On the Depth between Beam Search and Exhaustive Search for Text Generation

Anonymous ACL submission

Abstract

Beam search and exhaustive search are two extreme ends of text decoding algorithms with respect to the search depth. Beam search is limited in both search width and depth, whereas exhaustive search is a global search that has no such limitations. Surprisingly, beam search is not only computationally cheaper but also performs better than exhaustive search despite its higher search error. Plenty of studies have reported that moderate search widths work the best, but little has been investigated regarding the search depth. Based on the success of the moderate beam width, we examine a range of search depths to see its effect on performance. To this end, we introduce Lookahead Beam Search (LBS), a multi-step lookahead search that optimizes the objective considering a fixed number of future steps. Beam search and exhaustive search are special cases of LBS where the lookahead depth is set to 0 and ∞, respectively. We empirically evaluate LBS with the lookahead depth of up to 3 and show that it improves upon beam search. Although LBS is not a practical algorithm on its own because of its computational complexity, the results indicate that beam search with moderate widths still has room for improvement by searching deeper.

1 Introduction

The goal of natural language generation is to generate text representing structured information that is both fluent and contains the appropriate information. One of the key design decisions in text generation is the choice of decoding strategy. The decoding strategy is the decision rule used to generate strings from a probabilistic model (e.g., Transformer; Vaswani et al., 2017).

A straightforward solution is to **exhaustively search** for the strings with the highest probability with respect to the model. This is known as **maximum a posteriori (MAP) decoding**. Not only exhaustive search is computationally infeasible, but surprisingly, it is known to produce low-quality text (Murray and Chiang, 2018; Cohen and Beck, 2019). For example, Stahlberg and Byrne (2019) reports that in machine translation tasks, the highest-probability string is often the empty string.

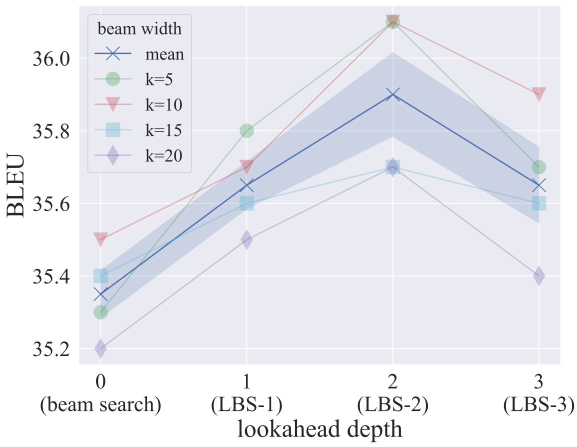


Figure 1: Results on machine translation by beam search and lookahead beam search (LBS) with lookahead depth 1, 2, and 3. The bold line represents the mean and the shaded area shows the standard error. Evaluated on the first 100 sentences of WMT'14 En-Fr dataset.

Beam search has been the go-to strategy in sequence generation. Beam search is a **local search** that greedily optimizes the local objective at each step with constraints on search depth and beam width. It is used in many state-of-the-art NLP applications, including machine translation (Wu et al., 2016; Ott et al., 2019; Wolf et al., 2020), text summarization (Rush et al., 2015; Narayan et al., 2018), and image captioning (Anderson et al., 2017). However, beam search is known to have high search error (Stahlberg and Byrne, 2019) due to the nature of local search. For example, Welleck et al. (2020) reports that beam search can yield infinite-length outputs that the model assigns zero probability to.

Prior work has studied the two extreme ends of the search in terms of search depth. Beam search is a one-step local search without any consideration of the future step. Exhaustive search optimizes the global objective without regard to local optimality at each step. Plenty of studies have investigated the effect of beam width on the search procedure and reported that a beam width that is neither too large nor too small is effective (Koehn and Knowles, 2017; Stahlberg and Byrne, 2019; Meister et al., 2020a). However, in terms of search depth, little has been investigated between the two extreme ends. The research question we investigate is whether there is a better trade-off between the two ends in terms of search depth.

To analyze the effect of the search depth on the quality of the generated sequences, we introduce **Lookahead Beam Search** (**LBS**), a variant of beam search with multiple steps lookahead to improve the estimate of the next step. Beam search and exhaustive search is a special case of LBS with lookahead depth of 0 and ∞, respectively. We empirically evaluate the performance of LBS in machine translation tasks. The results show that LBS with up to 3-step lookaheads outperforms the performance of beam search and exhaustive search overall using Transformer-based models (Figure 1).

2 Neural Text Generation

Sequence-to-sequence generation is the task of generating an output sequence $\mathbf{y}$ given an input sequence $\mathbf{x}$. Probabilistic text generators define a probability distribution $p_\theta(\mathbf{y}|\mathbf{x})$ over an output space of hypotheses $\mathcal{Y}$ conditioned on an input $\mathbf{x}$. The set of complete hypotheses $\mathcal{Y}$ is:

$$\mathcal{Y} := \{\mathrm{BOS} \circ \mathbf{v} \circ \mathrm{EOS} | \mathbf{v} \in \mathcal{V}^*\}, \qquad (1)$$

where $\circ$ is a string concatenation and $\mathcal{V}^*$ is the Kleene closure of a set of vocabulary $\mathcal{V}$. In practice, we set the maximum sequence length to $n_{\max}$ to limit the hypothesis space to $\mathcal{V}^{n_{\max}}$. The goal of decoding is to find the highest-scoring hypothesis for a given input.

