Probabilistically-sound beam search with masked language models

Anonymous

Abstract

Beam search with masked language models (MLMs) is challenging in part because joint probability distributions over sequences are not readily available, unlike for autoregressive models. Nevertheless, estimating such distributions has applications in many domains, including protein engineering and ancient text restoration. We present probabilistically-sound methods for beam search with MLMs. First, we clarify the conditions under which it is theoretically-sound to perform text infilling with MLMs using standard beam search. When these conditions fail, we provide a probabilistically-sound modification with no additional computational complexity and demonstrate that it is superior to the aforementioned beam search in the expected conditions. We then present empirical results comparing several infilling approaches with MLMs across several domains.

1 Introduction

Autoregressive language models (LMs) have showcased exceptional ability across many tasks. Yet, in specific contexts where bidirectionality is crucial, such as protein language modeling and ancient text restoration, masked language models (MLMs) remain prevalent. However, MLMs still face a significant challenge in these settings: while MLMs learn conditional distributions of single tokens, applications to the aforementioned domains often require knowledge of the probability of multiple tokens jointly.

The key challenge lies in computing the joint distribution $p(\mathbf{x})$ over sequences $\mathbf{x}$ given only the MLM-learned conditionals $p(x_i|\mathbf{x}_{-i})$, where $\mathbf{x}_{-i}$ denotes the context sequence $\mathbf{x}$ with the entry at index i removed. The Hammersley-Clifford-Besag (HCB) theorem yields a direct algebraic construction of a joint distribution $p(\mathbf{x})$ which is valid in

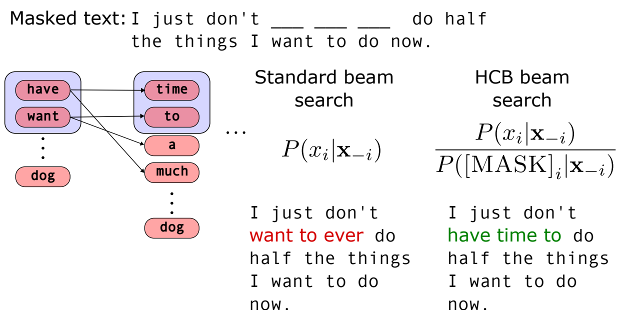

Figure 1: Overview of the proposed HCB beam search compared to standard beam search for text infilling.

the case when the MLM-learned distributions are *compatible*—that is, there exists a joint distribution which factors into the conditionals exactly (Hennigen and Kim, 2023).

In practical applications, a heuristic approximation based on the chain rule of autoregressive LMs is used for infilling with MLMs (Shen et al., 2020; Assael et al., 2022; Cowen-Breen et al., 2023b; Tran and Hy, 2023):

$$p(\mathbf{x}) \approx \prod_{i=1}^{n} p(x_i|\mathbf{x}_{:i}, [\mathbf{M}]_{i:}) \tag{1}$$

where the notation $[\mathbf{M}]_{i:}$ indicates that mask tokens are present from indices i onwards.

Our first contribution is demonstrating that Equation 1 is valid if and only if a *conditional independence assumption* about the MLM-learned conditionals is satisfied and providing the conditions under which this assumption holds, assuming compatibility. We state these conditions in Theorem 1:

Theorem 1 (Informal). *Suppose that p represents a model which achieves zero training loss on the MLM objective. Then, on the training distribution, the learned conditionals are both* compatible *and satisfy the* conditional independence assumption.

We hypothesize that there may be two regimes, as a consequence of theorem 1: a regime where training loss is small enough that the heuristic approach of Equation 1 may be reasonable, and a regime where the conditional independence assumption may not hold, in which another approach is needed.

Our second contribution is providing a modification of Equation 1 which holds for the second, possibly under-trained regime. Based on the HCB theorem, this modification relaxes the conditional independence assumption by including an adjustment term to correct for possible dependencies, and it requires no additional forward passes through the model.

Evidently, not all MLMs of interest lie in the regime where Equation 2 is reasonable: we find empirically that our modification outperforms standard beam search for certain models, including BERT-base. In developing RoBERTa, Liu et al. (2019) suggest that BERT was under-trained, which aligns with our experimental observation that the conditional independence assumption seems more reasonable for RoBERTa than for BERT.

2 Text infilling

Empirically, we evaluate Equation 1 and our modification of it on the task of *text infilling* involving predicting a missing span of text given its context, which is well-motivated in the domains of protein language modeling and ancient text restoration (Zhu et al., 2019).

Much existing work in text infilling focuses on developing custom architectures and training from scratch (Sun et al., 2017; Ippolito et al., 2019; Shen et al., 2020; Donahue et al., 2020). In some domains, however, pretrained MLMs may be available in instances where limits on available data or compute prohibit training new text infilling models. Manuscript restoration is a prototypical example: training data is often restricted by copyright (Graziosi et al., 2023), and full trainings can be computationally expensive, yet various pretrained MLMs for ancient languages are available (Bamman and Burns, 2020; Assael et al., 2022; Cowen-Breen et al., 2023a; Riemenschneider and Frank, 2023). Another example is protein language models (PLMs): the computational footprint of fine-tuning PLMs becomes a barrier for many research groups (Hu et al., 2021; Sledzieski et al., 2023). Despite the challenges posed by data and compute, effective infilling remains an important task for ancient text restoration and protein engineering.

Here, we examine the capabilities of MLMs to infill directly, primarily by infilling tokens sequentially through an adaptation of beam search to MLMs, although we compare additionally to other sampling schemes, such as nucleus sampling and sampling with temperature.

It should be noted that our methods only apply to settings where the number of missing tokens is fixed, as opposed to the more general setting addressed by Shen et al. (2020), among others. That said, fixing the number of missing tokens is reasonable in the case of infilling damaged texts or missing amino acids.[1]

3 Background and related work

3.1 Beam search

Beam search is a form of decoding which incrementally adds successive tokens to a set of candidate token sequences, maintaining a fixed number of candidate sequences $\mathbf{x} = (x_1, \ldots, x_n)$ of highest joint probability $p(\mathbf{x})$. For autoregressive models, the chain rule,[2]

$$\log p(\mathbf{x}) = \sum_{i=1}^{n} \log p(x_i | \mathbf{x}_{:i}), \qquad (2)$$

allows for the implementation of beam search shown in Algorithm 1.

