A Sampling-Based Exploration of Neural Text Generation Models

Bryan Eikema

A Sampling-Based Exploration of Neural Text Generation Models

Bryan Eikema

A Sampling-Based Exploration of Neural Text Generation Models

Bryan Eikema

A Sampling-Based Exploration of Neural Text Generation Models

ILLC Dissertation Series DS-2026-01

Institute for Logic, Language and Computation

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Science Park 107
1098 XG Amsterdam
phone: +31-20-525 6051
e-mail: illc@uva.nl
homepage: http://www.illc.uva.nl/

Printed and bound by Ipskamp Printing.

ISBN: 978-94-6536-012-6

Cover illustration: *In de maand juli: een molen aan een poldervaart* by Paul Joseph Constantin Gabriël (circa 1889).

The research for publication of this doctoral thesis received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825299 (GoURMET).

A Sampling-Based Exploration of Neural Text Generation Models

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op dinsdag 10 februari 2026, te 13.00 uur

door Bryan Eikema
geboren te Purmerend

Promotiecommissie

Promotor:	prof. dr. K. Sima'an	Universiteit van Amsterdam
Copromotor:	dr. W. Ferreira Aziz	Universiteit van Amsterdam
Overige leden:	dr. A.F.T. Martins	Instituto Superior Técnico de Lisboa
	prof. dr. W. Byrne	University of Cambridge
	prof. dr. C. Monz	Universiteit van Amsterdam
	prof. dr. E. Kanoulas	Universiteit van Amsterdam
	dr. I.A. Titov	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

Contents

Acknowledgments

Pursuing a PhD is not always easy. There have been moments of frustration and despair that I would not have been able to get through without the support of the people around me.

First, I am deeply grateful to my wife, Iulia, who has been with me every step of this journey. She has had to endure me during stressful deadlines and moments of disappointment, and was there for moments of celebration. Her unwavering support has been my bedrock.

I also want to thank my old office mates—Jasmijn, Samira, Sophie and Bas—with whom I have shared much joy and laughter. Going to the office would not have been as fun without you. That also extends to my (former) ILLC colleagues with whom I have spent much time at "the good" coffee machine, our own self-managed Nespresso coffeemaker. Thank you Iacer, Miguel, Víctor, Sandro, Marco, Mario, Ece and Tom.

Barely a year into my PhD the world went into a lockdown during the COVID pandemic. This meant working from home from a $28m^2$ student apartment together with my now-wife, taking simultaneous Zoom calls barely 3 meters apart from each other. I am very grateful for my in-laws, Ovidiu and Mihaela, for letting us stay with them during this period. From this time, I also fondly remember our Probabll lab Zoom meetings with Wilker, Lina and Eelco, such as our delivered-at-home Christmas dinners.

Returning to the office after lockdown has been a relief, nonetheless. The Probabll lab has expanded since with Joris, Pedro and Evgenia. I want to thank you all for enlivening the office again after the lockdown. I am very grateful for our many inspiring chats and joyful lunch conversations.

During my internship at NAVER Labs Europe, I had the privilege of collaborating with Marc Dymetman, Germán Kruszewski and Hady Elsahar. It has been a great pleasure to work with you and I have learned a tremendous amount from you.

I also wish to thank my Promotor Khalil Sima'an for the insightful chats we

have had and the feedback you have provided to my thesis.

Above all, I owe an enormous debt of gratitude to my adviser, Wilker. Thank you for believing in me and taking me on as your PhD student. I began my PhD on the day that you started as Assistant Professor, and it's been incredible to witness the Probabll lab grow to what it is today. You have always been positive, patient and a source of inspiration. Your mentorship has shaped me as a researcher, and I am profoundly grateful for your support.

Lastly, to my family—mom, dad, Patrick, Sammy, Ovidiu, Mihaela, Livia, Victor, and Adrian—and friends—Marco, Damiaan, Sander, Alex, Rens, Tyvar, Viet, and Omid—you have always reminded me of what truly matters in life. Thank you for your love and support throughout the years.

Amsterdam Bryan Eikema
January, 2025.

Chapter 1

Introduction

Neural text generation models[1] are at the basis of most modern-day natural language processing (NLP) systems. In recent years many important innovations to neural network architectures and training paradigms have appeared such as attention mechanisms, Transformers, and pre-training and data augmentation strategies that have accelerated the performance of NLP systems tremendously. At their core, however, these models have not changed their probabilistic formulation since the original neural text generation models were first described. Nor is there necessarily a strong motivation to do so: the current probabilistic formulation permits efficient training of the model as well as efficient generation through sampling or search, and makes no independence assumptions across tokens in the output sequence. As a result, this probabilistic approach to text generation is widespread in natural language processing tasks and approaches.

However, the probabilistic nature of these models is often quickly forgotten after the model has been trained. For many natural language processing tasks such as machine translation, for example, where translation accuracy (typically against a single reference translation) is given priority over generation variety, deterministic search algorithms are employed to extract a single "best" generation from the model. In such cases, the probabilistic model is only used to score partial subsequences during generation to find the highest scoring sequence, *i.e.* the sequence with highest probability under the distribution, also known as the mode of the distribution. Assuming that a well-trained text generation model indeed places data-like sequences[2] at its conditional modes, such an approach

[1] We will use the term neural text generation interchangeably with (neural) sequence generation throughout the dissertation, text consisting of a sequence of discrete tokens (*e.g.* words or sub-words). We opt not to use "natural language generation", as the focus is on the algorithms and models behind text generation, and not so much any particular collection of tasks. We discuss the mathematical formulation of neural text generation models in Section 2.1.

[2] Data-like is referring to generations similar to sequences in the training data. We can consider training data to come from a distribution implied by the data collection and human language production processes that we wish to capture in our text generation models.

seems intuitive when high accuracy is the primary goal.

Nonetheless, such an assumption should be verified before committing to a particular generation strategy. In fact, a well-known observation in machine translation (the "beam search curse", Koehn and Knowles (2017)) seems to suggest that the highest probability generations of machine translation models are not at all data-like. A better understanding of the sequence distributions that our neural networks predict allows us to make better-informed decisions about what kind of generation strategy is appropriate for our models.

Sampling is a natural way to explore the properties of the sequence distributions predicted by neural networks. By studying the properties of such samples we indirectly also study the properties of the sequence distributions we are working with. Samples can also be used to inform generation algorithms and for some tasks samples are even of direct interest themselves as outputs of text generation systems.

In this dissertation we will explore the use of sampling to better understand our text generation models and in order to inform novel generation algorithms. We will view commonly known pathologies and biases of text generation models under the lens of such a probabilistic exploration and provide a new perspective on their potential causes. We use these insights to propose and iterate on a sampling-based generation algorithm inspired by risk minimisation strategies. We also develop new sampling strategies altogether to sample from arbitrary distributions where a per-(sub)word factorisation does not exist.

1.1 Why Sampling-Based Generation?

Neural text generation models learn to predict (conditional) sequence distributions. At test-time we often wish to summarise these distribution objects into a single grammatically correct generation that is plausible given the context and task. Common ways to do this are to search for the mode of the distribution or bias the sampling algorithm towards higher probability generations.

While higher probability sequences can be a good representation of the sequence distribution, this does not necessarily have to be the case. When a distribution is high entropy and thus spreads probability mass over many sequences, the mode and other high probability sequences may obtain only a small amount of probability mass and may thus not be representative of the distribution as a whole. On the other hand, when the sequence distribution is very peaked around a single sequence, that single sequence is a much better summary of the sequence distribution. Therefore, it is important to know with what kind of distributions we are dealing with and if they are high entropy distributions we should reconsider the current focus on searching for high probability sequences.

But if the probability that the model assigns to any particular sequence given a context would be small, how can we even summarise the distribution into a sin-

gle sequence at all? While any particular sequence may not get much probability, certain properties may appear in many sequences that collectively receive a substantial amount of probability mass. By informing the generation algorithm of such properties encoded within the distribution, we can find a sequence that best represents the distribution as a whole. Samples inform us of these distributional properties, and will form the basis of the generation algorithms we study in this dissertation.

1.2 Beyond (Sub)Word Factorisation

Luckily for us, sampling from neural text generation models is usually trivial to do: text generation models use a per-(sub)word factorisation and we can simply build up a sequence of text units by sampling the next (sub)word from the next-token distributions predicted by the neural network, a process known as ancestral sampling (Bishop, 2006).

However, not all models permit easy sampling. An interesting class of models that allows for very flexible model specification and even the composition of multiple models are energy-based models. Energy-based models typically only score complete sequences and can thus not make use of any algorithms that rely on scoring partial sequences. Efficient generation from such models is a challenging task as neither search nor sampling is easily achieved. It is worthwhile to be able to explore such models as well through the act of sampling to study their behaviours as well as to inform sampling-based decoding algorithms. In this dissertation, we develop an approximate sampler specifically designed to sample from such models.

1.3 Contributions

The main contributions of this dissertation are an exploration of machine translation sequence distributions, a collection of sampling-based decoding algorithms based on minimum Bayes risk and an approximate sampling algorithm for sampling from energy-based models.

In the exploration of neural machine translation sequence distributions we find that they tend be high entropy, *i.e.* they tend to spread probability mass over many translations. This leads us to conclude that the existing generation paradigms based on mode-seeking search are sub-optimal for the sequence distributions we obtain in practice. We also explore the sequence distribution in more detail and find that most probability mass is put on a set of sequences that exhibit properties of good translations. On top of that we find that well-known pathologies and biases, such as a preference of machine translation models towards too short translations, are not clearly present in the sequence distribution, suggesting that they are introduced by the generation algorithm, *i.e.* mode-seeking search.

This leads us to propose a sampling-based generation algorithm. We adapt a generation algorithm previously popular in statistical machine translation (Kumar and Byrne, 2004), minimum Bayes risk, and propose a principled sampling-based approximation to it. The resulting algorithm uses expectations to effectively capture distributional properties and essentially re-scores sequences based on how well they summarise the sequence distribution. We provide multiple efficiency improvements and study its properties and effectiveness in neural machine translation.

Finally, we look into a case where obtaining samples is no trivial task: energy-based models. We study approximate sampling algorithms and look into how to efficiently obtain good (unconditional) proposal distributions for such samplers. A downside of existing approximate samplers is that it is difficult to estimate the approximation quality for any given hyperparameters to the sampling algorithm. We develop a sampler that does allow for such approximation quality estimates in terms of divergence measures. We show the effectiveness and advantages of this sampler over its competitors on a task where we put distributional constraints on the sequence distribution.

1.4 Thesis Overview

The remainder of this dissertation is laid out as follows.

Chapter 2 provides all the necessary background for understanding the rest of the dissertation. It assumes basic familiarity with statistical and natural language processing concepts. Concepts covered include the probabilistic model behind text generation systems, maximum-likelihood estimation, decision rules and their approximations, sampling algorithms, and approximate sampling techniques.

Chapter 3 explores the distributions learned by neural machine translation systems. We show that some of the biases claimed to be present in such systems are not convincingly present in the sequence distributions learnt by these models. Therefore, we argue that these biases are at least partially introduced by the generation algorithm. We further find that sequence distributions often spread probability mass and we argue that the mode is an inadequate target for generation algorithms. This is supported by other findings in the literature. We suggest an alternative generation objective and a straightforward sampling-based approximation of that we coin sampling-based minimum Bayes risk (MBR) decoding.

Chapter 4 expands on our sampling-based approximation to minimum Bayes risk for neural machine translation and studies its scaling properties and sensitivity to other hyperparameters of the decoding algorithm. We further propose more efficient approximations using a coarse-to-fine algorithm with a proxy objective to perform an initial filtering step. We discuss the impact of the work and findings outside of our own research, and discuss future directions for sampling-based

MBR decoding.

Chapter 5 looks into sampling from text generation models that do not admit efficient generation due to not being able to score partial sequences. We specifically focus on the task of controlled text generation from large language models, where we are interested on putting constraints on the sequence distribution as a whole rather than on individual generations. Energy-based models allow us to define distributions that meet constraints while staying as close as possible to the original distribution under a distribution divergence metric. We develop an approximate sampling algorithm that allows us to sample from such energy-based models, while being able to estimate how close our approximate samples are to the target distribution. Our proposed sampler allows for an informed trade-off between approximation accuracy and sampling efficiency. We also show how one can utilise modern advancements in in-context-learning and pre-trained neural text generation models to efficiently construct proposal distributions.

Chapter 6 provides concluding remarks for the thesis and discusses future directions.

1.5 Output

The core publications around which the contribution chapters in this dissertation are based are the following:

- Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics[3]

- Bryan Eikema and Wilker Aziz. 2022. Sampling-based approximations to minimum Bayes risk decoding for neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10978–10993, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics

- Bryan Eikema, Germán Kruszewski, Christopher R Dance, Hady Elsahar, and Marc Dymetman. 2022. An approximate sampler for energy-based models with divergence diagnostics. *Transactions on Machine Learning Research*

The following works were also partially developed during the course of my PhD, but did not end up in this dissertation, because many of the core developments happened before or after the PhD:

[3]This work was awarded the Best Paper Award at the Conference on Computational Linguistics (COLING) of 2020.

- Bryan Eikema and Wilker Aziz. 2019. Auto-encoding variational neural machine translation. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 124–141, Florence, Italy. Association for Computational Linguistics

- Bryan Eikema. 2024. The effect of generalisation on the inadequacy of the mode. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*, pages 87–92, St Julians, Malta. Association for Computational Linguistics

Chapter 2

Background

This chapter provides background knowledge essential to the remainder of the dissertation. In Section 2.1, we discuss neural text generation, covering its probabilistic formulation, training algorithms, generation algorithms and evaluation. Readers with a strong foundation in neural text generation can freely skip this section. Section 2.2 provides the necessary background knowledge for Chapter 5 of this dissertation. It discusses sampling techniques for models that do not permit straightforward sampling, covering both exact and approximate techniques for obtaining samples and computing expectations, as well as convergence guarantees and diagnostics. Readers already familiar with techniques such as Markov chain Monte Carlo, importance sampling and rejection sampling, and are familiar with f-divergences can confidently skip this section.

2.1 Neural Text Generation

Neural text generation consists of a neural network capable of generating sequences of discrete units that jointly form a string of text, typically a natural language sentence. Sequences are split up into smaller textual units, such as words, subwords or even characters. Generation is then considered a structured prediction task, where a sequence of such textual units is predicted in succession by the model, typically in a left-to-right manner. From a probabilistic perspective, we model a random sequence Y with implicit, but random (in the sense of a random variable), length $|Y|$. The split into smaller textual units is realised as a factorisation into random Categorical variables Y_j over a vocabulary of possible textual units. Using the chain rule of probability this can be achieved as:

$$P(Y = y|X = x) = \prod_{j=1}^{|y|} P(Y_j = y_j|Y_{<j} = y_{<j}, X = x) \tag{2.1}$$

$$\text{where } Y_j|Y_{<j} = y_{<j}, X = x \sim \text{Cat}\left(f(x, y_{<j}; \theta)\right) \tag{2.2}$$

Here, $y_{<j}$ denotes the sequence of textual units y_1 through y_{j-1} predicted before predicting y_j, or an empty set for the distribution over Y_1, and x is a collection representing the (potentially empty) context that is not directly modelled by the neural text generation system[1]. Eq. 2.2 states that the random variable Y_j follows a Categorical distribution with parameters determined by $f(.)$, where the function $f(.)$ is a neural network with parameters θ mapping from the context $(x, y_{<j})$ to the logits (*i.e.* parameters) of the Categorical distribution.

Notably, in the presented probabilistic model no independence assumptions are made and the factorisation is potentially exact. This is because modern neural networks for sequence generation such as recurrent neural networks (Sutskever et al., 2014) and Transformers (Vaswani et al., 2017) are powerful enough to encode $(x, y_{<j})$ well without suffering from traditional data sparsity problems incurred by such a factorisation. The neural network with parameters θ are then used to map ("encode") any context $y_{<j}$, sometimes along with additional context x, into a continuous space, followed by a projection onto the probability simplex over a predetermined set of textual units. Commonly a set of subwords is automatically learned using an algorithm such as byte pair encoding (BPEs; Sennrich et al., 2016), which ensures that frequent words are represented in a single token, while less frequent ones are split up into multiple tokens (subwords, or even characters). Most subword algorithms also allow for an *open vocabulary*, meaning that any piece of text can be encoded as a sequence of tokens, without the need for a special out-of-vocabulary token.

A short side note on notation

We have used uppercase letters to denote random variables and random sequences and lowercase letters to denote their instantiations. Throughout this dissertation, we will use the above notation as well as the shorter $P(y|x)$ to denote the probability mass function that preserves the distribution of $Y|X = x$, such that $P(Y = y|X = x) = P(y|x)$. We use uppercase $P(.)$ to denote probability mass functions over discrete random variables and lowercase $p(.)$ to denote probability densities over continuous random variables. When distributions are not normalised (*i.e.* they don't sum to 1), we denote that as $\tilde{P}(.)$ or $\tilde{p}(.)$. The support of random variables is denoted using script letters, e.g. $\mathcal{Y}$. Finally, we will sometimes use $P_\theta(.)$ to stress the dependence of the probability mass function on neural network parameters θ.

[1] Typically, this is a sequence as well, *e.g.* the sentence to translate in machine translation.

2.1.1 Training

As mentioned, each Categorical next-token distribution is parameterised in an amortised fashion using a single neural network that maps any arbitrary context $(x, y_{<j})$ to the probability simplex over vocabulary units. The parameters of this neural network, collectively referred to as θ, need to be estimated or *trained*. One principled approach to selecting θ is to select the θ that maximises the probability assigned by the model to the training data (also known as the model likelihood). Formally, this means solving the following optimisation problem:

$$\theta_{\text{MLE}} = \operatorname*{argmax}_{\theta} \prod_{n=1}^{N} \prod_{j=1}^{|y^{(n)}|} P_\theta(y_j^{(n)} | y_{<j}^{(n)}, x^{(n)}) \tag{2.3}$$

$$= \operatorname*{argmax}_{\theta} \sum_{n=1}^{N} \sum_{j=1}^{|y^{(n)}|} \log P_\theta(y_j^{(n)} | y_{<j}^{(n)}, x^{(n)}) \tag{2.4}$$

for N data points $(x^{(n)}, y^{(n)})$ where $x^{(n)}$ are collections of additional context elements depending on the particular generation task. This paradigm for selecting θ is known as maximum likelihood estimation (MLE) and is commonly used to train neural text generation models across natural language processing tasks. Oftentimes its negative is also posed as a loss function, then often referred to as the (Categorical) cross-entropy loss.

In particular, for neural networks, the backpropagation algorithm (Rumelhart et al., 1986) can be used to compute individual parameter updates across neural network layers. Furthermore, to make training feasible on large datasets, gradients are estimated against small subsets (mini-batches) of the training data instead of all N data points, in an algorithm called stochastic mini-batch gradient descent (Robbins and Monro, 1951; Bottou and Cun, 2004). Both optimisation have the same optima, but may still converge to different locally optimal solutions depending on initialisation and batch size.

2.1.2 Generation

A trained model provides us with a neural network that can take an arbitrary context $(x, y_{<j})$ and can produce a conditional probability distribution over Y_j. Following the factorisation of Equation 2.1, this means we have a neural network that can predict distributions over (natural language) sequences. At test-time, however, we are typically less interested in distributions, but rather would prefer to see a single generation, *e.g.* a translation of an input sentence or an answer to a question. This means that we have to select a single sequence from the sequence distribution as an output of the model at test-time.

Beam Search

Oftentimes this is achieved by searching for the highest probability sequence of the distribution, in an algorithm known as beam search. Beam search is a heuristic search algorithm that locally searches for the highest probability partial subsequences using the next-token distributions provided by the neural network, and keeping a memory of a fixed size containing the best scoring partial subsequences up to that point. At each step of the algorithm *the beam* (the memory of highest probability partial subsequences) is expanded using the next-token distributions for each subsequence (to $k \times |V|$ continuations) and filtering the continuations down to the k highest probability continuations, where k is the beam size and $|V|$ the vocabulary size. These steps repeat until all sequences in the beam end with an end-of-sequence symbol or a maximum length is reached.

Formally, we can view this as approximating a *decision rule* that dictates that the desired sequence from the distribution is the mode of the distribution. In natural language processing literature, this decision rule is known as *maximum-a-posteriori*, or MAP, decoding. Using a larger beam size, we arguably approximate this decision rule better, as we obtain higher probability sequences. Beam search has an exponential complexity increase as a function of beam size and thus, in practice, only approximations of the MAP decision rule with smaller beam sizes are tractable to compute.

It turns out, however, that for many text generation tasks larger beam sizes do not result in higher quality generations (Koehn and Knowles, 2017; Fan et al., 2018; Holtzman et al., 2020; Zhang et al., 2021). In fact, for many tasks such as machine translation and summarisation, there is negative correlation between beam size and quality beyond a certain point: an observation known as the *beam search curse* (Koehn and Knowles, 2017). Thus, in practice, heuristics are used in the beam search algorithm, such as using a beam size as small as $k = 5$ and combining model probability with additional "scoring functions" to counteract specific problems with higher probability sequences (*e.g.* a length penalty to prevent too short sequences from being output).

Sampling

In some, typically more open-ended tasks in NLP, beam search has been found to lead to especially poor quality generations, *e.g.* generations that contain repeating phrases or uninteresting generations such as an "I don't know" answer to questions (Holtzman et al., 2020). There is also oftentimes a desire for more diversity across generated outputs. For this reason, these fields often employ sampling to generate from their trained neural text generation models. The most straightforward way to do this is by simply following the generative story of the model: repeatedly draw $Y_j \sim \text{Cat}\left(f_\theta(x, y_{<j})\right)$ for increasing integers $j = 1 \ldots N_{\text{MAX}}$ until an end-of-sequence symbol is drawn or the maximum length N_{MAX} is reached.

This procedure is also known as *ancestral sampling* (Bishop, 2006). Following this procedure, and assuming N_{MAX} is large enough that no samples are cut off prematurely, sampled generations are distributed exactly according to the predicted sequence distribution. This procedure can therefore also be very useful to explore the properties of these distributions.

However, some works have found that the quality of generations for a particular task can improve by tweaking the predicted next-token distributions. For example, nucleus sampling (Holtzman et al., 2020) considers only the highest probability words that add up to some top-p amount of probability mass. The distribution over the resulting subset of the outcome space is re-normalised and continuations are sampled from the re-normalised next-token distributions. Top-k sampling (Fan et al., 2018) follows the same procedure except that a fixed number of k highest probability words are considered. Locally typical sampling (Meister et al., 2023) selects only words with a surprisal value within some range of the entropy of the next-token distributions. These methods cut-off a long tail of lower probability (sub)words and thus shrink the outcome space, as many sequences are now impossible to generate. Using a temperature τ to alter next-token distributions as $P_\theta(Y_j|x, y_{<j})^{1/\tau}$, is sometimes also used to sharpen or flatten the predicted distributions (but more typically sharpened), which does not have the aforementioned effect of shrinking the outcome space. As the above procedures alter the predicted next-token distributions, samples generated by them do no longer follow the sequence distributions predicted by the trained model. Therefore, we sometimes will also refer to these methods as *biased* samples and samples generated by ancestral sampling as *unbiased* samples to stress this fact.

2.1.3 Evaluation

In this dissertation we will mostly work on tasks where some form of ground truth sequence is available to compare against. Evaluation then involves comparison of a model generated output against an available ground truth sequence on a continuous scale. The gold-standard for evaluation would be a properly executed human evaluation using multiple experts judging the quality of the generated sequence against several aspects of generation quality. For this dissertation, following the majority of NLP literature, we will, however, have to be content with employing automatic evaluation metrics. Such metrics will compare a single generated sequence from the model against a ground truth sequence and assign it a continuous score.

BLEU (Papineni et al., 2002), for example, compares n-grams for $n \in [1, 4]$ between the generated and ground truth sequence and computes a weighted geometric mean of their precision. The granularity level of the used n-grams varies slightly from implementation to implementation, but we will frequently use sacreBLEU (Post, 2018), which uses a simple regex-based tokeniser to compute n-grams on a word-level. ChrF(++) (Popović, 2017) similarly computes a score

based on n-gram precision and recall, but rather operates on a character-level, therefore generally being more suitable towards morphologically rich languages that have many different forms for the same word in different contexts. A downside to these metrics is that they compare the exact wording used in the generation and reference and are thus not able to detect whether perhaps a synonym of a ground truth word is correctly used by the model. Therefore, metrics like METEOR (Denkowski and Lavie, 2011) operate similarly as BLEU and ChrF, but also make use of a list of synonyms available for a range of more high-resource languages. Other metrics such as BEER (Stanojević and Sima'an, 2014) use a number of word- and character-level features to create a trainable model, whose weights are then tuned such that its scores correlate well with human judgements.

Recent metrics have been based on large pre-trained neural network models, which have the advantage of being less sensitive to the exact wording of the sequences as the internal continuous representations are known to similarly encode semantically similar words (Mikolov et al., 2013). Similar to BEER, these models are often fine-tuned on datasets containing human judgements of generation quality and have been shown to correlate very well with them as a result. Therefore, such metrics are now considered the state-of-the-art. Examples that we will use in this dissertation are BLEURT (Sellam et al., 2020), based on the BERT (Devlin et al., 2019) pre-trained model, and COMET (Rei et al., 2020), built on XLM-R (Conneau et al., 2020) with variants based on other pre-trained language models.

2.2 Sampling Techniques

While most models we work with in this dissertation make use of the autoregressive factorisation outlined in Eq. 2.1, a model formulation that permits easy and efficient sampling (see Section 2.1.2), there exist models where this is not the case. In particular, in Chapter 5 we will work with an energy-based model (EBM) that encodes preferences over entire sequences. Such a model can *score* sequences, *i.e.* it can assign some numerical value that indicates a degree of preference of the model for that (complete) sequence over others, but it has no inherent mechanism to generate sequences, as the scores cannot be factorised. Such scores also imply a probability distribution. Restricting the scores to non-negative numbers[2], we can in principle compute a normalisation constant by summing the score for all sequences in the outcome space. We will denote such (unnormalised) scorers of a random variable (or sequence) X as $\tilde{P}(x)$. Normalised probabilities can then be

[2]One could also exponentiate the values assigned to an outcome, these possibly negative values are then called energies, and the exponentiated energies we will call scores.

obtained by dividing by the normalisation constant[3]:

$$P(x) \propto \tilde{P}(x) \tag{2.5}$$

$$Z = \sum_{x \in \mathcal{X}} \tilde{P}(x) \tag{2.6}$$

$$P(x) = \frac{\tilde{P}(x)}{Z} \tag{2.7}$$

Here, $\mathcal{X}$ denotes the support of random variable X. However, in practice the outcome space is typically too large to compute the normalisation constant (*e.g.* when X is a random variable over natural language sentences). Therefore, in order to obtain samples from $P(x)$, or compute expected values $\mathbb{E}_{P(x)}[h(x)]$ under it, we need to resort to sampling techniques that do not require access to a chain rule decomposition of $P(x)$. In this section, we will cover some more well-known ones that we also use in this thesis.

2.2.1 Rejection Sampling

Rejection sampling is a conceptually simple sampling technique that is able to sample exactly from the target distribution $P(x)$ when certain conditions are met. It makes use of the concept of a *proposal distribution*: a distribution different from the target distribution that does permit efficient sampling. Rejection sampling takes samples from this proposal distribution and accepts (*i.e.* output the sample) or rejects (*i.e.* throw away the sample) using a ratio that guarantees that the accepted samples are distributed according to the target distribution. Consider an unnormalised probability distribution $\tilde{P}(x)$ over random variable X from which we cannot tractably sample, but where we can assign non-negative scores to sequences $x \in \mathcal{X}$. If we can find an efficient-to-sample-from proposal distribution $Q(x)$, for which the support includes that of X, as well as a constant β such that $\tilde{P}(x) \leq \beta Q(x)$ over the entire support of X, we can sample from $P(x)$ accepting samples from $Q(x)$ with the following *acceptance ratio*:

$$r_x = \frac{\tilde{P}(x)}{\beta Q(x)} \tag{2.8}$$

We also illustrate this idea in Figure 2.1. While the main advantage of this method is that we obtain exact samples from the (normalised) target distribution $P(x)$, the main challenge is finding a suitable proposal distribution $Q(x)$ and constant β such that the rejection sampling condition is met. It may also be difficult to prove that $\tilde{P}(x) \leq \beta Q(x)$ for all $x \in \mathcal{X}$, or the acceptance ratio r_x

[3]In the continuous case the summation is replaced by an integral, but the reasoning is identical.

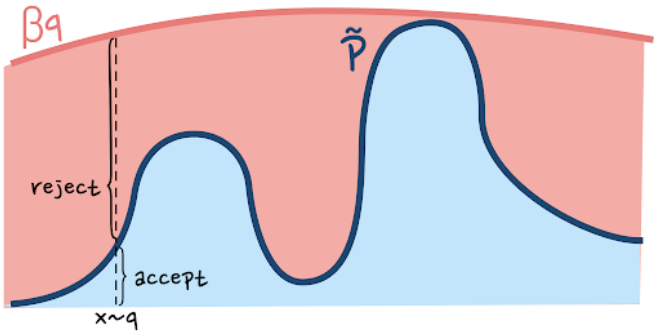


Figure 2.1: Rejection sampling, where $\tilde{p}(x)$ is a function that assigns non-negative scores to $x \in \mathcal{X}$ (continuous in this sketch), $q(x)$ is a proposal distribution, and β is a constant that guarantees that $\beta q(x) \geq \tilde{p}(x)$ for all $x \in \mathcal{X}$. If we sample from $q(x)$ and accept samples with the shown ratio, we obtain exact samples from $p(x)$.

may be too small on average to lead to an efficient-enough sampler. Because of this, we often resort to approximate sampling techniques, which is the topic of the next subsection.

2.2.2 Markov Chain Monte Carlo

When exact sampling with *e.g.* rejection sampling is infeasible, one can resort to *approximate sampling*. Approximate sampling means that we don't sample from the target distribution $P(x)$, but rather from a distribution as close as possible to $P(x)$. A popular collection of methods for performing approximate sampling is known as Markov chain Monte Carlo (MCMC). In MCMC, a Markov chain of samples is built up, such that the last sample in this Markov chain is distributed according to a distribution close to $P(x)$.

A Markov chain is a stochastic process that undergoes transitions from one state to another state stochastically. It is characterised by the Markov property, which dictates that a transition to a future state depends only on the present state and not the full sequence of states preceding it. Formally, for a sequence of states S_t where t indicates time:

$$P(S_t|S_{t-1}, S_{t-2}, \ldots, S_0) = P(S_t|S_{t-1}) \tag{2.9}$$

The probability from one state to another is known as a transition probability. For a finite number of states, we can also define a so-called *transition matrix*, containing the probabilities of reaching each state from every other state.

In Markov chain Monte Carlo we build such a Markov chain where each state S_t in the Markov chain consists of an outcome in $\mathcal{X}$, *e.g.* a full natural language sentence. In order to define the transition probabilities, MCMC makes

use of a proposal distribution, similar to rejection sampling explained in the previous section. For MCMC, however, oftentimes a *local* proposal distribution is employed, where the proposal distribution is conditioned on the previous state: $Q(x|x')$. This often is implemented as making small steps around the previous coordinates in a continuous space, or by making small edits in discrete space, *e.g.* by changing, adding or removing a single token in a sequence. A local proposal distribution could be easier to define and may have better convergence properties if no good global (*i.e.* unconditional) proposal distributions can be constructed. However, if a good global proposal distribution $Q(x)$ can be constructed, this can simply replace the local proposal distribution in the MCMC procedure.

A popular implementation of MCMC is the Metropolis-Hastings (MH) algorithm. In Metropolis-Hastings, we take a proposed transition obtained by sampling from $Q(x|x')$, and accept it with probability:

$$r_t = \min\left(1, \frac{\tilde{P}(x)Q(x'|x)}{\tilde{P}(x')Q(x|x')}\right) \tag{2.10}$$

If accepted, the new state S_{t+1} becomes the proposed state x. If rejected, the new state S_{t+1} rather becomes x', *i.e.* the previous state gets repeated. This process can be summarised as follows:

1. Initialise S_0 with some starting value x'.
2. Propose a new state x from the proposal distribution $Q(x|x')$.
3. Compute the acceptance probability r_t (Eq. 2.10).
4. Accept the new state x with probability r_t. If the transition is accepted, set $S_{t+1} = x$. Otherwise, set $S_{t+1} = x'$.
5. Repeat steps 2-4 until the Markov chain converges.

It can be shown (Robert and Casella, 2004) that if we build a long-enough Markov chain (such that it converges, see Section 2.2.5 for a discussion on how to measure convergence) using the the process described above, this collection of samples can be used to obtain approximate samples from $P(x)$ (by for example taking the last state of the Markov chain) as well as to compute expectations under $P(x)$ (by using all or some states of the Markov chain to compute sample averages).

2.2.3 Importance Sampling

Another important algorithm for estimating expected values under $P(x)$ if we can only sample from a proposal distribution $Q(x)$ is importance sampling (Robert

and Casella, 2004). Again, assume we do have a proposal distribution $Q(x)$ where we can efficiently sample from and whose support includes that of X, and we aim to compute an expectation under $P(x)$: $\mathbb{E}_{P(x)}[h(x)]$. The key idea in importance sampling is that we can use samples from $Q(x)$ and assign each sample a weight that accounts for discrepancy between $P(x)$ and $Q(x)$.

Consider the aforementioned expectation of a function $h(x)$ under the target distribution $P(x)$:

$$\mathbb{E}_{P(x)}[h(x)] = \sum_{x \in \mathcal{X}} h(x)P(x) \tag{2.11}$$

Using importance sampling, this expectation can be rewritten as:

$$\mathbb{E}_{P(x)}[h(x)] = \sum_{x \in \mathcal{X}} h(x)\frac{P(x)}{Q(x)}Q(x) = \mathbb{E}_{Q(x)}\left[h(x)\frac{P(x)}{Q(x)}\right] \tag{2.12}$$

Thus, using a standard Monte Carlo estimator, the expectation can be approximated using N samples $\{x_i\}_{i=1}^N$ drawn from the proposal distribution $Q(x)$:

$$\mathbb{E}_{P(x)}[h(x)] \approx \frac{1}{N}\sum_{i=1}^{N} h(x_i)\frac{P(x_i)}{Q(x_i)}. \tag{2.13}$$

The ratio $\frac{P(x_i)}{Q(x_i)}$ is known as the importance weight. It adjusts the contribution of each sample x_i to account for the fact that the samples are drawn from $Q(x)$ instead of $P(x)$. While importance sampling can be used with any $Q(x)$ whose support includes that of X, the choice of $Q(x)$ does impact the variance of the estimator. We briefly discuss this in Section 2.2.5.

2.2.4 f-Divergences

As we mentioned in Section 2.2.2, approximate sampling techniques like Markov chain Monte Carlo sample from some distribution close to the target distribution. In order to define "closeness" between distributions, often a *divergence measure* is used. A broad class that includes many commonly employed divergences is the f-divergences[4] (Polyanskiy, 2019).

Let $f : [0, \infty) \to \mathbb{R}$ be a convex function such that $f(1) = 0$, let $f(0) \doteq \lim_{t \to 0^+} f(t)$, and let P_1 and P_2 be probability distributions over a discrete sample space $\mathcal{X}$. The f-divergence of P_1 from P_2 is defined as:[5]

$$D_f(P_1, P_2) \doteq \mathbb{E}_{P_2(x)}\left[f\left(\frac{P_1(x)}{P_2(x)}\right)\right] \tag{2.14}$$

[4]The f in f-divergences has no relation with the $f(.)$ we occasionally use to refer to neural network computations, as in Eq. 2.2.

