Empirical Analysis of Beam Search Curse and Search Errors with Model Errors in Neural Machine Translation

Jianfei He[1], Shichao Sun[2], Xiaohua Jia[1], Wenjie Li[2]
[1] City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong
[2] The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
jianfeihe-2c@my.cityu.edu.hk, bruce.sun@connect.polyu.hk
csjia@cityu.edu.hk, wenjie.li@polyu.edu.hk

Abstract

Beam search is the most popular decoding method for Neural Machine Translation (NMT) and is still a strong baseline compared with the newly proposed sampling-based methods. To better understand the beam search, we investigate its two well-recognized issues, beam search curse and search error, not only on the test data as a whole but also at the sentence level. We find that only less than 30% of sentences in the WMT17 En–De and De–En test set experience these issues. Meanwhile, there is a related phenomenon. For the majority of sentences, their gold references get lower probabilities than the predictions from the beam search. We also test with different levels of model errors including a special test using training samples and models without regularization. In this test, the model has an accuracy of 95% in predicting the tokens on the training data. We find that these phenomena still exist even for such a model with very high accuracy. These findings show that it is not promising to improve the beam search by seeking higher probabilities and further reducing the search errors in decoding. The relationship between the quality and the probability at the sentence level in our results provides useful information to find new ways to improve NMT.

1 Introduction

Beam search has been the most popular decoding (inference) method for Neural Machine Translation (NMT) (Bahdanau et al., 2014). Fernandes et al. (2022)[1] and our experimental results (in Appendix A) show that the beam search is still a very strong baseline compared with the recently proposed sampling-based methods, including Top-k sampling, Nucleus (Top-p) sampling (Holtzman et al., 2019) and Minimum Bayes Risk (MBR) decoding (Eikema and Aziz, 2021; Freitag et al., 2022). This is verified with different evaluation methods: BLEU, Meteor, and Comet (Rei et al., 2020).

Meanwhile, there are still open issues deserving further exploration for the beam search.

One widely recognized issue is a phenomenon called *beam search curse* (Koehn and Knowles, 2017; Yang et al., 2018; Meister et al., 2020). Beam search tends to get worse performance when the beam size increases. This issue is counter-intuitive. Usually, it is expected that using a larger beam size finds a sequence with higher probability in the search space and gets better quality.

Another issue is *search error* (Stahlberg and Byrne, 2019; Shi et al., 2020), which means that the beam search as a heuristic method is not guaranteed to find the sequence with the largest probability in the search space. Stahlberg and Byrne (2019) implement *exact search* which can find the global maximum for experiments. They use it to assess the search errors in the beam search.

This paper aims to better understand these two

[1] Their conclusion is that MBR with Comet as the utility function outperforms the beam search if Comet is also used as the metrics. But if BLEU is used as the metrics, the beam search is still the best for the large models as shown in their Table 1 and Table 2.

issues via empirical analysis.

We look into *beam search curse* at the sentence level. Although the beam search curse is consistently verified on the whole test set at the corpus level, only a small portion of sentences suffer from this issue. One-sixth of sentences in WMT17 En–De and De–En test sets get worse translations when the beam size increases, meanwhile a similar number of sentences get better translations. One of the reasons for the beam search curse is *model error*, which means that the model is not well fitted to the data. We investigate the beam search curse using the model checkpoints with different validation accuracies. We find that there is no strong correlation between the beam search curse and model accuracy if the corpus BLEU score is used for evaluation. But there is an obvious correlation using the *oracle* BLEU score.

We assess *search error* using *exact search* with a length constraint. Exact search can be regarded as a beam search with its beam size as large as the size of vocabulary. We find that only less than 30% of sentences suffer from search errors using the beam search even with a small beam size like 5. For the majority of sentences, beam search can generate the sequences with the largest probability. We also compare exact search with beam search in terms of the quality of the predictions. Exact search gets significantly worse BLEU scores than beam search at the corpus level. At the sentence level, the number of sentences with worse quality from exact search is only slightly larger than those with better quality. This result is consistent with the experiments in the beam search curse issue.

Our experiments also demonstrate one phenomenon that is related to these two issues. The majority of the gold references get lower probabilities than the predictions from beam search. Although beam search seeks the sequences with high probability in principle, this result shows that it is the wrong direction to *further* pursue larger probabilities and smaller search errors.

To investigate how beam search performs under very low model errors, we test a special case. We use models without regularization which have an accuracy of around 95% on training data. The test data in this case are samples from training sets to reduce the mismatch of data distributions between training and testing. In this case, the phenomena about exact search and gold references are still observed.

These findings may contribute to future improvements in decoding and training methods.

2 Related Work

There are two approaches for decoding today: *mode-seeking* decoding and sampling-based stochastic decoding. *Mode-seeking* is also known as Maximum-A-Posteriori (MAP) decoding (Smith, 2011; Eikema and Aziz, 2020). Its objective is to predict a translation by searching a sequence $y^\star$ that maximizes $log\ P(y|src;\theta)$, where src is the source sentence and θ is the model parameter set. *Exact search* (Stahlberg and Byrne, 2019) aims to find the global maximum in the whole search space. Due to the vast search space, exact search is intractable in real application. Beam search (Lowerre, 1976; Graves, 2012) is used as a viable approximation by extending the N most probable partial solutions at each decoding step, where N is called *beam size*. Beam search is widely used for NMT.

Recently the sampling-based stochastic decoding (Fan et al., 2018; Holtzman et al., 2019; Eikema and Aziz, 2021; Freitag et al., 2022) is actively investigated. Sampling methods are used in decoding to get a set of candidate sequences, then a decision rule is used to choose the final prediction among these candidates. Although these methods are used for open-ended text generation tasks such as story generation, Fernandes et al. (2022) and our experimental results (in Appendix A) show that beam search is still a very strong baseline compared with these sampling-based methods for NMT.