Algorithm 1 Autoregressive beam search. Given a beam size $B > 0$, returns a collection of generated sequences S of length n.

Initialize $S = \{(0, \emptyset)\}$
for $i \in \{1, \ldots, n\}$ **do**
  for $(\ell, (x_1, \ldots, x_{i-1})) \in S$ **do**
    $f(\cdot) \leftarrow \log p(\cdot | x_1, \ldots, x_{i-1})$ (1 forward pass.[3])
    Append to S: $(\ell + f(x), (x_1, \ldots, x))$ for every x.
  end for
  $S \leftarrow$ {the B sequences $(x_1, \ldots, x_i)$ of S w/ highest ℓ}
end for
return S

3.2 Challenges with MLM beam search

A major barrier to conducting beam search with MLMs is that MLMs are not language models *a priori*, and thus it is less obvious that a joint distribution $p(\mathbf{x})$ exists and can be computed in terms

[1] In the case of damaged inscriptions, domain experts posit an estimated number of missing characters based on physical distance (Bruun and Edmondson, 2014).

[2] In what follows, we generally omit right-context tokens from $p(\cdot|\cdot)$ for readability, but we note that modified forms of all of the following equations hold when p is further conditioned on right-context.

of known quantities. If one is willing to tolerate compatibility and a **conditional independence assumption**[4] of the form

$$p(x_i|\mathbf{x}_{:i}, [\mathbf{M}]_{i:}) \approx p(x_i|\mathbf{x}_{:i}) \tag{3}$$

then Equation 2 implies an expression for the joint:[5]

$$\log p(\mathbf{x}) \approx \sum_{i=1}^{n} \log p(x_i|\mathbf{x}_{:i}, [\mathbf{M}]_{i:}) \tag{4}$$

where we use the notation $[\mathbf{M}]_{i:k}$ to indicate that [MASK] tokens occupy the indices from i to k. Equation 4 is the foundation for the implementation of beam search shown in Figure 2 with the standard scoring function.

The approximation in Equation 3 is equivalent to the assumption that the distribution of x_i conditioned on the given context $\mathbf{x}_{:i}$ is independent of the information that mask tokens occupy the indices from i to n. It is unlikely that this assumption holds true in practice,[6] as passing mask tokens to the model will alter the output distribution in general.

Therefore, to be probabilistically sound, this equation should include a term correcting for the potential dependency between x_i and $[\mathbf{M}]_{i:}$. Including this term conveniently incurs almost no additional computational cost, to be described in section 4.

3.3 Constructing joint distributions from conditionals

Approximating the joint distributions of MLMs is an active area of research. Hennigen and Kim (2023) compare several joint distribution approximation schemes and find that when the MLM-learned conditional distributions are compatible—that is, there exists a joint distribution which factors into the conditionals exactly—the HCB theorem provides a direct algebraic construction of a joint distribution $p(\mathbf{x})$ from a set of conditionals $p(x_i|\mathbf{x}_{-i})$, up to a normalizing constant.

Theorem 2 (HCB). *Suppose that p is a probability distribution with full support over the space A^n of n-element sequences over an arbitrary alphabet A. Then for any two sequences $\mathbf{x}, \mathbf{y} \in A^n$,*

$$\frac{p(\mathbf{x})}{p(\mathbf{y})} = \prod_{i=1}^{n} \frac{p(x_i|x_1, \ldots, x_{i-1}, y_{i+1}, \ldots, y_n)}{p(y_i|x_1, \ldots, x_{i-1}, y_{i+1}, \ldots, y_n)}$$

Theorem 2 yields an immediate expansion for $\log p(\mathbf{x})$ which is analogous to Equation 2:

$$\log p(\mathbf{x}) \sim \sum_{i=1}^{n} \log p(x_i|\mathbf{x}_{:i}, \mathbf{y}_{i:}) - \log p(y_i|\mathbf{x}_{:i}, \mathbf{y}_{i:}) \tag{5}$$

where $\sim$ indicates equality up to addition of a constant in $\mathbf{x}$. Following Hennigen and Kim (2023), we refer to $\mathbf{y}$ as the **pivot**. When the conditional distributions are compatible, Equation 5 should yield orderings of sequences $\mathbf{x}$ by their probabilities p(x) in a manner which is consistent across choice of pivots.

In actuality, the conditional distributions learned by BERT do not appear to be compatible (Young and You, 2023; Hennigen and Kim, 2023). When the MLM-learned conditional distributions are not compatible, the Arnold-Gokhale (AG) construction provides an algorithm for returning the joint distribution which *most nearly* factors into the learned conditionals (Arnold and Gokhale, 1998). Hennigen and Kim (2023) find that the AG construction achieves the lowest perplexity when compared to a number of baselines; however, it is severely limited by the computational cost it incurs: memory requirements of V^n for a vocabulary of size V and n missing tokens.

4 Methods

Our primary observation is that the HCB theorem (Theorem 2) yields a straightforward correction to the standard beam search induced by Equation 4 which incurs almost no additional computational cost.[7] Although the conditional distributions learned by BERT are not exactly compatible, they are empirically compatible enough to improve the accuracy of beam search in certain instances. Our hope is that these methods will be useful for infilling when the implementation of the AG construction is intractable.

[4] This is an assumption that we investigate theoretically in section 7. Note that, while Equation 3 may seem intuitively clear, it might be similarly intuitive that the conditionals learned by BERT are compatible, but this is far from true (Young and You, 2023).

[5] For a more detailed derivation, see Appendix A.

[6] See theorem 3 for a sufficient condition.