[5]We will assume that the support of P_1 is included in that of P_2 for the remainder of this dissertation, leading to this slightly simplified definition of f-divergences.

By varying the definition of $f(t)$ we can obtain many popular divergence measures, such as the Kullback-Leibler (KL) divergence, total variation distance (TVD) and χ^2 divergence. A number of useful properties for f-divergences that can be shown are that $D_f(P_1, P_2) \geq 0$ and, for $f(t)$ strictly convex at $t = 1$, $D_f(P_1, P_2) = 0$ iff $P_1 = P_2$ (Liese and Vajda, 2006).

In this dissertation, we mainly work with the total variation distance

$$\mathrm{TVD}(P_1, P_2) \doteq \sum_{x \in \mathcal{X}} |P_1(x) - P_2(x)|/2, \tag{2.15}$$

obtained with $f(t) = |1 - t|/2$, and KL divergence

$$\mathrm{KL}(P_1, P_2) \doteq \sum_{x \in \mathcal{X}} P_1(x) \log \frac{P_1(x)}{P_2(x)} \tag{2.16}$$

$$= \mathbb{E}_{P_1(x)} \left[\log \frac{P_1(x)}{P_2(x)} \right], \tag{2.17}$$

which has $f(t) = t \log t$. For both it holds that they equal 0 iff $P_1 = P_2$. While TVD is symmetric ($\mathrm{TVD}(P_1, P_2) = \mathrm{TVD}(P_2, P_1)$), KL divergence is not and so $\mathrm{KL}(P_1, P_2) \neq \mathrm{KL}(P_2, P_1)$.

2.2.5 Convergence

While sampling techniques allow us to work with distributions that we would otherwise not be able to sample from and / or compute expectations under, for any practical implementations of the algorithms these are approximations with some amount of error. In this section we will discuss some of the theoretical guarantees of the Metropolis-Hastings algorithm and importance sampling and their convergence properties.

Theoretical Guarantees of Metropolis-Hastings

Robert and Casella (2004, Theorem 7.4, p. 274) prove the following theorem, with P the target distribution.

Theorem 2.1. *Suppose that the Metropolis-Hastings Markov chain $(X^{(n)})$ is P-irreducible.*

(i) If h is an P-integrable function, then

$$\lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} h(X^{(n)}) = \int h(x) P(x) \, \mathrm{d}x \qquad \textit{almost surely.}$$

(ii) If, in addition, $(X^{(n)})$ *is aperiodic, then*

$$\lim_{N\to\infty} \mathrm{TVD}\left(P, \int K^N(x,\cdot)\,\mu(\mathrm{d}x)\right) = 0$$

for every initial distribution μ*, where* $K^N(x,\cdot)$ *denotes the kernel for* N *transitions.*

Let's explain some of the terminology here. P-irreducible means that we should design the Markov chain (using the proposal distribution) such that we can reach every state in the support of $P(x)$ with positive probability eventually down the Markov chain. In property (i), P-integrability means that the integral in property (i) should be finite, and almost surely means that the convergence happens with probability 1.[6] In property (ii), aperiodicity requires that the Markov chain does not get trapped in cycles. Common choices of local proposal distributions ensure that these properties hold.

Now as to what this theorem states. Property (i) says that the average over the N first elements of a *single* chain converges to the expectation of $h(x)$ for $x \sim P(x)$, as N increases. Property (ii) is concerned with the TVD between the target distribution $P(x)$ and the distribution obtained by repeatedly running an N-step chain and outputting the N^{th} element. This distance converges to zero as N increases. Meaning that for sufficiently long Markov chains, using the Metropolis-Hastings algorithm we can compute expectations under $P(x)$ and/or obtain samples from $P(x)$. While we can ensure that implementations of MH have these theoretical guarantees, it is often intractable to assess precisely how close we are to the target distribution for any particular finite N.

Convergence Diagnostics for MCMC

In order to determine a practical N for which we can rely on the accuracy of results, convergence diagnostics can provide insights into whether the Markov chain has converged to the target distribution (*i.e.* whether expectations computed using the produced samples are close to those under the target distribution). Common techniques include running multiple Markov chains from different starting points and assessing whether expectations computed under them converge to the same value, a process often quantified by the Gelman-Rubin statistic ($\hat{R}$; Gelman and Rubin, 1992; Vehtari et al., 2021). Furthermore, the autocorrelation of samples from the Markov chain is often assessed, as due to the nature of MCMC algorithm neighboring states are often highly correlated (especially with local proposal distributions).[7] A metric that is used to quantify this, is the effective sample size (ESS), which estimates the number of independent samples in

[6] The set of Markov chains for which convergence does not occur is possibly non-empty, but have 0 probability collectively.

[7] Therefore, oftentimes only every k-th state from the Markov chain is used, a procedure known as thinning.

the chain after accounting for autocorrelation (Gamerman and Lopes, 2006). We will not go into depth in these diagnostics as we will not be using them in this dissertation. It suffices to know that these diagnostics are mostly heuristic, and while they can be effective, they do not provide concrete estimates of how close our expectations or samples actually are to the target distribution.

Importance Sampling

Importance sampling, under some mild conditions, provides an unbiased estimator of the target expectation. Meaning that as the number of samples N increases, the estimate converges to the true expectation by the law of large numbers (Tao, 2008). Variance of the estimator also decreases with larger N, but is heavily influenced by the choice of proposal distribution $Q(x)$. In particular, if the ratio $\frac{P(x)}{Q(x)}$ has high variance, convergence may be slow. For example, if there is a large discrepancy between $P(x)$ and $Q(x)$ for some portions of the outcome space, the ratio $\frac{P(x)}{Q(x)}$ can blow up for some samples. Therefore, ideally $Q(x)$ would match $P(x)$ closely, leading to consistent importance ratios.

Chapter 3

The Inadequacy of the Mode

Neural machine translation (NMT) uses neural networks to predict distributions over translations given a source sentence. Crucially, the predictions of these models are (conditional) probability distributions over sequences. In order to elect a single candidate translation a choice of decision rule is required and typically also some tractable approximation to it using a decoding algorithm, given the complexity and vastness of the output space. The decision rule of choice in NMT is to elect the mode of the distribution, also known as maximum-a-posteriori (MAP), and its approximation, beam search, which searches for the mode of the sequence distribution. In this chapter we question this choice of decision rule, showing it to be suboptimal for the sequence distributions we obtain in practice. We show that many pathologies and biases typically observed in NMT are at least partially due to this choice of decision rule. We further show the distributions obtained do have desirable properties, capturing statistics of good translations. This suggests that a decoding algorithm that makes use of the sequence distribution holistically to generate translations could be fruitful. The latter will be the main focus of the next chapter, though we already propose a concrete direction, sampling-based minimum Bayes risk, along with some preliminary experiments at the end of this chapter. Finally, we also explore using the Bayesian framework for evaluating the data fit of our translation models. We argue that a distributional evaluation of translation models is crucial for obtaining insights in the effects of model and architecture changes, rather than purely focusing on improving the outcomes of beam search. The contents of this chapter are based on Eikema and Aziz (2020), published at the 28th International Conference on Computational Linguistics (COLING'2020), where it was awarded Best Paper.

Chapter Highlights

Problem Statement

- Neural machine translation systems suffer from a number of pathologies

and biases such as generating too short translations, producing copies of the input or generating hallucinated content.

- Beam search decoding suffers from a so-called *beam search curse*, meaning better search through a larger beam size often results in worse translation quality.

- The true mode of the translation distribution has been found to often be the empty sequence, an obviously inadequate translation.

Contributions

- Prior to this work, most research turned to the model to find a culprit for observed pathologies and biases. We show that targeting the mode during decoding through the use of MAP is at least partially responsible for a number of these pathologies and biases commonly observed in NMT systems.

- We show that samples from NMT models trained through maximum likelihood estimation (MLE) reproduce statistics of the data well, not showing some pathologies and biases commonly thought to be inherent to NMT models, such as producing too short translations.

- We argue that MAP is not well-suited as a decision rule for modern neural machine translation models and we suggest an alternative decision rule, minimum Bayes risk along with a proof-of-concept sampling-based approximation, as more suitable for NMT models.

3.1 Introduction

Numerous findings in neural machine translation (NMT) suggest that modern translation systems have serious flaws. This is based on observations such as: *i)* translations produced via beam search typically under-estimate sequence length (Sountsov and Sarawagi, 2016; Koehn and Knowles, 2017), the *length bias*; *ii)* translation quality generally deteriorates with better approximate search (Koehn and Knowles, 2017; Murray and Chiang, 2018; Ott et al., 2018; Kumar and Sarawagi, 2019), the *beam search curse*; *iii)* the true most probable translation under the model (*i.e.*, the mode of the distribution) is empty in many cases (Stahlberg and Byrne, 2019) and a general negative correlation exists between model probability and quality beyond a certain probability value (Ott et al., 2018), we call this the *inadequacy of the mode*.

A number of hypotheses have been formulated to explain these observations. They mostly suggest there is something fundamentally wrong with NMT as a model (*i.e.*, its factorisation as a product of locally normalised distributions) or its most popular training algorithm (*i.e.*, regularised maximum likelihood estimation, MLE for short). These explanations make an unspoken assumption, namely, that identifying the mode of the distribution, also referred to as maximum *a posteriori* (MAP) decoding (Smith, 2011), is in some sense the obvious decision rule for predictions. While this assumption makes intuitive sense and works well in unstructured classification problems, it is less justified in NMT, where oftentimes the most probable translations together account for very little probability mass, a claim we shall defend conceptually and provide evidence for in experiments. Unless the translation distribution is extremely peaked about the mode for every plausible input, criticising the model in terms of properties of its mode can at best say something about the adequacy of MAP decoding. Unfortunately, as previous research has pointed out, this is seldom the case (Ott et al., 2018). Thus, pathologies about the mode cannot be unambiguously ascribed to NMT as a model nor to MLE, and inadequacies about the mode cannot rule out the possibility that the model captures important aspects of translation well in expectation.

In this chapter, we criticise NMT models as probability distributions estimated via MLE in various settings: varying language pairs, amount of training data, and test domain. We observe that the induced probability distributions represent statistics of the data well in expectation, and that some length and lexical biases are introduced by approximate MAP decoding. We demonstrate that beam search outputs are rare outcomes, particularly so when test data stray from the training domain. The empty string, shown to often be the true mode (Stahlberg and Byrne, 2019), too is an infrequent outcome. Finally, we show that samples obtained by following the model's own generative story are of reasonable quality, which suggests we should base decisions on statistics gathered from the distribution holistically. One such decision rule is minimum Bayes risk (MBR) decoding (Goel and Byrne, 2000; Kumar and Byrne, 2004), of which we propose a proof-of-concept sampling-based approximation.

3.2 Observed Pathologies in NMT

Many studies have found that NMT suffers from a *length bias*: NMT underestimates length which hurts the adequacy of translations. Cho et al. (2014) already demonstrate that NMT systematically degrades in performance for longer sequences. Sountsov and Sarawagi (2016) identify the same bias in a chat suggestion task and argue that sequence to sequence models underestimate the margin between correct and incorrect sequences which they attribute to local normalisation. Later studies have also confirmed the existence of this bias in NMT (Koehn and Knowles, 2017; Stahlberg and Byrne, 2019; Kumar and Sarawagi, 2019).

Notably, all these studies employ beam search decoding. In fact, some studies link the length bias to the *beam search curse*: the observation that large beam sizes hurt performance in NMT (Koehn and Knowles, 2017). Sountsov and Sarawagi (2016) already note that larger beam sizes exacerbate the length bias. Later studies have confirmed this connection (Blain et al., 2017; Murray and Chiang, 2018; Yang et al., 2018; Kumar and Sarawagi, 2019). Murray and Chiang (2018) attribute both problems to local normalisation which they claim introduces label bias (Lafferty et al., 2001) to NMT. Yang et al. (2018) show that model probability negatively correlates with translation length. These findings suggest that the mode might suffer from length bias, likely thereby failing to sufficiently account for adequacy. In fact, Stahlberg and Byrne (2019) show that oftentimes the true mode is the empty sequence.

The connection with the length bias is not the only reason for the beam search curse. Ott et al. (2018) find that the presence of copies in the training data cause the model to assign too much probability mass to copies of the input, and that with larger beam sizes this copying behaviour becomes more frequent. Cohen and Beck (2019) show that translations obtained with larger beam sizes often consist of an unlikely prefix with an almost deterministic suffix and are of lower quality. In open-ended generation, Zhang et al. (2021) correlate model probability with human judgements for a fixed sequence length, thus eliminating any possible length bias issues. They find that probability generally correlates positively with human judgements, up until an inflection point, after which the correlation becomes negative. An observation also made in translation with BLEU rather than human judgements (Ott et al., 2018). We call this general failure of the mode to represent good translations in NMT the *inadequacy of the mode problem*.

3.3 NMT and its Many Biases

It has been said that due to certain design decisions NMT suffers from a number of biases. We review those biases here and then discuss in Section 3.4 one bias that has received very little attention and which, we argue, underlies many biases in NMT and explains some of the pathologies discussed in Section 3.2.

Exposure bias. MLE parameters are estimated conditioned on observations, a human-produced translation given a source sentence, sampled from the training data. Clearly, those are not available at test time, when we search through the learnt distribution. This mismatch between training and test, known as exposure bias (Ranzato et al., 2016), has been linked to many of the pathologies of NMT and motivated modifications or alternatives to MLE aimed at exposing the model to its own predictions during training (Bengio et al., 2015; Ranzato et al., 2016; Shen et al., 2016; Wiseman and Rush, 2016; Zhang et al., 2019). While exposure bias has been a point of critique mostly against MLE, it has

only been studied in the context of approximate MAP decoding. The use of MAP decoding and its approximations shifts the distribution of the generated translations away from data statistics (something we provide evidence for in later sections), thereby exacerbating exposure bias.

Non-admissible heuristic search bias. In beam search, partial translations are ranked in terms of log-probability without regards to (or with crude approximations of) their future score, which may lead to good translations being pruned too early. This corresponds to searching with a non-admissible heuristic (Hart et al., 1968), that is, a heuristic that may underestimate the probability of completing a translation. This biased search affects statistics of beam search outputs in unknown ways and may well account for some of the pathologies of Section 3.2, and has motivated variants of the algorithm aimed at comparing partial translations more fairly (Huang et al., 2017; Shu and Nakayama, 2018). This problem has also been studied in parsing literature, where it's known as imbalanced probability search bias (Caraballo and Charniak, 1996; Stanojević and Steedman, 2020).

Label bias. Where a conditional model makes independence assumptions about its inputs (*i.e.*, variables the model does not generate), local normalisation prevents the model from revising its decisions, a problem known as *label bias* (Bottou, 1991; Lafferty et al., 2001). This is a model specification problem which limits the distributions a model can represent (Andor et al., 2016). While this is the case in incremental parsing (Stern et al., 2017) and simultaneous translation (Gu et al., 2017), where inputs are incrementally available for conditioning, this is *not* the case in standard NMT (Sountsov and Sarawagi, 2016, Section 5), where inputs are available for conditioning in all generation steps. It is plausible that local normalisation might affect the kind of local optima we find in NMT, but that is orthogonal to label bias.

3.4 Biased Statistics and the Inadequacy of the Mode

In most NMT research, criticisms of the model are based on approximations of the mode obtained using beam search. The mode, however, is not an unbiased summary of the probability distribution that the model learnt. That is, properties of the mode say little about properties of the learnt distribution (*e.g.*, a short mode does not imply the model underestimates average sequence length). MAP decoding algorithms and their approximations bias the statistics by which we criticise NMT. They restrict our observations about the model to a single or a handful of outcomes which on their own can be rather rare. To gain insight about

the model as a distribution, it seems more natural to use all of the information available to us, namely, all samples we can afford to collect[1], and search for frequent patterns in these samples. Evidence found that way more faithfully represents the model and its beliefs.

On top of that, the sample space of NMT is high-dimensional and highly structured. NMT models must distribute probability mass over a massive sample space (effectively unbounded). While most outcomes ought to be assigned negligible mass, for the total mass sums to 1, the outcomes with non-negligible mass might still be too many. The mode might only account for a tiny portion of the probability mass, and can actually be extremely unlikely under the learnt distribution. Using the mode for predictions makes intuitive sense in unstructured problems, where probability distributions are likely very peaked, and in models trained with large margin methods (Vapnik, 1998), since those optimise a decision boundary directly. With probability distributions that are very spread out, and where the mode represents only a tiny bit of probability mass, targeting at the mode for predictions is much less obvious, an argument that we shall reinforce with experimental results throughout this analysis.[2]

At the core of our analysis is the concept of an unbiased sample from the model, which we obtain by ancestral sampling: iteratively sampling from distributions of the form $\text{Cat}(f(x, y_{<j}; \theta))$, each time extending the generated prefix $y_{<j}$ with an unbiased draw, until the end-of-sequence symbol is generated. By drawing from the model's probability distribution, unlike what happens in MAP decoding, we are imitating the model's training procedure. Only we replace samples from the data by samples from the model, thus shedding light onto the model's fit. That is, if these samples do not reproduce statistics of the data, we have an instance of poor fit.[3] Crucially, ancestral sampling is not a pathfinding algorithm, thus the non-admissible heuristic search bias is not a concern. Ancestral sampling is *not* a decision rule either, thus returning a single sample as a prediction is not expected to outperform MAP decoding (or any other rule). Samples can be used to diagnose model fit, as we do in Section 3.6, and to approximate decision rules, as we do in Section 3.8.4. In sum, we argue that MAP decoding is a source of various problems and that it biases conclusions about NMT. Next, we provide empirical evidence for these claims.

[1]Considering that we are dealing with an unbounded space of sequences, we cannot study the probability mass function directly. However, we can interact with it through simulation via ancestral sampling (see Section 2.1.2) and study the properties of the samples we obtain.

[2]This perhaps non-intuitive notion that the most probable outcomes are rare and do not summarise a model's beliefs well enough is also common in a popular information-theoretic concept, that of typicality (MacKay, 2003, Section 4.4). It's not clear whether such a typical set also exists in sequence-to-sequence models like NMT.

[3]Where one uses (approximate) MAP decoding instead of ancestral sampling this is known as exposure bias.

3.5 Data & System

We train NMT systems on German-English (de-en), Sinhala-English (si-en), and Nepali-English (ne-en), in both directions. For German-English we use all available WMT'18 (Bojar et al., 2018) parallel data, except for Paracrawl, amounting to about 5.9 million sentence pairs, and train a Transformer base model (Vaswani et al., 2017). For Sinhala and Nepali, for which very little parallel data are available, we mimic the data and system setup of Guzmán et al. (2019). As we found that the data contained many duplicate sentence pairs, we removed duplicates, but left in those where only one side (source or target) of the data is duplicate to allow for paraphrases. For all language pairs, we do keep a portion of the training data (6,000 sentence pairs) separate as held-out data for the analysis. In this process we also removed any sentence that corresponded exactly to either the source or target side of a held-out sentence from the training data. To analyse performance outside the training domain, we use WMT's *newstest2018* for German-English, and the FLORES datasets collected by Guzmán et al. (2019) for the low-resource pairs. Our analysis is focused on MLE-trained NMT systems. However, as Transformers are commonly trained with label smoothing (LS) (Szegedy et al., 2016), we do additionally report automatic quality assessments of beam search outputs on LS-trained systems.

3.6 Assessing the Fit of MLE-Trained NMT

We investigate the fit of the NMT models of Section 3.5 on a held-out portion of the training data. This allows us to criticise MLE without confounders such as domain shift. We will turn to data in the test domain (*newstest2018*, FLORES) in Section 3.8. We compare unbiased samples from the model to gold-standard references and analyse statistics of several aspects of the data. If the MLE solution is good, we would expect statistics of sampled data to closely match statistics of observed data.

We obtain statistics from reference translations, ancestral samples, and beam search outputs and model them using hierarchical Bayesian models. For each type of statistic, we formulate a joint model over these three groups and inspect the posterior distribution over the parameters of the analysis model. We also include statistics extracted from the training data in our analysis, and model the three *test groups* as a function of posterior inferences based on training data statistics. Our methodology follows that advocated by Gelman et al. (2013) and Blei (2014). In particular, we formulate separate hierarchical models to inspect length, lexical, and word order statistics: sequence length, unigram and bigram counts, and skip-bigram counts, respectively.[4] In the next section, we describe

[4]Skip-bigrams are pairs of tokens drawn in the same order as they occur in a sentence, but without enforcing adjacency.

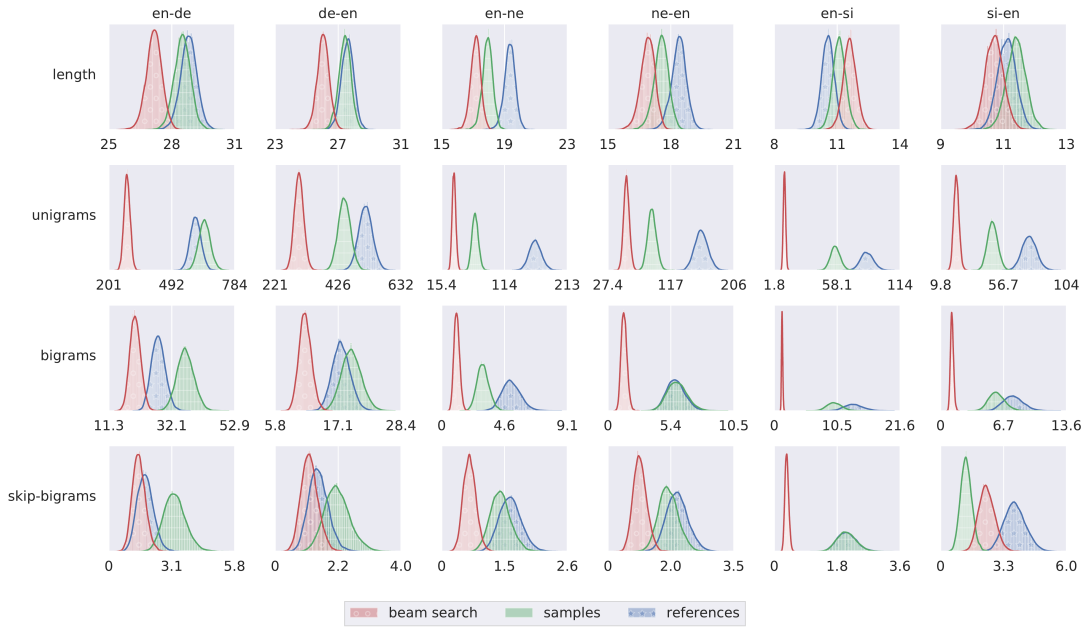


Figure 3.1: A comparison using hierarchical Bayesian models of statistics extracted from beam search outputs, samples from the model and gold-standard references. We show the posterior density on the y-axis, and the mean Poisson rate (length) and agreement with training data (unigrams, bigrams, skip-bigrams) on the x-axis for each group and language pair.

all the analysis models, inference procedures, and predictive checks that confirm their fit.

For length statistics, we look at the expected posterior Poisson rate for each group, each rate can be interpreted as that group's average sequence length. Ideally, the expected Poisson rates of predicted translations are close to those of gold-standard references. Figure 3.1 (top row) shows the inferred posterior distributions for all language pairs. We observe that samples generated by NMT capture length statistics reasonably well, overlapping a fair amount with the reference group. In almost all cases we observe that beam search outputs stray away from data statistics, usually resulting in shorter translations.

For unigrams, bigrams, and skip-bigrams, we compare the posterior agreement with training data of each group (this is formalised in terms of a scalar concentration parameter whose posterior we can plot). Higher values indicate a closer resemblance to training data statistics. For each statistic, the posterior distribution for gold-standard references gives an indication of ideal values of this agreement variable. Figure 3.1 (rows 2–4) show all posterior distributions. In most cases the gold-standard references agree most with the training data, as expected, followed by samples from the model, followed by beam search outputs.

For nearly all statistics and language pairs beam search outputs show least agreement with the training data, even when samples from the model show similar agreement as references do. Whereas samples from the model do sometimes show less similarity than references, in most cases they are similar and thus lexical and word order statistics are captured reasonably well by the NMT model. Beam search on the other hand again strays from training data statistics, compared to samples from the model.

3.7 Analysis Models

In this section we will go into detail on the Bayesian data analysis models employed in the previous section. On a first read, this section can be considered optional as it is not required for understanding the remainder of the thesis.

3.7.1 Length Analysis

We model length data from the training group using a hierarchical Gamma-Poisson model. Each target sequence length is modelled as being a draw from a Poisson distribution with a Poisson rate parameter specific to that sequence. All Poisson rates share a common population-level Gamma prior with population-level parameters α and β. The population-level parameters are given fixed Exponential priors set to allow for a wide but reasonable range of Poisson rates *a priori*.

$$\begin{aligned} \alpha &\sim \text{Exp}(1) & \beta &\sim \text{Exp}(10) \\ \lambda_i &\sim \text{Gamma}(\alpha, \beta) & y_i &\sim \text{Poisson}(\lambda_i) \end{aligned}$$

Here, i indexes one particular data point. This model is very flexible, because we allow the model to assign each datapoint its own Poisson rate. We model test groups as an extension of the training group. Test group data points are also modelled as draws from a Gamma-Poisson model, but parameterised slightly differently.

$$\begin{aligned} \mu &= \mathbb{E}\left[\text{Gamma}(\alpha, \beta | \mathcal{D}_T)\right] & \eta &\sim \text{Exp}(1.) \\ s_g &\sim \text{Exp}(\eta) & t_g &= 1/\mu \\ \lambda_{gi} &\sim \text{Gamma}(s_g, t_g) & y_{gi} &\sim \text{Poisson}(\lambda_{gi}) \end{aligned}$$

Here, i again indexes a particular data point, g a group in {reference, sampling, beam}, and $\mathcal{D}_T$ denotes the data of the training group. All Poisson rates are individual to each datapoint in each group. The Poisson rates do share a group-level Gamma prior, whose parameters are s_g and t_g. s_g shares a prior among all

test groups and therefore ties all test groups together. t_g is derived from posterior inferences on the training data by taking the expected posterior Poisson rate in the training data and inverting it. This is done such that the mean Poisson rate for each test group is $s_g \cdot \mu$, where s_g can be seen as a parameter that scales the expected posterior training rate for each test group individually. We infer Gamma posterior approximations for all unknowns using stochastic variational inference (SVI). After inferring posteriors, we compare predictive samples to the observed data in terms of first to fourth order moments to verify that the model fits the observations well.

3.7.2 Lexical & Word Order Analyses

We model unigram and (skip-)bigram data from the training group using a hierarchical Dirichlet-Multinomial model as shown below:

$$\begin{aligned} \alpha &\sim \text{Gamma}(1,1) & \beta &\sim \text{Gamma}(1,1) \\ \theta &\sim \text{Dir}(\alpha) & \psi_u &\sim \text{Dir}(\beta) \\ u &\sim \text{Multinomial}(\theta) & b|u &\sim \text{Multinomial}(\psi_u) \end{aligned}$$

Here, we have one Gamma-Dirichlet-Multinomial model to model unigram counts u, and a separate Dirichlet-Multinomial model for each u (the first word of a bigram) that b (the second word of a bigram) conditions on, sharing a common Gamma prior that ties all bigram models. This means that we effectively have $|V|+1$ Dirichlet-Multinomial models (where $|V|$ is BPE vocabulary size) in total to model the training group, where the $|V|$ bigram models share a common prior.

We model the three test groups using the inferred posterior distributions on the data of the training group $\mathcal{D}_T$. We compute the expected posterior concentration of the Dirichlets in the training group models and normalise it such that it sums to 1 over the entire vocabulary. The normalisation has the effect of spreading the unigram and bigram distributions. The test groups are modelled by scaling this normalised concentration parameter using a scalar. In order for test-groups to recover the training distribution the scaling variable needs to be large to undo the normalisation. This scalar, s_g for unigrams or m_g for bigrams, can be interpreted as the amount of agreement of each test group with the training group. The higher this scalar is, the more peaked the test group Multinomials will be about the training group lexical distribution.

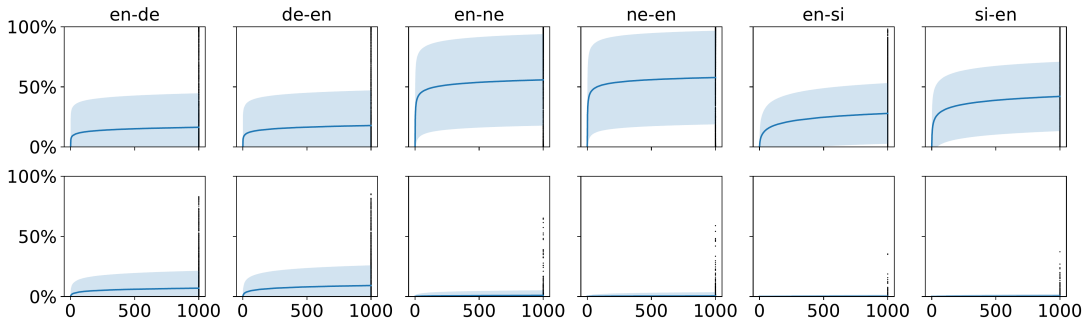


Figure 3.2: Cumulative probability of the unique translations in 1,000 ancestral samples on the held-out (top), and *newstest2018* / FLORES(bottom) test sets. The dark blue line shows the average cumulative probability over all test sentences, the shaded area represents 1 standard deviation away from the average. The black dots to the right show the final cumulative probability for each individual test sentence.

$$
\begin{aligned}
\mu(\alpha) &= \mathbb{E}\left[\alpha|\mathcal{D}_T\right]. & \mu(\beta) &= \mathbb{E}\left[\beta|\mathcal{D}_T\right] \\
\eta_s &\sim \text{Gamma}(1, 0.2) & \eta_m &\sim \text{Gamma}(1, 0.2) \\
s_g &\sim \text{Gamma}(1, \eta_s) & m_g &\sim \text{Gamma}(1, \eta_m) \\
\theta_g &\sim \text{Dir}(s_g \cdot \mu(\alpha)) & \psi_g &\sim \text{Dir}(m_g \cdot \mu(\beta)) \\
u_g &\sim \text{Multinomial}(\theta_g) & b_g|u_g &\sim \text{Multinomial}(\psi_g) \\
g &\in \{\text{reference}, \text{sampling}, \text{beam}\}
\end{aligned}
$$

We do collapsed inference for each Dirichlet-Multinomial (as we are not interested in assessing θ_g or ϕ_g), and infer posteriors approximately using SVI with Gamma approximate posterior distributions. To confirm the fit of the analysis model, we compare posterior predictive samples to the observed data in terms of absolute frequency errors of unigrams and bigrams as well as ranking correlation.

3.8 Examining the Translation Distribution

The NMT models of Section 3.5 specify complex distributions over an unbounded space of translations. Here, we examine properties of these distributions by inspecting translations in a large set of unbiased samples. To gain further insight we also analyse our models in the test domain (*newstest2018*, FLORES).

3.8.1 Number of Probable Translations

NMT, by the nature of its model specification, assigns probability mass to each and every possible sequence consisting of tokens in its vocabulary. Ideally, however, a well-trained NMT model assigns the bulk of its probability mass to good translations of the input sequence. We take $1,000$ unbiased samples from the model for each input sequence and count the cumulative probability mass of the unique translations sampled. Figure 3.2 shows the average cumulative probability mass for all test sentences with 1 standard deviation around it, as well as the final cumulative probability values for each input sequence. For the held-out data we observe that, on average, between 16.4% and 57.8% of the probability mass is covered. The large variance around the mean shows that in all language pairs we can find test sentences for which nearly all or barely any probability mass has been covered after $1,000$ samples. That is, even after taking $1,000$ samples, only about half of the probability space has been explored. The situation is much more extreme when translating data from the test domain (see bottom half of Figure 3.2).[5] Naturally, the NMT model is much more uncertain in this scenario, and this is very clear from the amount of probability mass that has been covered by $1,000$ samples: on average, only between 0.2% and 0.9% for the low-resource pairs and between 6.9% and 9.1% for English-German of the probability space has been explored. This shows that the set of likely translations under the model is very large and the probability distribution over those sentences mostly quite flat, especially so in the test domain. In fact, if we look at each input sequence individually, we see that for 37.0% (en-de), 35.5% (de-en), 18.5% (en-ne), 15.7% (ne-en), 9.2% (en-si), and 3.3% (si-en) of them all $1,000$ samples are unique. On the test domain data these numbers increase to 46.7% (en-de), 41.5% (de-en), 52.1% (en-ne), 86.8% (ne-en), 84.6% (en-si), and 87.3% (si-en). For these input sequences, the translation distributions learnt are so flat that in these $1,000$ samples no single translation stands out over the others.

3.8.2 Sampling the Mode

As the predominant decision rule in NMT is MAP decoding, which we approximate via beam search, it is natural to ask how frequently it is that the beam search output is observed amongst unbiased samples. We find that the beam search output is contained within $1,000$ unbiased samples for between 54.3% and 92.2% of input sequences on the held-out data. In the test domain, we find that on English-German for between 44.3% and 49.3%, and in the low-resource setting for between 4.8% and 8.4% of the input sequences the beam search output is contained in the set. This shows that for a large portion of the input sequences, the beam search solution is thus quite a rare outcome under the model.

[5]For English-German and German-English the test domain would not be considered out-of-domain here, as both training and test data concern newswire.

	Training Domain					Test Domain				
Task	LS	beam	sample	MBR	Oracle	LS	beam	sample	MBR	Oracle
en-de	19.5	19.5	15.3	19.2	22.7	35.2	34.9	20.5	31.5	35.7
de-en	26.6	26.8	21.9	26.2	29.4	39.6	39.4	26.6	37.3	41.0
en-ne	36.7	37.3	36.1	40.5	43.4	32.5	31.3	30.6	34.9	37.0
ne-en	30.6	29.8	26.7	30.2	35.4	19.2	17.2	12.8	16.6	20.1
en-si	34.2	34.3	31.0	36.3	41.5	31.8	30.3	30.3	34.8	36.8
si-en	29.1	28.9	24.3	29.1	36.2	20.0	18.4	13.7	17.7	21.6
High-resource	23.1	23.1	18.6	22.7	26.0	37.4	37.1	23.6	34.4	38.3
Low-resource	32.7	32.6	29.5	34.0	39.1	25.9	24.3	21.8	26.0	28.9
All	29.5	29.4	25.9	30.2	34.8	29.7	28.6	22.4	28.8	32.0

Table 3.1: METEOR scores under different strategies for prediction: beam search, single sample, MBR, and an oracle rule. MBR and the oracle both use 30 ancestral samples and sentence-level METEOR as utility, but the oracle has access to the reference. To show that our MLE-trained systems are competitive with LS-trained systems, we list the LS column (using beam search). The sample columns show average scores of 30 independent samples from the model. All standard deviations were below 0.2.

Stahlberg and Byrne (2019) showed that oftentimes the true mode of a trained NMT system is the empty sequence. This is worrying since NMT decoding is based on mode-seeking search. We find that for between 7.2% and 29.1% of input sequences for held-out data and between 2.8% and 33.3% of input sequences in the test domain an empty sequence is sampled at least once in $1,000$ samples. When an empty sequence is sampled it only occurs on average 1.2 ± 0.5 times. Even though it could well be, as the evidence that Stahlberg and Byrne (2019) provide is strong, that often the true mode under our models is the empty sequence, the empty string remains a rather unlikely outcome under the models.