Beam search curse is recognized as one of six challenges in NMT (Koehn and Knowles, 2017). Murray and Chiang (2018) and Yang et al. (2018) attribute its root cause to the *length ratio* problem via empirical study. With beam size increasing, beam search tends to get shorter predictions and results in lower BLEU due to the *brevity penalty* in the definition of BLEU scores. But it is a usual practice using length normalization methods and the issue of short predictions is significantly mitigated. On the other hand, the beam search curse also consistently exists with other evaluation methods such as Meteor and Comet. Cohen and Beck (2019) investigate the *discrepancy gap* which is defined as the difference in log-probability between the most likely token and the chosen token. They find that the majority of discrepancy happen in early positions and increasing the beam width leads to more early discrepancies. We investigate the beam search curse at the sentence level, which is orthogonal to their conclusion about the position of tokens.

Search error in NMT is intensively investigated by Stahlberg and Byrne (2019). They use an algorithm based on the deep first search to explore whether there is a sequence with a higher probability than the prediction from beam search. They also implement the exact search to find the sequence with the largest probability in the search space.

In these research, the beam search curse and the search error are mainly investigated on the whole test set at the corpus level, not at the sentence level. And it's not investigated how these issues are related to *model errors*. The model error means that the model is not well fitted to the data.

3 Methodology

We choose the widely used language pairs: En–De and De–En. Besides a standard test, we conduct a special *cleanroom* test to investigate the issues with very low model errors. Figure 1 depicts the distribution of sentence length in all test sets. Comparing it with our experimental results, it shows that the sentence length is not an influential factor in the conclusions.

Standard test In this test, we use Transformer Big and Transformer Base models and use the corpora from WMT17[2]: Europarl v7, News-commentary-v12 and Common Crawl for training, Newstest2014 for validation, Newstest2017 for the test which has 3004 sentence pairs.

Cleanroom test In this test, we investigate how the decoding methods work when the model is fitted well to the test data. The model errors are very small in this test. For this purpose, we randomly select 2000 sentences from the training set and use them as the test data. To further reduce the model errors in this test, we use models without regularization. Dropout (Srivastava et al., 2014) and label smoothing (Szegedy et al., 2016)) are used in Transformer as regularization methods to prevent neural networks from overfitting. The models that we used in this test are trained with both methods turned off.

Models We use the notations below for three models in our experiments.

- *Base* and *Big* for the normal Transformer Base and Transformer Big models. They use regularization methods.
- *NoReg* are based on Transformer Big except that they are trained with dropout and label smoothing turned off. These models have an accuracy larger than 95% on the training data.

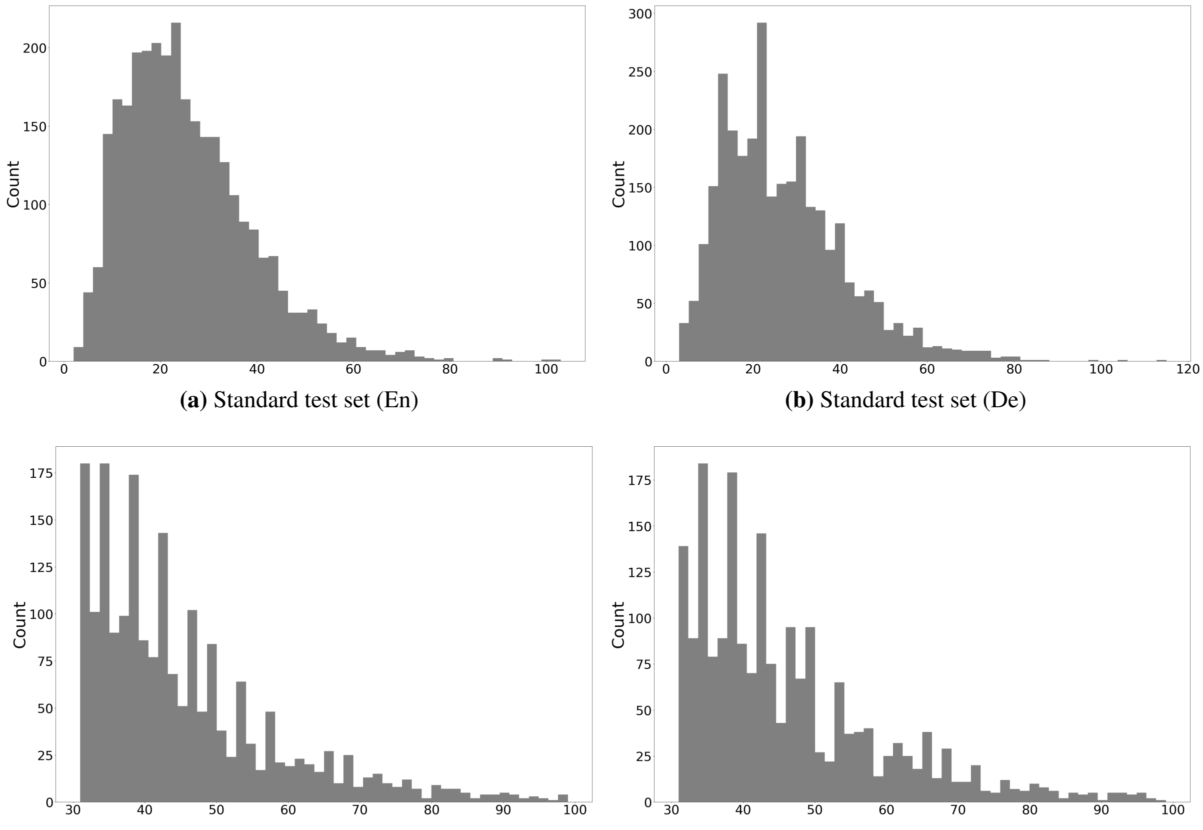
Decoding methods For *beam search*, we use two beam sizes and compare their results to investigate the issue of beam search curse. One is 5 and the other is 100. For *exact search*, we reimplement the algorithm in Stahlberg and Byrne (2019). In this algorithm, the search only extends a partial sequence if its probability is larger than a baseline value. A large baseline value can speed up the exact search. We get the probabilities of the predictions from the beam search with a series of beam sizes: 1–20, 50, and 100. We also get the probability of the gold reference under the model. Then we get the largest probability among these 23 instances for each sentence in the test set and use it as the baseline value for the exact search. We sort the test sets with the baseline values in descending order so that sentences with higher baseline values are translated before those with lower baseline values. We continue to run the search on one Nvidia GF1080Ti GPU for nearly 100 days. Table 3 lists how many sentences are translated using the exact search. We apply one of the length constraints used by Stahlberg and Byrne (2019) for exact search: the length of the target sentences is constrained to be no less than 1/4 of the length of their source sentences. Stahlberg and Byrne (2019) also use some tighter constraints to further mitigate the search errors. We aim to investigate the details at the sentence-level in the exact search. Therefore we choose a loose and practical constraint.