[7] Code for our experiments can be found at https://anonymous.4open.science/r/hcb_beam_search/

Algorithm 2 Infilling beam search with a MLM. Given $\mathbf{x}$, a sequence of length n with masked positions $j, ..., k$ and a beam size $B > 0$, return a collection of generated sequences S with masked positions filled in from vocabulary V. Uses a scoring function $f(\cdot)$ to evaluate a candidate beam extension.

```
Initialize S = {(0, ∅)}
for j ∈ {i, ..., k} do
    for (ℓ, (x_i, ..., x_{j−1})) ∈ S do
        Append to S: (ℓ + f(x), (x_i, ..., x)) for every x ∈ V.
    end for
    S ← {the B sequences (x_j, ..., x_i) of S w/ highest ℓ}
end for
return S
```

Scoring functions f for various beam search implementations:

$$\begin{aligned} \text{Standard}: \quad & f(\cdot) = \log p(\cdot|\mathbf{x}_{:i}, [\mathbf{M}]_{i:k}, \mathbf{x}_{k:}) \\ \text{HCB}: \quad & f(\cdot) = \log p(\cdot|\mathbf{x}_{:i}, [\mathbf{M}]_{i:k}, \mathbf{x}_{k:}) - \log p([\mathbf{M}]_i|\mathbf{x}_{:i}, [\mathbf{M}]_{i:k}, \mathbf{x}_{k:}) \\ \text{HCB with pivot } \mathbf{y}_{i:k}: \quad & f(\cdot) = \log p(\cdot|\mathbf{x}_{:i}, \mathbf{y}_{i:k}, \mathbf{x}_{k:}) - \log p(\mathbf{y}_i|\mathbf{x}_{:i}, \mathbf{y}_{i:k}, \mathbf{x}_{k:}) \end{aligned}$$

Based on Theorem 2 and Equation 5, we propose the following modification to Equation 3:

$$\log p(\mathbf{x}) \sim \sum_{i=1}^{n} \log p(x_i|\mathbf{x}_{:i}, [\mathbf{M}]_{i:}) - \log p([\mathrm{M}]|\mathbf{x}_{:i}, [\mathbf{M}]_{i:}) \tag{6}$$

where $\sim$ again indicates equality up to addition of a constant. This correction term guarantees probabilistic soundness of the type ensured by Equation 2 and requires no additional forward passes to compute, as the tensor $p(\cdot|\mathbf{x}_{:i}, [\mathbf{M}]_{i:})$ is computed with a single forward pass. Although the value of $p([\mathrm{M}]|\mathbf{x}_{:i}, [\mathbf{M}]_{i:})$ does not have an immediately obvious intuitive meaning, this quantity empirically contains enough signal[8] to demonstrate improvements over baselines and several ablations, as we demonstrate in section 6.

4.1 Choosing a pivot

For the purposes of sampling and infilling, the pivot $\mathbf{y}$ can be any sequence in the support of p. However, we find that some pivots lead to better performance than others (Figure 3). During the computation of Equation 6, the pivot $\mathbf{y}$ is injected into the text as context. Therefore, we find it important that the pivot is reasonably in-distribution as context for the model, regardless of the position where it is injected.

The MLM training procedure makes one particular choice of pivot especially in-distribution for any context: the sequence of mask tokens $\mathbf{y} = ([\mathrm{M}], \ldots, [\mathrm{M}])$. Throughout MLM training, mask locations are re-randomized, meaning that the MLM is likely to encounter sequences of masks as context in many positions across all examples in the train set. The probability $p([\mathrm{M}]|\mathbf{x}_{<i}, \mathbf{y}_{>i})$ decreases drastically during training, as $[\mathrm{M}]$ never occurs as a label.[9] Nonetheless, using this quantity improves infilling accuracy across different models in various domains. We perform ablation studies to verify this probability captures genuine information about the context, rather than random noise.

5 Experimental setup

5.1 Models

For English text data, we use the MLMs (# params) BERT-base (110M), BERT-large (340M), DistilBERT (66M), and RoBERTa (125M) (Devlin et al., 2019; Sanh et al., 2019; Liu et al., 2019). For ancient texts, we use the MLMs Ithaca (49M) and Desformers (126M) (Assael et al., 2022; DeVaul, 2023), two character-level BERT models trained on ancient Greek. For protein language modeling, we use ESM2 (8M) (Lin et al., 2023).

5.2 Metrics

To assess text infilling results, we use top-k accuracy, BLEU score[10], and BERTScore (Papineni

[8] In particular, we find that it depends strongly on the context, $\mathbf{x}_{:i}$.

[9] See Figure 7 for how $p([\mathrm{M}])$ changes during BERT training

[10] When there are only $k < 4$ tokens to infill, we employ BLEU-k.

et al., 2001; Zhang et al., 2020). We do not compute perplexity, which, for arbitrary scoring schemes, requires calculation of an intractable normalization constant. In contrast, top-k accuracy and BLEU are directly computable. Following Shen et al. (2020), we use top-1 predictions of each infilling scheme and the ground truth span to compute BLEU score and BERTScore.

5.3 Datasets

In English, we perform infilling experiments on three datasets: the Brown corpus (Nelson Francis and Kucera, 1979), the Stanford Natural Language Inference dataset (SNLI) (Bowman et al., 2015), and the extreme summarization dataset (XSUM) (Narayan et al., 2018). This allows us to additionally test varying amounts of context: the datasets have an average of 452, 19, and 28 tokens of context per example, respectively. For ancient language models, we perform infilling experiments on the Packhard Humanities Institute's database of ancient Greek inscriptions[11], collected by Sommerschield* et al. (2021). For protein language models, we perform infilling experiments on a subset of protein sequences from UniProt (UniProt, 2008).

Each experiment consists of selecting a random subset of k contiguous indices from a test example and performing infilling according to HCB beam search. We run experiments with varying values of k, beam size, and pivot choices. Additionally, we take advantage of MLMs' non-autoregressive nature and infill tokens in order of highest MLM confidence (best-to-worst), following Schick and Schütze (2021) and Assael et al. (2022).

5.4 Baselines

We compare HCB beam search (both left-to-right and best-to-worst) to a number of zero-shot infilling baselines. The first is the standard MLM beam search of Algorithm 2 with the "Standard" f, for which we also consider both left-to-right and best-to-worst beam searches.

We also compare our method to adapted versions of several popular sampling schemes including nucleus sampling (Holtzman et al., 2020) and sampling with temperature (Ackley et al., 1985). Since such schemes are designed to *sample*, instead of to *search*, they inherently involve fewer forward passes through the model than beam search. We therefore specifically compare HCB beam search with beam size B to these hybrid sampling-search schemes in which we sample and store in memory B candidate samples for each token. This approach ensures that the same number of forward passes is used by each scheme, therefore resulting in the same time and space complexity of each method.