3.8.3 Sample Quality

The number of translations that an NMT model assigns non-negligible probability mass to can be very large as we have seen in Section 3.8.1. We now investigate what the average quality of these samples is. For quality assessments, we compute METEOR (Denkowski and Lavie, 2011) using the `mteval-v13a` tokeniser.[6] We translate the test sets using a single ancestral sample per input sentence and repeat the experiment 30 times to report the average in Table 3.1 (sample). We also report beam search scores (beam). We see that, on average, samples of the model always perform worse than beam search translations. This is no surprise,

[6]For our analysis, it is convenient to use a metric defined both at the corpus and at the segment level. We use METEOR, rather than BLEU (Papineni et al., 2002), for it outperforms (smoothed) BLEU at the segment-level (Ma et al., 2018).

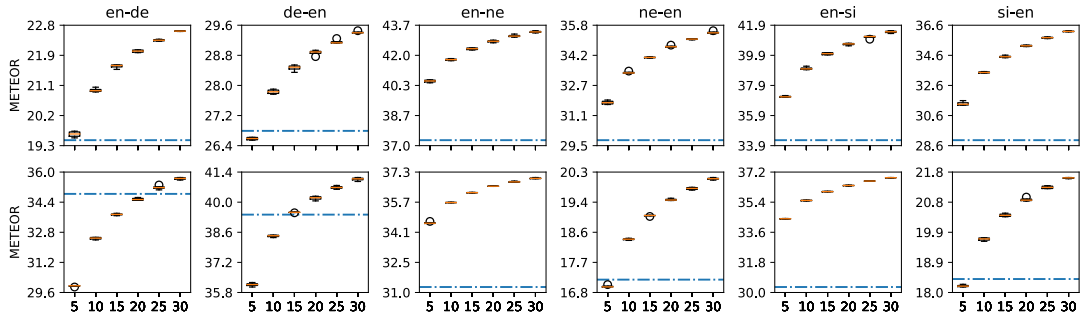


Figure 3.3: METEOR scores for oracle-selected samples as a function of sample size on the held-out data (top) and *newstest2018* / FLORES(bottom) test sets. For each sample size we repeat the experiment 4 times and show a box plot per sample size. Dashed blue lines show beam search scores.

of course, as ancestral sampling is not a fully fledged decision rule, but simply a technique to unbiasedly explore the learnt distribution. Moreover, beam search itself does come with some adjustments to perform well (such as a specific beam size and length penalty). The gap between sampling and beam search is between 0 and 14.4 METEOR. The gap can thus be quite large, but overall the quality of an average sample is reasonable compared to beam search. We also observe that the variance of the sample scores is small with standard deviations below 0.2.

Next, we investigate the performance we would achieve if we could select the best sample from a set. For that, we employ an oracle selection procedure using sentence-level METEOR with the reference translation to select the best sample from a set of samples. We vary sample size from 5 to 30 samples and repeat each experiment four times. Figure 3.3 plots the results in terms of corpus-level METEOR. Average METEOR scores for oracle selection out of 30 samples are shown in Table 3.1. METEOR scores steadily increase with sample size. For a given sample size we observe that variance is generally very small. Only between 5 and 10 samples are required to outperform beam search in low-resource language pairs and English-German in the training domain, but surprisingly 15 to 25 samples are necessary for English-German in the test domain. Still, this experiment shows that samples are of reasonable and consistent quality with respect to METEOR. For fewer than 30 random samples the model could meet or outperform beam search performance in most cases, if we knew how to choose the best sample from the set. This is a motivating result for looking into sampling-based decision rules.

3.8.4 Minimum Bayes Risk Decoding

We have seen that translation distributions spread mass over a large set of likely candidates, oftentimes without any clear preference for particular translations within the set (Section 3.8.1). Yet, this set is not arbitrary, it captures various statistics of the data well (Section 3.6) and holds potentially good translations (Section 3.8.3). Even though the model does not single out one clear winner, the translations it does assign non-negligible mass to share statistics that correlate with the reference translation. This motivates a decision rule that exploits all information we have available about the distribution. In this section we explore one such decision rule: minimum Bayes risk (MBR) decoding. We will propose to use sampling-based approximation to it and perform a small proof-of-concept experiment to motivate further investigation in the next chapter of the thesis.

For a given *utility function* $u(y, h)$, which assesses a translation candidate h against a reference y, statistical decision theory (Bickel and Doksum, 1977) prescribes that the optimum decision $y^\star$ is the one that maximises expected utility (or minimises expected loss) under the model: $y^\star = \text{argmax}_{h \in \mathcal{H}(x)} \ \mathbb{E}_{p(Y|x,\theta)}[u(Y, h)]$, where the maximisation is over the entire set of possible translations $\mathcal{H}(x)$. Note that there is no need for a human-annotated reference, expected utility is computed by having the model *fill in* reference translations. This decision rule, known as MBR decoding in the NLP literature (Goel and Byrne, 2000), is especially suited where we trust a model in expectation but not its mode in particular (Smith, 2011, Section 5.3).[7] MBR decoding, much like MAP decoding, is intractable. We can at best obtain unbiased estimates of expected utility via Monte Carlo (MC) sampling, and we certainly cannot search over the entirety of $\mathcal{H}(x)$. Still, a tractable approximation can be designed, albeit without any optimality guarantees. We use MC both to approximate the support $\mathcal{H}(x)$ of the distribution and to estimate the expected utility of a given translation candidate. In particular, we maximise over the support $\bar{\mathcal{H}}(x)$ of the empirical distribution obtained by ancestral sampling:

$$y^\star = \underset{h \in \bar{\mathcal{H}}(x)}{\text{argmax}} \ \frac{1}{S} \sum_{s=1}^{S} u(y^{(s)}, h) \quad \text{for } y^{(s)} \sim p(y|x, \theta) \ , \tag{3.1}$$

which runs in time $\mathcal{O}(S^2)$. Though approximate, this rule has interesting properties: MC improves with sample size, occasional pathologies in the set pose no threat, and there is no need for incremental search.

Note that whereas our translation distribution might be very flat over a vast number of translations, not showing a clear ordering in terms of relative frequency

[7] MAP decoding is in fact MBR with a very strict utility function which evaluates to 1 if a translation exactly matches the reference, and 0 otherwise (Kumar and Byrne, 2004, Eq. 5). As a community, we acknowledge by means of our evaluation strategies (manual or automatic) that exact matching is inadequate for translation, unlike many unstructured classification problems, admits multiple solutions.

within a large set of samples, this need not be the case under expected utility. For example, in Section 3.8.2 we found that for some input sequences the empty sequence is contained within the 1,000 samples in our set and appears in there roughly once on average. If all the 1,000 samples are unique (as we found to often be the case in Section 3.8.1), we cannot distinguish the empty sequence from the other 999 samples in terms of relative frequency. However, under most utilities the empty sequence is so unlike the other sampled translations that it would score very low in terms of expected utility.

Here, we chose METEOR as utility function for it, unlike BLEU, is well-defined at the sentence level.[8] We estimate expected utility using $S = 30$ ancestral samples, and use the translations we sample to make up an approximation to $\mathcal{H}(x)$. Results are shown in Table 3.1. As expected, MBR considerably outperforms the average single sample performance by a large margin and in many cases is on par with beam search, consistently outperforming it in low-resource pairs. For English-German in the test domain, we may need more samples to close the gap with beam search. In the next chapter, we will explore sampling-based minimum Bayes risk in much greater detail. However, in this experiment we see that sampling-based MBR produces promising results. Crucially, it shows that exploring the model as a probability distribution holds great potential.

3.9 Related Work

Some of our observations have been made in previous work. Ott et al. (2018) observe that unigram statistics of beam search stray from those of the data, while random samples do a better job at reproducing them. Holtzman et al. (2020) find that beam search outputs have disproportionately high token probabilities compared to natural language under a sequence to sequence model. Our analysis is more extensive, we include richer statistics about the data, more language pairs, and vary the amount of training resources, leading to new insights about MLE-trained NMT and the merits of mode-seeking predictions.

Ott et al. (2018) also observe that NMT learns flat distributions, they analyse a high-resource English-French system trained on 35.5 million sentence pairs from WMT'14 and find that after drawing $10,000$ samples from the WMT'14 validation set less than 25% of the probability space has been explored. Our analysis shows that even though NMT distributions do not reveal clear winners, they do emphasise translations that share statistics with the reference, which motivates us to look into MBR.

MBR decoding is old news in machine translation (Kumar and Byrne, 2004;

[8] Even though one can alter BLEU such that it is defined at the sentence level (for example, by adding a small positive constant to n-gram counts), this "smoothing" in effect biases BLEU's sufficient statistics. Unbiased statistics are the key to MBR, thus we opt for a metric that is already defined at the sentence level.

Tromble et al., 2008), but it has received little attention in NMT. Previous approximations to MBR in NMT employ beam search to define the support and to evaluate expected utility (with probabilities renormalised to sum to 1 in the beam), these studies report the need for very large beams (Stahlberg et al., 2017; Blain et al., 2017; Shu and Nakayama, 2017). They claim the inability to directly score better translations higher is a deficiency of the model scoring function. We argue this is another piece of evidence for the inadequacy of the mode: by using beam search, they emphasise statistics of high-scoring translations, potentially rare and inadequate ones. Very recently, Borgeaud and Emerson (2020) present a voting-theory perspective on decoding for image captioning and machine translation. Their proposal is closely-related to MBR, but motivated differently. Their decision rule too is guided by beam search, which may emphasise pathologies of highest-probability paths, but they also propose and investigate stronger utility functions which lead to improvements w.r.t. length, diversity, and human judgements.

The only instance that we are aware of where unbiased samples from an NMT model support a decision rule is the concurrent work by Bhattacharyya et al. (2021). The authors make the same observation that we make in Section 3.8.3, namely that an oracle selection from a small set of samples of an NMT model shows great potential, greatly outperforming beam search. Motivated by this observation, the authors propose a re-ranking model that learns to rank sampled translations according to their oracle BLEU. Using the trained model to re-rank a set of 100 samples from the NMT model they find strong improvements over beam search of up to 3 BLEU points, again showing the potential of sampling-based decision rules.

3.10 Consequent Work

Since the publication of Eikema and Aziz (2020) a number of works have attempted to better understand the inadequacy of the mode across text generation tasks. Stahlberg et al. (2022) show that the inadequacy of the mode, the entropy of output distributions and the ability of beam search to find the exact mode are all tied to the aleatoric uncertainty that is present at the task or sequence level. They find that tasks or input conditionals that allow for more valid answers, like machine translation (many valid translations) versus grammatical error correction (one or a few valid corrections), increases the degree of the inadequacy of the mode, the flatness of the output distribution and the amount of search errors that are made by beam search. Before that, Forster et al. (2021) already observed adequate modes for character-level grammatical error correction models and conjectured that the aleaoric uncertainty present in more complex text generation tasks may be an important predictor for the inadequacy of the mode. Riley and Chiang (2022) show something similar by smoothly reducing the fraction of the in-

put sequence that is available, thereby reducing the context available to the model and increasing the degree of aleatoric uncertainty. They show that repetition and length bias, common problems with high probability translations, are worsened when less context is available, suggesting that even less constrained tasks than machine translation, e.g. fully open-ended generation, suffers from an even less adequate mode, explaining why beam search doesn't work well for these type of tasks (Holtzman et al., 2020). Meister et al. (2022) propose an explanation for the inadequacy of the mode in the form of the expected information hypothesis, which states that human-like text should have information values (negative log probabilities) close to the entropy of the distribution of natural language. They base this around the observation that the information content of human text, as assessed by NLG models, often falls around the entropy of sequence models. Interestingly, MBR outputs also fall around the entropy of those same models in their experiments across tasks, whereas mode-seeking generation methods such as beam search and nucleus sampling only do for select tasks (Meister et al., 2022, Figure 9). Yoshida et al. (2024) suggest that the inadequacy of the mode may be caused by the presence of consistent noise in the training data, arguing that at relatively small rates of noise even optimally trained models may place such noise at the mode of the distribution. They also show that the inadequacy of the mode has not been resolved by the recent enormous increases in model size and training data size, finding that even recent large language models suffer from the inadequacy of the mode.

3.11 Conclusion

In this chapter, we discussed the inadequacy of the mode in NMT and questioned the appropriateness of MAP decoding. We showed that for such a high dimensional problem as NMT, the probability distributions obtained with MLE are spread out over many translations, and that the mode often does not represent any significant amount of probability mass under the learnt distribution. We therefore argue that MAP decoding is not suitable as a decision rule for NMT systems. Whereas beam search performs well in practice, it suffers from biases of its own (*i.e.*, non-admissible heuristic search bias), it shifts statistics away from those of the data (*i.e.*, exposure bias and other lexical and length biases), and in the limit of perfect search it falls victim to the inadequacy of the mode. Instead, we advocate for research into decision rules that take into account the probability distribution more holistically. Using ancestral sampling we can explore the learnt distribution in an unbiased way and devise sampling-based decision rules. Ancestral sampling does not suffer from non-admissibility, and, if the model fit is good, there is no distribution shift either.

We further argue that criticisms about properties of the mode of an NMT system are not representative of the probability distributions obtained from MLE

training. While this form of criticism is perfectly reasonable where approximations to MAP decoding are the only viable option, there are scenarios where we ought to criticise models as probability distributions. For example, where we seek to correlate an observed pathology with a design decision, such as factorisation, or training algorithm. In fact, we argue that many of the observed pathologies and biases of NMT are at least partially due to the use of (approximate) MAP decoding, rather than inherent to the model or its training objective.

Even though NMT models spread mass over many translations, we find samples to be of decent quality and contain translations that outperform beam search outputs even for small sample sizes, further motivating the use of sampling-based decision rules. We show that an approximation to one such decision rule, MBR decoding, shows competitive results. This confirms that while the set of likely translations under the model is large, the translations in it share many statistics that correlate well with the reference.

MLE-trained NMT models admit probabilistic interpretation and an advantage of the probabilistic framework is that a lot of methodology is already in place when it comes to model criticism as well as making predictions. We therefore advocate for criticising NMT models as probability distributions and making predictions using decision rules that take into account the distributions holistically. In the next chapter, we will look at the properties of sampling-based minimum Bayes risk decoding in NMT and explore various scaleable approximations for expected utility.

Chapter 4

Sampling-Based Minimum Bayes Risk

In this chapter we take a closer look at minimum Bayes risk (MBR) decoding and the sampling-based proof-of-concept approximation that we proposed for it in the previous chapter. We will explore the properties of this decision rule and approximation strategy as well as explore more efficient approximations to expected utility to combat the increased computational complexity of sampling-based MBR decoding compared to standard beam search decoding. We will explore strategies for generating candidates for MBR decoding and motivate why even then unbiased estimates of expected utility are crucial. The contents of this chapter are based around Eikema and Aziz (2022), published at the Empirical Methods in Natural Language Processing (EMNLP) conference in Abu Dhabi in 2022.[1]

Chapter Highlights

Problem Statement

- A sampling-based minimum Bayes risk approximation was proposed as more suitable alternative to beam search decoding in Eikema and Aziz (2020). However, little is known about its properties, such as how it scales with more computation.
- Sampling-based MBR decoding is considerably more expensive than standard beam search decoding, which uses a small beam size.
- Uses of MBR decoding prior to Eikema and Aziz (2022) made use of biased approximations using beam search with a large beam size or biased sampling techniques to generate candidates *and* to estimate expected utility.

[1] A preprint of this work was already available in 2021 (Eikema and Aziz, 2021).

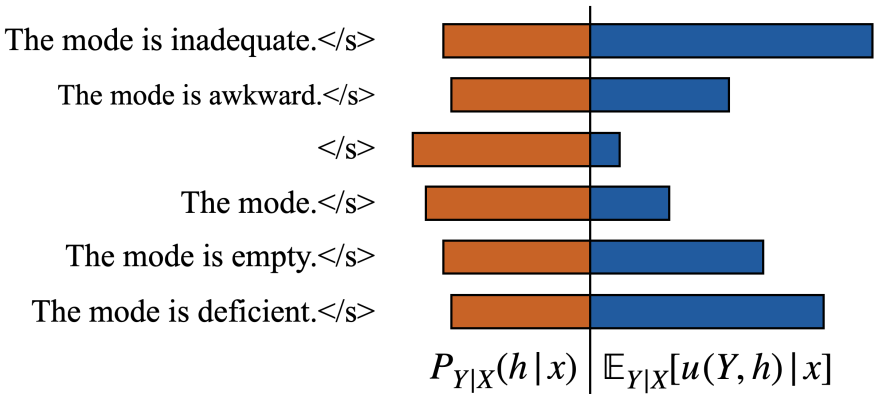


Figure 4.1: NMT spreads probability roughly uniformly over a large set of promising hypotheses (left). MBR (right) assigns hypotheses an expected utility, revealing clear preferences against those that are too idiosyncratic.

Contributions

- We establish that sampling-based minimum Bayes risk decoding does not suffer from an equivalent of the beam search curse. In fact, we find that better search to the exact MBR solution, in our experiments, always leads to better or equal translation performance. This solidifies expected utility as a robust criterion for making decisions.
- We propose more efficient approximations that allow performance in MBR with equivalent computational budget, bringing sampling-based MBR closer to being practically viable.
- We show that we can decouple candidate generation from estimation of expected utility, and show that mode-seeking strategies can still be useful for generating candidates in MBR, as unbiased estimates of expected utility are robust enough to filter out any idiosyncracies in the candidate space.

4.1 Introduction

NMT systems (Sutskever et al., 2014; Bahdanau et al., 2015) are trained to predict a conditional probability distribution over translation candidates of any given source sentence. After training, choosing a translation for a given input requires a decision rule: a criterion to elect a 'preferred' translation. MAP decoding, the most common decision rule in NMT, seeks the most probable translation under the model (*i.e.*, the mode of the distribution).

MAP decoding and its approximations such as beam search (Graves, 2012) have been under scrutiny. Stahlberg and Byrne (2019) show that the true mode is oftentimes inadequately short or empty. Better approximate search is known to hurt quality (Koehn and Knowles, 2017; Murray and Chiang, 2018; Kumar and Sarawagi, 2019), a problem known as the *beam search curse*. The success of beam search depends on search biases introduced by hyperparameters such as beam size and length normalisation, which are tuned not to correlate with the objective of MAP decoding, but rather to strike a compromise between mode-seeking search and properties of reasonable translations. Despite its success, a number of problems have been observed such as a length bias (Cho et al., 2014; Sountsov and Sarawagi, 2016), a word frequency bias (Ott et al., 2018), susceptibility to copy noise (Khayrallah and Koehn, 2018; Ott et al., 2018), and hallucination under domain shift (Lee et al., 2019; Müller et al., 2020; Wang and Sennrich, 2020).

In the previous chapter we argued that the inadequacy of the mode in NMT does not have to mean that our models are bad. We showed that distributions predicted by NMT do reproduce various statistics of observed data, but they tend to spread probability mass almost uniformly over a large space of translation candidates. This makes their precise ranking in terms of probability mass a fragile criterion for prediction. While some of these candidates are possibly inadequate (*e.g.*, the empty sequence), most of them are similar to one another and exhibit appreciable structural similarity to reference translations. To make better use of the statistics predicted by NMT models, we recommended MBR decoding (Kumar and Byrne, 2004), a decision rule that seeks the translation candidate which maximises an external notion of utility (*e.g.*, an MT evaluation metric) in expectation under the model distribution. While MBR decoding promises robustness to idiosyncratic translations, it remains intractable, much like MAP decoding. We specifically proposed an approximation based on Monte Carlo (MC) sampling, which although tractable in principle, requires a prohibitive number of assessments of the utility function.

In this work, we first analyse the procedure introduced in the previous chapter and establish that it does not suffer from a counterpart to the beam search curse. That is, better search does not hurt translation quality. The proposed approximation is, however, computationally expensive, requiring a number of assessments of the utility function that is quadratic in sample size. In this chapter, we propose algorithms that scale linearly, allowing us to explore large hypothesis spaces, and considerably improve upon existing approximations to MBR with less computation. Finally, we find that mode-seeking strategies such as nucleus sampling and beam search can still aid MBR decoding by constructing compact sets of high expected utility hypotheses, relying on MBR to filter idiosyncratic translations that may be present.

4.2 Minimum Bayes Risk Decoding

In Section 3.8.4 we proposed a sampling-based approximation to minimum Bayes risk (MBR) decoding. Here, we recap and elaborate on that approach and introduce the necessary notation for this chapter.

Minimum Bayes risk decoding stems from the principle of maximisation of expected utility (Berger, 1985). A utility function $u(y, h)$ measures the benefit in choosing $h \in \mathcal{Y}$ when $y \in \mathcal{Y}$ is the ideal decision. When forming predictions, we lack knowledge about ideal translations and must decide under uncertainty. MBR lets the model fill in 'ideal decisions' probabilistically as we search through the space of candidates for the one which is assigned highest utility *in expectation*:

$$y^{\text{MBR}} = \operatorname*{argmax}_{h \in \mathcal{Y}} \underbrace{\mathbb{E}[u(Y, h) \mid \theta, x]}_{=:\mu_u(h; x, \theta)} \,. \tag{4.1}$$

Note that Y is a random sequence, and we take the expectation with respect to the sequence distribution $P_\theta(Y|x)$.

MBR has a long history in parsing (Goodman, 1996; Sima'an, 2003), speech recognition (Stolcke et al., 1997; Goel and Byrne, 2000), and machine translation (Kumar and Byrne, 2002, 2004). In machine translation, u can be an instance-level evaluation metric (*e.g.*, METEOR (Denkowski and Lavie, 2011) or sentence-BLEU (Chen and Cherry, 2014)). Intuitively, whereas the MAP prediction is the translation to which the model assigns highest probability, no matter how idiosyncratic, the MBR prediction is the translation that is closest (under the chosen u) to all other probable translations. See Figure 4.1 for an illustration of this concept. Seeking support for a prediction not only in terms of probability but also in terms of utility makes MBR decoding robust to situations where inadequate translations are assigned high probability, as it often happens with the empty string (Stahlberg and Byrne, 2019), when the training data are noisy (Ott et al., 2018), too small (Eikema and Aziz, 2020) or distant from the test domain (Müller and Sennrich, 2021).

It is a well-known result that for the 'exact match' utility, $u(y, h) \coloneqq \mathbf{1}_{\{y\}}(h)$, the expected utility of h is $p_{Y|X}(h|x, \theta)$, hence MBR and MAP decoding have the same optimum under this choice (Kumar and Byrne, 2002). This view justifies MAP decoding as an instance of MBR, where decisions are optimised with respect to a strict notion of translational equivalence. In machine translation evaluation, exact match is a questionable choice of utility function. It, for example, is unable to capture paraphrases or any other form of semantic equivalence.

Like in MAP decoding, exhaustive enumeration of all hypotheses is impossible, we must resort to a finite subset $\bar{\mathcal{H}}(x)$ of candidates. Unlike MAP decoding, the objective function $\mu_u(h; x, \theta)$ *cannot* be evaluated exactly. Most approximations to MBR decoding, from Kumar and Byrne (2004) to recent instances (Stahlberg et al., 2017; Shu and Nakayama, 2017; Blain et al., 2017), use k-best

lists from beam search for $\bar{\mathcal{H}}(x)$ and to form a biased estimate of expected utility. In the previous chapter, we used unbiased samples from the model for both approximations: *i)* we followed the generative story in Equation (2.2) to obtain N independent samples $y^{(n)}$, a procedure known as ancestral sampling (Bishop, 2006); then, *ii)* for a hypothesis h, we computed an MC estimate of $\mu_u(h; x, \theta)$:

$$\hat{\mu}_u(h; x, N) \overset{\text{MC}}{\coloneqq} \frac{1}{N} \sum_{n=1}^{N} u(y^{(n)}, h) \ , \tag{4.2}$$

which is unbiased for any sample size N. We used the same N samples as candidates and approximated Equation (4.1) by

$$y^{\text{N-by-N}} \coloneqq \underset{h \in \{y^{(1)}, \ldots, y^{(N)}\}}{\operatorname{argmax}} \hat{\mu}_u(h; x, N) \ . \tag{4.3}$$

We note that the candidates do not need to be obtained using ancestral sampling, and investigate alternative strategies in Section 4.5.4. It is important, however, to use ancestral samples to obtain an unbiased estimate of expected utility as we show in Section 4.5.1. We call this class of MBR algorithms using unbiased MC estimation instances of *sampling-based MBR decoding*.

4.3 Coarse-to-Fine MBR Decoding

A big disadvantage of $\text{MBR}_{\text{N-by-N}}$ is that it requires N^2 assessments of the utility function. If U is an upperbound on the time necessary to assess the utility function once, then $\text{MBR}_{\text{N-by-N}}$ runs in time $\mathcal{O}(N^2 \times U)$. For a complex utility function, this can grow expensive even for a modest hypothesis space. As NMT distributions have been shown to be high entropy (Ott et al., 2018; Eikema and Aziz, 2020), the quadratic cost prevents us from sufficiently exploring the space of translations. Therefore, we investigate and propose more flexible algorithms.

An important property of sampling-based MBR decoding is that MC estimation of expected utility, Equation (4.2), and approximation of the hypothesis space in Equation (4.3) really are two independent approximations. Tying the two is no more than a design choice that must be reconsidered. We start by obtaining N translation candidates from the model, which will form the hypothesis space $\bar{\mathcal{H}}(x)$. Then, we use any number $S < N$ of ancestral samples for approximating expected utility in Equation (4.2).[2] We call this version $\text{MBR}_{\text{N-by-S}}$, which takes time $\mathcal{O}(N \times S \times U)$. Compared to $\text{MBR}_{\text{N-by-N}}$, this variant is able to scale to much larger hypothesis spaces $\bar{\mathcal{H}}(x)$. In practice, however, robust MC estimation for the utility of interest may still require S that is too large for the N we are interested in.

[2] In practice, for efficiency we will use a fixed set of S samples to estimate expected utility for each candidate.

src	Convercent erhielt $10 Millionen bei der Finanzierung im Februar von Firmen wie Sapphire Ventures und Tola Capital, womit das gesamte Kapital auf $47 Millionen angehoben wurde.
ref	Convercent raised $10 million in funding in February from firms such as Sapphire Ventures and Tola Capital, bringing its total capital raised to $47 million.

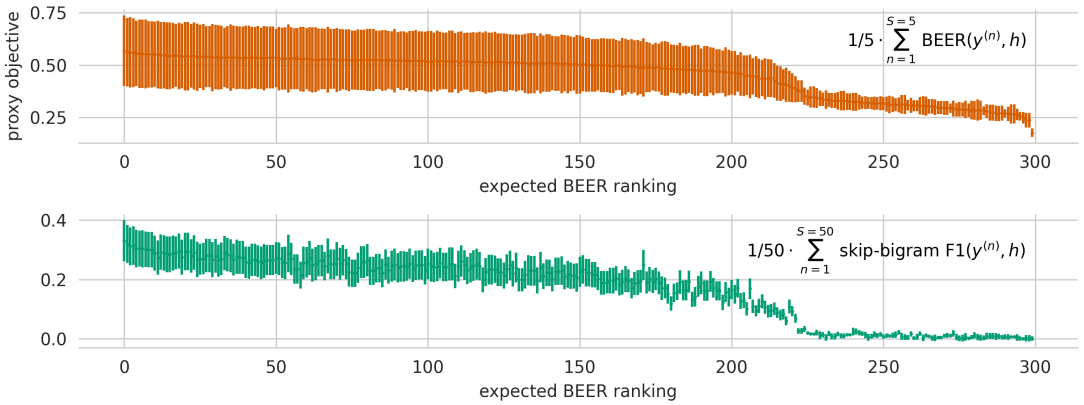


Figure 4.2: Motivation for coarse-to-fine MBR. We sort 300 candidates sampled from the model along the x-axis from best to worst according to a robust MC estimate (using 1,000 samples) of expected BEER under the model. Left: feasible MC estimates (5 samples) of each candidate's expected BEER. Right: robust and inexpensive MC estimates (100 samples) of expected utility w.r.t. a simpler metric (skip-bigram F1). As estimates are stochastic, we perform 100 repetitions and plot mean ± two deviations. We can see that the robust estimates (right) correlate fairly well with the expensive ranking we intend to approximate (x-axis), despite of the simpler utility. As we can afford more evaluations of the proxy utility, we obtain estimates of reduced variance, which leads to safer pruning.

An idea that we explore in this work is to make use of a proxy utility that correlates with the target utility but is cheaper to compute. Even when those do not correlate perfectly, we can make use of the proxy utility to filter the hypothesis space to a manageable size T on which we can perform robust MC estimation of expected utility. We coin this approach coarse-to-fine MBR decoding (or MBR_{C2F}), which filters the hypothesis space to a manageable size in the coarse step, and performs robust MC estimation of expected utility in the fine step:

$$y^{\text{C2F}} \coloneqq \operatorname*{argmax}_{h \in \bar{\mathcal{H}}_T(x)} \hat{\mu}_{u_{\text{target}}}(h; x, L) \tag{4.4a}$$

$$\bar{\mathcal{H}}_T(x) \coloneqq \operatorname*{top\text{-}T}_{h \in \bar{\mathcal{H}}(x)} \hat{\mu}_{u_{\text{proxy}}}(h; x, S)\ . \tag{4.4b}$$

Upper-bounding the complexity of the proxy utility by U_{proxy}, the target utility by U_{target}, using S samples for MC estimation in the coarse step (4.4b) and L in the fine step (4.4a), the complexity of this algorithm is $\mathcal{O}(N \times S \times U_{\text{proxy}} + T \times$

$L \times U_{\text{target}}$). MBR_{C2F} decouples robust MC estimation (large L) from exploration (large N) and the cost of exploration from the cost of the target utility.

As illustrated in Figure 4.2, we can find proxy utilities that correlate reasonably well with our target utility and are able to give us a rough—but useful—ordering of the hypothesis space. Rather than using a proxy utility, we could use the target utility itself in the coarse-step provided we pick a small S. This, however, most likely leads to too high variability in the ranking, as shown in Figure 4.2 (left).

4.4 Data, Systems and Utilities

We perform experiments on three language pairs with varying amount of resources for training: English into and from German, Romanian and Nepali. For German-English (de-en) we use all available WMT'18 (Bojar et al., 2018) news data except for Paracrawl, resulting in 5.9 million sentence pairs. We train a Transformer base model (Vaswani et al., 2017) until convergence and average the last 10 epoch checkpoints to obtain our final model. We test our models on `newstest2018`. For Romanian-English (ro-en) we use all available WMT'16 (Bojar et al., 2016a) news data amounting to 565k sentence pairs. We train a Transformer base model until convergence and pick the best epoch checkpoint according to the validation loss. We test our models on `newstest2016`. Finally, for Nepali-English (ne-en) we use the data setup by Guzmán et al. (2019). We apply the pre-processing step of removing duplicates as in the previous chapter. This results in 235k sentence pairs. We test our models on the FLORES test set, which is of a widely different domain than the training data. We mimick the training setup and models used in Guzmán et al. (2019). In all models we disable label smoothing, as we previously found this to negatively impact model fit, which would compromise the performance of MBR.

For computational efficiency, we opt for non-neural evaluation metrics for use as utility function in MBR. BEER (Stanojević and Sima'an, 2014) is a non-neural trained metric that has shown good correlation with human judgements in previous WMT metrics shared tasks (Macháček and Bojar, 2014; Stanojević et al., 2015; Bojar et al., 2016b). In experiments shown in Table 4.1 in Section 4.4.1 we find that using BEER as utility function performs well at pushing translation performance higher across a range of automatic evaluation metrics. We therefore use BEER as the utility of choice in our experiments and as a consequence will consistently report corpus-level BEER scores of MBR translations as well. We also report SacreBLEU (Papineni et al., 2002; Post, 2018) scores where relevant to be able to detect overfitting to the utility and for comparison with other works.

Task	Utility	BEER	BLEU	METEOR	ChrF++
en-de	BEER	**64.3**	37.0	56.6	61.3
	sentence-BLEU	63.3	**37.5**	55.9	60.2
	METEOR	62.5	33.4	**57.8**	60.5
	ChrF++	63.2	34.9	56.9	**61.4**
de-en	BEER	**64.9**	38.0	39.3	61.0
	sentence-BLEU	64.3	**38.3**	38.9	60.3
	METEOR	63.5	36.1	**39.7**	59.8
	ChrF++	64.4	37.2	39.5	**61.5**
en-ro	BEER	**54.8**	21.0	33.9	47.8
	sentence-BLEU	54.4	**21.3**	40.4	47.4
	METEOR	54.5	20.9	**40.9**	47.7
	ChrF++	54.2	20.2	40.3	**48.0**
ro-en	BEER	**58.4**	27.5	32.4	52.0
	sentence-BLEU	57.8	**27.8**	32.2	51.4
	METEOR	57.5	26.6	**32.9**	51.5
	ChrF++	58.0	27.1	32.7	**52.6**
en-ne	BEER	**38.4**	**3.4**	11.0	26.1
	sentence-BLEU	34.9	3.0	10.9	22.7
	METEOR	37.3	**3.4**	**13.2**	25.3
	ChrF++	36.8	2.6	12.3	**26.6**
ne-en	BEER	**42.7**	**6.0**	17.0	31.2
	sentence-BLEU	39.9	5.7	15.1	28.4
	METEOR	40.4	4.6	**17.3**	30.8
	ChrF++	40.6	4.8	17.0	**32.0**

Table 4.1: Comparing BEER, sentence-BLEU, METEOR and ChrF++ as utility functions in $\text{MBR}_{\text{N-by-S}}$ using $N = 405$ and $S = 100$.

4.4.1 Comparing Target Utilities

We compare a number of utility functions for use in MBR decoding. In principle any function that measures some notion of similarity across sequences and can be reliably assessed on the sentence-level is suitable as a utility function for MBR. As BLEU is the predominant automatic evaluation metric on which translation quality is assessed, we experiment with a smoothed version of BLEU (Papineni et al., 2002) that can work on the sentence-level: sentence-BLEU (Chen and Cherry, 2014) using the default parameters in Post (2018). We further try METEOR (Denkowski and Lavie, 2011) as we previously found this to produce good results.[3] BEER (Stanojević and Sima'an, 2014) is a character-based metric that has shown to correlate well with human judgements in many WMT metrics tasks (Macháček and Bojar, 2014; Stanojević et al., 2015; Bojar et al., 2016b). Finally, we also explore ChrF++ (Popović, 2017), another character based metric that is an improved version of ChrF (Popović, 2015).

We perform $\text{MBR}_{\text{N-by-S}}$ with $N = 405$ and $S = 100$ in order to perform the comparisons. We measure the performance of each utility on BEER, BLEU, METEOR and ChrF++. Our results are shown in Table 4.1. As expected, using a certain utility achieves the best performance under the lens of that metric as well. Sometimes we find a small deviation from this when BEER or METEOR outperforms sentence-BLEU in terms of BLEU score. This is likely due to sentence-BLEU only being an approximation to BLEU itself. We find that overall BEER seems to do best across metrics followed by ChrF++. One attempt to quantify this more clearly is by normalising the scores per language pair and evaluation metric compared to the maximum score obtained by the best scoring system for that metric and language pair. This leads to the following average performances per evaluation metric: BEER 0.978, METEOR 0.968, ChrF++ 0.964, and sentence-BLEU 0.955. This indeed shows a slight edge of BEER over the other utilities tested in pushing scores across our evaluation metrics. Therefore, we have used BEER as the utility of choice. The finding that BEER works well as a utility function in MBR was also made before in the work of Blain et al. (2017).