Training and Evaluations Our implementation is based on the OpenNMT-tf toolkit (Klein et al., 2020) with a typical configuration[3]. The Base models are trained for 200,000 steps on 4 GPUs, while the Big and NoReg are trained for 300,000 steps on 8 GPUs. All GPUs are Nvidia GF1080Ti. We use the unigram (Kudo, 2018) in SentencePiece[4] for subwords with 32,000 updates and use a

[2] http://www.statmt.org/wmt17

[3] https://opennmt.net/OpenNMT-py/FAQ.html\#how-do-i-use-the-transformer-model

[4] https://github.com/google/sentencepiece



(a) Standard test set (En)

(b) Standard test set (De)

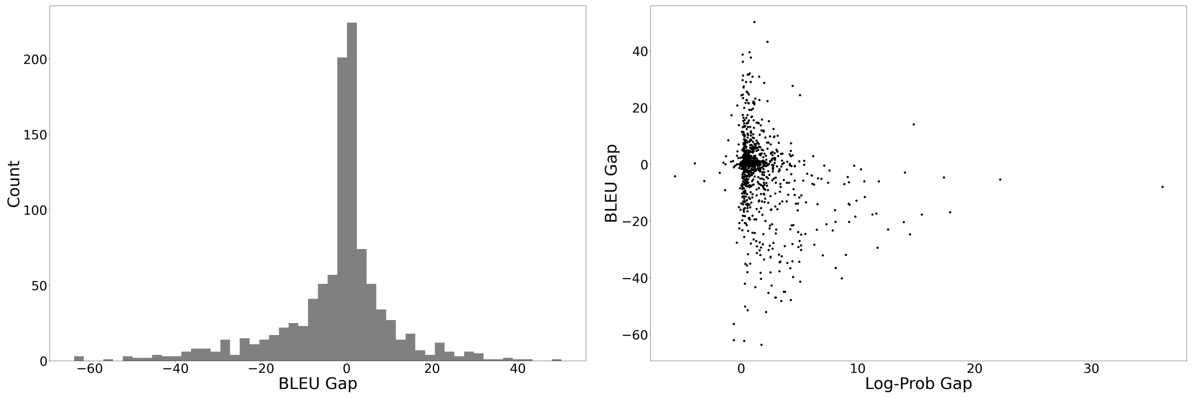
(c) 2000 samples from training set (En)

(d) 2000 samples from training set (De)

Figure 1: The histograms of sentence length for test sets. The number of subwords are counted for each sentence.

	En–De						De–En					
Model	**Base**			**Big**			**Base**			**Big**		
Metrics	BLEU	Meteor	Comet	BLEU	Meteor	Comet	BLEU	Meteor	Comet	BLEU	Meteor	Comet
Beam5	**28.2**	**29.1**	**0.490**	**28.9**	**29.2**	**0.498**	**33.5**	**36.5**	0.520	**33.8**	**36.7**	**0.539**
Beam100	27.7	26.0	0.450	27.4	28.8	0.426	**33.5**	**36.5**	**0.521**	33.2	36.5	0.527

Table 1: Performance of the beam search using beam size 5 and 100, denoted as *Beam5* and *Beam100* respectively.



(a) Gap of sentence BLEU: Beam100 minus Beam5

(b) Gap of log-probability as the x-axis and gap of sentence BLEU as the y-axis: Beam100 minus Beam5

Figure 2: Investigate the beam search curse at sentence level for En–De.

shared vocabulary for source and target. For evaluation, we use BLEU, Meteor, and Comet to compare the beam search with sampling-based stochastic decoding methods. Since the results are consistent, we stick to BLEU in the investigation of the beam search. For BLEU, We use SacreBLEU [5] (Post, 2018) [6]. For Meteor[7], we use version 1.5. For Comet[8], we use the *wmt20-comet-da* model.

4 Beam Search Curse

4.1 Only a Small Portion of Sentences Experience Beam Search Curse

The beam search curse has been consistently verified at the corpus level. Our results in Table 1 demonstrate this issue using the comparison between beam size 5 and beam size 100, denoted as *Beam5* and *Beam100* respectively.