[11] https://inscriptions.packhum.org/

5.5 Ablations

To test that the estimated probability of the mask token $p([\mathrm{M}])$ is not simply random noise, as we found a pivot of mask tokens is crucial for the success of HCB beam search, we perform two ablations.

Ablation 1 (Context Scramble): First, we test the hypothesis that $p([\mathrm{M}]|\mathbf{x})$ is sensitive to the given context $\mathbf{x}$. To do so, we track the values of the last 1,000 calls to $p([\mathrm{M}]|\mathbf{x}^{(i)})$ for the most recent contexts seen $(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(1{,}000)})$, and for the ablation, we replace the value of $p([\mathrm{M}]|\mathbf{x})$ in Algorithm 2 ("HCB" f) with a random element of the previous 1,000 calls.

Ablation 2 (Random Token Swap): Second, we test the hypothesis that $p([\mathrm{M}]|\mathbf{x})$ is sensitive to the input $[\mathrm{M}]$. To do so, we replace $[\mathrm{M}]$ with a completely random token y, and replace the computation of $p([\mathrm{M}]|\mathbf{x})$ in HCB beam search with $p(y|\mathbf{x})$.

6 Results

6.1 English language models

In our English language experiments with BERT-base, we observe a consistent relative ranking of methods: HCB Best-to-Worst > HCB Left-to-Right > Standard Best-to-Worst > Standard Left-to-Right. This ranking persists across metrics (top-k accuracy and BLEU; Table 3), number of missing tokens (2 through 5; Table 2), datasets (Brown, SNLI, XSUM; Table 1), and beam sizes (5 and 20; Figure 2).

We observe that BERTScore values are largely consistent across beam search methods (Table 3), likely due to the relatively small number of tokens being infilled in each example, but HCB consistently outperforms standard beam search.

Note that standard left-to-right beam search outperforms our nucleus sampling-beam search hybrid, consistent with the findings of (Shaham and Levy, 2022), as well as all other sampling-based methods.

Model		BERT-base	RoBERTa
$\mathbb{B}$	HCB	**11.43**	25.29
	Standard	10.63	**25.81**
$\mathbb{S}$	HCB	**7.49**	13.66
	Standard	7.08	**14.56**
$\mathbb{X}$	HCB	**9.86**	23.21
	Standard	9.18	**24.52**

Table 1: Comparison of HCB vs. standard beam search on the Brown $\mathbb{B}$, SNLI $\mathbb{S}$ and XSUM $\mathbb{X}$ datasets, across models. We only show best-to-worst, since left-to-right was strictly worse in each case.

Across models, we note that the various beam search methods rank similarly when using BERT-large and DistilBERT, but RoBERTa stands out as a case where standard beam search outperforms HCB beam search. For a given model, we also find that the ranking of methods is consistent across values of k for top-k accuracy (Figure 2).

6.2 Domain-specific language models

Ancient texts. In experiments infilling two missing characters with Desformers, standard beam search outperforms HCB beam search in top-1 accuracy, but HCB shows significant improvements for larger top-k values (Table 5). However, HCB's performance with this model decreases when infilling larger gaps relative to the standard method.[12] With Ithaca, HCB beam search shows consistent improvement over standard beam search across different gap sizes (Table 5).

Protein sequences. In experiments infilling between two and five missing amino acids in a protein sequence using ESM2, we find that HCB beam search shows comparable, albeit lower performance to standard beam search (Appendix B, Table 9). Thus, it appears that for protein sequences, the conditional independence assumption may hold better than for human language, leading to relatively strong standard beam search performance; this is further explored in section 7.

6.3 Pivot design

To explore how performance of HCB beam search depends on the choice of pivot, we extensively tested all two-character pivots for seven-character infills with a beam size of 20 for Ithaca. Testing every possible pivot choice is made possible by Ithaca's small 34-token vocabulary. With over 800 trials per pivot, we observe that the special token "-", which corresponds to a missing character, is very clearly the best performing pivot choice, achieving accuracy similar to standard beam search. Two other special tokens "#" and "<" perform about 20% worse as HCB pivots, and the 31 remaining tokens perform over 50% worse than standard beam search (Figure 3). We also exhaustively tested all pivots for ESM2, observing that the best choice of pivot appears to be the token "[CLS]".

Notably, Ithaca's architecture does not include the mask token in its output probability distribution; thus the missing character token, which is scattered randomly throughout train and test examples, is the token which remains most in-distribution when injected as a pivot into random positions. Similarly, for ESM2, we observe that some special tokens perform noticeably better as pivots than regular tokens. These results could suggest that special tokens, which do not add disruptive context, are better candidates for pivots.

6.4 Ablations

Ablation 1 dramatically worsens the performance of HCB beam search, as we show in Table 7. Ablation 2 also deteriorates the performance of HCB beam search, although by less than that of ablation 1.

We conclude that $p([\mathrm{M}]|\mathbf{x})$ is sensitive to both the context $\mathbf{x}$ and input token $[\mathrm{M}]$, and hypothesize that this contributes to the success of HCB beam search. Since results are far worse when we randomly replace $\mathbf{x}$ than when we randomly replace $[\mathrm{M}]$, we hypothesize that the context $\mathbf{x}$ is a more important factor to the success of HCB beam search.

7 Discussion

In Section 6, we see that HCB beam search achieves comparable performance to standard beam search across languages and that superiority of either method is usually model-dependent, rather than data-dependent: for instance, HCB beam search consistently outperforms standard beam search with BERT-base, but consistently underperforms with RoBERTa, across datasets, as shown by Table 1.

One plausible hypothesis is that each beam search variant performs well when certain assumptions hold, and the validity of these assumptions is determined by the training procedure. For instance,

[12] On gap sizes larger than three characters, HCB is inferior to standard beam search across all experiments with this model.