4.5 Experiments

4.5.1 Estimation of Expected Utility

We start by motivating the importance of unbiased estimates of expected utility using ancestral samples (*i.e.* sampling-based MBR). In Figure 4.3 we verify the biasedness of alternatives to ancestral sampling for this computation: nucleus

[3]We use a slightly different version of METEOR than in the previous chapter. We now use language-specific versions rather than a language-agnostic version used there.

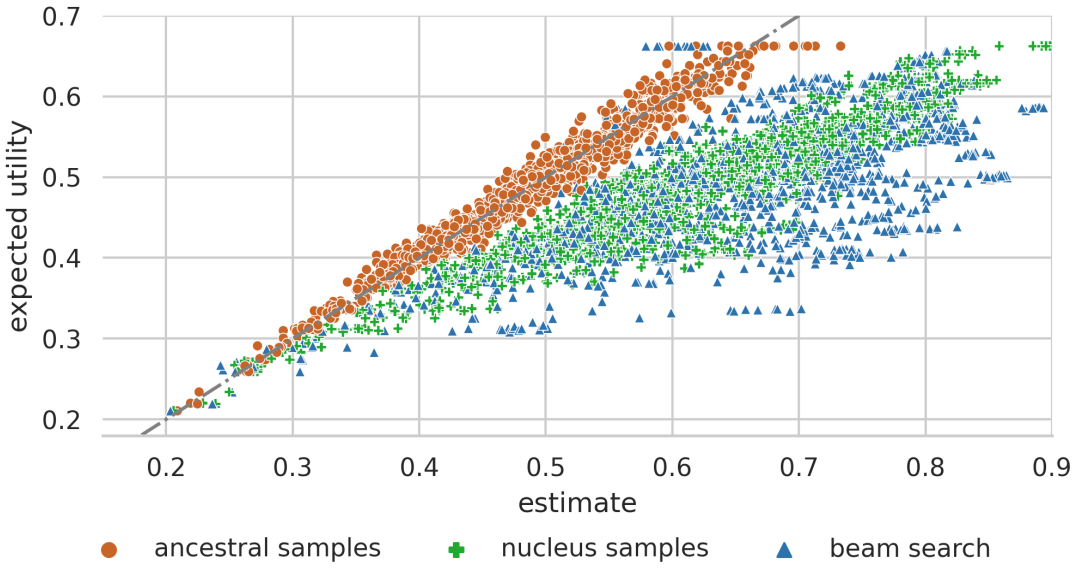


Figure 4.3: Estimates of expected utility for various hypotheses. We plot practical estimates of expected utility (x-axis) using either 100 ancestral, nucleus or 'beam' samples against an accurate MC estimate using 1,000 ancestral samples. The gray line depicts a perfect estimator.

sampling (Holtzman et al., 2020) and 'beam sampling' (*i.e.*, using k-best outputs from beam search for estimating expected utility; Blain et al. (2017)). We can see, rather clearly, that estimates using nucleus samples or beam search bias away from expected utility under the model, while ancestral sampling is unbiased by design and hence should be preferred when approximating the objective function in search. Therefore, in all experiments that follow, we shall use ancestral samples for making unbiased estimates of expected utility, even when different methods are used to construct the hypothesis space.

4.5.2 N-by-N MBR

Now, we look into scaling $\text{MBR}_{\text{N-by-N}}$. Previously, we only explored 30 by 30 approximations to the MBR objective. Here, our aim is to investigate whether MBR decoding is indeed able to scale to better translation performance with more computation. In Figure 4.4, we explore N from 30 to 405.[4] As MBR optimises a specific utility (we use BEER), we report translation quality along both BEER and BLEU to detect overfitting to the metric.

We find that MBR steadily improves across language pairs as N grows larger. BLEU scores improve at a similar rate to that of BEER, showing no signs of

[4]A batch size of 15 is convenient on our hardware, which is why we work with multiples of 15 in most experiments.

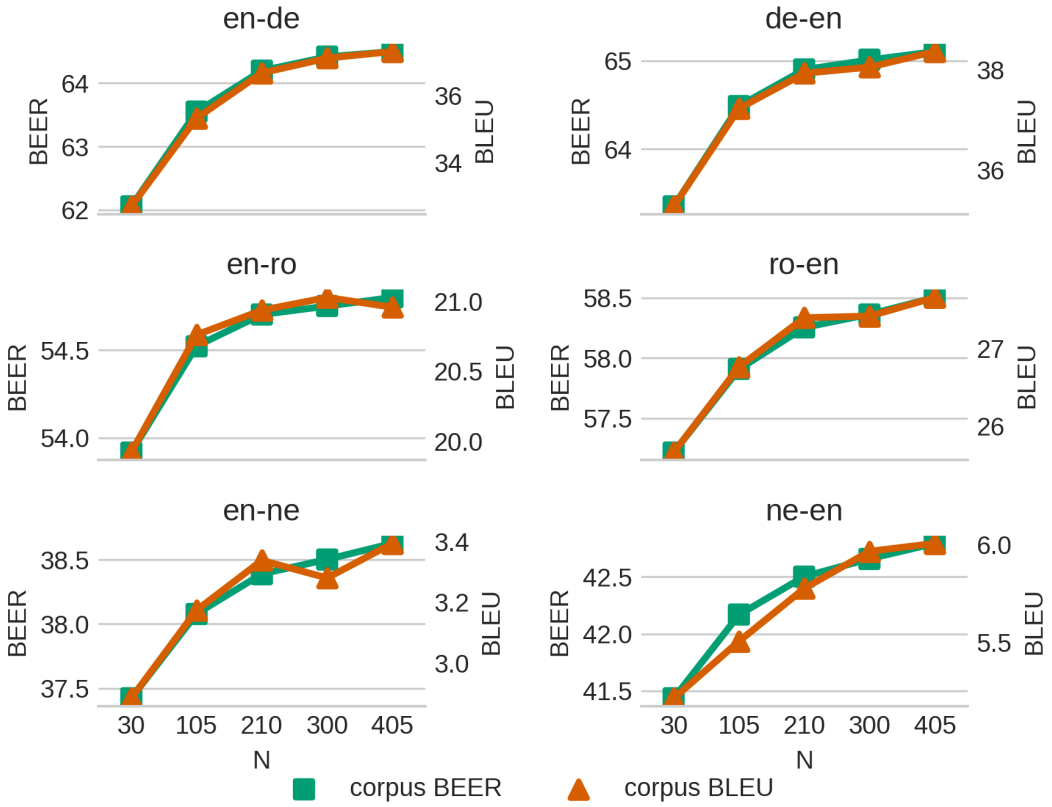


Figure 4.4: $\text{MBR}_{\text{N-by-N}}$ for various sizes of N using BEER as target utility. We report both BEER and BLEU scores.

overfitting to the utility. This is strong empirical evidence that *sampling-based* MBR has no equivalent to the beam search curse. We see this as an important property of a decoding objective.

4.5.3 N-by-S MBR

$\text{MBR}_{\text{N-by-N}}$ couples two approximations, namely, tractable exploration and unbiased estimation of expected utility are based on the same N ancestral samples. Our aim is to learn more about the impact of these two approximations, for which we look into $\text{MBR}_{\text{N-by-S}}$. Moreover, with less than N^2 assessments of utilities per decoding, we can also investigate larger $\bar{\mathcal{H}}(x)$. We explore N ranging from 210 to 1005, while keeping the number of samples used for approximating expected utility of each hypothesis smaller, with S ranging from 10 to 200. We argue that S does not need to grow at the same pace as N, as MC estimates should stabilize after a certain point.[5] See our results in Figure 4.5.

We find that growing N beyond 405 improves translation quality further, even

[5]The standard error of the mean scales with the inverse square root of the sample size.

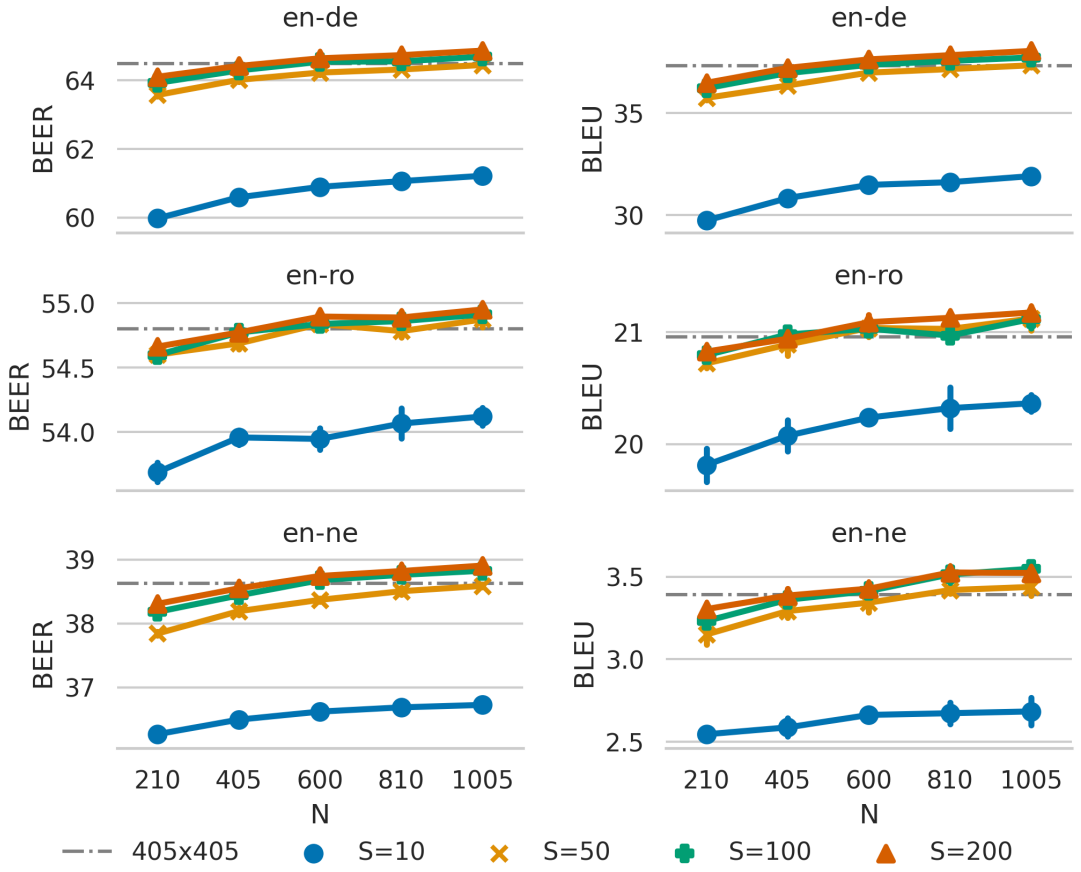


Figure 4.5: $\text{MBR}_{\text{N-by-S}}$: we estimate the expected utility of N hypotheses using S samples. We show average performance over 3 runs with 1 standard deviation. The dashed line shows $\text{MBR}_{\text{N-by-N}}$ performance at $N = 405$.

when the estimates of expected utility are less accurate. Increasing S also steadily improves translation quality, with diminishing returns in the magnitude of improvement. On the other hand, smaller values of S lead to notable deterioration of translation quality and we note higher variance in results. For all language pairs it is possible to improve upon the best $\text{MBR}_{\text{N-by-N}}$ results by considering a larger hypothesis spaces and smaller S. This experiment shows that the two approximations can be controlled independently and better results are within reach if we explore more. On top of that, the best setting of $\text{MBR}_{\text{N-by-N}}$ takes 164,025 utility assessments per decoding, $\text{MBR}_{\text{N-by-S}}$ with $S = 100$ brings this number down to 100,500 for the largest N considered, while improving BEER scores on all language pairs. We note that again increasing either N or S generally improves translation quality in our experiments. This further strengthens our previous finding that sampling-based MBR does not seem to have an equivalent of the beam search curse.

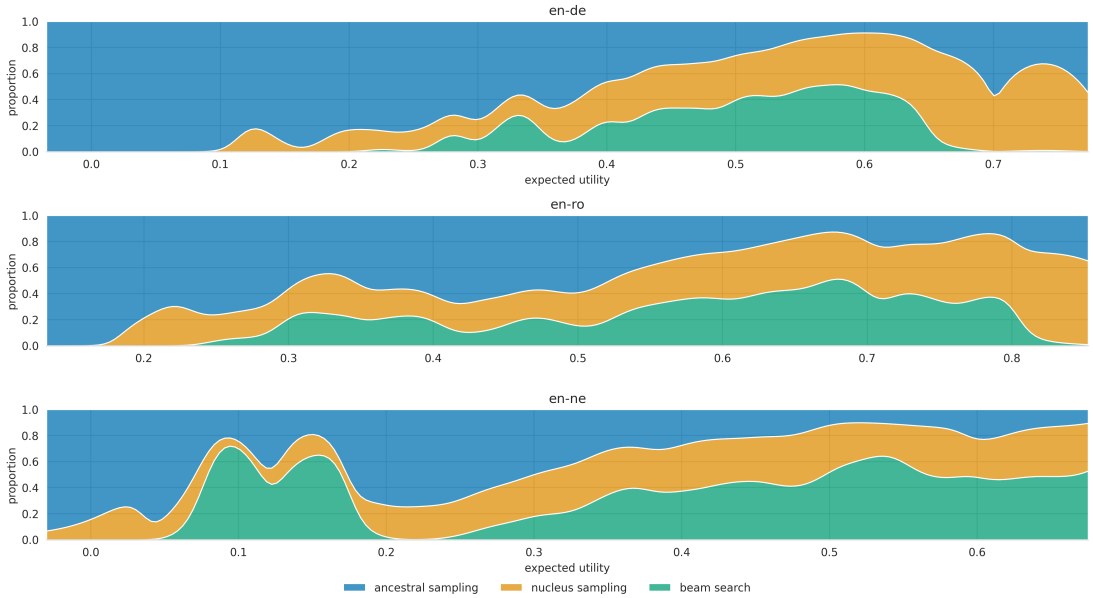


Figure 4.6: Proportion plots of expected utility for 3 strategies for constructing $\bar{\mathcal{H}}(x)$, using 100 translation candidates per strategy. We estimate expected utility using 1,000 samples. Results are aggregated over 100 source sentences.

4.5.4 Choice of Hypothesis Space

While our focus thus far has been on reducing the number of target utility calls, allowing the exploration of larger $\bar{\mathcal{H}}(x)$, one should also take sampling time in consideration. For example, we found that in $\text{MBR}_{\text{N-by-N}}$ with $N = 100$, sampling time made up about 60% of the total translation time for our setup. Therefore, it is computationally attractive to construct compact $\bar{\mathcal{H}}(x)$ with promising translation candidates. Ideally, for better search in MBR, we enumerate a set of high expected utility hypotheses. Up until now we have constructed $\bar{\mathcal{H}}(x)$ using ancestral samples. Strategies like nucleus sampling and beam search are known empirically to produce higher quality translations than ancestral sampling on average and might therefore also enumerate outcomes that have high expected utility. We explore ancestral sampling, nucleus sampling and beam search. In a hyperparameter search we found $p = 0.7$ for nucleus sampling to work best. For beam search we use a length penalty of 1.2 (ne) or 0.6 (de, ro). We compare each strategy by the expected BEER values of the translations generated, using accurate estimates of expected BEER (using 1,000 samples for MC estimation). We show results in Figure 4.6.

We find ancestral sampling to produce hypotheses across the entire range of expected BEER scores. Nucleus sampling and beam search generally produce translations at the higher end of expected BEER. Therefore, these seem more suitable for generating effective $\bar{\mathcal{H}}(x)$ at smaller N. Nucleus sampling seems to

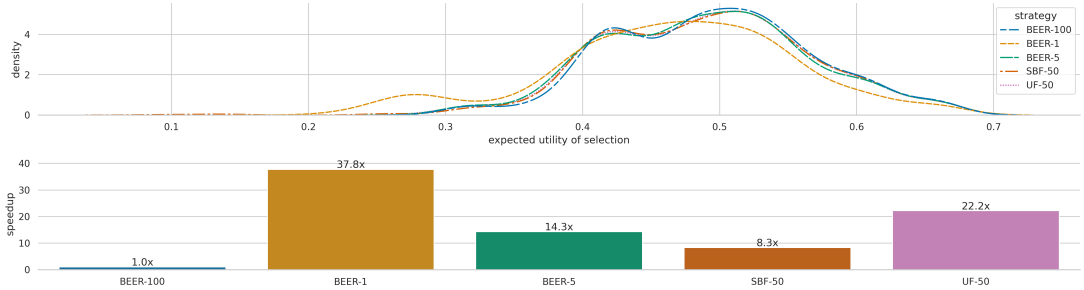


Figure 4.7: Comparison of proxy utilities on English to German: BEER using 1, 5 or 100 samples for MC estimation, and unigram F1 (UF) and skip-bigram F1 (SBF) each using 50 samples for MC estimation. We use each proxy utility to filter a top-20 from 100 ancestral samples. We show the resulting expected target utilties (BEER, an accurate estimate) (top), as well as a runtime comparison (bottom). Results are aggregated over 100 source sequences.

lead to the largest proportion of high expected utility translations across language pairs. Beam search has a noticeably high proportion of poor translations for English-Nepali, a low-resource language pair where mode-seeking search has been observed to be less reliable. Results in the opposite direction were similar. We explore both nucleus sampling and beam search for constructing $\bar{\mathcal{H}}(x)$ in the next experiment, as well as combining all three strategies together.

4.5.5 Coarse-to-Fine MBR

We now turn to the coarse-to-fine procedure (MBR_{C2F}) described in Section 4.3.

Choice of Proxy Utility

We compare various proxy utilities by their effectiveness as filtering strategies in obtaining high expected utility sets, where we again use accurate estimates of expected utility using 1,000 samples for MC estimation. We filter the top-20 hypotheses from an initial 100 hypotheses obtained using ancestral sampling. This ensures a high variety of expected utilities in the initial set. We also compare each proxy utility on their runtime performance. We compare both cheap estimates of expected BEER using either 1 or 5 samples for MC estimation (BEER-1 and BEER-5 respectively) as well as cheap-to-compute proxy metrics: unigram F1 using 50 samples for MC estimation (UF-50) and skip-bigram F1 using 50 samples for MC estimation (SBF-50).[6] We use expected BEER using 100 samples for MC estimation (BEER-100) as a reference point. See our results on the English-German system in Figure 4.7.

[6]Skip-bigrams are bigrams that do not enforce adjacency.

MBR	$\bar{\mathcal{H}}$	en-de		en-ro		en-ne	
		BEER	BLEU	BEER	BLEU	BEER	BLEU
NxS	N	64.3	38.0	54.9	21.4	38.9	3.6
C2F	N	+1.1	+1.9	+0.4	+0.2	+0.4	+0.2
	B	+0.9	+1.5	**+0.5**	**+0.5**	+0.5	**+0.5**
	all	**+1.3**	+2.4	**+0.5**	+0.4	**+0.6**	**+0.5**
BS	-	+0.9	**+2.8**	-0.1	+0.1	-0.8	+0.2

MBR	$\bar{\mathcal{H}}$	de-en		ro-en		ne-en	
		BEER	BLEU	BEER	BLEU	BEER	BLEU
NxS	N	64.8	38.7	58.5	28.0	43.1	6.3
C2F	N	+0.9	+1.1	+0.5	+0.7	+0.5	+0.2
	B	**+1.0**	**+1.5**	**+0.7**	**+1.2**	+0.5	**+0.9**
	all	**+1.0**	+1.4	+0.6	+1.1	**+0.8**	+0.8
BS	-	+0.5	+1.2	-0.0	+0.8	-1.0	+0.4

Table 4.2: Comparing $\text{MBR}_{\text{N-by-S}}$, MBR_{C2F} and beam search (BS) in terms of BEER and BLEU performance. We use BEER as utility, UF-50 as proxy utility, set top-$T = 50$ and use $L = 100$ samples for MC estimation. We use various strategies for constructing $\bar{\mathcal{H}}(x)$: 405 nucleus samples (N), the 405-best list from beam search (B) and combining both of these along with 1,005 ancestral samples (all). We use $S = 13$ in $\text{MBR}_{\text{N-by-S}}$ to mimic the computational cost of MBR_{C2F} at $N = 405$. The last row shows standard beam search performance using a typical beam size of 4 or 5 depending on the language. MBR results are averaged over 3 runs. Standard deviations for BEER/BLEU scores are below 0.1/0.2 (NxS), 0.1/0.1 (C2F) and 0 (BS).

We surprisingly find nearly all strategies to lead to equally good filtered sets as BEER-100 in terms of expected BEER of the filtered set. The only strategy that performs slightly worse than the others is BEER-1, which is likely too noisy to be a reliable filtering strategy. We observed very similar results for the other five language pairs. In terms of runtime performance we find BEER-1 to be fastest followed by UF-50 at a 22.2x performance increase over BEER-100.[7] In follow-up experiments, we will use UF-50 as a proxy utility, providing high quality filtered sets at good runtime performance.

Coarse-to-Fine MBR Results

In Table 4.2 we compare MBR_{C2F} with $\text{MBR}_{\text{N-by-S}}$ using $N = 405$ nucleus samples ($p = 0.7$) to construct the hypothesis space. We filter the top-$T = 50$ hypotheses using UF-50 as proxy utility and use $L = 100$ samples for MC estimation of the top-set, following our findings in Sections 4.5.5 and 4.5.3 respectively. For $\text{MBR}_{\text{N-by-S}}$ we set $S = 13$ to roughly match the amount of computation available to MBR_{C2F}, based on a 22.2x speed-up of UF-50 relative to BEER-100 observed in Figure 4.7. We find that across language pairs MBR_{C2F} consistently outperforms $\text{MBR}_{\text{N-by-S}}$ showing improvements between +0.4 and +1.1 BEER and +0.2 to +1.9 BLEU. MBR_{C2F} thus is effective at obtaining higher translation quality than $\text{MBR}_{\text{N-by-S}}$ at the same amount of computation available for MBR.

We also explore the effects on translation quality of changing and combining strategies for constructing $\bar{\mathcal{H}}(x)$. We find that using a beam of $N = 405$ (using the same length penalty as in Section 4.5.4) to construct $\bar{\mathcal{H}}(x)$ produces better results than nucleus sampling for most language pairs. Notably, re-ordering a large beam considerably improves over standard beam search decoding (using the usual beam size of 5 (ro, ne) or 4 (de)) for all language pairs in terms of BEER and for most language pairs in terms of BLEU scores. Combining all strategies for creating hypothesis spaces: ancestral sampling, nucleus sampling and beam search leads to the best results overall. For all language pairs both BEER and BLEU scores either improve or remain similar. This is more empricial evidence that expected utility is a robust and reliable criterion for picking translations: enlarging the hypothesis space or improving MC estimation under reasonable choices of hyperparameters seemingly never unreasonably hurts translation quality, but generally improves it.

A Multi-Reference Test Set We also test three systems from Table 4.2 (NxS, C2F and beam search) on a multi-reference test set. We use the English to German systems trained on WMT18 news data and translate `newstest2021`, which has three separate translations for each source sentence (we use translators A, C and D). We show results in Table 4.3. We find a similar pattern to that of Table 4.2. MBR_{C2F} greatly outperforms $\text{MBR}_{\text{N-by-S}}$ given the same amount of available compute (see Section 4.5.5) for details). MBR_{C2F} outperforms beam search results in terms of BEER, but is much closer to beam search this time in terms of BLEU.

4.5.6 Runtime

We measure runtime performance on hypothesis generation, sampling for MC estimation of expected utilities and decoding time seperately for various algo-

[7] Our Python implementations of unigram and skip-bigram F1 are not optimized and we deem it likely that a greater speed-up is possible with a more efficient implementation.

newstest2021	BEER	BLEU
NxS	63.4	40.9
C2F	**64.5**	42.8
BS	63.7	**43.0**

Table 4.3: English to German $\text{MBR}_{\text{N-by-S}}$ and MBR_{C2F} results on the newstest2021 multi-reference test set. We use $N = 405$ nucleus samples as hypothesis space and use the same hyperparameters as in Table 4.2.

MBR	hyp. generation	sampling	decoding
NxN	6,241s	7,739s	23,156s
NxS	6,241s	383s	746s
$\text{NxS}_{\text{large}}$	6,241s	1,825s	5,358s
C2F	6,241s	1,825s	726s
BS	-	-	194s

Table 4.4: A runtime comparison of MBR variants and beam search. We separate the time taken for *i)* hypothesis generation *ii)* sampling (for estimation of expected utility) and *iii)* running the decoder itself. We use $N = 405$ nucleus samples, $S = 13$ and $\text{S}_{\text{large}} = 100$ ancestral samples for NxS variants, and the hyperparameter settings for C2F as used in Table 4.2.

rithms explored in this work on the English to German language pair. We run all experiments on an Intel Xeon Bronze 3104 Processor and a single NVIDIA GeForce 1080Ti GPU. For generating samples and beam search outputs we set the batch size to as large as possible, constrained by the available GPU memory. MBR using BEER as utility runs on CPU, while sampling and beam search run on GPU. We mimic the $\text{MBR}_{\text{N-by-N}}$ and MBR_{C2F} setups from Table 4.2 using a hypothesis space of 405 nucleus samples. We also additionally include runtime results for $\text{MBR}_{\text{N-by-N}}$ with $N = 405$ and a more expensive $\text{MBR}_{\text{N-by-S}}$ variant with $S = 100$ ($\text{NxS}_{\text{large}}$). For beam search we report results for a beam size of 4, as has been used throughout the chapter for this language pair. Results are shown in Table 4.4. As can be seen, collecting hypotheses and unbiased sampling makes up for a large part of the total decoding time in MBR algorithms. We do note that sampling operations are easily parallelisable and can be split across multiple GPUs when available. In terms of the decoding time itself, we can see that we greatly reduced the amount of computation needed to perform MBR going from 23,156 seconds of decoding time for $\text{MBR}_{\text{N-by-N}}$ to only 726 seconds of decoding time for MBR_{C2F}. This can be attributed to the great reduction in number of utility calls in our proposed approximations.

4.6 Related Work

4.6.1 MBR Decoding in NMT

In recent NMT literature MBR has started being explored either in combination with MAP decoding or replacing it altogether. Stahlberg et al. (2017) adapt

lattice minimum Bayes risk decoding (Tromble et al., 2008) on SMT translation lattices to be incorporated in left-to-right beam search decoding in NMT, thereby proposing a hybrid decoding scheme. They adapt lattice MBR to work on partial hypotheses and perform beam search to find translations that are both high probability under the NMT model and have high expected utility under the SMT model. Shu and Nakayama (2017) also combine beam search with MBR decoding to find low risk hypotheses, after which they re-rank all hypotheses with MBR again. They report having to restrict the number of hypotheses as not to degrade the effectiveness of MBR re-ranking, a finding that is likely due to biased estimation of expected utility, as in our work we find that increasing the number of hypotheses always improves translation quality. Blain et al. (2017) explore the quality of k-best lists obtained from beam search in NMT models and find that while MAP is not a good criterion for ranking the resulting hypotheses, re-ranking using MBR with BEER as a utility leads to improvements on top of standard beam search decoding (with a small beam size), in terms of both BLEU scores as well as human evaluation scores. Borgeaud and Emerson (2020) approach decoding from a voting theory perspective and derive a decoding strategy similar to MBR. They explore a range of utility functions, achieving similar BLEU scores to beam search, but showing improvements in terms of length, diversity and human judgement.

All of the above works make use of beam search to provide both the hypothesis space as well as to make a biased estimate of expected utility. Eikema and Aziz (2020), the basis of the previous chapter, is the first work in NMT that propose to use sampling from the model to both make unbiased estimates of expected utility, the importance of which we confirm here again in experiments, and to form the hypothesis space. In the previous chapter we only explored $\text{MBR}_{\text{N-by-N}}$, however, and never explored hypothesis spaces larger than $N = 30$ samples. In this chapter, we show that it is beneficial to scale MBR to much larger hypothesis spaces and that it can be beneficial to construct them using mode-seeking strategies.

4.6.2 Approximations to MBR

Most instances of MBR decoding in machine translation, from the original work of Kumar and Byrne (2004) to recent instances in NMT (Stahlberg et al., 2017; Shu and Nakayama, 2017; Blain et al., 2017), approximate the objective function by computing expectations not w.r.t. the model distribution, but rather, w.r.t. a proxy distribution. This proxy is obtained by enumeration via beam-search of a subset of the sample space (*e.g.*, a k-best list), and renormalisation of the probabilities of the outcomes in this subset. This has the undesirable effect of exaggerating differences in probability due to underestimation of the normalisation constant, and, like MAP decoding, it over-represents pathologies around the mode. Similarly, most prior work uses mode-seeking search to explore a tractable subset of the hypothesis space. Mode-seeking approximations bias the decoder

towards the mode making MBR decoding less robust to idiosyncratic outcomes in the hypothesis space (Eikema and Aziz, 2020). This is in stark contrast with our work, where we sample from the model to construct unbiased estimates of expected utility, as well as to enumerate a tractable hypothesis space.

There are cases in statistical machine translation (SMT) where the computation of expected utility can be factorised along a tractable directed acyclic graph (DAG) via dynamic programming (Tromble et al., 2008; Zhang and Gildea, 2008; DeNero et al., 2009; Kumar et al., 2009). In such cases, the DAG contains a much larger subset of the sample space than any practical k-best list, still some pruning is necessary to construct a compact DAG containing only the most probable outcomes. These strategies are only available for models and utility functions that make strong Markov assumptions. For example, Tromble et al. (2008) and DeNero et al. (2009) develop linearisation strategies for BLEU, and Zhang and Gildea (2008) maximise expected trigram counts as a proxy to BLEU proper. The idea of utilising a proxy utility is something we also explore in this chapter, though only as an intermediate step to decoding with the target utility.

In some (rarer) cases, unbiased (or asymptotically unbiased) samples have been used to approximate the MBR objective and/or to reduce the search space. For example, Stanojević and Sima'an (2015) use ancestral sampling in MBR decoding for permutation-trees-based reordering models, and Arun et al. (2009) use Gibbs sampling for MBR decoding in phrase-based MT. Unbiased samples for estimation of expected utility or exploration of a tractable hypothesis space are simply not common in machine translation. In SMT, the reason is a technical one, most SMT models are not based on a left-to-right factorisation of the joint distribution, thus unbiased sampling requires MCMC (DeNero et al., 2008; Blunsom et al., 2009) or expensive adaptive rejection sampling (Aziz et al., 2013). This limitation does not extend to NMT models, but NMT most likely simply inherited from SMT the practice of using beam-search-based approximations, at least until we proposed the use of Monte Carlo estimation in Eikema and Aziz (2020).

4.6.3 Tackling the Inadequacy of the Mode

In the previous chapter, we linked the inadequacy of the mode in NMT to the entropy of the conditional distribution, or, more precisely, to the fact that NMT models tend to spread probability mass over large subsets of the sample space (Ott et al., 2018; Eikema and Aziz, 2020). It is plausible that strategies to concentrate probability mass (*e.g.*, reducing entropy or pruning the support of the model) will do so by making inadequate translations less probable. For example, Forster et al. (2021) find that the inadequacy of the mode problem does not seem to affect sequence-to-sequence models of morphological inflection, an essentially deterministic task, whose combinatorial space is built upon a smaller vocabulary (*i.e.*, characters instead of sub-word units), and whose observations are typically

very short (*i.e.*, words rather than sentences). Peters and Martins (2021) train sparse sequence-to-sequence models (Peters et al., 2019) which assign zero probability to many outcomes dramatically reducing the support of the conditional distribution over complete sequences. They show that sparsity leads to inadequate candidates such as the empty string being pruned out of the support. They also find that label smoothing increases the rate at which the empty string is more probable than the beam-search output.

Meister et al. (2020) interprets the algorithmic approximations of beam search as an inductive bias towards outputs with uniform information density (Jaeger and Levy, 2007). They develop variants of beam search where this preference is a tunable hyperparameter and show that deviating from the mode with this type of bias can lead to improved translation quality. Another way to deviate from the mode is to augment the decoding objective with an auxiliary model. Li and Jurafsky (2016) re-rank a k-best list using a combination of two model probabilities, namely, $p_{Y|X}(h|x, \theta_{\text{fwd}})$ and $p_{X|Y}(x|h, \theta_{\text{bwd}})$. They think of this as maximising the mutual information (MI) between source and translation. The motivation is that the target-to-source component will push against inadequate candidates, as those are unlikely to be mapped back to the source with high probability. Bhattacharyya et al. (2021) find that 100 samples from an NMT model contain better candidates (measured in terms of BLEU) than the output of beam search (an observation we previously also made based on 30 samples and METEOR, instead). They propose to rerank these samples using an energy-based model trained to order candidates as sentence-BLEU would. Like these works, sampling-based MBR decoding, can be seen as a form of *explore and rank* approach, however, the ranking function in MBR is derived from the NMT model itself, whereas both MI- and EBM-based re-ranking involve an auxiliary trained model. For the EBM, in particular, in the limit of a too large hypothesis space, the beliefs of the NMT model are completely overwritten by the EBM. MBR, instead, does not overwrite the model's beliefs, it re-expresses those beliefs in terms of utility.

Leblond et al. (2021) recast NMT as a reinforcement learning problem and learn both a policy (*i.e.*, a mechanism to explore the space of translations one word at a time from left-to-right) and a value function (*i.e.*, an estimate at the expected reward of finishing a given prefix translation). For reward they investigate what they call privileged metrics, which require access to references (*e.g.*, sentence-level BLEU), and unprivileged metrics, which do not use references but access the source (*e.g.*, a quality estimation score). Compared to sampling-based MBR, their work tightly integrates search and value estimation, thus going beyond ranking a fixed set of candidates. The objective function of MBR can be thought of as an 'unprivileged metric' in their terminology, one that is based on the NMT model itself (and a choice of utility). But, the policy in sampling-based MBR (*i.e.*, the NMT model) is not trained to be aware of the evaluation metric.

4.7 Consequent Work

Since the publication of Eikema and Aziz (2020) and Eikema and Aziz (2022) quite a number of works have explored the use of sampling-based minimum Bayes risk decoding across text generation tasks, oftentimes showing impressive results.

Müller and Sennrich (2021) study the properties of $\text{MBR}_{\text{N-by-N}}$ and explore hypothesis spaces up to a size of $N = 100$ as well as multiple utility functions. They find that MBR decoding outputs exhibit a similar but smaller bias towards short translations and frequent tokens compared to beam search, but do observe that this is dependent on the choice of utility function. They further find that MBR decoding mitigates spurious copying and hallucinations under domain shift. Similar to our results here, they find that MBR decoding scales well with larger hypothesis spaces and better estimation of expected utility.

Freitag et al. (2022) explore the use of large hypothesis spaces and a range of utilities, including neural utilities, on the $\text{MBR}_{\text{N-by-N}}$ approximation. They find that using BLEURT as utility leads to significantly better translations in a human evaluation, while producing considerably lower probability translations than beam search. Interestingly, BLEU favors beam search decoding over MBR decoding, not agreeing with human judgements.