However, our experiments reveal that this issue is not ubiquitous at the sentence level.

We investigate the gap of the *sentence BLEU* score between Beam100 and Beam5 for each sentence. The results from a standard test using the Big model are shown in Table 2. It illustrates how many sentences in the standard test set get *larger*, *equal*, and *smaller* sentence BLEU scores from Beam100 compared with Beam 5. Smaller sentence BLEU scores from Beam100 imply the beam search curse for these sentences. It shows that only about one-sixth of sentences have this issue. For En–De, the number of sentences with the beam search curse is less than those that Beam100 gets better performance than Beam5.

	Total Sent.	**>Beam5**	**=Beam5**	**<Beam5**
En–De	3004	506	1968	530
De–En	3004	515	1976	513

Table 2: The number of sentences that Beam100 gets *larger*, *equal* and *smaller* sentence BLEU compared with Beam 5, denoted as *>Beam5*, *=Beam5* and *<Beam5* respectively.

Figure 2a illustrates the gap of sentence BLEU scores for En–De. The sentences with a zero BLEU gap are not counted in this figure.

We also investigate the relationship between the gap of sentence BLEU and the gap of log-probability for each sentence, as illustrated in Figure 2b. For most sentences, Beam100 gets larger log-probabilities than Beam5. Beam search with a larger beam size has more opportunities to find sequences with larger log-probabilities. The majority of sentences have small log-probability gaps. For these sentences, the gap of sentence BLEU has a similar probability to be positive or negative. When the log-probability gap increases, the BLEU gap tends to be more negative. This small portion of sentences result in worse quality at the corpus level. Potentially we can find a way to identify these sentences and apply a small beam size for them. Meanwhile, we can use a large beam size to improve the quality of other sentences. The sentences with a zero log-probability gap are not counted in this figure.

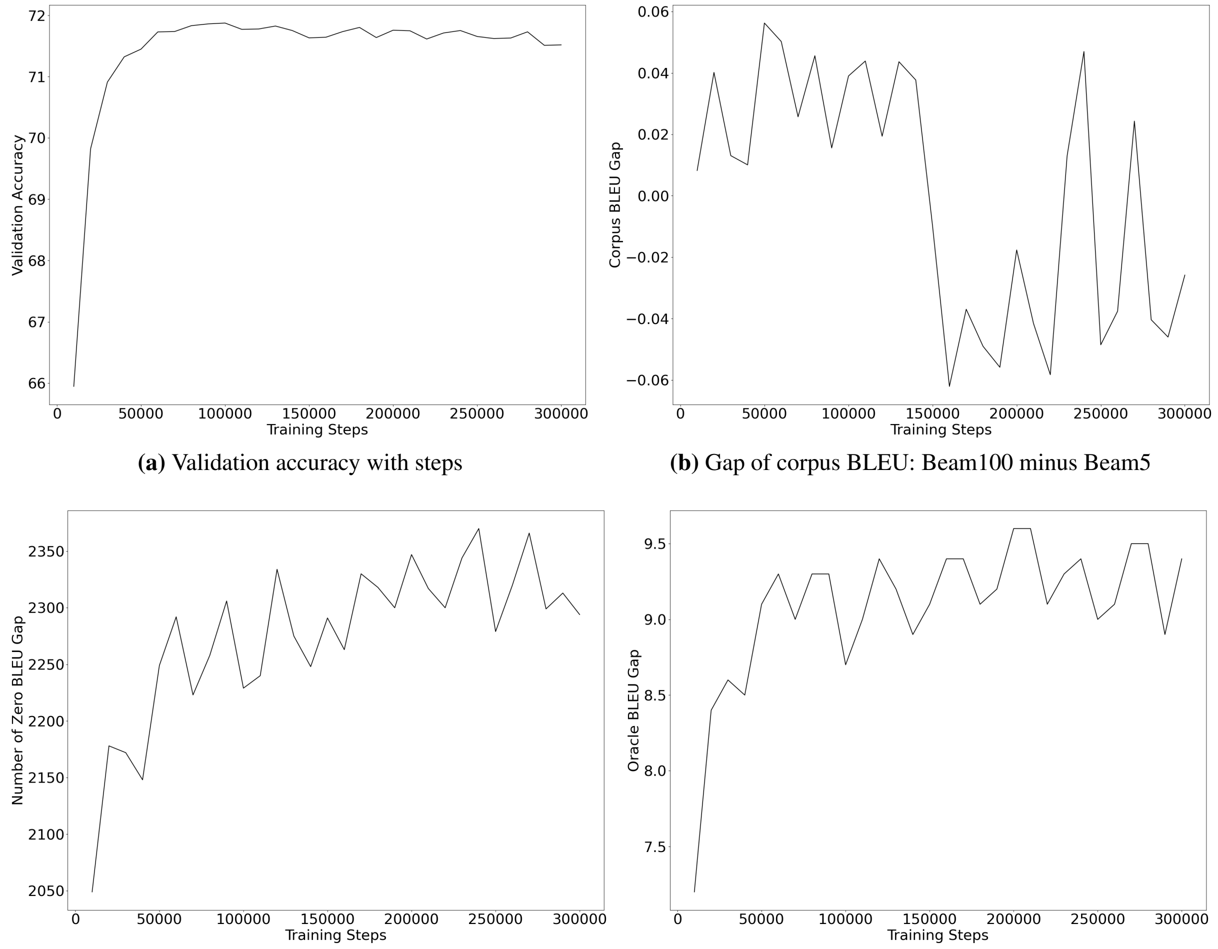
We conduct experiments using out-of-domain test sets and get consistent results which are illustrated in Appendix B.