2.1 Exhaustive Search

One of the most important objectives is the maximum a posterior (MAP) objective to find the most probable hypothesis among all:

$$\mathbf{y}^* := \operatorname*{arg\,max}_{\mathbf{y} \in \mathcal{Y}} \log p_\theta(\mathbf{y}|\mathbf{x}). \qquad (2)$$

We consider standard left-to-right autoregressive models for the model p_θ:

$$p_\theta(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{|\mathbf{y}|} p_\theta(y_t|\mathbf{x}, \mathbf{y}_{<t}). \qquad (3)$$

where each $p_\theta(y_t|\mathbf{x}, \mathbf{y}_{<t})$ is a distribution with support over a set of vocabulary and the EOS: $\bar{\mathcal{V}} = \mathcal{V} \cup \{\mathrm{EOS}\}$.

A straightforward solution to this problem is to maximize the MAP objective by exhaustively enumerating all possible hypotheses in $\mathcal{Y}$. Although it seems intuitive to use exhaustive search, prior work has pointed out several problems with this strategy. First, since the size of hypotheses set $|\mathcal{Y}|$ is extremely large, exhaustive search over $\mathcal{Y}$ is computationally infeasible. In fact, solving Eq. 2 is shown to be NP-hard (Chen et al., 2018). Second, even if we solve it optimally, the MAP objective often leads to low-quality results (Stahlberg and Byrne, 2019; Holtzman et al., 2020; Meister et al., 2020a).

2.2 Beam Search

A common heuristic to solve the decoding problem is greedy search, a local search with a greedy procedure. Greedy search sequentially chooses the token y_t at each time step t that maximizes $p(y_t|\mathbf{y}_{<t}, \mathbf{x})$ until the EOS token is generated or the maximum sequence length $n_{\max}$ is reached. Beam search is a generalization of greedy search where it selects the top k tokens at each step.

Let Y_t be the set of hypotheses at t-th step. Beam search is expressed as the following recursion:

$$\begin{aligned} Y_0 &= \{\mathrm{BOS}\}, \\ Y_t &= \operatorname*{arg\,topk}_{\mathbf{y} \in \mathcal{B}_t}(\log p_\theta(\mathbf{y}|\mathbf{x})) \end{aligned} \qquad (4)$$

where the candidate set $\mathcal{B}_t$ is defined as:

$$\mathcal{B}_t = \{\mathbf{y}_{<t} \circ y_t | y_t \in \bar{\mathcal{V}} \wedge \mathbf{y}_{<t} \in Y_{t-1}\}, \qquad (5)$$

for each $t > 0$. Beam search runs the recursion for a fixed number of iterations $n_{\max}$ and returns the set of hypotheses $Y_{n_{\max}}$. The most probable hypothesis (Eq. 2) in $Y_{n_{\max}}$ is the output of the decoding.

Many of the decoding strategies used in statistical machine learning systems are variants of beam search (Vijayakumar et al., 2018; Meister et al., 2021a; Anderson et al., 2017; Hokamp and Liu, 2017; King et al., 2022; Wan et al., 2023). Although beam search does not solve Eq. 2 exactly, it is a surprisingly useful strategy for NLP models. In many settings, beam search outperforms exhaustive search in terms of downstream evaluation (Stahlberg and Byrne, 2019; Holtzman et al., 2020; Meister et al., 2020a).

The drawback of beam search is that it is known to have high search errors due to the nature of local search (Stahlberg and Byrne, 2019). For example, previous work has reported degenerations such as repetitions and infinite-length outputs (Holtzman et al., 2020; Welleck et al., 2020).

2.3 Uniform Information Density

Meister et al. (2020a) explains the effectiveness of beam search by introducing the Uniform Information Density (UID) hypothesis. The UID hypothesis claims that communicative efficiency is maximized when information is distributed as uniformly as possible throughout the sequence (Levy, 2005; Levy and Jaeger, 2006). They study the information density of sentences generated by NMT systems quantitatively by measuring the amount of information conveyed by a word as surprisal (Hale, 2001). The surprisal u using a statistical language model is defined as follows:

$$u_0(BOS) = 0,$$
$$u_t(y) = -\log p_\theta(y|\mathbf{x}, \mathbf{y}_{<t}).$$

Meister et al. (2020a) shows that the variance of surprisals and BLEU have a strong relationship in their empirical evaluation of NMT models. They hypothesize that while restricting beam search leads to high search error in beam search, it also induces an inductive bias that may be related to promoting uniform information density, leading to the generation of higher quality sequences.