	Top-1 Accuracy (%)				Top-5 Accuracy (%)			
Number of missing tokens	2	3	4	5	2	3	4	5
HCB Left-to-Right	28.84	10.46	3.23	0.92	38.88	15.01	4.75	1.36
HCB Best-to-Worst	**29.76**	**11.29**	**3.66**	**1.02**	**39.73**	**16.2**	**5.4**	**1.56**
Standard Left-to-Right	26.43	8.93	2.69	0.74	36.8	13.77	4.3	1.21
Standard Best-to-Worst	28.27	10.21	3.18	0.86	38.18	15.08	4.88	1.36
Pure sampling	23.71	6.77	1.58	0.3	28.78	7.92	1.73	0.33
Sampling with T=0.25	26.1	8.41	2.46	0.68	34.23	11.08	3.18	0.84
Sampling with T=0.50	25.89	8.23	2.22	0.68	33.38	10.61	2.96	0.78
Sampling with T=0.75	25.09	7.63	1.99	0.46	31.54	9.42	2.43	0.53
Nucleus sampling, p=0.9	24.31	7.23	1.87	0.4	27.91	8.49	2.08	0.45

Table 2: Top-k % accuracy at infilling consecutive missing tokens on 100K examples from Brown corpus, using BERT-base with beam size 5 and number of missing tokens ranging from 2 to 5.

	BLEU				BERTScore F1			
Number of missing tokens	2	3	4	5	2	3	4	5
HCB Left-to-Right	28.74	10.33	3.25	0.93	**99.47**	**99.02**	98.66	**98.36**
HCB Best-to-Worst	**29.55**	**11.13**	**3.63**	**1.03**	99.46	**99.02**	**98.72**	98.32
Standard Left-to-Right	26.42	8.94	2.68	0.74	99.39	98.96	98.63	98.29
Standard Best-to-Worst	28.15	10.05	3.12	0.89	99.43	98.98	98.64	98.29

Table 3: BLEU score at infilling a random number (uniform between 2 and 5) of missing tokens on 100K examples from Brown corpus, across models, with beam size 5.

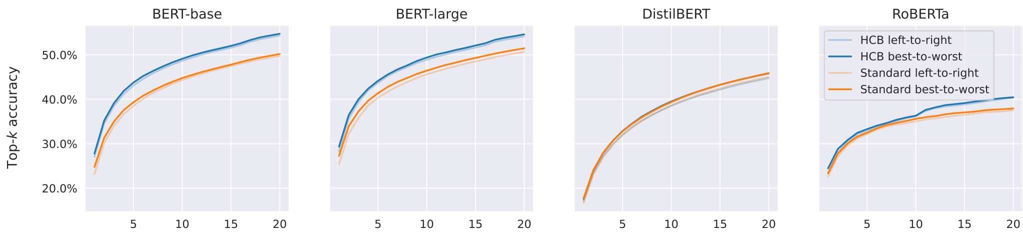

Figure 2: Performance of HCB beam search versus standard beam search methods across three English MLMs and DistilBERT, with beam size 20. Note that RoBERTa appears to be suffering from the *beam search curse*, as noted by Meister et al. (2020).

Model	BERT-base	BERT-large	DistilBERT	RoBERTa
HCB Left-to-Right	10.86	11.92	6.12	24.74
HCB Best-to-Worst	**11.43**	**12.58**	**6.44**	25.29
Standard Left-to-Right	9.7	11.31	5.87	24.98
Standard Best-to-Worst	10.63	12.32	6.39	**25.81**
Ablation 1 (Context Scramble)	4.63	4.08	2.06	9.86
Ablation 2 (Random Token Swap)	9.71	11.16	5.23	17.61

Table 4: Top-1 % accuracy at infilling a random number (uniform between 2 and 5) of missing tokens on 100K examples from Brown corpus, across models, with beam size 5. For ablations, we show only best-to-worst results since they strictly outperformed left-to-right results.

	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-10 Accuracy (%)		
Beam size	10	15	20	10	15	20	10	15	20
Desformers HCB	51.66	52.18	52.59	**63.11**	**62.58**	**62.27**	**67.88**	**69.67**	**66.56**
Desformers Standard	**55.21**	**56.08**	**56.62**	60.79	60.80	60.98	64.88	66.29	63.43

	Top-1 Accuracy (%)			Top-10 Accuracy (%)			Top-20 Accuracy (%)		
Number of missing tokens	5	6	7	5	6	7	5	6	7
Ithaca HCB	**68.91**	**61.97**	**53.82**	**85.33**	**79.42**	**72.33**	**87.71**	81.81	75.14
Ithaca Standard	68.87	61.86	53.75	85.24	79.38	71.21	87.68	**81.83**	**75.17**

Table 5: Top-k accuracy for infilling tasks on inscription datasets. (Top) Desformers top-k accuracy on 10K examples infilling two missing tokens using [MASK] pivot. (Bottom) Ithaca top-k accuracy on 6K examples infilling using beam size 20 and “-” pivot

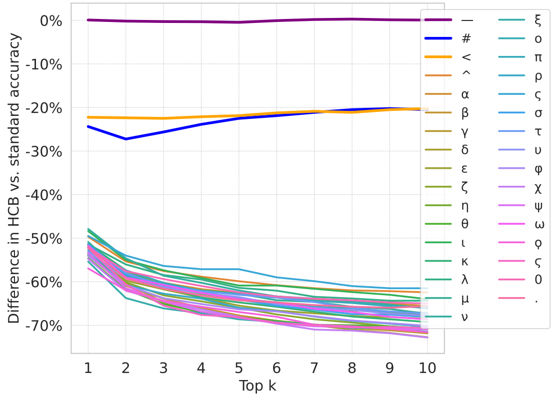

Figure 3: Difference in Ithaca infilling accuracy across all choice of pivots

given a probability distribution p on fixed-length sequences $\mathbf{x}$, the claim that

$$\log p(\mathbf{x}) = \sum_{i=1}^{n} \log p(x_i|\mathbf{x}_{:i}, [\mathrm{M}]_{i:})$$

holds for all n and all $\mathbf{x}$ is equivalent to the claim that x_i is conditionally independent of $[\mathrm{M}]_{i:}$ given $\mathbf{x}_{<i}$. Therefore, standard beam search considers the true joint probability $p(\mathbf{x})$ exactly when this conditional independence holds.