4.2 Correlation between Beam Search Curse and Model Accuracy

It is an interesting question whether the beam search curse is mitigated for a model with higher accuracy. We record the checkpoints at every 10,000 steps till 300,000 steps in training the Big model. The values of their validation accuracy are depicted in Figure 3a. As shown in Figure 3b, we surprisingly find that there is no strong correlation between model accuracy and beam search curse in terms of the corpus BLEU.

However, we find two correlations related to the model accuracy. One is the number of sentences with zero gap. When the model accuracy increases, Beam100 and Beam5 tend to have more sentences that have the same BLEU scores, as illustrated in Figure 3c. The other is *oracle corpus BLEU*, which is calculated given that the gold references are used to pick the best predictions from candidates. More candidates usually contain better oracle hypotheses. It is not surprising that Beam100 has much better oracle BLEU scores than Beam5. The interesting result in Figure 3d is the strong correlation between the gap of the oracle corpus BLEU and the model accuracy. This means that there are better candidates in the top 100 candidates with higher model accuracy. But current Beam100 cannot make use of it to make better predictions because the usual beam search method uses the probabilities of candidates to decide the final output. Better candidates do not necessarily have the larger probabilities. They

[5] https://github.com/mjpost/sacreBLEU
[6] case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.14
[7] http://www.cs.cmu.edu/~alavie/METEOR/
[8] https://github.com/Unbabel/COMET



(a) Validation accuracy with steps

(b) Gap of corpus BLEU: Beam100 minus Beam5

(c) Number of sentences with a zero BLEU gap

(d) Gap of oracle BLEU: Beam100 minus Beam5

Figure 3: Investigate the correlation between beam search curse and model accuracy

		Total Sent.	**Exact**	**Beam5**	Δ	**<Beam5**	**=Beam5**	**>Beam5**
Std+Big	**En–De**	2319	27.33	30.49	-3.16	431	1638	250
	De–En	2375	32.80	35.70	-2.90	424	1701	250
Sample+NoReg	**En–De**	2000	52.47	53.80	-1.33	259	1606	135
	De–En	2000	58.51	60.23	-1.72	264	1623	113

Table 3: Corpus BLEU of exact search (denoted as *Exact*) and comparison with Beam5. *Total Sent* is the total number of sentences that the exact search finishes translation. Columns *<Beam5*, *=Beam5* and *>Beam5* are how many sentences that exact search gets lower, equal, and greater BLEU compared with Beam5.

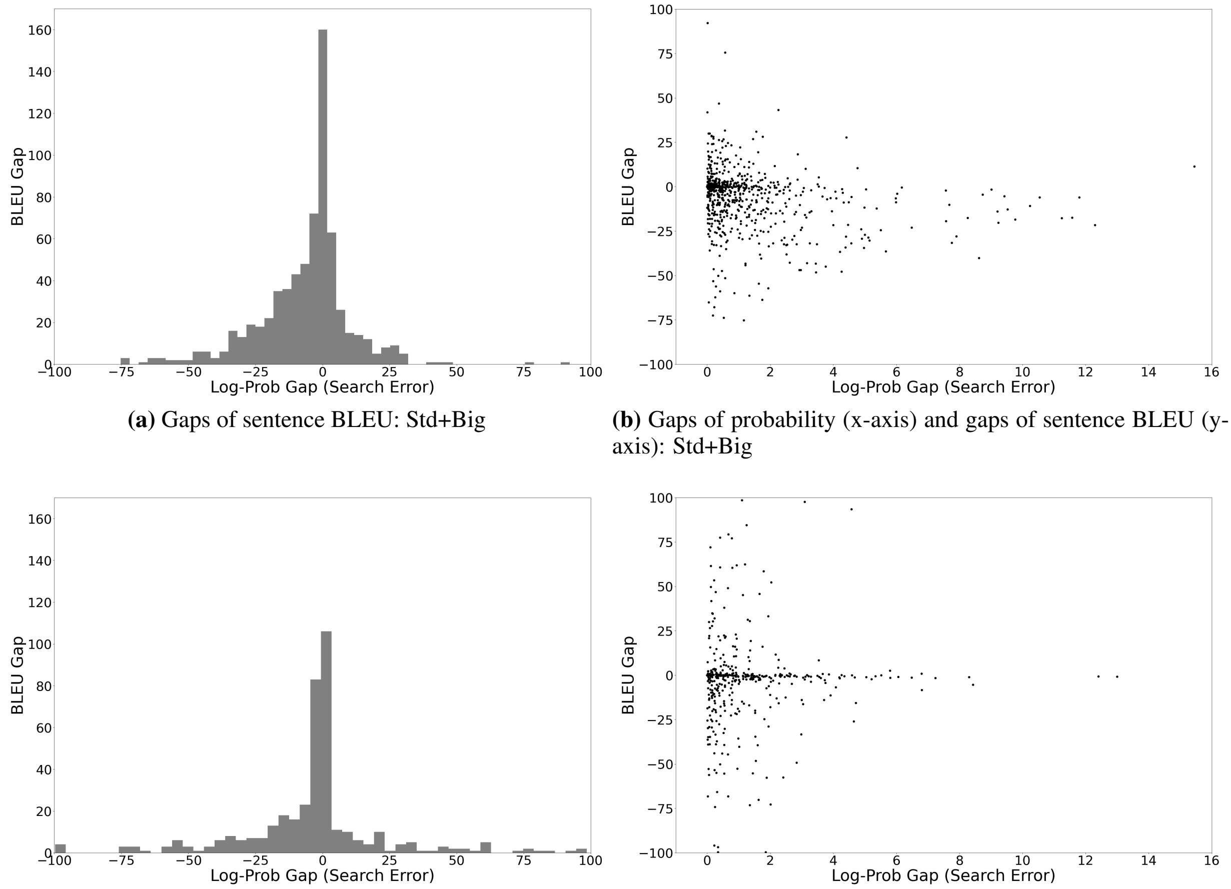
are probably discarded in the final decision. This implies a potential solution to improve the beam search. Beam search may benefit from the models with lower model errors in case that we have a suitable reranking method on the candidates.