3 Lookahead Beam Search

To study the effect of search depth, we first introduce Lookahead Beam Search (LBS). LBS is a simple extension of beam search that deploys a lookahead strategy to optimize the multi-step score instead of the immediate score (Figure 2). In addition to the score given by the current partial hypothesis, LBS-d incorporates the maximum possible score achievable in the d-step future. We replace Eq. 4 with the following:

$$Y_0 = \{\text{BOS}\},$$
$$Y_t = \underset{\mathbf{y} \in \mathcal{B}_t}{\arg\operatorname{topk}}(\log p_\theta(\mathbf{y}|\mathbf{x}) + h_d(\mathbf{y})), \quad (6)$$

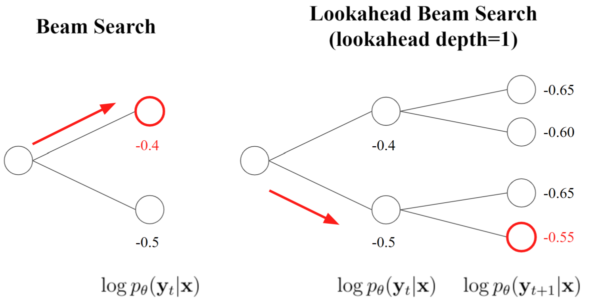


Figure 2: Comparison of Lookahead Beam Search and beam search. While beam search chooses the next hypotheses according to the current score of the hypothesis, lookahead beam search chooses them according to the current score plus the highest possible score achievable within d-step future.

where $h_d(\mathbf{y})$ is the highest score achievable of d-step future starting from $\mathbf{y}$. $h_d(\mathbf{y})$ is defined as:

$$h_d(\mathbf{y}_{1:t}) = \max_{\mathbf{y}_{1:t+d} \in \mathcal{B}_t^d} \log p_\theta(\mathbf{y}_{1:t+d}|\mathbf{x}, \mathbf{y}),$$
$$\mathcal{B}_t^d = \{\mathbf{y}_{1:t} \circ y_{t+1} \circ \ldots \circ y_{t+d} | \; y_{t+1}, \ldots, y_{t+d} \in \bar{\mathcal{V}}\}. \quad (7)$$

The lookahead depth d is the hyperparameter of the algorithm to control the locality of the search. The search becomes more local and shallow as d becomes smaller. In particular, if $d = 0$, it recovers beam search. The search becomes more exhaustive with larger d, and $d \geq n_{\max}$ recovers exhaustive search.

Proposition 1. *Lookahead Beam Search (LBS) is a generalization of beam search and exhaustive search. That is,*

1. *LBS-0 recovers beam search.*
2. *LBS-d with $d \geq n_{\max}$ recovers exhaustive search.*

The proof is immediate from the definition of LBS.

3.1 Implementation

A straightforward implementation to compute $h_d(\mathbf{y})$ is by a breadth-first search which needs to call the scoring function for $k|\bar{\mathcal{V}}|^d$ times per step. This is prohibitively expensive because the vocabulary size $|\mathcal{V}|$ is large in many tasks (e.g. >30000). To reduce the computation time, we implement the evaluation of h_d by best-first branch-and-bound search. Algorithm 1 describes the procedure of lookahead beam search. Since the scoring function

Algorithm 1: Lookahead Beam Search-d

Input: a set of hypotheses Y_{t-1} of length $t-1$
Output: a set of hypotheses Y_t of length t

$\mathcal{B} = \{\mathbf{y}_{t-1} \circ y | y \in \bar{\mathcal{V}}\}$
$\{\mathbf{y}_t^1, \mathbf{y}_t^2, ..., \mathbf{y}_t^b\} = \text{sort}(\mathcal{B})$ in a descending order of $p(\mathbf{y}_t^i|\mathbf{x})$
$Y' \leftarrow \emptyset$
$b \leftarrow |\bar{\mathcal{V}}|$
for $i \in \{1, ..., b\}$ **do**
  if $\log p_\theta(\mathbf{y}_t^i|\mathbf{x}) < \min \text{topk}_{\mathbf{y}\in Y'}(f(\mathbf{y}))$ **then**
    return $Y_t = \arg \text{topk}_{\mathbf{y}\in Y'}(f(\mathbf{y}))$
  end if
  $f(\mathbf{y}_t^i) \leftarrow \text{Eval}(\mathbf{y}_t^i, d, \min \text{topk}_{\mathbf{y}\in Y'}(f(\mathbf{y})))$
  if $f(\mathbf{y}_t^i) > \min \text{topk}_{\mathbf{y}\in Y'}(f(\mathbf{y}))$ **then**
    $Y' \leftarrow Y' \cup \{\mathbf{y}_t^i\}$
  end if
end for
return $Y_t = \arg \text{topk}_{\mathbf{y}\in Y'}(f(\mathbf{y}))$