Freitag et al. (2023) explore different candidate generation strategies and find a mode-seeking sampling strategy called epsilon-sampling (Hewitt et al., 2022) to perform best in a human evaluation. They also confirm once again that MBR decoding with BLEURT (Sellam et al., 2020) as utility can outperform beam search in a human evaluation even though BLEU prefers beam search. They also find that adding a temperature to the sampling procedure can improve performance for small candidate sets. Whereas for large candidate sets they find it's better to not use a temperature, confirming the robustness of unbiased estimates of expected utility for ranking a diverse candidate set. Yan et al. (2024) also explore the use of adding a temperature to the ancestral sampling procedure, but with different motivations. Instead, they intend to address the failure of MBR decoding on models trained with label smoothing. Similarly to what we found when trying to implement MBR for the first time, they show that even though label smoothing only has a minor effect on token-level distributions, it has a large (negative) impact on the sequence-level distribution. They aim to undo this post-training by adding a temperature to the sampling procedure. They show that doing this they can achieve constant performance of MBR decoding across models trained with various degrees of label smoothing, outperforming beam search in terms of automatic evaluation metrics across all variants.

With the advent of using large language models for virtually all language tasks, sampling-based MBR has also proven to be a promising decoding algorithm showing strong results. These models are oftentimes trained without label smoothing and sampling-based methods have been used over beam search for a long time (though still using mode-biased sampling methods like nucleus sampling and top-k

sampling). Suzgun et al. (2023) show competitive performance using sampling-based MBR with `code-davinci-002` (Chen et al., 2021) and using BERTScore (Zhang et al., 2020b) as utility function on summarization, data-to-text generation, translation, textual style transfer, and image captioning. They also find state-of-the-art performance on the WMT16 translation task (Bojar et al., 2016a) and the WebNLG (Castro Ferreira et al., 2020) data-to-text generation dataset. Garcia et al. (2023) also apply sampling-based MBR for few-shot translation using LLMs and convincingly show better translation performance over beam search. They explore the English to German, Chinese and Icelandic language pairs and find that their few-shot translation models outperform commercial systems and performs on par with recent WMT submissions. Johnson et al. (2023) employ sampling-based MBR in an interesting way. Instead of using it as an algorithm for generation, they employ clever utility design and optimize for expected utility to find an optimal sequence of uncertainty annotations of an existing generation. Doing this they are able to annotate generated code from a code generation model with uncertain subregions of the generation that a user would have to more closely look at, or similarly, stop a generation when continuations become too uncertain.

Recently, a number of prominent works have appeared that have focused on making sampling-based MBR decoding more efficient at test-time. Cheng and Vlachos (2023) use confidence intervals of expected utility estimates to dynamically prune unpromising hypotheses. Vamvas and Sennrich (2024) aggregate statistics of groups of sampled pseudo-references in the MBR algorithm and only compare hypotheses against the aggregate statistics. Jinnai and Ariu (2024) run multiple coarse-to-fine steps with increasingly accurate estimates of the expected utility (using the target utility). Finkelstein and Freitag (2024) distil performance improvements of MBR decoding by fine-tuning on MBR translations of monolingual source-side data. Yang et al. (2024) also distil MBR performance improvements using model fine-tuning, but instead use direct preference optimization (DPO; Rafailov et al., 2023) to distil MBR ranking preferences in the model. Wu et al. (2024) show success in applying MBR to instruction following tasks using an expensive LLM-as-a-judge model as utility function (Prometheus; Kim et al., 2024b), and apply the DPO fine-tuning approach to successfully distil performance improvements without additional test-time compute burden.

4.8 Conclusion

We have shown MBR to be a robust decision rule for NMT that can find high quality translations. In particular, we have found that MBR, under reasonable hyperparameter choices, generally leads to improved translation quality with more computation (*i.e.*, searching a larger search space and/or using more samples for more accurate MC estimation). Big challenges in decoding with MBR are constructing the hypothesis space and keeping computational cost of estimating

expected utility tractable. We have proposed effective strategies for both, by exploring more efficient ways of forming the hypothesis space and proposing an approximation to MBR that is linear in the size of this hypothesis space. Our coarse-to-fine MBR procedure is able to considerably reduce the number of calls to the utility function without compromising translation quality. We have shown that sampling-based MBR in general can outperform beam search on all the language pairs we explored and can continue to improve with better and more accurate search. We believe sampling-based MBR to be a promising, albeit still more expensive, alternative to beam search decoding. Unlike beam search, where it is not obvious how to further improve translation quality, sampling-based MBR is likely to benefit from improvements to different aspects of the algorithm. Works from other labs have demonstrated the effectiveness of sampling-based minimum Bayes risk decoding across a wide variety of text generation tasks. Still, computational efficiency is the main bottleneck in using MBR decoding in production environments. Therefore, we believe fruitful avenues of research to be among *i)* clever algorithms for constructing hypothesis spaces, *ii)* more robust estimates of expected utility using fewer samples, *iii)* clever utility design and *iv)* improving the modelling capacity of NMT systems. We hope that these results motivate researchers and practitioners to make more conscious considerations of the choice of decision rule and that it paves the way for use of tractable sampling-based MBR decoding in NMT.[8]

[8]An implementation of sampling-based MBR decoding and the approximation strategies proposed in this chapter is available at `github.com/roxot/mbr-nmt`.

Chapter 5

Quasi-Rejection Sampling

In this chapter, we move away from machine translation and autoregressive text generation altogether. Instead of considering the easy-to-sample-from autoregressive factorisation of typical neural text generation models, we turn our attention to energy-based models (EBMs). Energy-based models are a class of models that assign arbitrary non-negative scores to outcomes within its support, in this chapter being sequences of natural language tokens. As these scores do not necessarily need to be normalised, this allows for a very flexible definition of models. Generation from such models, however, is not always trivial as we cannot exploit an autoregressive factorisation to generate one token at a time. In this chapter we have a look at energy-based models for controlled text generation, allowing for placing distribution-level constraints on the generated text, such as generating biographies that are not biased towards generating male biographies. To sample from these models, we propose an algorithm that we coin quasi-rejection sampling (QRS), which can produce approximate samples from arbitrary energy-based models while being able to monitor how good the approximation to the target distribution (the distribution implied by the EBM) is, and also allowing for trading off the approximation quality with sampling efficiency. While the application of this chapter is controlled text generation, the main contribution is rather the sampling technique (QRS) itself. Hence, this will be the focus of this chapter. A technical background on sampling techniques is presented in Section 2.2. This work is based on Eikema et al. (2022), published in the Transactions on Machine Learning Research (TMLR) journal.

Chapter Highlights

Problem Statement

- Energy-based models (EBMs) permit flexible specifications of probability distributions. For example, we can use EBMs to define constraints on the outcome distributions of neural text generation models. Generation from

such models, however, is non-trivial. Typical generation techniques such as beam search and ancestral sampling (an essential component of sampling-based MBR decoding) are not easily applicable as they rely on the autoregressive factorisation of neural text generation models.

- Approximate samples from such models can be obtained using Markov chain Monte Carlo (MCMC) techniques. However, those are not trivial to apply either, as most instantiations rely on local (*i.e.* conditional) proposal distributions that approximate the target distribution locally, for which the options in discrete outcome spaces are limited.[1]

- MCMC techniques also typically do not provide quality estimates of the approximate samples, *i.e.* how close the realised sampling distribution is from the target distribution implied by the EBM.

Contributions

- We propose quasi-rejection sampling (QRS), a relaxation of rejection sampling that produces approximate samples from an energy-based model provided that a reasonable easy-to-sample-from proposal distribution is available, *e.g.* an autoregressive model approximating the energy-based model.

- We focus on the ease with which we can obtain global (*i.e.* unconditional) proposal distributions for text generation. We show how one can construct them by making use of recent advances in large language models: prompting, training objectives for approximating EBMs, and the widespread availability of pre-trained neural text generation models.

- We show that unlike for many other approximate sampling algorithms, we can estimate the quality of the approximate samples (*i.e.* how close the sampling distribution is to the energy-based model) in the form of f-divergences.

- We show that QRS has desirable properties for an approximate sampler: providing guarantees on the sampling quality in the form of upper-bounds on the total variation distance and showing that f-divergence is a monotonic function of QRS's tuning parameter.

- We demonstrate the effectiveness of QRS in controlled text generation, sampling from energy-based models dictating distribution-level constraints. We show that we can trade-off sampling efficiency and quality arbitrarily by changing QRS's core parameter.

[1] It is difficult to define local neighbourhoods in discrete spaces such as those over natural language sentences, but examples do exist such as using delete, insert and replace operations of single tokens to define a local neighbourhood around an existing natural language sequence.

- We compare QRS to Metropolis-Hastings methods (a popular instantiation of MCMC) and show that QRS outperforms local variants and performs on par with global variants on proxy metrics. We also exploit the data-processing inequality to provide a lower bound on the divergence of the Metropolis-Hastings samplers and compare it to the precise divergence estimate for QRS.

5.1 Introduction

Generating samples from a probabilistic model is a fundamental part of many machine learning tasks, as we have already seen in sampling-based MBR decoding. Sometimes, the relation between the probabilistic model and the associated generative process is direct: for instance, as we've seen in language modelling, an autoregressive model can both generate a sequence by ancestral sampling and compute its probability. However, imposing preferences on such models is not trivial. We may, for example, wish to impose a preference upon our models to produce generations with particular properties, such as generated biographies of a biography-generation language model to be about scientists. A family of models with much greater representational freedom is the family of *energy-based models* (EBMs; LeCun et al., 2006). Such models map elements x of the sample space to real-valued "energies" $E(x)$, or, equivalently, to non-negative scores $\tilde{P}(x) = \exp(-E(x))$ which can be seen as an unnormalised probability distribution, where $P(x) \propto \tilde{P}(x)$. However, EBMs can be difficult to sample from as we do not have a clear generative process to follow.

In this chapter, we address the problem of sampling from such EBMs, with a particular focus on discrete spaces of sequences over a finite vocabulary, and study applications to text generation. A popular approach to sampling from complex, unnormalised, probability distributions, such as EBMs, consists in applying Markov chain Monte Carlo (MCMC) techniques, which are guaranteed to converge to the target distribution in the limit (of infinitely long chains), under mild regularity conditions (Robert and Casella, 2004), also see Section 2.2.5 of this dissertation. In practice, however, the length of the Markov chain is finite, in which case often only approximate samples are obtained. In order to know whether the samples are representative of the target distribution, ideally, one should quantify the divergence of the MCMC sampling distribution from the target distribution $P(x)$ in terms of well-established metrics (e.g. an f-divergence such as the total variation distance or KL divergence). Unfortunately, evaluating convergence is often challenging (Cowles and Carlin, 1996; Roy, 2020), especially if one makes no assumptions about the sample space. For instance, popular convergence assessments such as effective sample size (ESS; Gamerman and Lopes, 2006) or $\hat{R}$ (Gelman and Rubin, 1992; Vehtari et al., 2021) require Euclidean structure

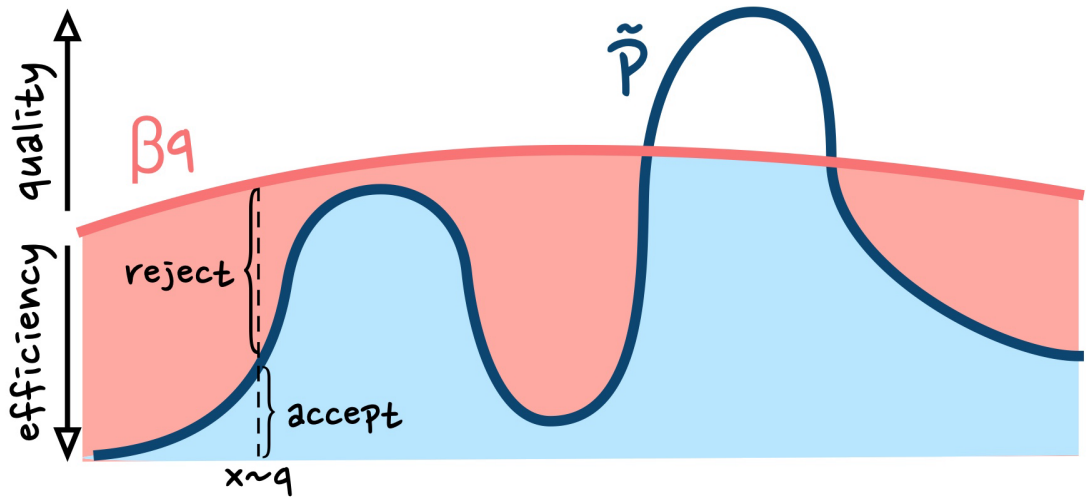


Figure 5.1: Quasi-rejection sampling **(QRS)** approximates a target distribution p as defined by unnormalised $\tilde{p}$ (continuous in this sketch) with a truncated distribution p_β (the blue shaded area). This QRS distribution is defined in terms of a global proposal distribution q and a scalar parameter β that controls the quality of the approximation.

for computing variances and correlations of *real-valued* variables (or vectors in multivariate generalizations; Vats et al., 2019; Brooks and Gelman, 1998), which are not intrinsically defined on *discrete* spaces only endowed with a probabilistic structure. For this reason, prior work on sampling from discrete sequence spaces often relies on proxy metrics (such as perplexity, diversity metrics and constraint satisfaction). Such proxies can be insufficient to assess how representative the samples are of the target distribution, and therefore can be misleading, as we will see in Section 5.3.4.

In this chapter, we introduce a simple approximate sampling technique, quasi-rejection sampling (QRS), which provides explicit estimates of the divergence from the target distribution for the general class of f-divergences, which includes the total variation distance (TVD), forward and reverse KL, Jensen-Shannon, and χ^2-divergence (also see Section 2.2.4). This is possible because QRS associates explicit probability scores with samples: a property generally lacking in MCMC samplers. One may use such divergence estimates to tune the sampler, controlling the trade-off between efficiency (acceptance rate) and quality (divergence).

QRS is a relaxation of rejection sampling that obtains approximate samples from a target distribution (see Figure 5.1). This requires access to a global (or unconditional) proposal distribution that, ideally, produces samples close to the target distribution. Traditionally, MCMC methods work with *local* proposals (conditional distributions defined around a given point, for instance a distribution defined by performing a set of local edits), since high-quality global proposals have been hard to construct. Fortunately, with the advent of powerful neural network

training techniques, this situation is rapidly changing and obtaining powerful global proposals is now possible as we show in our experiments for NLP tasks.

We demonstrate the effectiveness of QRS on controlled text generation. Large pre-trained language models are becoming increasingly useful general purpose tools and generation is typically accomplished by sampling, albeit biased towards the mode (Nadeem et al., 2021). Controlling the distribution of these language models to accommodate human preferences can be difficult, but EBMs have shown to be a promising way to achieve this (Khalifa et al., 2021). However, sampling from EBMs defined over a discrete sequence space is non-trivial, making it a challenging task to benchmark QRS. In this chapter, we experiment with EBMs resulting from restricting a GPT-2 language model (Radford et al., 2019) in some way: either the model is restricted to only generate sequences containing a specific term; or generations from model are restricted to meet particular conditions at a specific rate, for example debiasing a distribution over biographies to consist of 50% female biographies. We explore a variety of ways to construct proposal distributions for QRS. In particular, we explore prompting a pre-trained language model, as well as training an autoregressive model to approximate the EBM (Khalifa et al., 2021).[2] We also experiment with a paraphrase generation task in which we use off-the-shelf machine translation models as proposal distributions. Results show that we are able to approximate the target distributions to any desired level in exchange for sampling efficiency. Finally, we include experimental comparisons with both local (random-walk) and global MCMC methods, showing that QRS performs comparably or better across several dimensions, while providing stronger guarantees.

5.2 Formal Approach

We consider a discrete (i.e. countable) sample space $\mathcal{X}$. We are given a nonnegative real function — such as an EBM — $\tilde{P}(x)$ over $\mathcal{X}$, such that the partition function $Z \doteq \sum_{x \in \mathcal{X}} \tilde{P}(x)$ is strictly positive and finite. We can then associate with P a normalised probability distribution $P(x) \doteq \tilde{P}(x)/Z$. Our goal is to define a "sampler" ω, that is a generator of elements from $\mathcal{X}$, such that ω produces a sample x with a probability $\omega(x)$ as close as possible to our target $P(x)$, in terms of distance measures such as KL divergence $\mathrm{KL}(P, \omega)$ and total variation distance $\mathrm{TVD}(P, \omega)$, and more generally the large family of f-divergences. To help us achieve this goal, we assume that we have at our disposal a *global proposal distribution* $Q(x)$ such that *i)* we can effectively compute $Q(x)$ (i.e. *score* x) for any $x \in \mathcal{X}$, *ii)* we can efficiently generate samples from Q, and *iii)* the support of Q includes the support of P, i.e. $P(x) > 0 \;\Rightarrow\; Q(x) > 0$.

[2] Khalifa et al. (2021) are able to train an autoregressive model to approximate EBMs of the sort that we see in this chapter reasonably well, but not perfectly. Hence, sampling algorithms can help to improve the approximation even further.

5.2.1 Quasi-Rejection Sampling (QRS)

Algorithm 1 QRS

1: **Require:** (possibly unnormalised) target $\tilde{P}(x)$, proposal Q, parameter β, number of required samples N $\{0 < \beta < \infty\}$
2: $n \leftarrow 0$
3: **while** $n < N$ **do**
4: $x \sim Q$
5: $r_x \leftarrow \min\left(1, \tilde{P}(x)/(\beta Q(x))\right)$ {Acceptance prob.}
6: $u \sim U_{[0,1]}$ $\{U_{[0,1]}$: unif. dist. over $[0,1]\}$
7: **if** $u \leq r_x$ **then**
8: output x
9: $n \leftarrow n + 1$
10: **end if**
11: **end while**

We propose quasi-rejection sampling (QRS), as shown in Algorithm 1. In addition to $\tilde{P}(x)$ and Q, QRS requires the input of a finite positive number β. For a given β, the QRS sampler, which we will denote by P_β, produces an independent and identically distributed (i.i.d.) values x (line 8), with a probability mass function that we denote by $P_\beta(x)$. If β is a global upper bound on the ratio $\tilde{P}(x)/Q(x)$, then the behaviour of the QRS algorithm is identical of that of the classical rejection sampling (RS) algorithm (von Neumann, 1963). However, QRS does not require β to be an upper bound, and the acceptance probability r_x in line 5 is an extension of that used in RS to situations where $\tilde{P}(x) > \beta q(x)$. In such situations, the sample x is always accepted at line 7.

QRS has crucial practical advantages over rejection sampling. It is well known that for rejection sampling, with β a finite global upper bound, we have $P_\beta = P$: in other words, rejection sampling is a perfect sampler for P (Robert and Casella, 2004). This is of course a major advantage, however it comes with serious theoretical and practical limitations: there may not exist such a finite upper bound, and even if one exists, its value may not be known. Furthermore, even if such a bound could be found, the resulting sampler could be extremely inefficient: the "acceptance rate" of rejection sampling is proportional to $1/\beta$, which can be very small. By relaxing the requirement that β is a global upper bound, QRS sacrifices the identity between P_β and P. However, QRS becomes much more broadly applicable, and crucially, allows an explicit trade-off between the sampling efficiency of P_β and its approximation quality.

5.2.2 Explicit f-divergence Diagnostics for QRS

Let $\tilde{P}_\beta(x) \doteq \min(\tilde{P}(x), \beta Q(x))$, and let $Z_\beta \doteq \sum_{x\in\mathcal{X}} \tilde{P}_\beta(x)$ be the associated partition function. Also, define the *acceptance rate* AR_β of the QRS sampler P_β as the proportion of samples from Q, in line 4 of the algorithm, that are accepted on line 7, a proportion that provides a measure of the efficiency of the algorithm. We then have the following properties, proven in Eikema et al. (2022):

$$P_\beta(x) = \frac{\min\left(\tilde{P}(x), \beta Q(x)\right)}{Z_\beta} = \frac{\tilde{P}_\beta(x)}{Z_\beta,} \tag{5.1}$$

$$\mathrm{AR}_\beta = \mathbb{E}_{x\sim Q}\left[\min\left(1, \frac{\tilde{P}(x)}{\beta Q(x)}\right)\right] = \frac{Z_\beta}{\beta.} \tag{5.2}$$

Eq. 5.1 provides an explicit form for P_β as the normalised distribution associated with P_β, while Eq. 5.2 shows that the acceptance rate is a nonincreasing function of parameter β. Thus, similarly to rejection sampling, QRS has an acceptance rate proportional to $\frac{1}{\beta}$. The explicit form of $P_\beta(x)$ given in Eq. 5.1 enables us to directly compute empirical estimates of the f-divergence of the target from P_β:

$$D_f(P, P_\beta) = \mathbb{E}_{x\sim P_\beta}\left[f\left(\frac{P(x)}{P_\beta(x)}\right)\right] \tag{5.3}$$

Crucially, to estimate this quantity given a collection of samples from P_β we need to compute (or at least approximate) the two values $\tilde{P}(x)$ and $P_\beta(x)$ for any given x. And this is something that we *can* do with QRS thanks to the explicit form of P_β of Eq. 5.1 and given that we can estimate the partition functions Z and Z_β — see Eqs. 5.4 and 5.6 below.

The contrast here with a typical Markov chain based sampler ω is striking: it is usually unfeasible to estimate the probability $\omega(x)$ for a given x (even one sampled from ω): to do so, one might estimate the chain's transition matrix or kernel, and repeatedly multiply by it, but this matrix is usually huge, or even infinite. In other words, unlike QRS, these samplers are not "scorers", making it impractical to estimate f-divergences $D_f(p, \omega)$ as in Eq. 5.3.

5.2.3 Divergence Estimates via Importance Sampling

In order to compute the quantities mentioned above, we need to estimate the partition functions of P and P_β. Also, computing the f-divergence from Eq. 5.3 would require a separate estimate for each value of β. Using importance sampling (IS, see Section 2.2.3 for a short primer), we can use a single collection $\{x_1, \ldots, x_N\}$ of i.i.d. draws from Q (rather than from P_β) and make importance

sampling estimates:

$$Z \approx \frac{1}{N}\sum_{i=1}^{N}\frac{\tilde{P}(x_i)}{Q(x_i)}, \qquad Z_\beta \approx \frac{1}{N}\sum_{i=1}^{N}\frac{P_\beta(x_i)}{Q(x_i)} \tag{5.4}$$

$$\mathrm{AR}_\beta \approx \frac{1}{N}\sum_{i=1}^{N}\min\left(1, \frac{\tilde{P}(x_i)}{\beta Q(x_i)}\right) \tag{5.5}$$

$$D_f(P, P_\beta) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{P_\beta(x_i)}{Z_\beta Q(x_i)}\, f\left(\frac{Z_\beta\ \tilde{P}(x_i)}{Z\ P_\beta(x_i)}\right) \tag{5.6}$$

In Eq. 5.6, we exploit the fact that $P_\beta(x) = \min(\tilde{P}(x), \beta Q(x))$ is known explicitly.

In this work, from the f-divergences we will only use the KL divergence and total variation distance (TVD). Again, using importance sampling, we can estimate these as:

$$\mathrm{TVD}(P, P_\beta) \approx \frac{1}{2N}\sum_{i=1}^{N}\left|\frac{P_\beta(x_i)}{Z_\beta\ Q(x_i)} - \frac{\tilde{P}(x_i)}{ZQ(x_i)}\right| \tag{5.7}$$

$$\mathrm{KL}(P, P_\beta) \approx \log\frac{Z_\beta}{Z} + \frac{1}{N}\sum_{i=1}^{N}\frac{\tilde{P}(x_i)}{ZQ(x_i)}\log\frac{\tilde{P}(x_i)}{P_\beta(x_i)}. \tag{5.8}$$

5.2.4 Partition Function Estimates

In Eq 5.4 we estimate the partition functions Z and Z_β. As these estimates are required for further estimates of acceptance rate and f-divergences, it is crucial that we estimate these with reasonable accuracy. As the sample mean is an unbiased estimator of the mean these estimates are unbiased, meaning these estimates converge (almost surely) to Z and Z_β for $N \to \infty$, a consequence of the strong law of large numbers (Tao, 2008), whether or not the random variables $\frac{\tilde{P}(x)}{Q(x)}$ and $\frac{P_\beta(x)}{Q(x)}$ have finite variances. However, in order to provide guarantees about the accuracy of these estimates, *e.g.* in terms of confidence bounds, one would need to estimate these variances. A practical approach consists in providing empirical variance estimates based on the same N samples, and this is what we will do in several experiments, comforting us about the practical accuracy of our estimates of Z and Z_β. However, in theory, the empirical variance estimates could themselves be wrong, resulting in an estimation circularity. The only way to avoid this circularity, that we are aware of, consists in cases where the RVs $\frac{P(x)}{q(x)}$ and $\frac{P_\beta(x)}{q(x)}$ can be bounded *a priori*. In these cases the variances of these RVs can also be formally bounded using Popoviciu's inequality (Popoviciu, 1935), which states that for m and M a lower- and upper-bound on the values of a random variable X, its variance is bounded as:

$$\text{Var}\left[X\right] \leq \frac{(M-m)^2}{4} \tag{5.9}$$

Interestingly, the random variable $\frac{P_\beta(x)}{Q(x)} = \min(\tilde{P}(x)/Q(x), \beta) \leq \beta$ appearing in Z_β is bounded *by construction*[3], so that one can always provide formal guarantees about the Z_β estimate:

$$\text{Var}\left[\frac{P_\beta(x)}{Q(x)}\right] \leq \frac{\beta^2}{4} \tag{5.10}$$

and thus, the IS estimator of Z_β has variance

$$\text{Var}\left[\frac{1}{N}\sum_{i=0}^{N}\frac{P_\beta(x_i)}{Q(x_i)}\right] = \frac{1}{N^2}\sum_{i=0}^{N}\text{Var}\left[\frac{P_\beta(x_i)}{Q(x_i)}\right] \tag{5.11}$$

$$= \frac{1}{N^2}\cdot N \cdot \text{Var}\left[\frac{P_\beta(x)}{Q(x)}\right] \tag{5.12}$$

$$\leq \frac{\beta^2}{4N} \tag{5.13}$$

By contrast, in the case of Z, while one may find proposals Q for which a bound on $\frac{\tilde{P}(x)}{Q(x)}$ is known[4], such proposals can be unacceptably inefficient, while other proposals closer to the target are much more efficient in practice, but eschew strict formal guarantees. We provide variance estimates of our partition function estimates throughout the experimental section, and will discuss some provable bounds on Z for some of the EBMs that we consider in Section 5.3.2.

5.2.5 QRS Properties

We provide some interesting properties of QRS samplers. As mentioned before, ideally an approximate sampler samples from a (sampling) distribution as close to the target distribution as possible. In QRS, we have the hyperparameter β that allows for trading off efficiency (lower β, see Eq. 5.15) and better approximation quality (higher β). The latter is shown with the following theorem, proven in Eikema et al. (2022):

Theorem 5.1. *Let* $0 < \beta < \beta' < \infty$. *Then* $D_f(P, P_{\beta'}) \leq D_f(P, P_\beta)$

[3]Also note that any partition function is lower-bounded by $m = 0$.

[4]Note that if we would know a bound on this ratio, we could construct a (standard) rejection sampling sampler.

Theorem 5.1 guarantees that increasing the value of parameter β never increases the f-divergence of the target distribution from the QRS distribution. We will refer to this as the monotonicity property of QRS for all f-divergences.

We also provide an upper-bound on the total variation distance between the sampling and target distributions. We refer the reader again to Figure 5.1 for an intuition. For outcomes where the ratio $\tilde{P}(x)/Q(x) > \beta$ the sample is always accepted, but these outcomes possibly receive too little probability mass compared to the target distribution. Let these regions where the classical rejection sampling bound is violated be:

$$\bar{A}_\beta \doteq \{x \in \mathcal{X} : \frac{\tilde{P}(x)}{Q(x)} > \beta\} \tag{5.14}$$

It is intuitive that the "fewer" violators there are, the closer P_β is to the target P (*i.e.* the less white area in Figure 5.1). The following theorem, proven in Eikema et al. (2022), makes this statement precise, in terms of the actual probability mass under the target distribution P of the violators:

Theorem 5.2. $\text{TVD}(P, P_\beta) \leq P(\bar{A}_\beta)$.

In other words, the TVD (a value between 0 and 1), is upper-bounded by the probability of the set of outcomes, under the target distribution P, that violate the bound. Reducing the size of this set of violators typically lowers the upper-bound on the TVD between the sampling and target distribution. Also, as a corollary of this, observing that $\lim_{\beta\to\infty} P(\bar{A}_\beta) \to 0$, one sees that P_β converges to P for $\beta \to \infty$. In experiments we will estimate this quantity using an importance sampling estimate:

$$P(A_\beta) = \sum_{x\in X} P(x)\mathbb{1}[x \in A_\beta] \approx \frac{1}{N}\sum_{i=1}^{N} \frac{\tilde{P}(x_i)}{ZQ(x_i)}\mathbb{1}[x_i \in A_\beta], \tag{5.15}$$

where $x_i \sim Q$, *i.e.* we use samples x_i from the proposal distribution Q.

5.2.6 Estimating the Mapping Between β and AR

Eq. 5.15 provides a way to estimate the acceptance rate (AR) given a value of β and a set of samples from a proposal Q. How can we go in the opposite direction and estimate β for a target AR value? One way to do so, is to estimate the full mapping from AR to β, and interpolate it at the target AR. Algorithm 2 estimates this mapping efficiently, based on the observation that Eq. 5.15 can be rewritten as a sum of two terms:

$$\text{AR}_\beta \approx (a_i + b_i)/N \qquad \text{at } \beta = \beta_i \tag{5.16}$$

where

$$\beta_i \doteq \tilde{P}(x_i)/Q(x_i), \qquad a_i \doteq \sum_{j=1}^{N} \mathbb{1}\left[\beta_j \leq \beta_i\right] \beta_j/\beta_i, \qquad b_i \doteq \sum_{j=1}^{N} \mathbb{1}\left[\beta_j > \beta_i\right], \tag{5.17}$$

noting that both a and b can be computed efficiently given a sorted list of β_i values. (In the case that $\beta_i = \beta_j$ for some $j \neq i$, the output will contain repeated values, which are easily filtered out, if necessary). We present the algorithm we will employ in experiments to get the desired ranges of β values in Algorithm 2 below.

Algorithm 2 Estimate AR $\rightarrow \beta$ mapping

Require: $\tilde{P}$, Q, N
$S \leftarrow [\,]$
for $i = 1, 2, \ldots, N$ **do**
  $x_i \sim Q$
  $\beta_i \leftarrow \tilde{P}(x_i)/Q(x_i)$
  $S[i] \leftarrow \beta_i$
end for
$S_s \leftarrow \texttt{SortAscending}(S)$ {Array of sorted β_i}
$a_{aux}[0] \leftarrow 0$
for $i = 1, 2, \ldots, N$ **do**
  $\beta_i \leftarrow S_s[i]$
  $a_{aux}[i] \leftarrow a_{aux}[i-1] + \beta_i$ {$a[i] = \sum_{j:\beta_j \leq \beta_i} \beta_j$}
  $b[i] \leftarrow N - i$ {$b[i] = \sum_j \mathbb{1}\left[\beta_j > \beta_i\right]$}
end for
for $i = 1, 2, \ldots, N$ **do**
  $\beta_i \leftarrow S_s[i]$
  $a[i] \leftarrow a_{aux}[i]/\beta_i$ {$a[i] = \sum_{j:\beta_j \leq \beta_i} \beta_j/\beta_i$}
  $AR[i] \leftarrow (a[i] + b[i])/N$
end for
return AR and S_s {$S_s[i]$ is the β at which $AR_\beta = AR[i]$}

5.3 Experiments

We will now verify the effectiveness of our sampler experimentally. We start with a toy setting in Section 5.3.1, where we wish to sample from a *known* Poisson target distribution using samples from a different Poisson distribution as proposal generations. In Section 5.3.2 we turn to neural text generation again, sampling from EBMs that encode constraints on the sequence distribution as a whole,

exploring various ways of constructing proposal distributions. We continue in Section 5.3.3 with a small paraphrase generation experiment that combines two machine translation models in a round-trip fashion as proposal distribution. In Section 5.3.4 we compare the QRS sampler with MCMC techniques using various proxy metrics as well as using a lower bound on divergence measures. Finally, in Section 5.3.5 we return to the Poisson toy example, which allows us to estimate exact f-divergences for MCMC samplers, for comparison with QRS.

5.3.1 Sampling From a Poisson Distribution

To demonstrate the usage of QRS we start with a toy setting using two Poisson distributions. The goal is to sample from a target Poisson distribution P with rate $\lambda_p = 11$ using samples from a proposal Poisson distribution Q with rate $\lambda_q = 10$. Rejection sampling is *not* possible in this setting as the ratio $P(x)/Q(x) = (e^{-11}11^x/x!)/(e^{-10}10^x/x!) = e^{-1}\,1.1^x$ can take arbitrarily large values when x increases, i.e. the ratio is unbounded. However, it is possible to use QRS here.

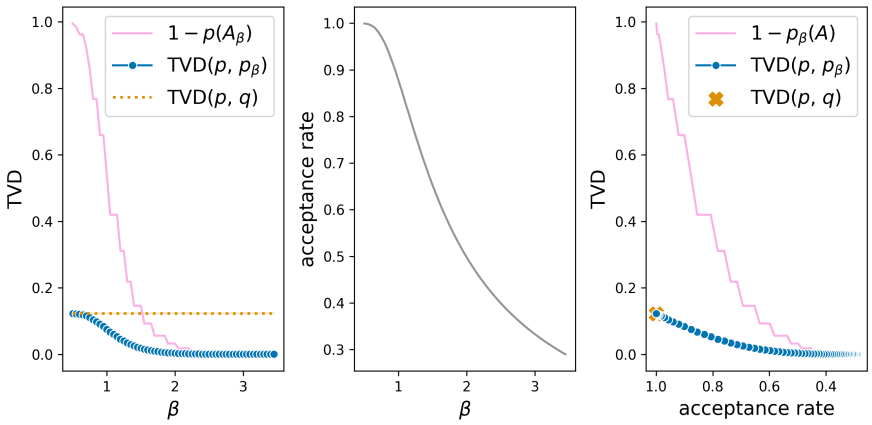
We perform ten independent experiments in which we sample 10^4 elements from Q that we use to compute the quality of the approximation by estimating: $\mathrm{KL}(P, P_\beta)$, $\mathrm{TVD}(P, P_\beta)$ and its upper bound $P(\bar{A}_\beta)$. We use β values in the interval $[0.5, 3.5]$[5]. Furthermore, we compute the sampler's efficiency by estimating the acceptance rate (AR) for each value of β (Eq. 5.2). As described previously, we compute estimates for all these metrics using importance sampling. Results for TVD and its upper-bound are displayed in Figure 5.2. As shown, using higher values of β improves the TVD, even though this comes at the cost of lower acceptance rate. In particular, with $\beta = 3.5$, the TVD is tiny ($\approx 10^{-4}$), yet the acceptance rate is moderate (0.3, i.e. 30% of proposal samples are accepted). Notably, we can ease the visualisation of the trade-off between quality and efficiency by reparametrising the divergence metrics in terms of the acceptance rate (last panel of Figure 5.3a). We use the procedure to map acceptance rates to a corresponding β formerly discussed in Section 5.2.6. We use this concise reparameterised representation to plot subsequent results.

In Figure 5.3 we show the same plots for KL-divergence. Similarly to the TVD results, we see that the divergence quickly converges to zero as β increases.