5 Zero Search Error Gets Worse Quality

We compare the BLEU scores from *exact search* with Beam5 at both the corpus level and the sentence level. In our experiments, we find that a zero gap of the sentence BLEU score usually implies a zero probability gap as well, which means *zero search error* for Beam5.

The results at the sentence level in Table 3 reveal that the beam search works quite well in terms of the search error. Even with a small size like 5, beam search is capable to find the sequences with the largest probability for about 70% of sentences.

Table 3 also shows that the exact search gets significantly worse corpus BLEU scores than Beam5. Figure 4a and Figure 4b shows the results of the



(a) Gaps of sentence BLEU: Std+Big

(b) Gaps of probability (x-axis) and gaps of sentence BLEU (y-axis): Std+Big

(c) Gaps of sentence BLEU: Sample+NoReg

(d) Gaps of probability (x-axis) and gaps of sentence BLEU (y-axis):Sample+NoReg

Figure 4: Comparison between exact search and Beam5: En–De. All gaps are exact search minus Beam5.

standard test with the Big model. Figure 4c and Figure 4d show the results of the training samples with the NoReg model. In this case that the model errors are very small, the gap of the corpus BLEU score is mitigated. But in both cases, when the gap of log-probability between two methods increases, the gap of BLEU is more likely to be negative.

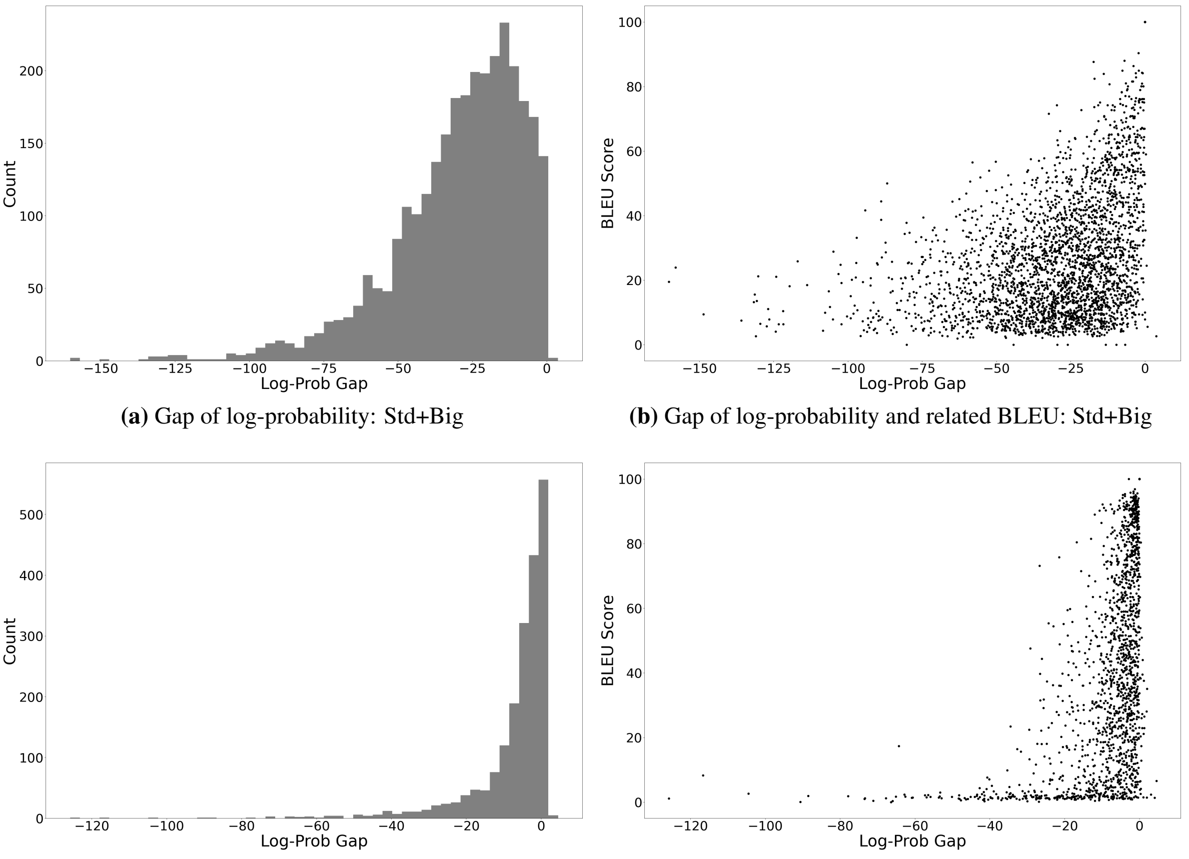
In all these four figures, sentences having a zero BLEU gap are not counted.

6 Gold References Get Lower Probability than Predictions from Beam Search

The experiments above show that sequences with higher log-probabilities do not necessarily get better BLEU scores. This leads us to investigate the log-probabilities of gold references. We find that gold references get lower log-probability than the predictions from the beam search even with very low model errors.