Algorithm 2: $\text{Eval}(\mathbf{y}_t, d, f_{\max})$

Input: a hypothesis $\mathbf{y}_t$, a depth d, and a threshold $f_{\max}$
Output: a score of the hypothesis $h_d(\mathbf{y})$

if $d = 0$ **then**
  return $\log p_\theta(\mathbf{y}_t|\mathbf{x})$
end if
$\mathcal{B} = \{\mathbf{y}_t \circ y | y \in \bar{\mathcal{V}}\}$
$\{\mathbf{y}_{t+1}^1, \mathbf{y}_{t+1}^2, ..., \mathbf{y}_{t+1}^b\} = \text{sort}(\mathcal{B})$ in a descending order of $\log p_\theta(\mathbf{y}_{t+1}^i|\mathbf{x})$
for $i \in \{1, ..., b\}$ **do**
  if $\log p_\theta(\mathbf{y}_{t+1}^i) < f_{\max}$ **then**
    return $f_{\max}$
  end if
  $f_i \leftarrow \text{Eval}(\mathbf{y}_{t+1}^i, d-1, f_{\max})$
  if $f_i > f_{\max}$ **then**
    $f_{\max} \leftarrow f_i$
  end if
end for
return $f_{\max}$

is monotonically decreasing (Meister et al., 2020b), we can prune a partial hypothesis that is lower than the current k-th largest score before expanding the hypothesis further. The $\min \text{topk}$ returns the k-th largest score among Y' if $|Y'| \geq k$ and negative infinity otherwise. We explore the candidates in best-first order – the hypothesis with the highest score is explored first. In this way, we have a higher chance of pruning the less promising hypothesis, thus reducing computation. Because it only prunes paths which has no chance of getting into the top-k, it is guaranteed to find the same h_d as breadth-first search.

4 Experiments

To study the effect of search depth, we evaluate LBS on decoding neural machine translation (NMT) models. Experiments are performed on WMT'14 En-Fr and En-De datasets (Bojar et al., 2014). We evaluate the text quality by BLEU (Papineni et al., 2002) using the SacreBLEU system (Post, 2018).[1] For reproducibility, we use the Transformer-based pretrained models provided by fairseq (Ott et al., 2019).[2] We build the decoding framework in SGNMT (Stahlberg et al., 2017).[3] Due to the long duration (Table 6) and computational constraints, we present the evaluation on the first 100 sentences. We evaluate with a beam width of $k \in \{5, 10, 15, 20\}$. To reduce the computational load of the experiment, we prune lookahead branches except for the top-k_l scoring branches. Although it no longer guarantees to find the h_d, we observe that the BLEU score of LBS-1 with $k_l = 3k$ is the same as the LBS-1 with $k_l = \infty$ for $k = 1, 5$ using the first 10 sentences of WMT'14 En-Fr, so we expect it to be a valid approximation of the exact LBS-1.

[1] https://github.com/mjpost/sacrebleu
[2] https://github.com/facebookresearch/fairseq/tree/main/examples/translation
[3] https://github.com/ucam-smt/sgnmt

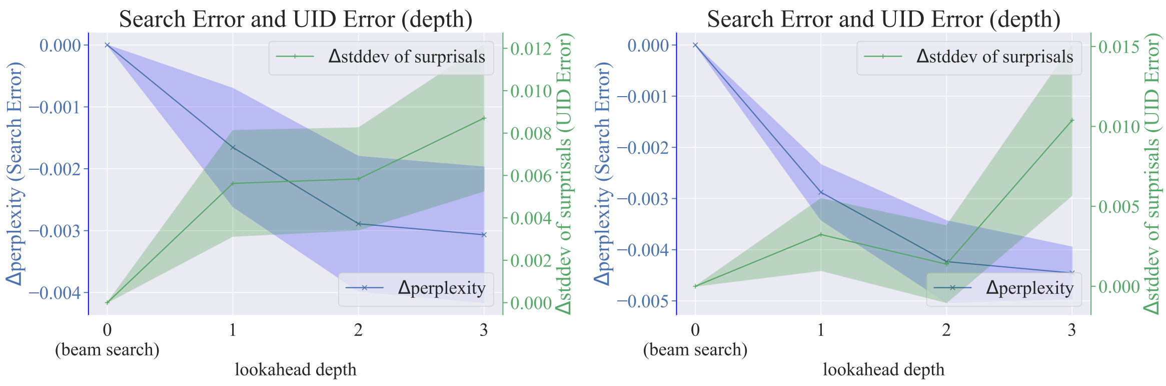
4.1 Analysis of Search Depth

The summary of the analysis is as follows.

- BLEU scores are slightly improved with $d = 1, 2$, and 3 lookaheads (Figure 1). However, a lookahead depth of 3 has diminished return compared to $d = 2$.

- We observe a trade-off between **search error** and **UID error** with varying lookahead depth (Figure 4). Although search error is decreased with larger lookahead depth, UID error is increased at the same time. This is analogous to the observation of beam width by Meister et al. (2020a).

- Lookahead depths of up to 3 have little effect on sequence length, while beam width has a strong negative correlation with it (Figure 7).



(a) WMT'14 En-Fr

(b) WMT'14 En-De

Figure 3: Difference in perplexity (search error) and average standard deviation of surprisals per sequence (UID error) of lookahead beam search (LBS) compared to beam search. The bold line represents the mean over the beam widths ($k \in \{5, 10, 15, 20\}$). The shaded area shows the standard error. Evaluated on the first 100 sentences of WMT'14 En-Fr and En-De.