For this reason, we might expect standard beam search to function well exactly when this conditional independence holds. BERT's training procedure is known to encourage this conditional independence (Devlin et al., 2019), and so we might expect independence to be a more reasonable assumption for more finely optimized BERT models, such as RoBERTa. Indeed, this hypothesis is consistent with Table 1, which finds that standard beam search outperforms HCB beam search over all datasets considered when RoBERTa is used, but not when BERT-base is used. We hypothesize this may also explain why standard beam search outperforms HCB beam search for ESM2.

Theorem 3. *Suppose that $\mathbf{x}^1, \ldots, \mathbf{x}^m$ are sequences of length n, and let $\mathbb{P}$ be their empirical distribution. Let $\{Z_{ij}\}_{i \le m, j \le n}$ be auxiliary i.i.d. Bernoulli random variables with parameter $p > 0$. For each ground truth sequence $\mathbf{x}^i$, define a "partially masked" sequence $\mathbf{y}^i$ such that*

$$y^i_j = \begin{cases} x^i_j & Z_{ij} = 0 \\ [\mathrm{M}] & Z_{ij} = 1 \end{cases}$$

where $[\mathrm{M}]$ is an arbitrary symbol not contained in the alphabet over which $\mathbb{P}$ has support.

Consider the (standard) MLM loss function[13] on functions p:

$$L(p) = \mathbb{E}_Z \left[\sum_{i,j: Z_{ij}=1} -\log p(x^i_j | \mathbf{y}^i_{-j}) \right]$$

Then if p minimizes the training loss $L(p)$, p satisfies the following conditional independence law:

$$p(x_j|\mathbf{x}_{<j}, [\mathrm{M}]_{>j}) = p(x_j|\mathbf{x}_{<j});$$

and the conditional distributions $p(x_j|\mathbf{x}_{-j})$ are compatible.

Proof. See Appendix A. □

8 Conclusion

In this work, we develop and apply theoretically-sound methods to use pretrained MLMs for text infilling, a task with important applications spanning a wide variety of domains. We clarify the conditions under which it is theoretically sound to perform text infilling with MLMs using standard beam search. For instances when these conditions do not hold, we introduce HCB beam search as a probabilistically-justified modification with no additional computational complexity and demonstrate its superiority to standard beam search in the expected conditions. Future work exploring HCB beam search in the context of other MLMs and in other domains can help further elucidate the contexts in which HCB beam search is beneficial, ultimately facilitating the adaptation of MLMs for text infilling and related tasks in low data or compute settings.

9 Limitations

One limitation of this method is that we do not explicitly know whether conditional independence assumptions hold or fail for a given model, despite hypothesized heuristics. Another limitation is the lack of intuition regarding the use of the mask token as a pivot – future work should more rigorously investigate optimal pivot choices and the interpretability of these entities. Additionally, further understanding the objective and training conditions

[13] In addition to masking random tokens, some MLM training objectives (e.g. BERT) include random replacement of tokens. This property can be included by simply including all such random transformations as elements of the training dataset $\mathbf{x}^1, \ldots, \mathbf{x}^m$.

under which an MLM satisfies conditional independence assumptions is also needed. Finally, the proposed HCB beam search relies on the presence of some token in a given model's output distribution which flexibly behaves as in-distribution; while some MLMs maybe include special characters in their output layers, others may not. A potential risk of our method is its ability to use MLMs outside of their original contexts, which could have unforeseen consequences and lead to unpredictable behavior.

References

David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. 1985. A learning algorithm for boltzmann machines*. *Cognitive Science*, 9(1):147–169.

Barry C. Arnold and D. V. Gokhale. 1998. Distributions most nearly compatible with given families of conditional distributions. *Test*, 7(2):377–390.

Yannis Assael, Thea Sommerschield, Brendan Shillingford, Mahyar Bordbar, John Pavlopoulos, Marita Chatzipanagiotou, Ion Androutsopoulos, Jonathan Prag, and Nando de Freitas. 2022. Restoring and attributing ancient texts using deep neural networks. *Nature*, 603(7900):280–283.

David Bamman and Patrick J. Burns. 2020. Latin bert: A contextual language model for classical philology.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, page 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Christer Bruun and Jonathan Edmondson. 2014. *The Oxford handbook of Roman epigraphy*. Oxford University Press.

Charlie Cowen-Breen, Creston Brooks, Johannes Haubold, and Barbara Graziosi. 2023a. Logion: Machine-learning based detection and correction of textual errors in greek philology. In *Ancient Language Processing*.

Charlie Cowen-Breen, Creston Brooks, Johannes Haubold, and Barbara Graziosi. 2023b. Logion: Machine Learning for Greek Philology. ArXiv:2305.01099 [cs].

Desmond DeVaul. 2023. Desformers. `https://huggingface.co/ddevaul/desformers`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv:1810.04805 [cs].

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 2492–2501, Online. Association for Computational Linguistics.

Barbara Graziosi, Johannes Haubold, Charlie Cowen-Breen, and Creston Brooks. 2023. Machine learning and the future of philology: A case study. *TAPA*, 153(1):253–284.

Lucas Torroba Hennigen and Yoon Kim. 2023. Deriving Language Models from Masked Language Models. ArXiv:2305.15501 [cs].

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. ArXiv:2106.09685 [cs].

Daphne Ippolito, David Grangier, Chris Callison-Burch, and Douglas Eck. 2019. Unsupervised hierarchical story infilling. In *Proceedings of the First Workshop on Narrative Understanding*, page 37–43, Minneapolis, Minnesota. Association for Computational Linguistics.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. 2023. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130. Publisher: American Association for the Advancement of Science.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Clara Meister, 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)*, page 2173–2185, Online. 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. ArXiv:1808.08745 [cs].

W. Nelson Francis and Henry Kucera. 1979. *Brown corpus manual*. Brown University.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 311, Philadelphia, Pennsylvania. Association for Computational Linguistics.

Frederick Riemenschneider and Anette Frank. 2023. Exploring large language models for classical philology. *arXiv preprint arXiv:2305.13698*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, page 2339–2352, Online. Association for Computational Linguistics.

Uri Shaham and Omer Levy. 2022. What do you get when you cross beam search with nucleus sampling? In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, page 38–45, Dublin, Ireland. Association for Computational Linguistics.