Figure 5a illustrates the gap of log-probability between the gold references and Beam5 for En–De. Only for a few sentences, the gold references have higher log-probabilities than the predictions of Beam5. Figure 5b demonstrates the strong correlation between the gap of log-probability (as the x-axis) and the sentence BLEU scores of Beam5 (as the y-axis). When the gold references get lower log-probabilities than Beam5, the sentence BLEU scores of Beam5 decrease. These two figures are results from the standard test with the Big model. We also test using the training samples with models without regularization. Results are illustrated in Figure 5c and Figure 5d. Comparing these two test cases, we find that the gaps are reduced when the model errors are smaller in the latter case. However, the correlation between the log-probability and the sentence BLEU still exists even for a model with an accuracy of 95% in the cleanroom test.

Case study and analysis Table 4 illustrates an example in the test using training samples and models without regularization. There is only one



(a) Gap of log-probability: Std+Big

(b) Gap of log-probability and related BLEU: Std+Big

(c) Gap of log-probability: Sample+NoReg

(d) Gap of log-probability and related BLEU: Sample+NoReg

Figure 5: The gap of log-probability between gold references and Beam5 for En–De. All gaps are gold reference minus Beam5.

Source	_Die _Aktionspläne _der _Hoch rang igen _Arbeitsgruppe _zielen _zwar _auf _die _zukünftige _Begrenzung _des _Einwanderung s strom s _ab , _doch _tragen _sie _in _keiner _Weise _zur _Verbesserung _der _Situation _hinsichtlich _der _Menschenrechte _und _der _Grundfreiheiten _sowie _der _wirtschaftliche n _Situation _der _betroffenen _Länder _bei .	
Prediction	_Although _action _plans _established _by _the _high - level _working _group _aim _to _limit _migratory _flows _in _the _future , _these _plans _do _nothing _to _improve _human _rights , _civil _liberties _and _the _economic _situation **_of** _the _countries _concerned .	Log Probablity: **-2.4142**
Gold Reference	_Although _action _plans _established _by _the _high - level _working _group _aim _to _limit _migratory _flows _in _the _future , _these _plans _do _nothing _to _improve _human _rights , _civil _liberties _and _the _economic _situation **_in** _the _countries _concerned .	Log Probablity: **-6.9390**

Table 4: An example that a gold reference gets a lower log-probability than Beam5. There is only one token that is different between the prediction of Beam5 and the gold reference.

token that is different between the gold reference and the prediction of Beam5. This small difference results in a significantly lower log probability for the gold reference.

This result can be explained by the objective in training.

We use s and t_i to denote the source sequence and the ground truth token at the target side for the step i. $t_i^{'}$ is a token different from t_i at step i. At step k, the usual training objective is to maximize $log\ P(t_k|s, t_1, ..., t_{k-1})$. If the model is effectively trained, it implies

$$log\ P(t_k|s, t_1, ..., t_{k-1})\ >\ log\ P(t_k^{'}|s, t_1, ..., t_{k-1}). \quad (1)$$

However, the inequality below is *not* part of the training objective:

$$log\ P(t_k|s, t_1, ..., t_{k-1}) > log\ P(t_k|s, t_1, ..., t_{k-1}^{'}) \quad (2)$$

This can lead to the phenomenon that gold references get lower probabilities than potential sequences in the search space even in a model with very small model errors.

7 Conclusion

Experiments show that the beam search still outperforms most stochastic decoding methods in NMT. We investigate the beam search in the details at the sentence level. We find that two well-recognized issues, beam search curse and search error, only happen to a small portion of sentences in the test set. Meanwhile, for the majority of sentences, their gold references get lower log-probabilities than the predictions from the beam search. We also test with different levels of model errors including a cleanroom test using training samples and models without regularization. The results show that these issues still exist even for a model with an accuracy of 95%. These findings show that we cannot improve the beam search by further seeking higher log-probability during the search. In other words, further reducing search errors are not promising. Our results about the relationship between the quality and the gap of log-probability provide useful information for two potential ways to improve NMT. One is to find better reranking methods or decision rules to find good translations among the candidates from the beam search. The other is to find a new way to train the model so that the sequences with higher log-probabilities get better performance.

References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Cohen, Eldan and Christopher Beck. 2019. Empirical analysis of beam search performance degradation in neural sequence models. In *International Conference on Machine Learning*, pages 1290–1299. PMLR.

Eikema, Bryan 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), December. International Committee on Computational Linguistics.

Eikema, Bryan and Wilker Aziz. 2021. Sampling-based minimum bayes risk decoding for neural machine translation. *arXiv preprint arXiv:2108.04718*.

Fan, Angela, 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.

Fernandes, Patrick, António Farinhas, Ricardo Rei, José GC de Souza, Perez Ogayo, Graham Neubig, and André FT Martins. 2022. Quality-aware decoding for neural machine translation. *arXiv preprint arXiv:2205.00978*.

Freitag, Markus, David Grangier, Qijun Tan, and Bowen Liang. 2022. High quality rather than high model probability: Minimum bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

Graves, Alex. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Holtzman, Ari, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Klein, Guillaume, François Hernandez, Vincent Nguyen, and Jean Senellart. 2020. The opennmt neural machine translation toolkit: 2020 edition. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (AMTA 2020)*, pages 102–109.

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

Kudo, Taku. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia, July. Association for Computational Linguistics.

Lowerre, Bruce T. 1976. *The harpy speech recognition system*. Carnegie Mellon University.

Meister, Clara, Ryan Cotterell, and Tim Vieira. 2020. If beam search is the answer, what was the question? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185.

Murray, Kenton and David Chiang. 2018. Correcting length bias in neural machine translation. *WMT 2018*, page 212.