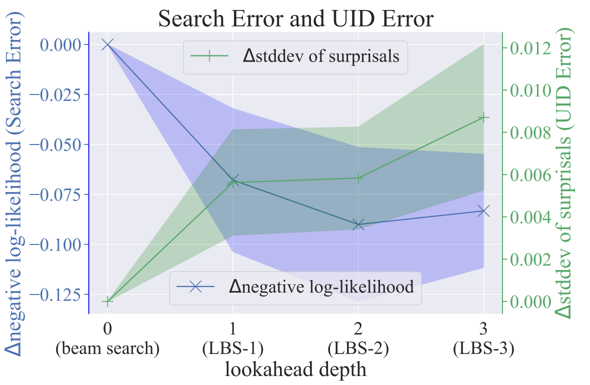


Figure 4: Difference in negative log-likelihood (search error) and average standard deviation of surprisals per sequence (UID error) of lookahead beam search (LBS) compared to beam search. The bold line represents the mean over the beam widths ($k \in \{5, 10, 15, 20\}$). The shaded area shows the standard error. Evaluated on the first 100 sentences of WMT'14 En-Fr.

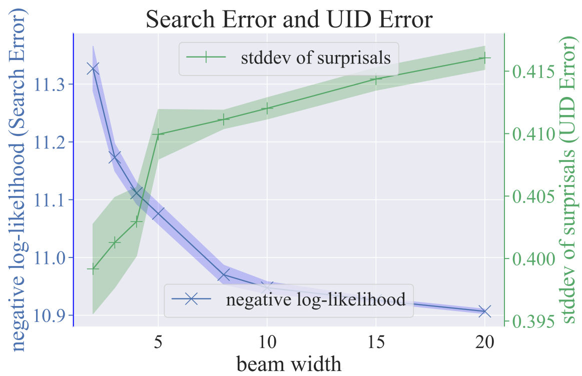


Figure 5: Negative log-likelihood (search error) and the average standard deviation of surprisals per sequence (UID error) by lookahead beam search (LBS). The bold line represents the mean over lookahead depth of $d \in \{0, 1, 2, 3\}$. The shaded area shows the standard error. Evaluated on the first 100 sentences of WMT'14 En-Fr.

4.1.1 BLEU Score

Figure 1 demonstrates how the lookahead strategy affects the quality of the results as the lookahead depth varies on WMT'14 En-Fr. In particular, LBS-2 achieves the best overall BLEU score. We observe a reduced improvement with a lookahead depth of 3 (LBS-3) compared to LBS-2. The BLEU score of the En-De dataset is present in Table 1. For En-De, the highest BLEU score is achieved with a beam width of 5 and a lookahead depth of 1 or 2. In both datasets, LBS achieves a better BLEU score than beam search in all widths. Interestingly, the advantage of LBS over beam search is reduced with $d = 3$. We also evaluate an exhaustive search (MAP decoding) which corresponds to LBS with $d = \infty$ (Table 2). As observed in previous work (Stahlberg and Byrne, 2019), the BLEU score drops significantly with an exhaustive search.

4.1.2 Why is there a "sweet spot" for lookahead depth?

We observe that a lookahead depth of $d = 2$ outperforms $d = 0, 1$, and 3 (Figure 1). The question is why there is a "sweet spot" for lookahead depth. Our hypothesis is that this phenomenon can be explained by the trade-off between the *search error* and the *UID error*. We measure the search error per token and per sentence using two metrics, the loss of perplexity (Figure 3) and the negative log-

WMT'14 En-Fr				
Decoder	$k = 5$	$k = 10$	$k = 15$	$k = 20$
beam	35.3	35.5	35.4	35.2
LBS-1	35.8	35.7	35.6	35.5
LBS-2	**36.1**	**36.1**	**35.7**	**35.7**
LBS-3	35.7	35.9	35.6	35.4
WMT'14 En-De				
Decoder	$k = 5$	$k = 10$	$k = 15$	$k = 20$
beam	22.7	21.9	22.0	21.8
LBS-1	**23.2**	22.6	22.2	21.5
LBS-2	**23.2**	**22.7**	22.6	22.6
LBS-3	23.0	**22.7**	**23.0**	**23.0**

Table 1: Evaluation of lookahead beam search on the first 100 sentences of WMT'14 En-Fr and En-De datasets. The best for each beam width is bolded. The best for each dataset is underlined.

Dataset	En-Fr	En-De
BLEU	2.2	6.0
sequence length	9.169	16.217
negative log-likelihood	8.195	8.246
stddev of surprisals	0.291	0.486

Table 2: Results of exhaustive search (i.e. LBS-∞) on the first 100 sentences of WMT'14 En-Fr and En-De datasets.

likelihood compared to beam search (4). We observe that increasing the lookahead depth reduces the search error measured by both the perplexity and the negative log-likelihood on both datasets. A prior study reports that the deviation from uniform information density measured by the standard deviation of surprisals has a negative correlation with the BLEU score (Meister et al., 2020a). We report the standard deviation of surprisals as UID error in Figure 3 and 4 (right axis). We observe a negative correlation between lookahead depth and the standard deviation of surprisals.

Overall, the result shows that deeper lookaheads improve the search error, but at the cost of higher UID error at the same time. We speculate that a lookahead depth of 2 happens to be a better trade-off between search error and UID error in our experimental setting. We also observe a similar trend for beam width (Figure 5), as indicated by Meister et al. (2020a).