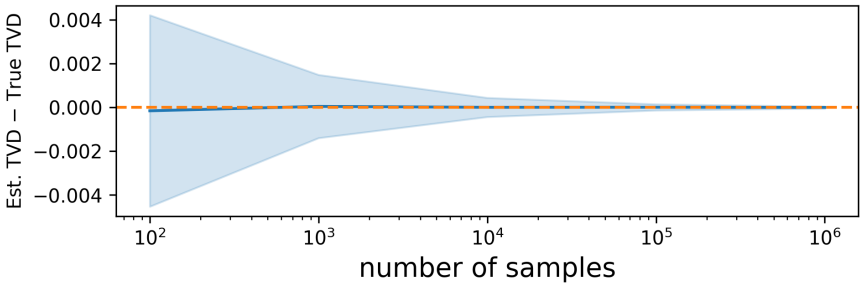
Empirical Estimates of Divergence Diagnostics

In the previous and following experiments, we use a sufficiently large sample size to obtain accurate estimates of the divergence diagnostics. However, it is reasonable to wonder about the bias and variance of these estimators for smaller sample sizes. While it is not possible to provide a definite answer for all EBMs, we can investigate this question by exploiting the fact that we can compute $D_f(P, P_\beta)$

[5] We use the procedure described in Section 5.2.6 to obtain these.



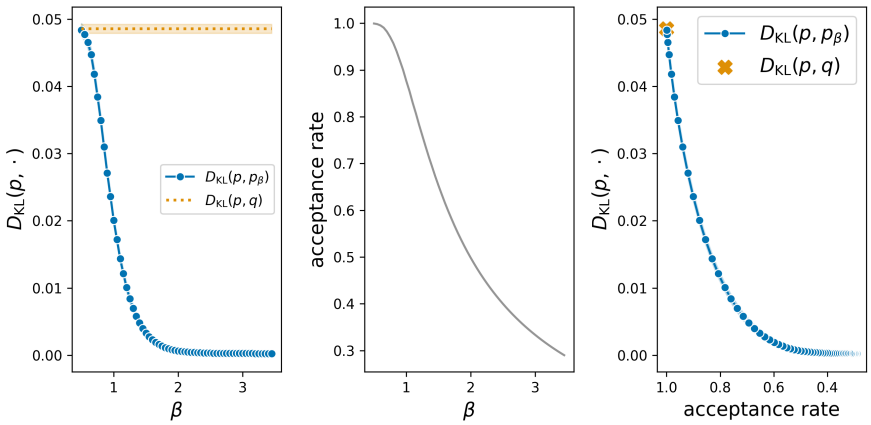
(a)



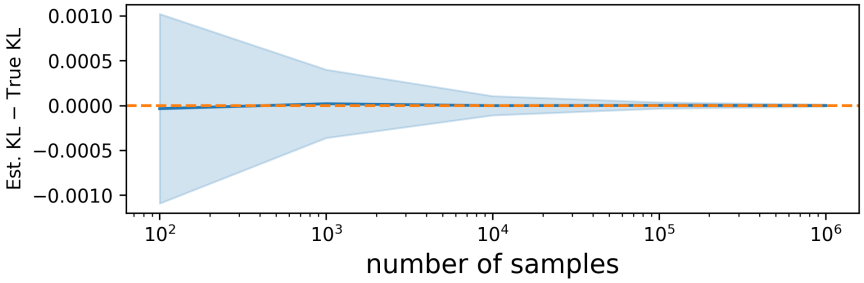
(b)

Figure 5.2: (a) Estimation of sampling quality as $\mathrm{TVD}(P, P_\beta)$, efficiency (acceptance rate), and the trade-off between them for a QRS sampler when using a proposal $Q = \mathrm{Poisson}(\lambda = 10)$ to approximate $P = \mathrm{Poisson}(\lambda = 11)$ computed in 10 independent experiments over 10^4 samples. (b) Differences between estimated $\mathrm{TVD}(P, P_\beta)$ and their true values for $\beta = 2$, computed 1000 times for each different number of samples used in the importance sampling estimate. The blue line is the mean and shaded areas represent one standard deviation. We note that the notation in this figure differs slightly from that used in the main text of the thesis, namely using lower-cased letters (p, p_β and q), for the discrete distributions P, P_β and Q.

with great precision when both P and Q are (known) Poisson distributions. We compare this approximation to the true value with the estimators proposed in Section 5.2 using sample sizes $n \in \{10^2, \ldots, 10^6\}$ and repeating the process 1000 times. Results for TVD and KL are shown in Figure 5.2b and 5.3b respectively. As can be seen, there is some small variance when only 100 samples are used for the estimation, and it quickly improves as more samples are used. Furthermore, the estimation bias is tiny even when using only 100 samples.



(a)



(b)

Figure 5.3: (a) Estimation of sampling quality as $\mathrm{KL}(P, P_\beta)$, efficiency (acceptance rate), and the trade-off between them for a QRS sampler when using a proposal $Q = \mathrm{Poisson}(\lambda = 10)$ to approximate $P = \mathrm{Poisson}(\lambda = 11)$, computed in 10 independent experiments over 10^4 samples. (b) Differences between estimated $\mathrm{KL}(P, P_\beta)$ and their true values for $\beta = 2$, computed 1000 times for each different number of samples used in the importance sampling estimate. Shaded areas represent one standard deviation. We note that the notation in this figure differs slightly from that used in the main text of the thesis, namely using lower-cased letters (p, p_β and q), for the discrete distributions P, P_β and Q.

5.3.2 Generation with Distributional Control

The following experiments focus on the task of generation with distributional control, introduced by Khalifa et al. (2021), a task that requires sampling from an EBM over sequences of discrete tokens, making it an ideal test bed for QRS. Given a language model $A(x)$, the goal of this task is to sample from a model $P(x)$ that, on the one hand, constrains the moments of a vector of n pre-defined features $\phi(x)$ to match some desired value $\bar{\mu}$ (i.e. $\mathbb{E}_{x\sim P}\phi(x) = \bar{\mu}$), while on the other hand minimising $\mathrm{KL}(P, A)$, a generalised version (Csiszar, 1975; Kullback and Khairat, 1966) of the maximum entropy approach (Jaynes, 1957; Rosenfeld,

prompt name	prompt
simple	Wikileaks.
multiple	Wikileaks, Wikileaks, Wikileaks.
knowledge	Here is what I know about Wikileaks:
jeopardy	This medium was founded by Julian Assange in 2006.
news	Here are the latest developments on Wikileaks:

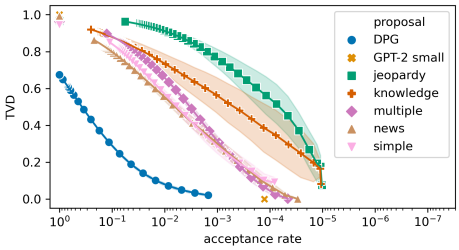


Figure 5.4: Comparing GPT-2, GPT-2 conditioned on various prompts, and a fine-tuned model (DPG) as proposals for generating sequences containing "Wikileaks". Disconnected points in the upper-left corner indicate $\mathrm{TVD}(P, Q)$ for each proposal, while the curves show $\mathrm{TVD}(P, P_\beta)$ as a function of the acceptance rate. Standard deviation bootstrap estimates are shown as shaded regions for every proposal. Ideal proposals concentrate in the lower left corner of this plot, achieving low TVD at high acceptance rates.

1996). For example, one might want to debias a language model trained on a corpus of biographies to produce biographies only of scientists, 50% of which should be female. Then $\phi_1(x)$ and $\phi_2(x)$ would be binary classifiers assessing whether a sentence speaks about a scientist or a female individual respectively, and the desired moments would be set to $\bar{\mu} = [1, 0.5]$.

The authors show that P can be expressed as an unnormalised EBM $\tilde{P} = A(x)b(x)$, and describe two choices of $b(x)$. On the one hand, they consider *pointwise* constraints, where $\bar{\mu} \in \{0, 1\}^n$. For instance, if there is a single binary feature for which we would like that $\forall x : \phi(x) = 1$, then b takes the form $b(x) = \phi(x)$. Otherwise, in the case of *distributional* constraints in which $\bar{\mu} \in \mathbb{R}^n$, they show that there is a vector $\lambda \in \mathbb{R}^n$ such that $b(x) = \exp(\lambda \cdot \phi(x))$ and $P(x) \propto A(x)b(x)$ fulfills the requirements of moment matching and minimal KL divergence from the original model. The vector λ is found using self-normalised importance sampling (Owen, 2013; Parshakova et al., 2019b) and stochastic optimisation.

Proposals for a pointwise constraint

We first experiment with constraining GPT-2 small (Radford et al., 2019) using one of the pointwise constraints ($\bar{\mu} = 1.0$) proposed in Khalifa et al. (2021), namely, $b(x) = \mathbb{1}[x \text{ contains "Wikileaks"}]$. In order to apply QRS we need to find a suitable proposal distribution. A possible candidate is GPT-2 small itself. An advantage of this proposal is that we can use pure rejection sampling with an upper-bound $\beta = 1$ to obtain exact samples from the EBM. This is because we can upper bound the ratio $\tilde{P}(x)/Q(x) = A(x)b(x)/A(x) = b(x) \leq 1$. In fact, for $b(x) \in \{0, 1\}$ this process reduces to "naively" filtering out all samples for which $b(x) = 0$. However, a serious disadvantage is that the acceptance rate will be

given by the natural frequency of the constraint, *i.e.* the frequency by which the constraint will be observed by simply sampling from the model. Using QRS, we can employ proposal distributions leading to better efficiency at a small cost in quality of approximation to P. We explore two such options:

1. First, we make use of the model proposed by Khalifa et al. (2021), which consists of a fine-tuned autoregressive model obtained by applying the distributional policy gradient (DPG) algorithm (Parshakova et al., 2019a) to approximate the target EBM in a generic way. While this model is considerably better at satisfying the desired constraints, it does not match the desired distribution perfectly.

2. Second, in the spirit of "in-context learning" (Brown et al., 2020), we propose to condition $A(x)$ on a prompt with the aim of increasing the constraint satisfaction rate in the resulting conditional distributions. In contrast to the previous approach, this does not require the training of a new model, even though it does require the manual selection of promising prompts. We experiment with five such prompts, which we present in Figure 5.4.

Figure 5.4 shows the $\mathrm{TVD}(P, \cdot)$ as a function of acceptance rate for different samplers. In this and the following experiments, we chose a range of β values that yields acceptance rates in the range 10^0–10^{-5} (also see Table 5.1), using Algorithm 2 in Section 5.2.6 for this purpose. We first show the $\mathrm{TVD}(P, Q)$ for each proposal Q, at an acceptance rate of 1, before applying QRS (in the upper left corner). Then, we plot $\mathrm{TVD}(P, P_\beta)$ as a function of acceptance rate for each proposal distribution. We compute IS estimates of the TVD on 1M samples from each proposal distribution. Variance is estimated using the bootstrap estimator (Wasserman, 2010). Note that all samples obtained from QRS satisfy the constraints perfectly, as sequences that do not satisfy the constraint are always rejected, and for this reason the curves start with different acceptance rates. As expected, using GPT-2 small results in zero TVD, but comes at the cost of low efficiency, with an acceptance rate around 10^{-4}. Using prompting, we improve the constraint satisfaction of the resulting proposal distributions and trade-off approximation quality for greater efficiency using QRS: For instance, if a TVD of 0.3 can be tolerated, then some of the prompt proposals provide a 10-fold higher acceptance rate with respect to the base GPT-2 model with no further training. Some prompts work notably better than others and we do not exclude the possibility of there existing prompts that perform even better than the ones we tested; we leave a more extensive exploration of prompting to create proposal distributions for future work. The autoregressive policy obtained from the DPG algorithm is the best proposal distribution we tested. Notably, it allows one to obtain low TVDs at higher acceptance rates than is possible by naively filtering samples from the base language model. For example, we can obtain a TVD of 0.1 at 100× the acceptance rate.

Chandra Pradha Towni (born February 11, 1965) is a social scientist, activist, poet, and author living in Portugal. She is...

Enrella Carrière is a Canadian writer, translator, and philosopher specializing in the history of show business. She has covered topics such as the direction and psychology of television and the evolution of human...

Albert Fahn (born 1970) is an American scientist who focuses on algorithms for generating biomechanical data. Methods to generate and construct biomechanical data...

Wyndham Radnor (born 1946) is a British historian and criminologist specialising in the subject of labour law. He has written extensively on...

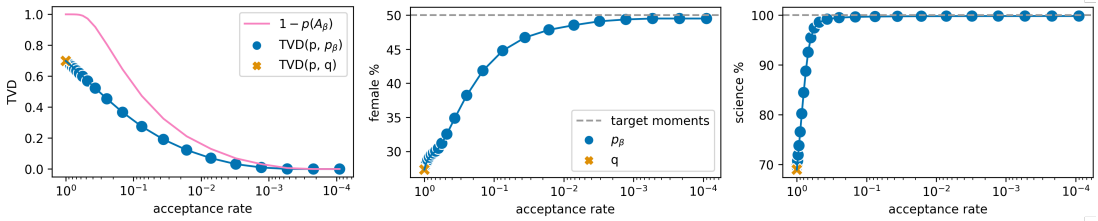


Figure 5.5: Estimation of the divergence from the EBM (TVD), and moments of features *female* and *science* in sampling debiased GPT-2 biographies talking about scientists. Variance is negligible as we show in Table 5.2 in Section 5.3.2. We also show samples from running the QRS sampler at an acceptance rate of 10^{-3}. Samples are cut off at 40 (subword) tokens and are manually chosen to show two male and two female biographies, for constraint satisfaction (moment matching) results refer to the graph. We color words that fire our female or science features. We note that the notation in this figure differs slightly from that used in the main text of the thesis, namely using lower-cased letters (p, p_β and q), for the discrete distributions P, P_β and Q.

Distributional constraints

We now turn to the task, also introduced by Khalifa et al. (2021), of generating biographies of scientists while debiasing the gender distribution to contain female scientists 50% of the time. For this we make use of GPT-2 Biographies ($A(x)$), a language model fine-tuned on Wikipedia biographies[6] and follow the same setup as the authors to define the binary classifiers identifying sequences talking about scientists or females[7] and infer an EBM that matches the distributional constraints with minimal deviation from the original model. The frequency with which the model $A(\cdot)$ generates scientist biographies is 1.8%, female biographies 7.5%, and the frequency with which it generates female scientist biographies is only 0.14%. As proposal distribution, we use the DPG model that Khalifa et al. (2021) trained to approximate the EBM, which reaches a constraint satisfaction of 69.0% scientist, 27.3% female and 19.6% female scientist biographies.

As before, we obtain 1M samples from the proposal distribution to compute

[6] https://huggingface.co/mkhalifa/gpt2-biographies

[7] Gender is estimated by the ratio of female to male pronoun counts, scientists are identified by the mention of at least one of multiple words associated with the profession.

importance sampling estimates of $\mathrm{TVD}(P, P_\beta)$, acceptance rate (AR_β), and the moments of the features that we wish to control. We show all metric curves as a function of acceptance rate of the QRS algorithm as well as some generation examples in Figure 5.5.

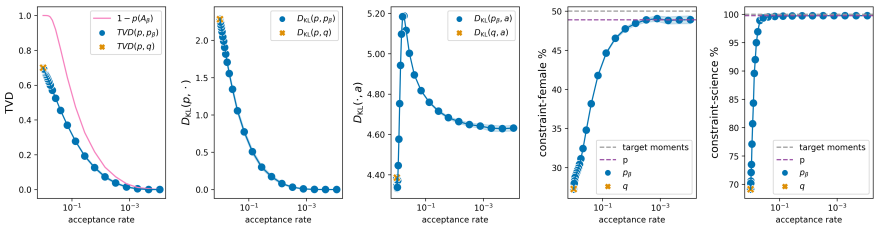
We find that both $\mathrm{TVD}(P, P_\beta)$ and the upper-bound on $\mathrm{TVD}(P, P_\beta)$ steadily converge to zero as the acceptance rate decreases, meaning that we can perfectly match the target EBM in exchange for sampling efficiency. As a result, at an acceptance rate of $\mathrm{AR}_\beta = 10^{-3}$ we nearly perfectly debias the original language model while exclusively generating biographies about scientists (49.5% female and 99.8% scientist biographies). We show some example generations at $\mathrm{AR}_\beta = 10^{-3}$ chosen manually to illustrate two male and two female biographies. Notably, we also achieve a good level of constraint satisfaction (48.4% female and 99.9% scientist biographies) and a TVD of 0.1 at $\mathrm{AR}_\beta = 10^{-2}$. This is a considerable improvement in quality with respect to the proposal distribution (with a TVD of 0.7), and in acceptance rate relative to directly rejecting from GPT-2 Biographies (which would result in $\mathrm{AR} \leq 2 \times 0.14\% = 2.8 \times 10^{-3}$).

Additional Constraints

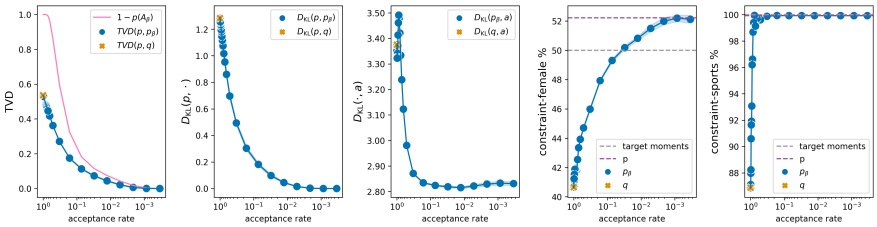
In this section, we repeat the previous experiments for some additional constraints, both pointwise and distributional, as well as show KL-divergence results for the previous constraints. In particular, we constrain the GPT-2 biographies model to contain (a) 50% female biographies about scientists, (b) 50% female biographies about sports, or (c) 50% female biographies without additional constraint. Also, we constrain GPT-2 small to exclusively generate sequences containing (d) the term "amazing", or (e) the term "Wikileaks". For each of these tasks, we obtained a fine-tuned model using DPG, which serves both as a baseline and as a proposal Q that we can sample from. In the case of pointwise constraints, we also consider a *naive filter* sampler Q_{proj} in which the proposal distribution is directly projected onto the constraint manifold by filtering out all samples that do not match the constraint. This sampler also assigns well-defined scores to the sequences that it samples, so we can compute estimates of the TVD and KL for it.

For each task, we again obtain 1M samples from the corresponding proposal, which we use to evaluate the proposal Q, the projected proposal Q_{proj} (only for the pointwise constraints), and QRS sampling (P_β) for a range of β values reported in Table 5.1. For all of these, we compute estimates of a number of metrics including those of the previous sections (i.e. $\mathrm{TVD}(P, P_\beta)$, $\mathrm{KL}(P, P_\beta)$, AR, reverse KL divergence from the base language model $\mathrm{KL}(\cdot, A)$, and the moments of the features that we wish to control).

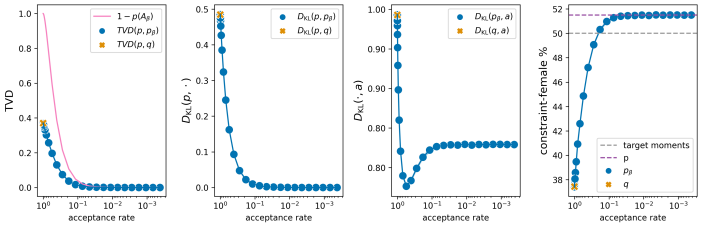
Our results are shown in Figure 5.6. As expected, the upper bound on the TVD between P_β and P, and the KL divergence of P from P_β both converge monotonically to zero as the acceptance rate decreases. For the constraints and



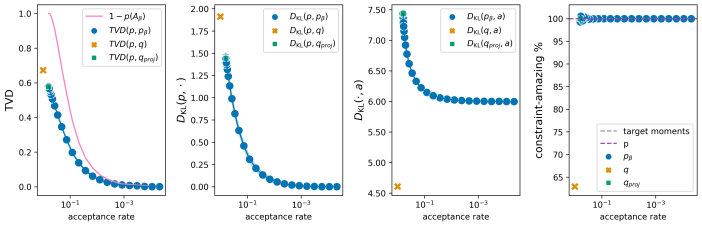
(a) 50% female scientist biographies



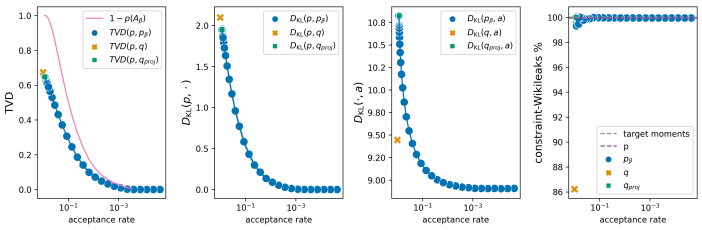
(b) 50% female sports biographies



(c) 50% female biographies



(d) 100% sequences containing “amazing”



(e) 100% sequences containing “Wikileaks”

Figure 5.6: We show IS estimates of $\mathrm{TVD}(P, \cdot)$, an upper-bound on $\mathrm{TVD}(P, P_\beta)$, $\mathrm{KL}(P, \cdot)$, $\mathrm{KL}(\cdot, A)$ and feature moments as a function of acceptance rate. We show three distributional constraints on GPT-2 biographies and two pointwise constraints on GPT-2 small. As proposal distribution, we use a DPG model trained for each constraint separately. We show separate lines for the target moments and the moments realized by the EBMs, revealing slight inaccuracies in the EBM moments for some constraints. We note that the notation in this figure differs slightly from that used in the main text of the thesis, namely using lower-cased letters (p, p_β, q and q_{proj}), for the discrete distributions P, P_β, Q and Q_{proj}.

corresponding proposal distributions shown here, it seems that an acceptance rate of 10^{-3} is sufficient to match the target EBM nearly perfectly. The feature moments also converge as the acceptance rate decreases, although in some cases the QRS sampler matches the target EBM so closely that small inaccuracies in the λ values obtained from the EBM estimation procedure (which is from Khalifa et al. 2021) become apparent. As for the divergence of the QRS distribution from the original language model $\mathrm{KL}(P_\beta, A)$, there is no obvious trajectory that it should follow other than a tendency to converge to the lowest possible value $\mathrm{KL}(P, A)$ when all constraints are satisfied (by definition, the EBM P is the distribution C which minimizes $\mathrm{KL}(C, A)$ among all distributions satisfying the constraints). Indeed, our results show that this metric is a non-monotonic function of AR. We observe that the moments computed downstream on QRS closely match the IS predictions, giving us confidence in the accuracy of those estimates. Finally, in the case of our pointwise "amazing" and "Wikileaks" constraints, we find that the naive filter strategy (q_{proj}) corresponds to running the QRS sampler at a high acceptance rate.

Experiment	$\beta_{\min}$	$\beta_{\max}$
50% female and 100% scientists	$1.0 \cdot 10^{-12}$	$9.3 \cdot 10^{6}$
50% female and 100% sports	$1.0 \cdot 10^{-12}$	$2.9 \cdot 10^{7}$
50% female	$4.0 \cdot 10^{-7}$	$4.0 \cdot 10^{3}$
100% "amazing"	$1.0 \cdot 10^{-12}$	$5.3 \cdot 10^{1}$
100% "Wikileaks"	$1.0 \cdot 10^{-12}$	6.0
simple	$1.0 \cdot 10^{-12}$	3.8
multiple	$1.0 \cdot 10^{-12}$	2.3
knowledge	$1.0 \cdot 10^{-12}$	24.0
jeopardy	$1.0 \cdot 10^{-12}$	29.0
news	$1.0 \cdot 10^{-12}$	3.0

Table 5.1: We report the range of β values used to obtain the range of acceptance rates in Figure 5.4, 5.5, and 5.6.

Importance Sampling Estimates

We report variances of all importance sampling estimates computed using the bootstrap estimator in Table 5.2. We report mean $\pm$ one standard deviation for β values within the range used in our experiments (also see Table 5.1). We find our estimates to be accurate within reasonable variance.

Secondly, as noted in Section 5.2.2, it is not generally possible to compute Z or its variance for any given EBM with certainty. This observation can put into question the validity of the above-described estimates. We note, however, that

P	β	AR	TVD(P, P_β)	KL(P, P_β)
100% amazing	0.02	$0.08 \pm 1.8 \times 10^{-4}$	$0.2 \pm 6.5 \times 10^{-3}$	0.3 ± 0.04
	0.3	$7.8 \times 10^{-3} \pm 3.2 \times 10^{-5}$	$0.04 \pm 8.0 \times 10^{-3}$	0.05 ± 0.02
	2	$1.2 \times 10^{-3} \pm 6.8 \times 10^{-6}$	$0.01 \pm 6.7 \times 10^{-3}$	$0.01 \pm 9.3 \times 10^{-3}$
100% wikileaks	7.4×10^{-4}	$0.1 \pm 2.1 \times 10^{-4}$	$0.3 \pm 5.2 \times 10^{-3}$	0.6 ± 0.03
	0.01	$8.5 \times 10^{-3} \pm 3.7 \times 10^{-5}$	$0.07 \pm 7.3 \times 10^{-3}$	0.09 ± 0.02
	0.2	$8.2 \times 10^{-4} \pm 6.6 \times 10^{-6}$	$9.7 \times 10^{-3} \pm 3.5 \times 10^{-3}$	$3.9 \times 10^{-3} \pm 1.9 \times 10^{-3}$
50% female	1×10^1	$0.1 \pm 1.3 \times 10^{-4}$	$0.02 \pm 6.2 \times 10^{-4}$	$0.01 \pm 1.2 \times 10^{-3}$
	1×10^2	$0.01 \pm 1.5 \times 10^{-5}$	$5.9 \times 10^{-4} \pm 2.9 \times 10^{-4}$	$3.3 \times 10^{-4} \pm 1.8 \times 10^{-4}$
	2×10^3	$1.1 \times 10^{-3} \pm 1.5 \times 10^{-6}$	$2.7 \times 10^{-7} \pm 4.4 \times 10^{-7}$	$-1.7 \times 10^{-8} \pm 9.4 \times 10^{-7}$
50% female + 100% science	7×10^3	$0.08 \pm 1.7 \times 10^{-4}$	$0.3 \pm 6.0 \times 10^{-3}$	0.5 ± 0.03
	1.1×10^5	$7.1 \times 10^{-3} \pm 3.5 \times 10^{-5}$	$0.07 \pm 7.1 \times 10^{-3}$	0.08 ± 0.02
	6.4×10^5	$1.3 \times 10^{-3} \pm 1.0 \times 10^{-5}$	$0.02 \pm 4.9 \times 10^{-3}$	$9.9 \times 10^{-3} \pm 4.8 \times 10^{-3}$
50% female + 100% sports	1.2×10^5	$0.07 \pm 1.3 \times 10^{-4}$	$0.1 \pm 4.2 \times 10^{-3}$	0.2 ± 0.01
	7.5×10^5	$0.01 \pm 3.5 \times 10^{-5}$	$0.04 \pm 3.8 \times 10^{-3}$	$0.04 \pm 6.2 \times 10^{-3}$
	1.2×10^7	$8.5 \times 10^{-4} \pm 4.6 \times 10^{-6}$	$4.2 \times 10^{-4} \pm 3.2 \times 10^{-4}$	$4.2 \times 10^{-5} \pm 3.6 \times 10^{-5}$

P	β	Z	Z_β
100% amazing	0.02	$2.5 \times 10^{-3} \pm 2.5 \times 10^{-5}$	$0.08 \pm 1.8 \times 10^{-4}$
	0.3		$7.8 \times 10^{-3} \pm 3.2 \times 10^{-5}$
	2		$1.2 \times 10^{-3} \pm 6.8 \times 10^{-6}$
100% wikileaks	7.4×10^{-4}	$1.4 \times 10^{-4} \pm 1.4 \times 10^{-6}$	$0.1 \pm 2.1 \times 10^{-4}$
	0.01		$8.5 \times 10^{-3} \pm 3.7 \times 10^{-5}$
	0.2		$8.2 \times 10^{-4} \pm 6.6 \times 10^{-6}$
50% female	1×10^1	$2 \pm 2.3 \times 10^{-3}$	$0.1 \pm 1.3 \times 10^{-4}$
	1×10^2		$0.01 \pm 1.5 \times 10^{-5}$
	2×10^3		$1.1 \times 10^{-3} \pm 1.5 \times 10^{-6}$
50% female + 100% science	7×10^3	$8 \times 10^2 \pm 9$	$0.08 \pm 1.7 \times 10^{-4}$
	1.1×10^5		$7.1 \times 10^{-3} \pm 3.5 \times 10^{-5}$
	6.4×10^5		$1.3 \times 10^{-3} \pm 1.0 \times 10^{-5}$
50% female + 100% sports	1.2×10^5	$1 \times 10^4 \pm 6 \times 10^1$	$0.07 \pm 1.3 \times 10^{-4}$
	7.5×10^5		$0.01 \pm 3.5 \times 10^{-5}$
	1.2×10^7		$8.5 \times 10^{-4} \pm 4.6 \times 10^{-6}$

Table 5.2: Means and standard deviation of IS estimates of acceptance rate, TVD with the target distribution and KL divergence to the target distribution for various β on various EBMs using a DPG fine-tuned proposal. We perform 5,000 bootstrap simulations using 1,000,000 samples each to compute the means and standard deviations. Values of β are chosen within the range used for our experiments as reported in Table 5.1.

when we can bound $\tilde{P}(x)/Q(x)$ we can have formal bounds on these quantities, allowing us to double-check the accuracy of these estimates. In particular, this is possible when $\tilde{P}(x) = A(x)b(x)$, $Q(x) = A(x)$ and $b(x) \in [0, M]$, as in that case we have $\tilde{P}(x)/Q(x) \leq M$. Using Popoviciu's inequality again (see Section 5.2.2), we find that the IS estimator of Z has variance

$$\text{Var}\left[\frac{1}{N}\sum_{i=0}^{N}\frac{\tilde{P}(x_i)}{Q(x_i)}\right] \leq \frac{M^2}{4N}. \tag{5.18}$$

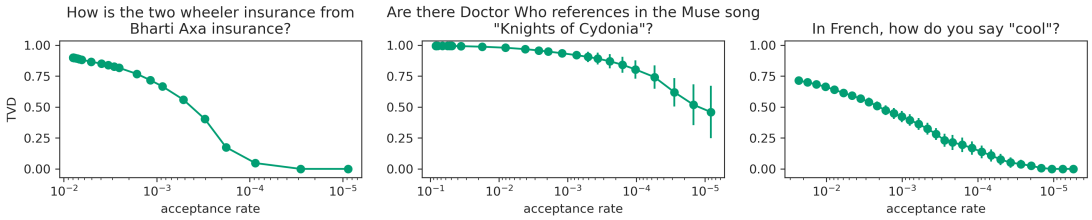
We generated 1 million samples with GPT-2, and used them to compute the partition functions of the EBMs for pointwise constraints, with provable bounds on their variance using the above equation. For the 100% "amazing" EBM, we obtained $Z = 2.5 \times 10^{-3} \pm 2.5 \times 10^{-7}$; whereas for 100% "wikileaks" we obtained $Z = 1.5 \times 10^{-4} \pm 2.5 \times 10^{-7}$. These are in agreement with the estimates in Table 5.2. However for our experiments with distributional constraints, we determinined the rather large bound $M = 304868$, and we would need to gather at least 10^{12} samples to obtain a reasonable bound on the variance. This highlights the need for better proposal distributions to compute these quantities, especially considering that the number of samples that IS needs to compute partition function of $\tilde{P}(x)$ is inversely proportional to $\exp(\text{KL}(P, Q))$ (Chatterjee and Diaconis, 2018). This is the reason why, *in practice*, a good proposal distribution can allow us to compute accurate estimates, even if that comes at the loss of the above-described formal bounds.

5.3.3 Paraphrase Generation

Inspired by Miao et al. (2019), we also perform proof-of-concept experiments on paraphrase generation by framing the task in terms of conditional EBMs. Specifically, given a sentence y to paraphrase, we define our EBM in terms of a language model $A(x) = \text{GPT-2}(x)$, and a pointwise constraint $b(x)$ given by a binary classifier that classifies a pair (x, y) as a paraphrase if the cosine similarity between their sentence embeddings is above 0.95. We obtain high-quality sentence embeddings from sentence-BERT[8] (Reimers and Gurevych, 2019). As proposal distribution we do not use GPT-2, but rather illustrate how we can use off-the-shelf deep learning models as proposal distributions for QRS. In particular, we use a round-trip machine-translation model, which is a well-known tool in generating paraphrases (Bannard and Callison-Burch, 2005; Mallinson et al., 2017). Specifically, we use the English-to-German and German-to-English models from Ng et al. (2019). We first obtain a translation into German using beam search,[9] and then define the proposal distribution as the German-to-English model con-

[8] We use https://huggingface.co/sentence-transformers/all-mpnet-base-v2.

[9] We use a beam size of 5.



Input Sequence	**Proposal Distribution**	**QRS samples** from p at AR $= 10^{-5}$
How is the two wheeler insurance from Bharti Axa insurance?	What about bicycle insurance from Bharti Axa insurance?	How is the Axa Bharti two-wheeler insurance policy?
	What about the Bharti Axa insurance?	How is Bharti Axa insurance for two-wheeler?
	What is the Bharti Axa insurance plan?	The Bharti Axa Two-wheeler insurance. How is it?
Are there Doctor Who references in the Muse song "Knights of Cydonia"?	Do you hear a hint of doctors in the Muse songs "Knights of Cydonia"?	Are there Doctor Who references in Muse's Knights of Cydonia?
	Can you find a hint at Doctor Who in the "Knights of Cydonia" line from the book's Muse song?	Does this Muse song 'Knights of Cydonia' have any references to Doctor Who?
	Are there any references to Doctor Who in a muse song, Knights of Cydonia?	Are there Doctor Who references in Muse's Knights of Cydonia?
In French, how do you say "cool"?	How do you call 'cool' in French?	How to Say "cool" in French
	How do you keep the language Cool in French?	How to Say 'Cool' in French
	How do you say 'cool' in French?	How do you say "cool" in French

Figure 5.7: TVD(P, P_β) running the QRS sampler at various acceptance rates to generate paraphrases of three sequences (**top**). We show some example paraphrases from both the proposal distribution $Q(x)$ (round-trip NMT) as well as the QRS sampler P_β at an acceptance rate of 10^{-5} (**bottom**).

ditioned on the obtained translation. We locally renormalise the model to do top-30 sampling (Fan et al., 2018).[10]

We show IS estimates of TVD(P, P_β) using 1M samples for three sequences in Figure 5.7 along with samples from both the proposal distribution and QRS at AR $= 10^{-5}$. The quality of the proposal distribution varies with the input sequence, as can be seen from the slope of the curve and the low-efficiency starting points of some curves (non-paraphrases are always rejected and so they have a big influence on the acceptance rate). Still, QRS gives excellent approximations to the target EBM in two out of the three examples (in the "insurance" and "cool" examples, the TVD is nearly zero), although the TVD in the "Doctor Who" ex-

[10]The decision to use a top-k sampling strategy to generate from the round-trip translation model violates the constraint we set in Section 5.2, namely that the support of P should be included in the support of Q. In Appendix D.2 of Eikema et al. (2022), we show that this affects the convergence property of QRS such that it becomes $\lim_{\beta\to\infty} P(\bar{A}_\beta) = 1 - P(\text{supp}(Q))$, *i.e.* the convergence of the probability mass on violators (and thus the TVD bound of Theorem 5.2) is limited by how much of the support of P is covered in Q. If the support of P is included in the support of Q, we can see that the convergence guarantee we originally showed in Section 5.2.5, $\lim_{\beta\to\infty} P(\bar{A}_\beta) = 0$, is recovered.

ample remains above 0.4 even for AR $= 10^{-5}$). Looking at the examples, we find that the proposal distribution produces decent paraphrases, but they are not always semantically equivalent or grammatically correct. The QRS samples are mostly semantically equivalent, though they still contain some mistakes ("Axa Bharthi" vs "Bharti Axa") and seem to be insensitive to the question mark and to the casing of words ("Cool", "Two-wheeler insurance"). Interestingly, this experiment illustrates how the presented approach could be employed to *disentangle* the questions of how to model a problem (by defining the corresponding EBM) and how to efficiently sample from it (by improving the proposal distributions), making it possible to work on each of these questions separately.

