reduced degradation of length modeling does correlate with the improvements in translation quality, although the degree of such correlation



(a) IWSLT'17 FR→EN (b) IWSLT'17 ZH→DE (c) IWSLT'17 DE→EN

(d) WMT'16 EN→DE (e) WMT'19 EN→DE (f) WMT'19 RU→EN

(g) WMT'19 DE→EN

Figure 5: BLEU score is measured on corresponding test sets. Decoding is done using beam search with beam sizes given in the legend. Section 5 provides more details on test sets and decoding hyper-parameters. Filled regions denote the standard deviation across training runs according to Section 5.

varies across language pairs and beam widths. We highlight two major aspects of the effect of regularization on the translation quality. First, the impact of regularization is only visible when the beam size is substantially larger than what is commonly used in practice. Second, the degradation of translation quality with a larger beam size lessens as oversmoothing does as well, but it does not eliminate the degradation fully. These observations imply that the effectiveness of approximate decoding in neural machine translation remains unsolved, despite our successful attempt at addressing the issue of oversmoothing.

7 Conclusion

In this work, we tackled a well-reported issue of oversmoothing in neural autoregressive sequence modeling, which has evaded rigorous characterization until now despite of its ubiquity. We characterized it by defining the oversmoothing rate. It computes how often the probability of the ground-truth sequence is lower than the probability of any of its prefixes. We confirmed that the oversmoothing rate is too high among well-trained neural machine translation systems and proposed a way to directly minimize it during training. We designed a differentiable upper bound of the oversmoothing rate called the oversmoothing loss. We experimented with a diverse set of neural machine translation systems to study the effect of the proposed regularization.

The experiments revealed several findings and takeaways. First, the oversmoothing loss is effective: we were able to monotonically decrease the oversmoothing rate by increasing the strength of the loss. Second, we found that this regularization scheme significantly expands the dynamic range of the log-probability of $\langle \text{eos} \rangle$ token and has even greater impact on its rank, without compromising on sequence modeling. Third, the proposed approach dramatically alters the behaviour of decoding when a large beam width was used. More specifically, it prevents the issue of degrading length ratio and improves translation quality. These effects were not as apparent with a small beam size though. The proposed notion of oversmoothing explains some of the degeneracies re-

ported earlier, and the proposed mitigation protocol alleviates these degeneracies. We, however, find that the proposed approach could not explain a more interesting riddle, that is, the lack of improvement in translation quality despite lower oversmoothing when beam search with a smaller beam was used. This unreasonable effectiveness of beam search continues to remain a mystery and needs to be investigated further in the future.

References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Bryan Eikema and Wilker Aziz. 2020. Is map decoding all you need? the inadequacy of the mode in neural machine translation. *arXiv preprint arXiv:2005.10283*.

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

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

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation.

Mathias Müller and Rico Sennrich. 2021. Understanding the properties of minimum bayes risk decoding in neural machine translation.

Benjamin Newman, John Hewitt, Percy Liang, and Christopher D. Manning. 2020. The eos decision and length extrapolation.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*.

Toan Q. Nguyen, Kenton Murray, and David Chiang. 2021. Data augmentation by concatenation for low-resource translation: A mystery and a solution.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018a. Scaling neural machine translation. *Proceedings of the Third Conference on Machine Translation: Research Papers*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018b. Scaling neural machine translation. *arXiv preprint arXiv:1806.00187*.

Ben Peters and André FT Martins. 2021. Smoothing and shrinking the sparse seq2seq search space. *arXiv preprint arXiv:2103.10291*.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, et al. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325.

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.

Xing Shi, Yijun Xiao, and Kevin Knight. 2020. Why neural machine translation prefers empty outputs.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Felix Stahlberg and Bill Byrne. 2019. On nmt search errors and model errors: Cat got your tongue? *arXiv preprint arXiv:1908.10090*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

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

Yu-Siang Wang, Yen-Ling Kuo, and Boris Katz. 2020. Investigating the decoders of maximum likelihood sequence models: A look-ahead approach.

Sean Welleck, Ilia Kulikov, Jaedeok Kim, Richard Yuanzhe Pang, and Kyunghyun Cho. 2020a. Consistency of a recurrent language model with respect to incomplete decoding. *arXiv preprint arXiv:2002.02492*.

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