Does Search error alone explain the results? We observe that increasing the lookahead depth tends to improve both the perplexity and negative log-likelihood (Figure 3 and 4). Therefore, search error, measured as both negative log-likelihood and perplexity, decreases with increasing lookahead depth. Thus, search error *alone* does not explain why $d = 2$ has the highest BLEU score.

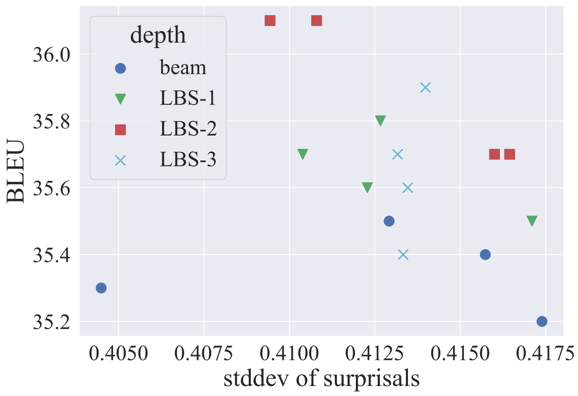


Figure 6: Average standard deviation of surprisals per sequence (UID error) and BLEU with lookahead beam search (LBS) for different beam widths and lookahead depths (WMT'14 En-Fr).

Does UID error alone explain the results? Figure 6 shows the standard deviation of surprisals and BLEU for different numbers of lookahead depths and beam widths. Although LBS has higher BLEU scores than beam search, it also has a higher average standard deviation of surprisals per sentence. Therefore, the UID error *alone* cannot account for the effect of lookahead depth on the BLEU scores.

4.1.3 Does searching deeper result in shorter output?

Previous studies reported that beam search with larger widths is likely to result in shorter sequences (Koehn and Knowles, 2017; Stahlberg and Byrne, 2019; Holtzman et al., 2020). To see the effect of search depth on length, we show the average length of the output sequences in Figure 7. We observe that while widening the beam reduces the output sequence length, deepening the lookahead by up to 3 steps does not. The correlation of beam width and lookahead depth with sequence length is -0.92 and 0.12, respectively. While beam width has a clear negative correlation with output sequence length, lookahead depth has little effect on sequence length. Thus, the length bias is unlikely to be the reason why BLEU score decreases with $d = 3$.

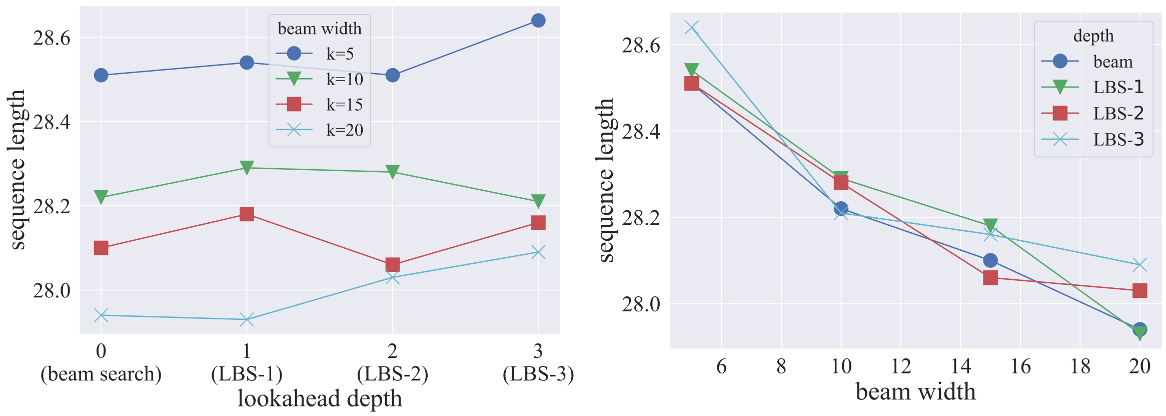


Figure 7: Average sequence length for varying lookahead depth and beam width (WMT'14 En-Fr). The correlation of lookahead depth and beam width with the average sequence length is 0.12 and -0.92, respectively.

Decoder	$k=1$	$k=2$	$k=5$	$k=10$
beam	33.6	34.5	**34.6**	34.9
LBS-1	**34.6**	34.8	34.1	**35.3**
LBS-2	33.9	**35.0**	**34.6**	35.0
LBS-3	33.4	34.0	34.4	34.5

Table 3: BLEU on the first 100 sentences of WMT'14 En-Fr using a fully convolutional decoder. The best for each beam width is bolded. The best score over all the conditions is underlined.

WMT'14 En-Fr				
Decoder	$k=1$	$k=5$	$k=10$	$k=15$
beam	34.8	35.8	36.0	**36.0**
LBS-1	**35.2**	**35.9**	**36.1**	35.9
WMT'14 En-De				
Decoder	$k=1$	$k=5$	$k=10$	$k=15$
beam	28.6	29.3	29.0	28.9
LBS-1	**28.8**	**29.4**	**29.2**	**29.0**

Table 4: BLEU on the entire dataset on WMT'14 En-Fr and En-De. The best for each beam width is bolded. The best for each dataset is underlined.