5.3.4 Comparison with MCMC techniques

We now compare QRS with MCMC samplers, for which we focus on the EBM for a *pointwise* constraint restricting GPT-2 to only generating sequences containing "amazing".

Baselines

We use the popular Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953; Hastings, 1970). This algorithm works by constructing a Markov chain of dependent samples, in which the next state of the chain is equal to a newly proposed sample with a certain acceptance probability, and otherwise the next state is the same as the current state. When the length n of the chain tends to infinity it can be proven (Robert and Casella, 2004, Theorem 7.4), see Section 2.2.5, that the average of a statistic on the elements of the chain converges to its expected value under the target distribution and, more importantly when focussing on sampling as we do, that the distribution of the n^{th} element of the chain converges (in total variation) to the target distribution. In practice, the chain is of finite length and only approximate samples are obtained.

Common practices are to discard the first few samples of the chain to reduce the effects of poor starting conditions (which is known as burn-in), and to only keep every t^{th} sample to reduce autocorrelations (which is known as thinning). We use both these heuristics in our experiments. We set a burn-in period of 1,000 steps and only keep every $1,000^{\text{th}}$ sample to attain an acceptance rate of 10^{-3}. Note that we chose not to include the burn-in period to compute the acceptance rate of MCMC samplers, as this period is constant and does not grow with sample size. We also experiment with a reset variant (-R) of the MH samplers that does away with autocorrelations among samples altogether (i.e. produces i.i.d. samples like QRS) by, instead of using thinning, resetting the chain after 1,000 steps and only retaining the last sample of the chain. This variant does not make use of a burn-in period.

We experiment with two proposal distributions for use in the MH samplers: *i)*

the global proposal distribution used in QRS, i.e. DPG, for independent Metropolis-Hastings (IMH; Robert and Casella, 2004) and *ii)* a local proposal distribution that makes local edits to an evolving sample in the chain for random-walk Metropolis-Hastings (RWMH). As far as we are aware, IMH is not commonly employed in the literature due to global proposals classically being difficult to come by. We stress that with the advances in neural network training techniques such as DPG, global proposals are more accessible than ever and we therefore include IMH as a strong baseline in our experiments. Our design of the local proposal is inspired by uses of MCMC in controlled text generation, in particular by Miao et al. (2019) and Goyal et al. (2021). The local proposal randomly performs either an insert, delete or replace operation on the token level, where insert and replace operations are performed by sampling from BERT (Devlin et al., 2019). Locations of insertions and deletions are chosen uniformly at random. To inform the local proposal distribution about the target distribution, we implement the insert and replace operations by randomly mask-filling using BERT (99% of the time) and filling with "amazing" (1% of the time). We initialize the chain using a sample from the global proposal distribution.[11]

Metrics

As no f-divergence estimates are available for the MCMC samplers, we instead resort to some proxy metrics specific to controlled text generation to measure the sample quality, extracted from 10^4 samples over ten independent experiments. In particular we measure constraint satisfaction (% amazing), perplexity (PPL) of our samples under GPT-2, diversity across samples with Self-BLEU-5 (Zhu et al., 2018) and percentage of unique samples (% Uniq), and finally, diversity within samples as given by the percentage of distinct bigrams (Dist-2; Li et al., 2016). However, we note that these metrics can easily be cheated. An example of this is if we would manually construct a sampler that would repeat a small set of reasonably diverse, high probability (under GPT-2) sequences that meet the target constraint. The resulting sampling distribution would not necessarily be close to the target EBM, but would score well on such proxy metrics. For QRS we do report the TVD to the target distribution, which cannot be gamed in this way. Notably, for MCMC we can also provide *lower bounds* on the TVD thanks to the data-processing inequality (DPI). The DPI tells us, informally, that the f-divergence of one distribution to another can only decrease by applying the same "projection" to the two distributions. A precise formulation of this theorem is provided as Theorem 6.2 of Polyanskiy and Wu (2017), but, for our purposes, the following special case ("lumping property") proven as Lemma 4.1 of Csiszár

[11]Note that localised MCMC methods can still take advantage of good starting points, especially the reset variants. We therefore also provide our variants of RWMH with a global proposal distribution for obtaining good starting points, the same global proposal as used for QRS.

Method	CS	PPL↓	Self-BLEU↓	%Uniq↑	Dist-2↑	TVD↓*
proposal	62.9 ± 0.4	61.7 ± 0.3	85.8 ± 0.1	100 ± 0.0	96.1 ± 0.0	0.67
RWMH	100 ± 0.0	-	99.8 ± 0.2	32.0 ± 33.7	83.8 ± 17	-
RWMH-R	100 ± 0.1	58.7 ± 3.3	87.6 ± 0.4	100 ± 0.0	92.0 ± 0.3	- (≥ 0.28)
IMH	100 ± 0.0	-	86.9 ± 0.3	98.7 ± 0.5	96.3 ± 0.1	-
IMH-R	100 ± 0.0	63.4 ± 1.5	86.7 ± 0.1	100 ± 0.0	96.3 ± 0.1	- (≥ 0.01)
QRS	100 ± 0.0	62.8 ± 1.6	86.6 ± 0.2	100 ± 0.0	96.3 ± 0.1	0.01

Table 5.3: Comparing with MCMC samplers for the constraint of including "amazing" in the sequence, CS denotes constraint satisfaction. We show mean ± one standard deviation over 10 runs. All samplers are run at an acceptance rate of 10^{-3}. *TVD is estimated on 10^6 independent samples, standard deviations are below 0.01.

and Shields (2004) will be sufficient:

Theorem 5.3. *Let P and Q be distributions over a sample space $\mathcal{X}$. Let $\{A_1, \ldots, A_k\}$ be a finite partition of $\mathcal{X}$. Then the distributions $P'(i) \doteq P(A_i)$ and $Q'(i) \doteq Q(A_i)$ over $\{1, \ldots, k\}$ satisfy $D_f(P', Q') \leq D_f(P, Q)$.*

While it is in general unfeasible to estimate the actual divergence of two distributions P, Q over a *large* space $\mathcal{X}$ based only on a moderate number of *samples* (as opposed to *scores*) from P and Q (Canonne, 2020), by projecting these samples onto a much smaller number k of "bins", it is possible to obtain precise estimates of the divergence of the "histogram" P' from Q', and therefore to *lower bound* the divergence of P from Q.

We use this result to obtain lower bounds on the TVD to the target distribution of RWMH-R and IMH-R, for which we have i.i.d. samples, whereas this is not the case for RWMH and IMH. After manually inspecting a single set of 10^4 samples out of ten independently sampled sets to look for salient defects, we chose a binning function that classifies a sample in two bins according to whether it contains a newline character or not. Then, we used it to bin the samples of the remaining nine sets and compute their mean TVD to the binned target distribution to obtain lower bounds on the TVD.

Results

Our results are shown in Table 5.3, and we also show the first 10 samples of each sampler in Tables 5.4 to 5.6. All samplers satisfy the constraint of only generating sequences containing the term "amazing". The EBM P, however, is defined to be, among all those distributions that satisfy the constraint, the one closest to the original language model A (GPT-2) under $\mathrm{KL}(P, A)$. Constraint satisfaction

QRS samples at 10^{-3} acceptance rate
• Cowgirl 87 Care Commission↵↵Header now available on our VIP web site.↵↵CowboyBillIssues AVAILABLE NOW...↵↵Start your website bio examining this amazing cam-
• take note. No wonder Vox isn't having an amazing day.↵↵Screen Snipe Dave ends Neo by christening it #DroppedSummer↵↵[] @cloudp
• I got an unprecedented amount of reviews years ago because of such an interesting movie that seemed Nude World will be released in some theaters soon. The movie itself was pretty amazing. I only had to hear
• Before Charlie Hebdo rendered its amazing save for lost lives, scholarship was blind to the brutal irony. Gregory Wade accidentally shot three bystanders, at least ten more wounded, and four more dead within three days of
• 4279 watts has passed the bar↵↵You can also check your save scores by clicking 'Save Data to PWSW Audience'.↵↵Shopping↵↵Welters also have amazing customer
• Sculpt your beard to stay stylish when you struggle to look like a nerd. This Collection makes all your lengthening amazing, so show it off { Ruler of Salt, Cone of Salt at Holl
• If you've contacted our sponsor, they know who you are and how hard you work. Unless months pass without a postcard. Best Served to meet everything you demand. You are amazing and are
• I'm sure a lot of people are most excited to see an F-35 Lightning II V2 today because of the amazing kick that's (understandably) in the engine room now that Benson
• Catherine is serviceable, happy, and successful. She loves teaching, having children, and saving time. She regularly works with can and is an amazing listener.↵↵Our community is full of
• The bellies of the runs was amazing. A ton of complements were good, including some neighboring dressing room appearances. So long worked up info, my 2014 Mustang had lots of special opening-specific

Table 5.4: The first 10 examples produced in the first run of QRS on the EBM filtering GPT-2 sequences containing "amazing".

alone thus does not tell us how well the samplers approximate the target EBM. For MCMC samplers we have to rely on proxy metrics. RWMH-R seems to excel in terms of perplexity while also obtaining competitive diversity metrics (Self-BLEU-5, % Uniq, and Dist-2). However, we can identify a large TVD between its sampling distribution and the target one, showcasing the failure of proxy metrics to reliably measure the approximation accuracy of our samplers, and demonstrating the benefits of having explicit estimates of divergence measures. After qualitatively inspecting samples of both local MCMC methods (see Table 5.5), we found many samples to contain repetitions, e.g. of punctuation marks. Also, RWMH seems to suffer from low diversity across samples (as evidenced by the high Self-BLEU-5 and low % Uniq scores), containing many repeated samples. On the other hand, IMH, IMH-R and QRS do not suffer from this lack of diversity. The variants of IMH and QRS seem to perform on par in terms of sample diversity, and QRS attains slightly lower perplexity than IMH-R.

An estimate of the TVD around 0.01 shows that QRS closely approximates the target EBM and shows considerably more diversity within and across samples. It might be that the IMH and IMH-R samplers would achieve similar results, but we cannot know as no divergence estimates are available. We only know that the binning strategy failed to detect any large divergences for IMH-R.

In Table 5.7 we additionally show KL-divergence and the Dist-1 and Dist-3 metrics (distinct unigrams and trigrams respectively) for the pointwise "amaz-

RWMH samples at 10^{-3} acceptance rate
• The OnePlus 2 is one of the most amazing smartphones we’ve ever tried. It’s an extremely powerful smartphone you’ve never heard of before, and carries an amazing battery life, or battery life I’m
• The OnePlus 2 is one of the most amazing smartphones we’ve ever tried. It’s an extremely powerful smartphone you’ve never heard of before, and carries an amazing battery life, or battery life I’m
• The OnePlus 2 is one of the most amazing smartphones we’ve ever tried. It’s an extremely powerful smartphone you’ve never heard of before, and carries an amazing battery life, or battery life I’m
• The OnePlus 2 is one of the most amazing smartphones we’ve ever tried. It’s an extremely powerful smartphone you’ve never heard of before, and carries an amazing battery life, or battery life I’m
• The OnePlus 2 is one of the most amazing smartphones we’ve ever tried. It’s an extremely powerful smartphone you’ve never heard of before, and carries an amazing battery life, or battery life I’m
• The OnePlus 2 is one of the most amazing smartphones we’ve ever tried. It’s an extremely powerful smartphone you’ve never heard of before, and carries an amazing battery life, or battery life I’m
• The OnePlus 2 is one of the most amazing smartphones we’ve ever tried. It’s an extremely powerful smartphone you’ve never heard of before, and carries an amazing battery life, or battery life I’m
• The OnePlus 2 is one of the most amazing smartphones we’ve ever tried. It’s an extremely powerful smartphone you’ve never heard of before, and carries an amazing battery life, or battery life I’m
• The OnePlus 2 is one of the most amazing smartphones we’ve ever tried. It’s an extremely powerful smartphone you’ve never heard of before, and carries an amazing battery life, or battery life I’m
• The OnePlus 2 is one of the most amazing smartphones we’ve ever tried. It’s an extremely powerful smartphone you’ve never heard of before, and carries an amazing battery life, or battery life I’m

(a) RMWH

RWMH-R samples at 10^{-3} acceptance rate
• yay!!! this game is amazing!!!!! a cool physical sci fi action game where you play a hardened junkie with an old girlfriend who’s tough
• in the immediate aftermath of the church shutdown, strange and often amazing events raised the question of whether the government had ever given up its restrictive control of religious provocations for political purposes and assumed a new political leadership
• ... had an amazing 7 out of 10, 8 out of 10, 9 out of 10, 6 out of 5... liked it... the story...
• I do have several solo masters who still have done amazing work to suit/complement orientated-character traits. In particular, I find myself building out secondary mazes based on their chunks iay
• "She and my son played "Friends," and used songs that I loved from that’s were amazing," Kleutz said. "They were acknowledging responsible themes. They felt I was so passionate that I
• useful for those in need or those who are distracted in the outside world. qualms (# 77574), pro / pro bizarre (# 776171), amazing 1 / 2 / 3 / 5 (# 776249)
• We are inspired by the amazing work of the Reader Space Foundation, which fosters artists who make an impact on a global audience through knowledge, support, and advocacy. It’s made us an un-
• the casanova is amazing the key to this being a classic, a true black lace bra made from dl’s, lace and all the fabrics that are available in pink and black. go on, have some fruit, sugar, vitamins, icing. whisper, wow, good for you
• and it is only using standard rules of thumb, but it is amazing that all the use of " " can be taken into account. " established.
• With so many amazing actors appearing on every level in a stone once it’s filmed, Star Wars Rebels follows the Rebellion throughout 2017 with an unusual archive of credits from the original and Rebels Two Casts.

(b) RMWH-R

Table 5.5: The first 10 examples produced in the first run of the local Metropolis-Hastings variants RWMH (a) and RMWH-R (b) on the EBM filtering GPT-2 sequences containing “amazing”.

IMH samples at 10^{-3} acceptance rate
• OF ALL The Verifiers who voted for Dragon Quest Revealed, two amazing choices are going to still be in the fire. To make up for the lost depth, Vulcans will make a busy
• A PO3 sniff tested, protecting the lab from detection by probiotic products, is amazing.↵↵5 years ago, scientists found the cannabinoid-blocking drug O3 responsible
• You have watched The Witcher 3 as a kid. You have eaten plenty of meat at your meals. You've seen a video game show before that has been amazing. Very few people have grown up without
• 11 am 2pm 958-701-99997 to Lift EcoTV, located in Washington, D.C. Do you have more questions about lifting something amazing in the Farm 600 in↵
• Bringing this gift is a unique opportunity to share with your loved ones about the home we love. Our family stunts cloud of feedback and the truly amazing handmade items we came upon included, providing a great
• Over a group of homebrew enthusiasts gathered for a beer sampling on Wednesday morning, they shared some amazing homebrew recipes with beer enthusiasts on the craft beer of the weekend, as well as some star brewers at home
• RIDER VIDEOTAPE | LEON BEVERLY RON OLD AFAIK proud to present you the amazing RADIO available from RON Old Hollywood: http://www.personalradio
• with comments, screenshots↵↵collaborating authors & planners↵↵Connecting friends & family, live with family of our wonderful amazing people.↵↵An Albany Folklore Center where we are based
• Could She Play The Same Sex Career?↵↵Missoula is full of different, amazing lesbian and gay couples to come. It's so amazing that she's so welcoming to them. They love
• amazing even or nerdyally↵↵I mean it's true that almost every person touching your skin is under their own skin, but when you have to contribute, you are competing against art first

(a) IMH

IMH-R samples at 10^{-3} acceptance rate
• Journey to Mars! Supermassive Black Hole Recovered From Big Bang Rosetta Sky↵↵It took an amazingly severe situation to produce an amazing HDR image through single recombination with massive radio uplink
• Early in the credits game, Sid Meier looked over Randi with his smugly excited face.↵↵"What if," he started, "if you think Randi's doing amazing things to me
• While Gilinda tells everybody what she has in mind in the past, notice that despite their competitiveness and amazing athleticism, they do outlive the Hokies and don't recognize one another.↵↵↵It
• ameral bairnsaucer, from Prague↵↵No, this wasn't a blast. It was awesome. My test team was amazing.↵↵If you wait against an scent that is
• Last day was so amazing. I was going to lay down for some lunch on the park deck and watch contestants go nuts. Happy morning.↵↵Psychos Solid: Winning the Super Bowl isn't
• Can Customers Generation Support Field Trawler?↵↵Well, it looks like it's gonna be well taken care of...Tousting can also apparently utilise compiled sound block and got some amazing gimm
• 64 Explicit Nov 16, 2016 FIVE US SECS CONCLUSIONS On Saturday night you'll find an amazing bunch of colleges and universities alike as they host the 'THE LAST TENSE' panel for
• Pick anything you wish for appears in new like Black Bag Revenge, Savage Comet, Bride of Waibou, Oinnibox and much more.↵↵Gamazing Of The Month Star Trade Reply
• At 6 miles in, the Gawdover Range is an amazingly located range within Markham county, NY. I was able to flag down one during the last few kilometres and here is the fun!
• I'm never happy about an OC for a while, since I can like them because the fact that the beautiful side has proven amazing makes sure I learned to love them more and remain satisfied when looking at

(b) IMH-R

Table 5.6: The first 10 examples produced in the first run of the global Metropolis-Hastings samplers IMH (a) and IMH-R (b) on the EBM filtering GPT-2 sequences containing "amazing".

ing" constraint of the former experiment, as well as for the EBM that produces 50% female scientist biographies. For the latter there is no obvious way to inform the proposal and therefore, we only perform mask-filling through insertions, replacements and deletions using BERT. We note the same patterns that we observed before for the 50% female scientists EBM, as well as for the KL-divergence (compared to TVD) in both EBMs.

5.3.5 Exact Divergence Estimates for MCMC

One case in which we can estimate f-divergences for IMH-R and RWMH-R variants is the Poisson example that we discussed in Section 5.3.1. We present a comparison between QRS and the reset variants of MH for this Poisson example here. As in Section 5.3.1, we use the proposal $Q = \text{Poisson}(\lambda = 10)$ and target $P = \text{Poisson}(\lambda = 11)$. To ensure accurate values for the KL divergence and TVD, we implement each method in closed form.

Calculating the probability mass functions of the samplers

For QRS, we compute the probability mass function P_β and acceptance rate using Eq. 5.1 and 5.2. For IMH-R, we take k samples from the proposal Q according to the generative process:

$$
\begin{aligned}
&X^{(0)} \sim Q, \\
&\textbf{for } t = 1, \ldots, k-1 \\
&\quad Y^{(t)} \sim Q \\
&\quad X^{(t)} = \begin{cases} Y^{(t)} & \text{with probability } \min\left(1, \frac{P(Y^{(t)})Q(X^{(t-1)})}{P(X^{(t-1)})Q(Y^{(t)})}\right) \\ X^{(t-1)} & \text{otherwise.} \end{cases}
\end{aligned}
$$

For nonnegative integer states $X^{(t-1)} = x_0$ and $X^{(t)} = x_1$, this corresponds to the transition matrix (or Markov kernel, noting that the state space is infinite):

$$
\begin{aligned}
T_{x_1,x_0} &= Q(x_1) \min\left(1, \frac{P(x_1)Q(x_0)}{P(x_0)Q(x_1)}\right), \qquad \forall x_1 \neq x_0 \\
T_{x_0,x_0} &= 1 - \sum_{x_1 \neq x_0} T_{x_1,x_0}.
\end{aligned}
$$

Using these transition probabilities, a finite chain length k, and by truncating the support of these Poissons to $x < 50$, we can compute the exact distributions obtained after performing k steps of Metropolis-Hastings, and compute divergence measures. For comparison with QRS, we compute the "acceptance rate" as $1/k$, i.e. the reciprocal of the number of times we sample from Q.

sampler	AR	%amazing	PPL↓	Self-BLEU-5↓	%Uniq↑	Dist-1↑	Dist-2↑	Dist-3↑	TVD(P, P_β) ↓*	KL(P, P_β) ↓*
DPG	1	62.9 ± 0.4	61.7 ± 0.3	85.8 ± 0.1	100.0 ± 0	89.3 ± 0.1	96.1 ± 0.0	94.1 ± 0.0	0.67 ± 0.00095	1.91 ± 0.04
IMH	10^{-1}	100.0 ± 0	-	92.0 ± 0.7	66.5 ± 3.9	89.9 ± 0.3	96.3 ± 0.1	94.3 ± 0.1	-	-
IMH-R	10^{-1}	100.0 ± 0	60.8 ± 1.4	87.1 ± 0.2	100.0 ± 0	89.8 ± 0.2	96.4 ± 0.1	94.4 ± 0.1	-	-
QRS	10^{-1}	100.0 ± 0	61.8 ± 0.8	86.8 ± 0.4	100.0 ± 0	89.9 ± 0.2	96.4 ± 0.1	94.3 ± 0.1	0.27 ± 0.0054	0.45 ± 0.045
RWMH-base	10^{-1}	80.0 ± 40.0	-	99.6 ± 0.9	21.8 ± 28.8	85.8 ± 7.6	91.1 ± 11.2	87.2 ± 14.6	-	-
RWMH-R-base	10^{-1}	63.2 ± 1.3	61.9 ± 1.2	86.5 ± 0.3	100.0 ± 0	88.5 ± 0.2	96.2 ± 0.1	94.3 ± 0.1	-	-
IMH	10^{-3}	100.0 ± 0	-	86.9 ± 0.3	98.7 ± 0.5	90.0 ± 0.2	96.3 ± 0.1	94.2 ± 0.1	-	-
IMH-R	10^{-3}	100.0 ± 0	63.4 ± 1.5	86.7 ± 0.1	100.0 ± 0	89.9 ± 0.2	96.3 ± 0.1	94.3 ± 0.1	-	-
QRS	10^{-3}	100.0 ± 0	62.8 ± 1.6	86.6 ± 0.2	100.0 ± 0	90.0 ± 0.2	96.3 ± 0.1	94.3 ± 0.0	0.01 ± 0.0067	0.011 ± 0.0093
RWMH	10^{-3}	100.0 ± 0	-	99.8 ± 0.1	41.0 ± 35.2	71.5 ± 36.0	69.4 ± 33.7	63.1 ± 31.1	-	-
RWMH-R	10^{-3}	100.0 ± 0	57.7 ± 3.9	87.8 ± 0.1	100.0 ± 0	82.9 ± 0.4	93.2 ± 0.4	92.4 ± 0.3	-	-

(a) 100% amazing

sampler	AR	%female	%science	PPL↓	Self-BLEU-5↓	%Uniq↑	Dist-1↑	Dist-2↑	Dist-3↑	TVD(P, P_β) ↓*	KL(P, P_β) ↓*
DPG	1	27.3 ± 0.4	69.1 ± 0.4	34.4 ± 0.2	89.8 ± 0.1	100.0 ± 0	89.7 ± 0.1	95.8 ± 0.0	93.1 ± 0.0	0.7 ± 0.00099	2.28 ± 0.03
IMH	10^{-1}	46.3 ± 4.3	99.8 ± 0.1	-	95.4 ± 0.5	55.7 ± 5.8	89.3 ± 0.7	96.0 ± 0.2	93.4 ± 0.2	-	-
IMH-R	10^{-1}	40.9 ± 1.5	99.5 ± 0.2	29.2 ± 0.8	91.3 ± 0.1	100.0 ± 0	89.5 ± 0.2	95.9 ± 0.1	93.3 ± 0.0	-	-
QRS	10^{-1}	44.2 ± 2.6	99.7 ± 0.1	29.9 ± 0.8	91.1 ± 0.2	100.0 ± 0	89.4 ± 0.2	96.0 ± 0.1	93.3 ± 0.0	0.37 ± 0.0049	0.77 ± 0.036
RWMH-base	10^{-1}	29.9 ± 45.6	70.0 ± 45.8	-	100.0 ± 0.0	7.6 ± 15.1	89.3 ± 4.7	96.1 ± 1.0	93.0 ± 1.0	-	-
RWMH-R-base	10^{-1}	28.0 ± 1.3	68.3 ± 1.5	34.9 ± 0.7	89.8 ± 0.2	100.0 ± 0	89.6 ± 0.2	95.8 ± 0.1	93.1 ± 0.1	-	-
IMH	10^{-3}	49.8 ± 2.1	99.8 ± 0.1	-	91.2 ± 0.2	97.4 ± 1.0	89.1 ± 0.2	95.9 ± 0.1	93.4 ± 0.0	-	-
IMH-R	10^{-3}	49.8 ± 1.6	99.8 ± 0.1	34.1 ± 0.8	90.9 ± 0.2	100.0 ± 0	89.1 ± 0.2	95.9 ± 0.1	93.4 ± 0.0	-	-
QRS	10^{-3}	49.4 ± 1.4	99.8 ± 0.1	34.1 ± 0.9	90.9 ± 0.2	100.0 ± 0	89.1 ± 0.2	95.9 ± 0.1	93.4 ± 0.0	0.012 ± 0.0055	0.0084 ± 0.0048

(b) 50% female scientists

Table 5.7: Further comparisons of samplers on an EBM with a pointwise constraint to include the word "amazing" in the sequence (a) and to produce 50% female scientist biographies (b). We do not compute perplexity for IMH and RWMH without reset as it does not yield i.i.d. samples. As noted, TVD and KL for MCMC methods are unknown (i.e. we have no way of estimating them). Where available, we show mean ± one standard deviation over 10 runs. *TVD and KL are estimated on independent sets of 10^6 samples.

For RWMH-R, we consider the chain:

$$
\begin{aligned}
&X^{(0)} \sim Q, \\
&\textbf{for } t = 1, \ldots, k-1 \\
&\qquad Y^{(t)} \sim \begin{cases} X^{(t-1)} - 1 & \text{with probability } \frac{1}{2} \\ X^{(t-1)} + 1 & \text{otherwise} \end{cases} \\
&\qquad X^{(t)} = \begin{cases} Y^{(t)} & \text{with probability } \min\left(1, \frac{P(Y^{(t)})}{P(X^{(t-1)})}\right) \\ X^{(t-1)} & \text{otherwise,} \end{cases}
\end{aligned}
$$

corresponding to the transition matrix:

$$
T_{x_1,x_0} = \begin{cases} \min\left(1, \frac{P(x_1)}{P(x_0)}\right) & \text{if } |x_1 - x_0| = 1 \\ 0 & \text{if } |x_1 - x_0| > 1 \\ 1 - \sum_{x_1 : |x_1 - x_0| = 1} T_{x_1,x_0} & \text{if } x_1 = x_0. \end{cases}
$$

Clearly, the performance of RWMH-R might be improved by using a different random walk: the ± 1 random walk is used here as it seems the simplest and most natural choice. Similarly to IMH-R, we can use this transition matrix to compute the exact distributions, and thus divergences for various chain lengths k.

Truncation errors

For all methods, to ensure practical computation, we truncate the distributions, working only with states $x < 50$. To assess the truncation error, we repeat the experiment, this time truncating with $x < 100$, and find that the largest relative error in KL, TVD or AR for any method is in the KL divergence of QRS for $\beta = 5$, which is

$$
\left| \frac{\mathrm{KL}(P, P_\beta)_{x<100} - \mathrm{KL}(P, P_\beta)_{x<50}}{\mathrm{KL}(P, P_\beta)_{x<100}} \right| = 1.48\cdots \times 10^{-10}.
$$

Results

Figure 5.8 presents the TVD and KL divergence as a function of AR, computed for parameter β in the range $[0.1, 5]$ for QRS, and for $k = 1, 2, \ldots 5$ iterations of IMH-R and RWMH-R. For acceptance rate AR $= 1$, all samplers give the same divergence, as they all return samples directly from the proposal Q. For each lower acceptance rate[12], the TVD of QRS is systematically lower than that of IMH-R, and the TVD of IMH-R is systematically lower than that of RWMH-R; and the

[12]The acceptance rates of QRS for $\beta = 2, 3, \ldots$ are nearly equal to $1/2, 1/3, \ldots$ because AR $= Z_\beta/\beta$ by Eq. 5.2, and for this choice of β, P and Q we have $Z_\beta \approx 1$.

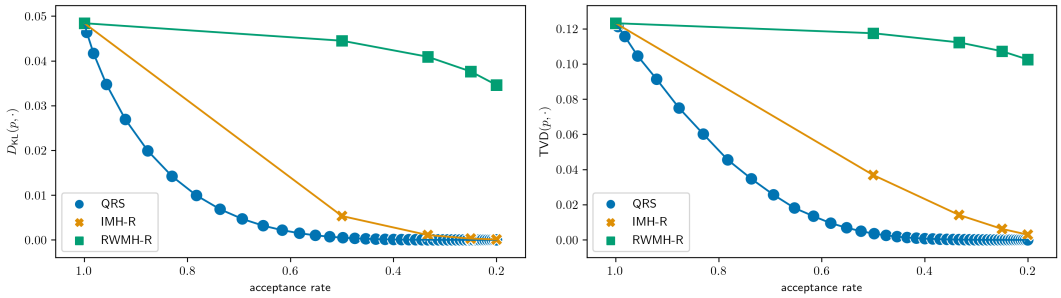


Figure 5.8: Quality-efficiency trade-off for QRS, IMH-R and RWMH-R samplers, when approximating the target distribution $P = \text{Poisson}(\lambda = 11)$. We use distribution $Q = \text{Poisson}(\lambda = 10)$ as proposal for QRS and IMH-R, and as initial distribution for RWMH-R. Quality is measured as $\text{TVD}(P, P_{\text{sampler}})$ in the left plot, and as $\text{KL}(P, P_{\text{sampler}})$ in the right plot. Efficiency is measured as the acceptance rate for QRS; and as the reciprocal of the number of sampling iterations for IMH-R and RWMH-R, which we call the "acceptance rate" in this context.

same holds for KL divergence. On investigating the ratios of the divergences, we find that these advantages increase as the acceptance rate decreases, and for $\text{AR} \approx 1/5$ we have

$$\frac{\text{TVD}(P, P_{\text{IMH-R},k=5})}{\text{TVD}(P, P_{\text{QRS},\beta=5})} = 2.24\cdots \times 10^3 \qquad \text{and} \qquad \frac{\text{KL}(P, P_{\text{IMH-R},k=5})}{\text{KL}(P, P_{\text{QRS},\beta=5})} = 3.60\cdots \times 10^2.$$

5.4 Related Work

The vast majority of approaches to approximate sampling from complex probability distributions have been based on MCMC. However, a few approaches have taken rejection sampling as their starting point. Like QRS, the method of rejection sampling chains (Tierney, 1992; Chib and Greenberg, 1995) does not require a global upper bound. This is a hybrid method that uses rejection sampling in a region satisfying a partial upper bound but combines it with IMH outside of that region to produce a Markov chain that converges to the correct stationary distribution. Caffo et al. (2002) propose empirical supremum rejection sampling, an algorithm that adaptively increases the β upper bound based on the maximum observed so far, with a focus on convergence in the limit rather than approximation quality.

Closer to our work, some researchers have observed before us that a partial bound β leads to the probability distribution presented in Eq. 5.1. Rejection control (Liu et al., 1998; Liu, 2004), in the context of particle filters, exploits this observation to accelerate the computation of an unbiased IS estimate of the expectation $\mathbb{E}_{x\sim p}[f(x)]$ in situations where computing $f(x)$ is expensive and must be done repeatedly. While the focus of that work was not to produce divergence

diagnostics, interestingly, on close examination, we find that one of their proofs (Liu, 2004) formulated a χ^2-divergence between a target distribution and a proxy distribution similar to P_β. Variational rejection sampling (Grover et al., 2018) uses a relaxation of Eq. 5.1 to better approximate the variational posterior in a variational inference setting. They aim to construct a differentiable sampler that can tighten the evidence lower bound, if additional computing power is available. The focus of our work is more general: we propose a generic sampler for which we can quantify a trade-off between sampling efficiency and approximation quality and we study the properties of this sampler.

While this work is concerned with generating samples from *discrete* EBMs, much research so far has been more concerned with *continuous* EBMs, in particular for applications in computer vision. Continuous EBMs have the advantage over discrete ones that it is possible to differentiate the EBM $p(x)$ with respect to x, and not only the approximating model π_θ relative to θ. This opens a range of optimized training techniques (see the survey by Song and Kingma 2021), including Langevin and Hamiltonian dynamics (Parisi, 1981; Duane et al., 1987), where the local Markov chain moves are informed by $\nabla_x \log p(x)$. While such techniques are not available for discrete EBMs, some recent efforts are trying to bridge the gap. For instance, *continuous relaxation* techniques (Han et al., 2020; Nishimura et al., 2020) relax the original discrete space into a continuous space, perform the sampling in this space, then map the samples back to the discrete space; while Grathwohl et al. (2021); Zhang et al. (2022); Rhodes and Gutmann (2022) sample directly in the discrete space, but inform the local moves of the chain through gradients computed in a continuous space. To the best of our knowledge, while Zhang et al. (2022) do experiment with "text infilling", the ability to replace some blanks in a given sentence by actual words, none of the above work has so far directly addressed text generation applications of the kind we have been considering here, in which the sample space is not only discrete, but composed of structured objects, namely word sequences of varying length, raising specific challenges.

Monte Carlo sampling techniques are popular for various NLP applications, in particular language modelling. For example, Miao et al. (2020) propose a sampler that mitigates poor estimation of probabilities due to overfitting. Deng et al. (2020) train globally normalised language models to combat negative effects of local normalisation, and use a form of sampling importance resampling (Rubin, 1987) to sample from the resulting EBM using an autoregressive proposal language model. Goyal et al. (2021) develop a Metropolis-Hastings algorithm to sample from masked language models. For controlled text generation Miao et al. (2019) propose a random-walk Metropolis-Hastings algorithm for sampling from an EBM that encodes sequence-level preferences on natural text. Their proposal distribution consists of local string editing operations on randomly selected words or positions. Zhang et al. (2020a) improve on this approach by making use of a tree-search algorithm to more efficiently explore the space of proposals, by al-

lowing several edits in a single step of the MH algorithm. In contrast to QRS, none of these approaches attempt to directly estimate the divergence between the sampler and the target EBM, but rely on unreliable proxy metrics.

5.5 Consequent Work

Block and Polyanskiy (2023) similarly study a relaxation of rejection sampling and provide various bounds on sampling complexity (*i.e.* the number of samples required from the proposal distribution) in order to obtain an f-divergence below a given ϵ, under various assumptions on the f-divergence itself. Kruszewski et al. (2023) use and implement our quasi-rejection sampling algorithm as part of the disco toolkit, a Python framework for imposing distribution-level constraints on pre-trained language models. Kim et al. (2024a, Appendix F) compare quasi-rejection sampling and an independent Metropolis-Hasting (IMH) sampler in a controlled text generation setting under binary constraints and find that QRS Pareto-dominates IMH in their setup, comparing a lower-bound on the KL divergence against inference time.