Tianxiao Shen, Victor Quach, Regina Barzilay, and Tommi Jaakkola. 2020. Blank language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 5186–5198, Online. Association for Computational Linguistics.

Samuel Sledzieski, Meghana Kshirsagar, Minkyung Baek, Bonnie Berger, Rahul Dodhia, and Juan Lavista Ferres. 2023. Democratizing Protein Language Models with Parameter-Efficient Fine-Tuning. Pages: 2023.11.09.566187 Section: New Results.

Thea Sommerschield*, Yannis Assael*, Brendan Shillingford, Mahyar Bordbar, John Pavlopoulos, Marita Chatzipanagiotou, Ion Androutsopoulos, Jonathan Prag, and Nando de Freitas. 2021. I.PHI dataset: ancient greek inscriptions. `https://github.com/sommerschield/iphi`.

Qing Sun, Stefan Lee, and Dhruv Batra. 2017. Bidirectional beam search: Forward-backward inference in neural sequence models for fill-in-the-blank image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 7215–7223, Honolulu, HI. IEEE.

Tibshirani, 2023. Stat 241B lecture notes. [link].

Trong Thanh Tran and Truong Son Hy. 2023. Protein design by directed evolution guided by large language models. *bioRxiv*, pages 2023–11.

UniProt. 2008. The universal protein resource (UniProt) - PubMed.

Tom Young and Yang You. 2023. On the inconsistencies of conditionals learned by masked language models.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. (arXiv:1904.09675). ArXiv:1904.09675 [cs].

Wanrong Zhu, Zhiting Hu, and Eric Xing. 2019. Text infilling. *arXiv preprint arXiv:1901.00158*.

Appendix

A Proofs

Proof of Theorem 3. Log-loss is a strictly proper scoring function (Tibshirani, 2023), in that it is uniquely minimized when $p = \mathbb{P}$.[14] From the fact that $\mathbb{P}$ is a probability distribution, it immediately follows that the conditionals $P(x_j|\mathbf{x}_{-j})$ are compatible. On the other hand, consider the events

$$P(x^i_j = x|\mathbf{y}^i_{<j} = \mathbf{z}_{<j}, \mathbf{y}^i_{>j} = [\mathrm{M}]_{>j})$$

for arbitrary x and $\mathbf{z}$. As the event $\{\mathbf{y}^i_k = [\mathrm{M}]\}$ occurs if and only if $Z_{ik} = 1$, and thus with probability p independently of $\mathbf{x}$ for all $i \in [m], j \in [n]$, the conditional independence claim follows. □

Proof that compatibility and conditional independence imply validity of Equation 4. By compatibility, there exists a joint distribution $p(\mathbf{x})$ whose conditional distributions are equal to those learned by the MLM. By the chain rule and the assumption of conditional independence,

$$\begin{aligned}\log p(\mathbf{x}) &= \sum_{i=1}^{n} \log p(x_i|\mathbf{x}_{<i}) \\ &= \sum_{i=1}^{n} \log p(x_i|\mathbf{x}_{:i}, [\mathbf{M}]_{i:})\end{aligned}$$

On the other hand, if

$$\log p(\mathbf{x}) = \sum_{i=1}^{n} \log p(x_i|\mathbf{x}_{:i}, [\mathbf{M}]_{i:})$$

for all n, then we recover the conditional independence assumption by induction on n. □

B Additional figures

Model	BERT-base	BERT-large	DistilBERT	RoBERTa
HCB Left-to-Right	15.0	16.24	9.05	31.85
HCB Best-to-Worst	**15.72**	**16.89**	9.62	32.47
Standard Left-to-Right	14.02	15.95	9.03	32.4
Standard Best-to-Worst	14.87	16.72	**9.65**	**33.15**
Pure sampling	9.69	11.56	4.67	22.76
Sampling with $T = 0.25$	12.33	14.14	7.69	28.57
Sampling with $T = 0.5$	11.93	13.67	7.11	27.63
Sampling with $T = 0.75$	10.98	12.86	6.05	25.84
Nucleus sampling, $p = 0.9$	9.73	11.3	5.02	24.56

Table 6: Top-5 accuracy at infilling a random number (uniform between 2 and 5) of missing tokens on 100K examples from Brown corpus, across models, with beam size 5.

[14] For instance, see the following blog post on a consistency theorem for BERT: https://machinethoughts.wordpress.com/2019/07/14/a-consistency-theorem-for-bert/

Model	BERT-base	BERT-large	DistilBERT	RoBERTa
HCB Left-to-Right	10.86	11.92	6.12	24.74
HCB Best-to-Worst	**11.43**	**12.58**	**6.44**	**25.29**
Ablation 1 Left-to-Right	4.1	3.83	1.86	9.42
Ablation 1 Best-to-Worst	4.63	4.08	2.06	9.86
Ablation 2 Left-to-Right	8.94	10.65	4.82	16.34
Ablation 2 Best-to-Worst	9.71	11.16	5.23	17.61

Table 7: Top-1 accuracy comparison of HCB beam search to ablations 1 and 2, across models, when infilling gaps consisting of a (uniformly random) number of tokens between 2 and 5.

Model	BERT-base	BERT-large	DistilBERT	RoBERTa
HCB Left-to-Right	10.86	11.92	6.12	24.74
HCB Best-to-Worst	**11.43**	**12.58**	**6.44**	25.29
Standard Left-to-Right	9.7	11.31	5.87	24.98
Standard Best-to-Worst	10.63	12.32	6.39	**25.81**
Pure sampling	8.09	9.57	4.0	20.72
Sampling w/ $T = 0.25$	9.41	10.97	5.66	24.59
Sampling w/ $T = 0.5$	9.25	10.74	5.4	24.03
Sampling w/ $T = 0.75$	8.79	10.31	4.85	22.89
Nucleus sampling, $p = 0.9$	8.45	9.99	4.3	22.57
Ablation 1 (Context Scramble)	4.63	4.08	2.06	9.86
Ablation 2 (Random Token Swap)	9.71	11.16	5.23	17.61

Table 8: Top-1 accuracy at infilling a random number (uniform between 2 and 5) of missing tokens on 100K examples from Brown corpus, across models, with beam size 5. For ablations, we show only best-to-worst results since they strictly outperformed left-to-right results.