4.1.4 Is the result specific to the Transformer model?

To test the effect of the lookahead strategy on non-Transformer models, we evaluate the performance of LBS on a fully convolutional decoder proposed by Gehring et al. (2017). For reproducibility, we use the pretrained model provided by fairseq.[4] Table 3 reports the BLEU score. We observe that LBS-1 with $k = 10$ achieves the best score. Similar to the results of Transformer models, LBS achieves the same or higher BLEU scores in all widths of beam search.

4.1.5 Extended Evaluation of LBS-1

To evaluate the lookahead strategy more precisely, we evaluate LBS-1 on the entire WMT'14 En-Fr and En-De dataset. Due to computational constraints, we present only the evaluation of LBS-1. Table 4 reports the BLEU score. We observe that LBS-1 achieves slightly higher BLEU compared to beam search except for En-Fr with $k = 15$. In both datasets, LBS-1 achieves the highest BLEU score.

4.2 Running Time

Table 5 reports the number of calls to the scoring function (e.g. probabilistic model) by lookahead beam search. We observe that the number of calls grows rapidly with increasing lookahead depth. The wall-clock time of LBS is also significantly larger than beam search especially when the lookahead depth is large. As the evaluation is the most time-consuming operation of the decoding, the wall-clock time is roughly proportional to the number of calls (Figure 8). Note that the wall-clock time is heavily dependent on the hardware, so the values should be taken as a reference point rather than an absolute measure. As a reference, all the experiments are performed on `g4dn.xlarge` instances on AWS EC2 (4 vCPU cores, 16 GB memory, and an NVIDIA T4 GPU).

[4] `https://github.com/facebookresearch/fairseq/tree/main/examples/translation`

Decoder	$k=5$	$k=10$	$k=15$	$k=20$
beam	145.86	291.38	436.07	580.99
LBS-1	718.02	1841.83	3142.28	4590.62
LBS-2	2103.73	6270.25	11630.90	17946.10
LBS-3	4656.60	14471.00	27364.80	42597.90

Table 5: Average number of calls to the scoring function (probabilistic model) per sentence (WMT'14 En-Fr).

Decoder	$k=5$	$k=10$	$k=15$	$k=20$
beam	2.04	3.98	5.44	7.59
LBS-1	22.93	58.55	90.45	138.31
LBS-2	53.91	165.13	302.62	482.58
LBS-3	102.71	329.69	640.80	999.73

Table 6: Average running time (sec) per sentence (WMT'14 En-Fr). Note that the wall-clock time is heavily dependent on the hardware.

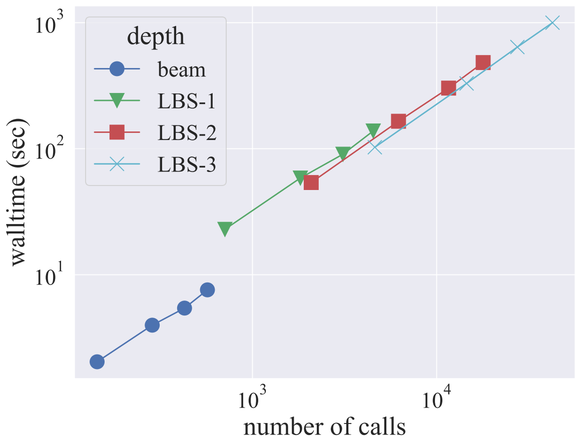


Figure 8: Comparison of the average number of calls to the scoring function to the wall-clock time (WMT'14 En-Fr).

5 Related Work

The phenomenon that using a larger beam leads to worse performance has been analyzed in a number of studies (Koehn and Knowles, 2017; Murray and Chiang, 2018; Yang et al., 2018; Stahlberg and Byrne, 2019; Cohen and Beck, 2019; Leblond et al., 2021). Many of the authors observe that widening the beam search degrades performance due to a bias in sequence models to favor shorter sequences even with a length penalty. Other authors have investigated why beam search successfully generates high quality sequences. The uniform information density hypothesis (Levy, 2005; Levy and Jaeger, 2006) is introduced to explain why beam search outperforms exhaustive search (Meister et al., 2020a, 2021b). They hypothesize that narrowing the width of beam search induces a bias in the decoding that enforces uniform information density, resulting in higher quality sequences. Although many have studied the width of the beam search, little is known about the depth of the search. Our work extends the analysis to the search depth and observes a similar trade-off between search and UID error, which is balanced by the lookahead depth parameter.

Some authors have studied lookahead strategies for decoding. Hargreaves et al. (2021) investigates the greedy roll-out strategy to apply reranking during decoding instead of only at the end. Lu et al. (2022) evaluated several lookahead strategies to estimate the future score of the given partial hypothesis. Several works have investigated the lookahead strategy for constraint sentence generation tasks using Monte Carlo sampling (Miao et al., 2019; Zhang et al., 2020; Leblond et al., 2021). Our analysis provides a fundamental insight into why these lookahead strategies can be effective.