Post, Matt. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Rei, Ricardo, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference* *on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November. Association for Computational Linguistics.

Shi, Xing, Yijun Xiao, and Kevin Knight. 2020. Why neural machine translation prefers empty outputs. *arXiv preprint arXiv:2012.13454*.

Smith, Noah A. 2011. Linguistic structure prediction. *Synthesis lectures on human language technologies*, 4(2):1–274.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Stahlberg, Felix 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, November. Association for Computational Linguistics.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Yang, Yilin, Liang Huang, and Mingbo Ma. 2018. Breaking the beam search curse: A study of (re-)scoring methods and stopping criteria for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3054–3059.

A Comparing Beam Search to other Decoding Methods

Table 6 shows the comparison between beam search and some of sampling-based decoding methods. We use the notations below for the decoding methods.

- Beam5: beam search, the beam size is 5.
- Top5k10 and Top5k30: Top-k sampling, using top 10 and top 30 for the range for sampling respectively, the beam size is 5.
- Top5p75 and Top5p90: Nucleus (Top-p) sampling, using 75% and 90% for the sampling probability mass respectively. The beam size is 5.
- MBR300: the MBR decoding using 300 candidates from the unbiased sampling. The decision rule (utility function) is the similarity in terms of the sentence BLEU score between any two candidates. Fernandes et al. (2022) also use other utility functions such as Comet. These methods use some pre-trained models and introduce extra knowledge in the decision rule. Since we focus on the comparison of different decoding methods, we only use the ngram-based decision rule for MBR in our experiments.

B Out-of-Domain Test sets

We use the test sets in EMEA [9] for out-of-domain (OOD) tests.

Figure 6a illustrates the gap of sentence BLEU scores for En–De. Figure 6b illustrates the relationship between the gap of sentence BLEU and the gap of log-probability for each sentence. Table 5 shows the number of sentences that Beam100 gets *larger*, *equal* and *smaller* sentence BLEU compared with Beam 5 These results are consistent with the in-domain tests, shown in Figure 2a, Figure 2b and Table 2 in Section 4.1 respectively.

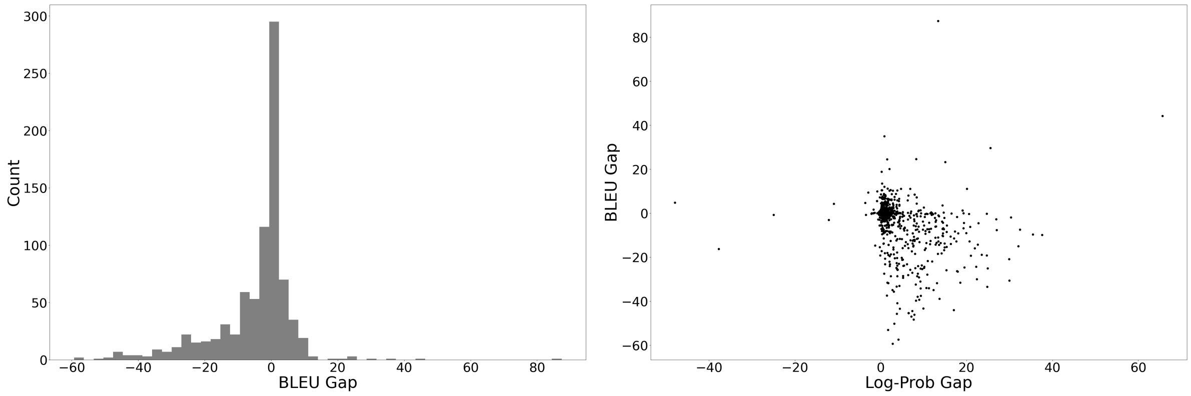
	Total Sent.	**>Beam5**	**=Beam5**	**<Beam5**
En–De	1267	347	434	486
De–En	1267	275	646	346

Table 5: Out-of-domain (OOD) tests: the number of sentences that Beam100 gets *larger*, *equal* and *smaller* sentence BLEU compared with Beam 5, denoted as *>Beam5*, *=Beam5* and *<Beam5* respectively.

[9] `http://https://opus.nlpl.eu/EMEA.php`

	En–De						**De–En**					
Model	**Base**			**Big**			**Base**			**Big**		
Metrics	BLEU	Meteor	Comet	BLEU	Meteor	Comet	BLEU	Meteor	Comet	BLEU	Meteor	Comet
Beam5	**28.2**	**29.1**	**0.490**	**28.9**	**29.2**	**0.498**	**33.5**	**36.5**	**0.520**	**33.8**	**36.7**	**0.539**
Top5k10	22.5	26.0	0.391	23.9	26.8	0.426	28.1	34.2	0.442	29.5	34.8	0.481
Top5k30	21.4	25.5	0.357	23.2	26.3	0.413	27.2	33.5	0.420	28.5	34.3	0.456
Top5p75	24.6	27.2	0.415	25.7	27.7	0.457	30.0	35.1	0.462	31.4	35.6	0.502
Top5p90	20.6	24.9	0.292	22.5	25.9	0.379	26.4	32.8	0.357	28.1	33.8	0.420
MBR300	24.9	27.0	0.181	26.5	27.9	0.298	30.7	34.2	0.301	31.9	35.0	0.377

Table 6: Comparison between beam search, Top-k sampling, Nucleus (Top-p) sampling and MBR decoding for En–De and De–En.



(a) Gap of sentence BLEU: Beam100 minus Beam5

(b) Gap of log-probability as the x-axis and gap of sentence BLEU as the y-axis: Beam100 minus Beam5

Figure 6: Out-of-domain (OOD) tests: investigate the beam search curse at sentence level for En–De.