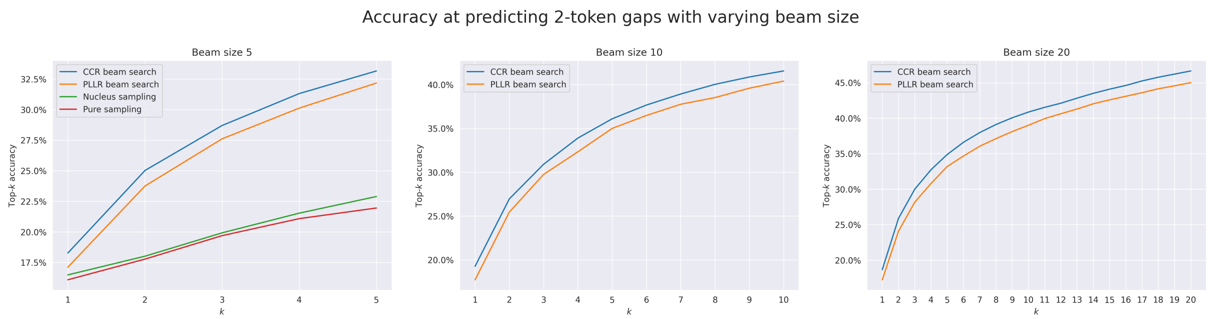

Figure 4: Comparison of HCB beam search with standard beam search, nucleus sampling, and pure sampling. Evaluated on 10,000 examples from the SNLI dataset. When comparing nucleus sampling to beam search with beam size B, we draw B samples for a fair comparison.

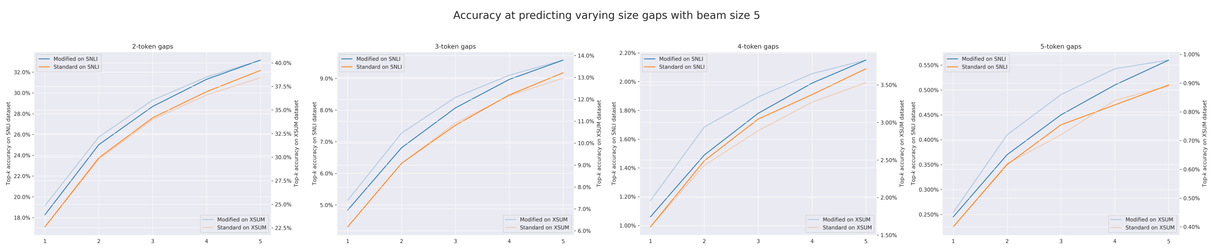

Figure 5: Comparison of HCB beam search with standard beam search. Evaluated on 10,000 examples each from the SNLI and XSUM datasets.

	Top-1 Accuracy (%)				Top-5 Accuracy (%)			
Number of missing tokens	2	3	4	5	2	3	4	5
HCB Left-to-Right	6.16	2.12	0.91	0.46	16.79	5.80	2.41	1.16
HCB Best-to-Worst	6.36	2.30	1.02	0.53	16.95	6.03	2.53	1.19
Standard Left-to-Right	6.57	2.49	1.29	0.71	17.64	6.51	2.92	1.50
Standard Best-to-Worst	**6.65**	**2.58**	**1.31**	**0.76**	**17.76**	**6.70**	**3.09**	**1.67**

Table 9: Top-k accuracy at infilling consecutive missing tokens on 10K examples from UniProt dataset, using ESM2 with beam size 5 and number of missing tokens ranging from 2 to 5.

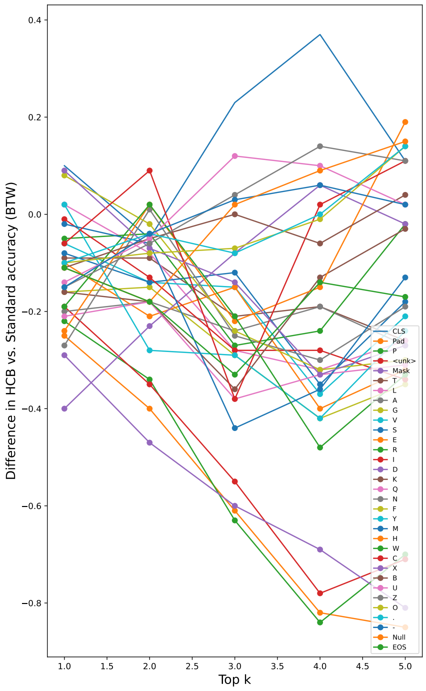

Figure 6: Performance of HCB beam search compared to standard beam search for various choices of pivot. All pivot experiments performed with beam size of 5 and gap size of 2.

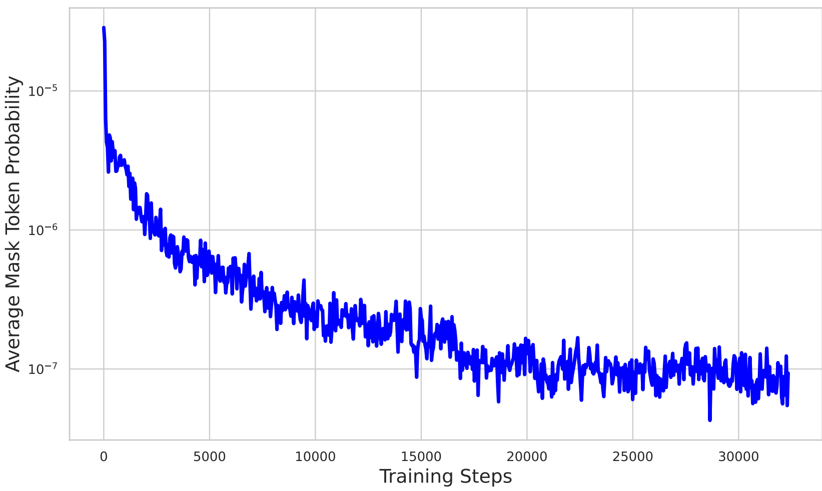

Figure 7: Average $P[\mathrm{M}]$ for fixed random masks across 50 test examples over the course of BERT training