This work focuses on the quality of the text evaluated by its similarity to the reference text. Previous work has investigated other factors such as diversity (Vijayakumar et al., 2018; Meister et al., 2021a), constraints (Anderson et al., 2017; Hokamp and Liu, 2017), or faithfulness (King et al., 2022; Wan et al., 2023). How the lookahead strategy affects these factors is an open question.

6 Conclusion

To study the effect of search depth on the performance of decoding strategies for text generation models, we introduce Lookahead Beam Search (LBS). LBS is a generalization of beam search and exhaustive search that allows control of the lookahead depth by its hyperparameter. We observe that increasing lookahead depth reduces search error but increases UID error, similar to the observation reported by Meister et al. (2020a) for increasing beam width. LBS with a lookahead depth of 1 to 3 slightly improves upon beam search in machine translation tasks. This is analogous to the empirical observation that a beam width of a certain size often improves upon beam width of 1 (i.e. greedy search). The results indicate room for improvement orthogonal to width by searching deeper.

7 Limitations and Risks

All the experiments are conducted on machine translation tasks. Although we expect the effect of the search depth is not specific to machine translation, it is not evaluated on other text generation tasks.

The primary focus of the study is on analyzing the effect of the lookahead strategy, not on proposing a new practically useful decoding algorithm. Because the inference of LBS is very slow compared to the beam search, it is not a practical option as is.

Due to limited computational resources, our experiments use only part of the dataset instead of the whole dataset. As a result, the scores are not directly comparable with the existing literature.

While language generation can be used for malicious purposes, we do not foresee any specific ethical concerns with the analysis in this paper beyond those discussed by Bender et al. (2021).

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark. Association for Computational Linguistics.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Ondřej Bojar, Christian Buck, Christian Federmann, Microsoft Research, Barry Haddow, Philipp Koehn, Jhu / Edinburgh, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut Google, and Lucia Specia. 2014. Findings of the 2014 workshop on statistical machine translation. In *the Workshop on Statistical Machine Translation*, pages 12–58.

Yining Chen, Sorcha Gilroy, Andreas Maletti, Jonathan May, and Kevin Knight. 2018. Recurrent neural networks as weighted language recognizers. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2261–2271, New Orleans, Louisiana. Association for Computational Linguistics.

Eldan Cohen and Christopher Beck. 2019. Empirical analysis of beam search performance degradation in neural sequence models. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1290–1299. PMLR.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

James Hargreaves, Andreas Vlachos, and Guy Emerson. 2021. Incremental beam manipulation for natural language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2563–2574, Online. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Daniel King, Zejiang Shen, Nishant Subramani, Daniel S. Weld, Iz Beltagy, and Doug Downey. 2022. Don't say what you don't know: Improving the consistency of abstractive summarization by constraining beam search. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 555–571, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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.

Rémi Leblond, Jean-Baptiste Alayrac, Laurent Sifre, Miruna Pislar, Lespiau Jean-Baptiste, Ioannis Antonoglou, Karen Simonyan, and Oriol Vinyals. 2021. Machine translation decoding beyond beam search. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8410–8434, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Roger Levy. 2005. *Probabilistic models of word order and syntactic discontinuity*. Ph.D. thesis, Stanford University.

Roger Levy and T. Florian Jaeger. 2006. Speakers optimize information density through syntactic reduction. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada*, pages 849–856. MIT Press.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. NeuroLogic a*esque decoding: Constrained text generation with lookahead heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 780–799, Seattle, United States. Association for Computational Linguistics.

Clara Meister, Ryan Cotterell, and Tim Vieira. 2020a. 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, Online. Association for Computational Linguistics.

Clara Meister, Martina Forster, and Ryan Cotterell. 2021a. Determinantal beam search. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6551–6562, Online. Association for Computational Linguistics.

Clara Meister, Tiago Pimentel, Patrick Haller, Lena Jäger, Ryan Cotterell, and Roger Levy. 2021b. Revisiting the Uniform Information Density hypothesis. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 963–980, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Clara Meister, Tim Vieira, and Ryan Cotterell. 2020b. Best-first beam search. *Transactions of the Association for Computational Linguistics*, 8:795–809.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. CGMH: constrained sentence generation by metropolis-hastings sampling. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA*, pages 6834–6842. AAAI Press.

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.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. 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, Eva Hasler, Danielle Saunders, and Bill Byrne. 2017. SGNMT – a flexible NMT decoding platform for quick prototyping of new models and search strategies. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 25–30, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

David Wan, Mengwen Liu, Kathleen McKeown, Markus Dreyer, and Mohit Bansal. 2023. Faithfulness-aware decoding strategies for abstractive summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2864–2880, Dubrovnik, Croatia. Association for Computational Linguistics.

Sean Welleck, Ilia Kulikov, Jaedeok Kim, Richard Yuanzhe Pang, and Kyunghyun Cho. 2020. Consistency of a recurrent language model with respect to incomplete decoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5553–5568, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Yilin Yang, 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, Brussels, Belgium. Association for Computational Linguistics.

Maosen Zhang, Nan Jiang, Lei Li, and Yexiang Xue. 2020. Language generation via combinatorial constraint satisfaction: A tree search enhanced Monte-Carlo approach. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1286–1298, Online. Association for Computational Linguistics.