5.6 Discussion

QRS is a simple-yet-powerful technique: why has it not previously been promoted as a practical sampling method? One possible reason is the limited repertoire of *global* surrogates to complex distributions. This lack has strongly motivated the development of MCMC techniques which can exploit simple *local* proposals to compute transition probabilities between samples. This is now rapidly changing with advances in neural training, such as the impressive ability of pretrained language models to exploit simple prompts to orient their productions towards certain desired outcomes (but without formal guarantees); or with certain recent *generic* techniques, such as DPG, for fine-tuning autoregressive models towards arbitrary EBMs (but without the ability to totally reproduce them).

We believe that, given such global proposals, QRS can be a strong competitor to MCMC approaches. In particular, QRS has strong theoretical guarantees, not shared by these approaches: *i)* the ability to estimate, for any value of the β parameter, the divergence of the target EBM from the QRS sampler P_β, for any member of the large class of f-divergences, including TVD and KL, *ii)* the ability to *tune* the sampler to attain a desired quality-efficiency trade-off, and the intuitive nature of this tuning process thanks to the monotonic relationship between parameter β and the f-divergence (Theorem 5.1), *iii)* the existence of a simple, intuitive bound on the TVD between the QRS sampler and the target, provided by Theorem 5.2, and *iv)* the fact that QRS directly produces i.i.d. samples, rather than the correlated samples of a typical MCMC method.

Our experimental results show that QRS achieves strong results on the task of controlled text generation, where for instance, the sampler achieves excellent debiasing of the language model for acceptance rates in the range 10^{-1} to 10^{-3}. We show the versatility of the approach by applying it to different sources of global proposals: proposals over EBMs obtained by generic DPG-style fine-tuning; proposals based on handcrafted prompts; and for paraphrase generation, proposals based on round-trip translation.

Finally, when comparing QRS to variants of Metropolis-Hastings, we find QRS outperforms the local variants (RWMH and RWMH-R) and performs on par with the global variants (IMH and IMH-R), according to the proxy metrics available for all samplers. Our results on RWMH and RWMH-R, however, show how proxy metrics can be deceiving and do not give us a full picture of the approximation accuracy of our samplers. Therefore, we stress the importance of well-founded divergence measures and, in this chapter, have proposed a sampler for which we can estimate these. And while we did not experiment with minimum Bayes risk decoding, the sampling techniques experimented with in this chapter can be used to obtain samples from a large class of models, which provides all the necessary ingredients for sampling-based MBR decoding.

Chapter 6

Conclusion

In this dissertation, we explored the use of sampling from neural text generation models to study properties of sequence distributions, inform decoding algorithms of such properties, and obtain generations, using quasi-rejection sampling, from energy-based models that did not inherently permit efficient generation.

We saw that sequence distributions in neural machine translation models are such that high probability translations tend to be inadequate in some way, and stray from data statistics. We showed that a number of biases that were commonly thought to be inherent to neural machine translation did not show up as clearly in unbiased samples from the model. Samples, while varying greatly in quality, did exhibit properties of good translations on average and amongst a number of samples, typically translations could be found that performed well on evaluation metrics. High probability translations turned out to be, in fact, rather infrequent on average. Meaning that while neural machine translation models, by definition, put most probability mass on the mode, the mode itself and by consequence other high probability translations, as well as their properties, typically are not that frequent in the context of the entire sequence distribution.

Related and consequent works to ours show that these observations are not limited to neural machine translation. In fact, collectively these observations seem to suggest that inadequate modes and inadequate high probability sequences in general[1] are a fundamental property of current neural text generation models. As of yet, while some works have come up with hypotheses, there is no consensus amongst the scientific literature of what the underlying reason for this is, and whether we can resolve it. One problematic aspect is that it is not easy to quantify the extent with which neural text generation models exhibit the inadequacy of the mode. In order for the scientific community to assess the efficacy of potential solutions to the problem, developing more extensive methodology for evaluating current models is essential.

[1]We will still refer to this property of both the mode and high probability sequences as "the inadequacy of the mode".

However, we also showed that the inadequacy of the mode does not necessarily mean that the model is poor. In fact, while spread over many translations, high probability mass is collectively put on properties of good translations of the input. Motivated by this observation, we proposed an alternative decision rule for such sequence distributions based on minimum Bayes risk decoding, as well as a sampling-based approximation of it. The results in this dissertation, as well as those in the many consequent works that have built on our findings, show the effectiveness of a decoding strategy informed by the sequence distribution holistically. This is especially reinforced by the fact that, in our experiments and in those of some related works, it is found that better approximations to the decision rule generally lead to better generation quality.

A major obstacle for widespread adoption of sampling-based MBR decoding, however, is its increased computational cost. In our work, we proposed a more efficient linear time approximation to the decision rule using a coarse-to-fine strategy, but in consequent works many greatly effective efficiency optimisations have been proposed, as well as strategies to distil the gains of MBR decoding through model fine-tuning. Nonetheless, future work in this direction could still prove fruitful, making sampling-based MBR more practical and testing its scaling limits.

More consideration into the chosen utility function in MBR decoding could also prove fruitful. In our work, and most other works on sampling-based MBR decoding, utilities have simply mimicked the evaluation metrics that work well for a particular application. The choice of utility is important for performance, as research in recent years has shown that with the advent of neural evaluation metrics, MBR has also greatly benefitted with great generation quality performance boosts. Yet, carefully crafted utility functions could potentially better exploit any discovered structure latent within the sequence distribution, or exploit particular aspects important to a given text generation task.

Finally, we looked at the flexible definitions of probability distributions permitted by energy-based models, at the cost of efficient inference. We saw how sampling techniques such as Markov chain Monte Carlo and quasi-rejection sampling allow us to draw samples from such distributions by using autoregressive neural text generation models to produce proposal generations. Quasi-rejection sampling in particular allowed us to make an informed trade-off between f-divergence with the target distribution and sampling efficiency. We explored various ways of obtaining global proposal distributions for text generation by either fine-tuning, in-context learning or model combination. With large language models becoming more capable by the day, proposal distributions obtained with in-context learning and model combinations make it increasingly more accessible to define energy-based models to one's desire and obtain reasonably efficient generations from them.

All in all, a sampling-based exploration of neural text generation models has proven fruitful in uncovering inadequacies of such models, guiding better decision-making, and obtaining generations from non-autoregressive variants. I hope that

the observations and developments in this dissertation and the works covered within it are insightful to the broader research community in development, evaluation and use of neural text generation models, and that they may inspire further advancements to our understanding and developments of the probabilistic models behind the automated production of natural language.

Bibliography

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2442–2452, Berlin, Germany. Association for Computational Linguistics.

Abhishek Arun, Chris Dyer, Barry Haddow, Phil Blunsom, Adam Lopez, and Philipp Koehn. 2009. Monte Carlo inference and maximization for phrase-based translation. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 102–110, Boulder, Colorado. Association for Computational Linguistics.

Wilker Aziz, Marc Dymetman, and Sriram Venkatapathy. 2013. Investigations in exact inference for hierarchical translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 472–483, Sofia, Bulgaria. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR, 2015*, San Diego, USA.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 597–604, Ann Arbor, Michigan. Association for Computational Linguistics.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1171–1179. Curran Associates, Inc.

James O Berger. 1985. *Statistical decision theory and Bayesian analysis; 2nd ed.* Springer Series in Statistics. Springer, New York.

Sumanta Bhattacharyya, Amirmohammad Rooshenas, Subhajit Naskar, Simeng Sun, Mohit Iyyer, and Andrew McCallum. 2021. Energy-based reranking: Improving neural machine translation using energy-based models. 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 4528–4537, Online. Association for Computational Linguistics.

Peter J Bickel and Kjell A Doksum. 1977. *Mathematical statistics: basic ideas and selected topics*. Holden-Day Inc., Oakland, CA, USA.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Frédéric Blain, Lucia Specia, and Pranava Madhyastha. 2017. Exploring hypotheses spaces in neural machine translation. *Asia-Pacific Association for Machine Translation (AAMT), editor, Machine Translation Summit XVI. Nagoya, Japan.*

David M Blei. 2014. Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application*, 1:203–232.

Adam Block and Yury Polyanskiy. 2023. The sample complexity of approximate rejection sampling with applications to smoothed online learning. In *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 228–273. PMLR.

Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. 2009. A Gibbs sampler for phrasal synchronous grammar induction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 782–790, Suntec, Singapore. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016a. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016b. Results of the WMT16 metrics shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 199–231, Berlin, Germany. Association for Computational Linguistics.

Sebastian Borgeaud and Guy Emerson. 2020. Leveraging sentence similarity in natural language generation: Improving beam search using range voting. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 97–109, Online. Association for Computational Linguistics.

Bottou. 1991. Une approche théorique de l'apprentissage connexioniste; applications à la reconnaissance de la parole.

Léon Bottou and Yann L. Cun. 2004. Large scale online learning. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 217–224. MIT Press.

Stephen P. Brooks and Andrew Gelman. 1998. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, P. Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, Tom Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *arXiv*, abs/2005.14165. GPT-3.

Brian S. Caffo, James G. Booth, and A. C. Davison. 2002. Empirical supremum rejection sampling. *Biometrika*, 89(4):745–754.

Clément L. Canonne. 2020. *A Survey on Distribution Testing: Your Data is Big. But is it Blue?* Number 9 in Graduate Surveys. Theory of Computing Library.

Sharon A. Caraballo and Eugene Charniak. 1996. Figures of merit for best-first probabilistic chart parsing. In *Conference on Empirical Methods in Natural Language Processing*.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Sourav Chatterjee and Persi Diaconis. 2018. The sample size required in importance sampling. *The Annals of Applied Probability*, 28(2):1099–1135.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367, Baltimore, Maryland, USA. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Julius Cheng and Andreas Vlachos. 2023. Faster minimum Bayes risk decoding with confidence-based pruning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12473–12480, Singapore. Association for Computational Linguistics.

Siddhartha Chib and Edward Greenberg. 1995. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Eldan Cohen and J. Christopher Beck. 2019. Empirical analysis of beam search performance degradation in neural sequence models. In *ICML*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Mary Kathryn Cowles and Bradley P. Carlin. 1996. Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434):883–904.

I. Csiszar. 1975. *I*-Divergence Geometry of Probability Distributions and Minimization Problems. *The Annals of Probability*, 3(1):146 – 158.

Imre Csiszár and Paul C. Shields. 2004. Information theory and statistics: A tutorial. *Commun. Inf. Theory*, 1(4):417–528.

John DeNero, Alexandre Bouchard-Côté, and Dan Klein. 2008. Sampling alignment structure under a Bayesian translation model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 314–323, Honolulu, Hawaii. Association for Computational Linguistics.

John DeNero, David Chiang, and Kevin Knight. 2009. Fast consensus decoding over translation forests. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 567–575, Suntec, Singapore. Association for Computational Linguistics.

Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. 2020. Residual energy-based models for text generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of WMT, 2011*, pages 85–91, Edinburgh, Scotland.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. 1987. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216 – 222.

Bryan Eikema. 2024. The effect of generalisation on the inadequacy of the mode. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*, pages 87–92, St Julians, Malta. Association for Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2019. Auto-encoding variational neural machine translation. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 124–141, Florence, Italy. Association for Computational Linguistics.

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

Bryan Eikema and Wilker Aziz. 2021. Sampling-based approximations to minimum bayes risk decoding for neural machine translation.

Bryan Eikema and Wilker Aziz. 2022. Sampling-based approximations to minimum Bayes risk decoding for neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10978–10993, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Bryan Eikema, Germán Kruszewski, Christopher R Dance, Hady Elsahar, and Marc Dymetman. 2022. An approximate sampler for energy-based models with divergence diagnostics. *Transactions on Machine Learning Research*.

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

Mara Finkelstein and Markus Freitag. 2024. MBR and QE finetuning: Training-time distillation of the best and most expensive decoding methods. In *The Twelfth International Conference on Learning Representations*.

Martina Forster, Clara Meister, and Ryan Cotterell. 2021. Searching for search errors in neural morphological inflection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1388–1394, Online. Association for Computational Linguistics.

Markus Freitag, Behrooz Ghorbani, and Patrick Fernandes. 2023. Epsilon sampling rocks: Investigating sampling strategies for minimum bayes risk decoding for machine translation.

Markus Freitag, 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.

Dani Gamerman and Hedibert F. Lopes. 2006. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Chapman and all/CRC.

Xavier Garcia, Yamini Bansal, Colin Cherry, George Foster, Maxim Krikun, Melvin Johnson, and Orhan Firat. 2023. The unreasonable effectiveness of few-shot learning for machine translation. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. 2013. Bayesian data analysis, third edition.

Andrew Gelman and Donald B. Rubin. 1992. Inference from iterative simulation using multiple sequences. *Statistical science*, pages 457–472.

Vaibhava Goel and William J. Byrne. 2000. Minimum bayes-risk automatic speech recognition. *Comput. Speech Lang.*, 14(2):115–135.

Joshua Goodman. 1996. Parsing algorithms and metrics. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 177–183, Santa Cruz, California, USA. Association for Computational Linguistics.

Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. 2021. Exposing the implicit energy networks behind masked language models via Metropolis-Hastings. *CoRR*, abs/2106.02736.

Will Grathwohl, Kevin Swersky, Milad Hashemi, David Duvenaud, and Chris Maddison. 2021. Oops I took a gradient: Scalable sampling for discrete distributions. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3831–3841. PMLR.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. In *Proceedings of ICML, Workshop on Representation Learning*.

Aditya Grover, Ramki Gummadi, Miguel Lazaro-Gredilla, Dale Schuurmans, and Stefano Ermon. 2018. Variational rejection sampling. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 823–832. PMLR.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of*

the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 1053–1062, Valencia, Spain. Association for Computational Linguistics.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. 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 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Jun Han, Fan Ding, Xianglong Liu, Lorenzo Torresani, Jian Peng, and Qiang Liu. 2020. Stein variational inference for discrete distributions. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4563–4572. PMLR.

P. E. Hart, N. J. Nilsson, and B. Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

W. Keith Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

John Hewitt, Christopher Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427, Abu Dhabi, United Arab Emirates. 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.

Liang Huang, Kai Zhao, and Mingbo Ma. 2017. When to finish? optimal beam search for neural text generation (modulo beam size). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2134–2139, Copenhagen, Denmark. Association for Computational Linguistics.

T. Jaeger and Roger Levy. 2007. Speakers optimize information density through syntactic reduction. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

E. T. Jaynes. 1957. Information theory and statistical mechanics. *Phys. Rev.*, 106(4):620–630.

Yuu Jinnai and Kaito Ariu. 2024. Hyperparameter-free approach for faster minimum Bayes risk decoding. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8547–8566, Bangkok, Thailand. Association for Computational Linguistics.

Daniel D. Johnson, Daniel Tarlow, and Christian Walder. 2023. R-u-SURE? Uncertainty-aware code suggestions by maximizing utility across random user intents. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15262–15306. PMLR.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. 2021. A distributional approach to controlled text generation. In *International Conference on Learning Representations*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Minbeom Kim, Thibaut Thonet, Jos Rozen, Hwaran Lee, Kyomin Jung, and Marc Dymetman. 2024a. Guaranteed generation from large language models.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024b. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. 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.

Germán Kruszewski, Jos Rozen, and Marc Dymetman. 2023. disco: a toolkit for distributional control of generative models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 144–160, Toronto, Canada. Association for Computational Linguistics.

S. Kullback and M. A. Khairat. 1966. A Note on Minimum Discrimination Information. *The Annals of Mathematical Statistics*, 37(1):279 – 280.

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv:1903.00802*.

Shankar Kumar and William Byrne. 2002. Minimum Bayes-risk word alignments of bilingual texts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 140–147. Association for Computational Linguistics.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Shankar Kumar, Wolfgang Macherey, Chris Dyer, and Franz Och. 2009. Efficient minimum error rate training and minimum Bayes-risk decoding for translation hypergraphs and lattices. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 163–171, Suntec, Singapore. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. 2006. A Tutorial on Energy-Based Learning. In *Predicting Structured Data*. MIT Press.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2019. Hallucinations in neural machine translation.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Jiwei Li and Dan Jurafsky. 2016. Mutual information and diverse decoding improve neural machine translation. *arXiv preprint arXiv:1601.00372*.

Friedrich Liese and Igor Vajda. 2006. On divergences and informations in statistics and information theory. *IEEE Transactions on Information Theory*, 52(10):4394–4412.

Jun S. Liu. 2004. *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer New York.

Jun S. Liu, Rong Chen, and Wing Hung Wong. 1998. Rejection control and sequential importance sampling. *Journal of the American Statistical Association*, 93(443):1022–1031.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the WMT18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 671–688, Belgium, Brussels. Association for Computational Linguistics.

Matouš Macháček and Ondřej Bojar. 2014. Results of the WMT14 metrics shared task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 293–301, Baltimore, Maryland, USA. Association for Computational Linguistics.

David J. C. MacKay. 2003. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893, Valencia, Spain. Association for Computational Linguistics.

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)*, pages 2173–2185, Online. Association for Computational Linguistics.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121.

Clara Meister, Gian Wiher, Tiago Pimentel, and Ryan Cotterell. 2022. On the probability–quality paradox in language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 36–45, Dublin, Ireland. Association for Computational Linguistics.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092.

Ning Miao, Yuxuan Song, Hao Zhou, and Lei Li. 2020. Do you have the right scissors? Tailoring pre-trained language models via Monte-Carlo methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 3436–3441. Association for Computational Linguistics.

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, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6834–6842. AAAI Press.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Mathias Müller, Annette Rios, and Rico Sennrich. 2020. Domain robustness in neural machine translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 151–164, Virtual. Association for Machine Translation in the Americas.

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

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

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. 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 5356–5371, Online. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2:*

Shared Task Papers, Day 1), pages 314–319, Florence, Italy. Association for Computational Linguistics.

Akihiko Nishimura, David B. Dunson, and Jianfeng Lu. 2020. Discontinuous Hamiltonian Monte Carlo for discrete parameters and discontinuous likelihoods. *Biometrika*, 107(2):365–380.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3956–3965, Stockholmsmässan, Stockholm Sweden. PMLR.

Art B. Owen. 2013. Importance Sampling. In *Monte Carlo Theory, Methods and Examples*, chapter 9. Unpublished Lecture Notes.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL, 2002*, pages 311–318, Philadelphia, USA.

G. Parisi. 1981. Correlation Functions and Computer Simulations. *Nuclear Physics B*, 180:378.

Tetiana Parshakova, Jean-Marc Andreoli, and Marc Dymetman. 2019a. Distributional reinforcement learning for energy-based sequential models. In *Proceedings of the OptRL Workshop (Optimization Foundations for Reinforcement Learning) at NeurIPS 2019*.

Tetiana Parshakova, Jean-Marc Andreoli, and Marc Dymetman. 2019b. Global Autoregressive Models for Data-Efficient Sequence Learning. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 900–909, Hong Kong, China. Association for Computational Linguistics.

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

Ben Peters, Vlad Niculae, and André F. T. Martins. 2019. Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1504–1519, Florence, Italy. Association for Computational Linguistics.

Yury Polyanskiy. 2019. MIT Information Theoretic Methods in Statistics and Computer Science, Lecture 1: f-Divergences.

Yury Polyanskiy and Yihong Wu. 2017. MIT, Yale and UIUC, Lecture Notes on Information Theory.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Tiberiu Popoviciu. 1935. Sur les équations algébriques ayant toutes leurs racines réelles. *Mathematica*, 9(129-145):20.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Ricardo Rei, 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. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Benjamin Rhodes and Michael U. Gutmann. 2022. Enhanced gradient-based MCMC in discrete spaces. *Transactions on Machine Learning Research*.

Darcey Riley and David Chiang. 2022. A continuum of generation tasks for investigating length bias and degenerate repetition. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 426–440, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407.

Christian P. Robert and George Casella. 2004. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Ronald Rosenfeld. 1996. A maximum entropy approach to adaptive statistical language modelling. *Comput. Speech Lang.*, 10(3):187–228.

Vivekananda Roy. 2020. Convergence diagnostics for Markov chain Monte Carlo. *Annual Review of Statistics and Its Application*, 7(1):387–412.

Donald B. Rubin. 1987. Comment on "The calculation of posterior distributions by data augmentation". *Journal of the American Statistical Association*, 82(398):543–546.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

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

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.

Raphael Shu and Hideki Nakayama. 2017. Later-stage minimum bayes-risk decoding for neural machine translation. *CoRR*, abs/1704.03169.

Raphael Shu and Hideki Nakayama. 2018. Improving beam search by removing monotonic constraint for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 339–344, Melbourne, Australia. Association for Computational Linguistics.

Khalil Sima'an. 2003. On maximizing metrics for syntactic disambiguation. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 183–194, Nancy, France.

Noah A. Smith. 2011. *Linguistic Structure Prediction*. Morgan and Claypool.

Yang Song and Diederik P. Kingma. 2021. How to train your energy-based models. *CoRR*, abs/2101.03288.

Pavel Sountsov and Sunita Sarawagi. 2016. Length bias in encoder decoder models and a case for global conditioning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1525, Austin, Texas. 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, Adrià de Gispert, Eva Hasler, and Bill Byrne. 2017. Neural machine translation by minimising the Bayes-risk with respect to syntactic translation lattices. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 362–368, Valencia, Spain. Association for Computational Linguistics.

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

Miloš Stanojević, Amir Kamran, Philipp Koehn, and Ondřej Bojar. 2015. Results of the WMT15 metrics shared task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 256–273, Lisbon, Portugal. Association for Computational Linguistics.

Miloš Stanojević and Khalil Sima'an. 2014. Fitting sentence level translation evaluation with many dense features. In *Proceedings of the 2014 Conference on*

Empirical Methods in Natural Language Processing (EMNLP), pages 202–206, Doha, Qatar. Association for Computational Linguistics.

Miloš Stanojević and Khalil Sima'an. 2015. Reordering grammar induction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 44–54, Lisbon, Portugal. Association for Computational Linguistics.

Miloš Stanojević and Mark Steedman. 2020. Max-Margin Incremental CCG Parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Mitchell Stern, Daniel Fried, and Dan Klein. 2017. Effective inference for generative neural parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1695–1700, Copenhagen, Denmark. Association for Computational Linguistics.

Andreas Stolcke, Yochai Konig, and Mitchel Weintraub. 1997. Explicit word error minimization in n-best list rescoring. In *Fifth European Conference on Speech Communication and Technology*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *NIPS, 2014*, pages 3104–3112. Montreal, Canada.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2023. Follow the wisdom of the crowd: Effective text generation via minimum Bayes risk decoding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4265–4293, Toronto, Canada. Association for Computational Linguistics.

Christian Szegedy, 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.

Terence Tao. 2008. The strong law of large numbers.

Luke Tierney. 1992. Exploring posterior distributions using Markov chains. Technical report, Minnesota University, Minneapolis Department of Statistics.

Roy Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. 2008. Lattice Minimum Bayes-Risk decoding for statistical machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 620–629, Honolulu, Hawaii. Association for Computational Linguistics.

Jannis Vamvas and Rico Sennrich. 2024. Linear-time minimum Bayes risk decoding with reference aggregation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 790–801, Bangkok, Thailand. Association for Computational Linguistics.

Vladimir Vapnik. 1998. Statistical learning theory Wiley. *New York*, 1:624.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 6000–6010.

Dootika Vats, James M. Flegal, and Galin L. Jones. 2019. Multivariate output analysis for Markov chain Monte Carlo. *Biometrika*.

Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. 2021. Rank-normalization, folding, and localization: an improved R for assessing convergence of MCMC (with discussion). *Bayesian Analysis*, 16(2):667–718.

John von Neumann. 1963. Various techniques used in connection with random digits. *John von Neumann, Collected Works*, 5:768–770.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, Online. Association for Computational Linguistics.

Larry Wasserman. 2010. *All of Statistics: A Concise Course in Statistical Inference*. Springer, New York.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, Austin, Texas. Association for Computational Linguistics.

Ian Wu, Patrick Fernandes, Amanda Bertsch, Seungone Kim, Sina Pakazad, and Graham Neubig. 2024. Better instruction-following through minimum bayes risk.

Jianhao Yan, Jin Xu, Fandong Meng, Jie Zhou, and Yue Zhang. 2024. DC-MBR: Distributional cooling for minimum Bayesian risk decoding. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4423–4437, Torino, Italia. ELRA and ICCL.

Guangyu Yang, Jinghong Chen, Weizhe Lin, and Bill Byrne. 2024. Direct preference optimization for neural machine translation with minimum Bayes risk decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 391–398, Mexico City, Mexico. Association for Computational Linguistics.

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.

Davis Yoshida, Kartik Goyal, and Kevin Gimpel. 2024. MAP's not dead yet: Uncovering true language model modes by conditioning away degeneracy. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16164–16215, Bangkok, Thailand. Association for Computational Linguistics.

Hao Zhang and Daniel Gildea. 2008. Efficient multi-pass decoding for synchronous context free grammars. In *Proceedings of ACL-08: HLT*, pages 209–217, Columbus, Ohio. Association for Computational Linguistics.

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

Maosen Zhang, Nan Jiang, Lei Li, and Yexiang Xue. 2020a. 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.

Ruqi Zhang, Xingchao Liu, and Qiang Liu. 2022. A langevin-like sampler for discrete distributions. In *International Conference on Machine Learning*, pages 26375–26396. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational*

Linguistics, pages 4334–4343, Florence, Italy. Association for Computational Linguistics.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1097–1100. ACM.

Samenvatting

Neurale tekstgeneratiemodellen staan aan de basis van de meeste hedendaagse systemen voor natuurlijke taalverwerking (*natural language processing*, NLP). In de afgelopen jaren zijn de prestaties van NLP-systemen in een stroomversnelling geraakt dankzij innovaties op het gebied van neurale netwerkarchitecturen en trainingsparadigma's zoals attentiemechanismen, Transformers en pre-trainingen data-augmentatiestrategieën. In de kern is de probabilistische formulering van deze modellen echter niet veranderd sinds de oorspronkelijke neurale tekstgeneratiemodellen werden geïntroduceerd.

De probabilistische aard van deze modellen wordt daarentegen vaak snel vergeten nadat het model getraind is. Voor veel natuurlijke taalverwerkingstaken, zoals machinevertaling, worden bijvoorbeeld deterministische zoekalgoritmen gebruikt om een enkele "beste" generatie uit het model te halen. In dergelijke gevallen wordt het probabilistische model alleen gebruikt om gedeeltelijke generaties een score toe te wijzen om zo de hoogst scorende complete generatie te vinden, oftewel de sequentie met tekstsymbolen met de hoogste kans binnen de kansverdeling, ook wel bekend als de modus van de kansverdeling. Dit gaat ervan uit dat neurale tekstgeneratiemodellen datasoortige generaties inderdaad op hun conditionele modi plaatsen. Een observatie die in literatuur wordt gemaakt in verscheidene textgeneratietaken suggereert echter dat dit niet het geval is.

Een beter begrip van de kansverdelingen over tekst die onze neurale netwerken voorspellen, stelt ons in staat beter geïnformeerde beslissingen te nemen over welke generatie-strategie geschikt is voor onze modellen. *Sampling*, het genereren van tekst zodat dit de kansverdeling volgt, is een natuurlijke manier om de eigenschappen van de kansverdelingen die door neurale netwerken worden voorspeld te verkennen. Door de eigenschappen van dergelijke *samples* te bestuderen, onderzoeken we indirect ook de eigenschappen van de kansverdelingen waarmee we werken. Samples kunnen ook worden gebruikt om generatie-algoritmen te informeren en voor sommige taken zijn samples zelfs de voorkeursuitvoer van het model.

In deze dissertatie zullen we het gebruik van sampling verkennen om onze tekstgeneratiemodellen beter te begrijpen en om generatie-algoritmen te informeren. We zullen algemeen bekende problemen van tekstgeneratiemodellen bekijken door de lens van een dergelijke probabilistische verkenning en een nieuw perspectief bieden op hun mogelijke oorzaken. We gebruiken deze inzichten om een generatie-algoritme op basis van sampling te ontwikkelen, geïnspireerd door risicominimisatiestrategieën. Daarnaast ontwikkelen we geheel nieuwe samplingstrategieën om samples te verkrijgen van willekeurige verdelingen waarbij een per-token- (*i.e.* autoregressieve) factorisatie niet bestaat.

Abstract

Neural text generation models are at the basis of most modern-day natural language processing (NLP) systems. In recent years many important innovations to neural network architectures and training paradigms have appeared such as attention mechanisms, Transformers, and pre-training and data augmentation strategies that have accelerated the performance of NLP systems tremendously. At their core, however, these models have not changed their probabilistic formulation since the original neural text generation models were first described.

The probabilistic nature of these models, however, is often quickly forgotten after the model has been trained. For many natural language processing tasks such as machine translation, for example, deterministic search algorithms are employed to extract a single "best" generation from the model. In such cases the probabilistic model is only used to score partial subsequences during generation as to find the highest scoring sequence, *i.e.* the sequence with highest probability under the distribution, also known as the mode of the distribution. This assumes that neural text generation models indeed put data-like sequences at its modes. A well-known observation across text generation tasks, however, seems to suggest that the highest probability generations of neural text generation models are not at all data-like.

A better understanding of the sequence distributions that our neural networks predict allows us to make better-informed decisions about what kind of generation strategy is appropriate for our models. Sampling is a natural way to explore the properties of the sequence distributions predicted by neural networks. By studying the properties of such samples we indirectly also study the properties of the sequence distributions we are working with. Samples can also be used to inform generation algorithms and for some tasks samples even are the outputs of choice from the model.

In this dissertation we will explore the use of sampling to better understand neural text generation models and in order to inform decoding algorithms. We will view commonly known pathologies and biases of neural text generation models

under the lens of such a probabilistic exploration and provide a new perspective on their potential causes. We use these insights to propose and iterate on a sampling-based decoding algorithm inspired by risk minimisation strategies, as well as develop new sampling strategies altogether to sample from arbitrary distributions where a per-token, *i.e.* autoregressive, factorisation does not exist.

Titles in the ILLC Dissertation Series:

ILLC DS-2020-17: **Francesca Zaffora Blando**
Patterns and Probabilities: A Study in Algorithmic Randomness and Computable Learning

ILLC DS-2021-01: **Yfke Dulek**
Delegated and Distributed Quantum Computation

ILLC DS-2021-02: **Elbert J. Booij**
The Things Before Us: On What it Is to Be an Object

ILLC DS-2021-03: **Seyyed Hadi Hashemi**
Modeling Users Interacting with Smart Devices

ILLC DS-2021-04: **Sophie Arnoult**
Adjunction in Hierarchical Phrase-Based Translation

ILLC DS-2021-05: **Cian Guilfoyle Chartier**
A Pragmatic Defense of Logical Pluralism

ILLC DS-2021-06: **Zoi Terzopoulou**
Collective Decisions with Incomplete Individual Opinions

ILLC DS-2021-07: **Anthia Solaki**
Logical Models for Bounded Reasoners

ILLC DS-2021-08: **Michael Sejr Schlichtkrull**
Incorporating Structure into Neural Models for Language Processing

ILLC DS-2021-09: **Taichi Uemura**
Abstract and Concrete Type Theories

ILLC DS-2021-10: **Levin Hornischer**
Dynamical Systems via Domains: Toward a Unified Foundation of Symbolic and Non-symbolic Computation

ILLC DS-2021-11: **Sirin Botan**
Strategyproof Social Choice for Restricted Domains

ILLC DS-2021-12: **Michael Cohen**
Dynamic Introspection

ILLC DS-2021-13: **Dazhu Li**
Formal Threads in the Social Fabric: Studies in the Logical Dynamics of Multi-Agent Interaction

ILLC DS-2021-14: **Álvaro Piedrafita**
On Span Programs and Quantum Algorithms

ILLC DS-2022-01: **Anna Bellomo**
Sums, Numbers and Infinity: Collections in Bolzano's Mathematics and Philosophy

ILLC DS-2022-02: **Jan Czajkowski**
Post-Quantum Security of Hash Functions

ILLC DS-2022-03: **Sonia Ramotowska**
Quantifying quantifier representations: Experimental studies, computational modeling, and individual differences

ILLC DS-2022-04: **Ruben Brokkelkamp**
How Close Does It Get?: From Near-Optimal Network Algorithms to Suboptimal Equilibrium Outcomes

ILLC DS-2022-05: **Lwenn Bussière-Carae**
No means No! Speech Acts in Conflict

ILLC DS-2022-06: **Emma Mojet**
Observing Disciplines: Data Practices In and Between Disciplines in the 19th and Early 20th Centuries

ILLC DS-2022-07: **Freek Gerrit Witteveen**
Quantum information theory and many-body physics

ILLC DS-2023-01: **Subhasree Patro**
Quantum Fine-Grained Complexity

ILLC DS-2023-02: **Arjan Cornelissen**
Quantum multivariate estimation and span program algorithms

ILLC DS-2023-03: **Robert Paßmann**
Logical Structure of Constructive Set Theories

ILLC DS-2023-04: **Samira Abnar**
Inductive Biases for Learning Natural Language

ILLC DS-2023-05: **Dean McHugh**
Causation and Modality: Models and Meanings

ILLC DS-2023-06: **Jialiang Yan**
Monotonicity in Intensional Contexts: Weakening and: Pragmatic Effects under Modals and Attitudes

ILLC DS-2023-07: **Yiyan Wang**
Collective Agency: From Philosophical and Logical Perspectives

ILLC DS-2023-08: **Lei Li**
Games, Boards and Play: A Logical Perspective

ILLC DS-2023-09: **Simon Rey**
Variations on Participatory Budgeting

ILLC DS-2023-10: **Mario Giulianelli**
Neural Models of Language Use: Studies of Language Comprehension and Production in Context

ILLC DS-2023-11: **Guillermo Menéndez Turata**
Cyclic Proof Systems for Modal Fixpoint Logics

ILLC DS-2023-12: **Ned J.H. Wontner**
Views From a Peak: Generalisations and Descriptive Set Theory

ILLC DS-2024-01: **Jan Rooduijn**
Fragments and Frame Classes: Towards a Uniform Proof Theory for Modal Fixed Point Logics

ILLC DS-2024-02: **Bas Cornelissen**
Measuring musics: Notes on modes, motifs, and melodies

ILLC DS-2024-03: **Nicola De Cao**
Entity Centric Neural Models for Natural Language Processing

ILLC DS-2024-04: **Ece Takmaz**
Visual and Linguistic Processes in Deep Neural Networks: A Cognitive Perspective

ILLC DS-2024-05: **Fatemeh Seifan**
Coalgebraic fixpoint logic Expressivity and completeness result

ILLC DS-2024-06: **Jana Sotáková**
Isogenies and Cryptography

ILLC DS-2024-07: **Marco Degano**
Indefinites and their values

ILLC DS-2024-08: **Philip Verduyn Lunel**
Quantum Position Verification: Loss-tolerant Protocols and Fundamental Limits

ILLC DS-2024-09: **Rene Allerstorfer**
Position-based Quantum Cryptography: From Theory towards Practice

ILLC DS-2024-10: **Willem Feijen**
Fast, Right, or Best? Algorithms for Practical Optimization Problems

ILLC DS-2024-11: **Daira Pinto Prieto**
Combining Uncertain Evidence: Logic and Complexity

ILLC DS-2024-12: **Yanlin Chen**
On Quantum Algorithms and Limitations for Convex Optimization and Lattice Problems

ILLC DS-2024-13: **Jaap Jumelet**
Finding Structure in Language Models

ILLC DS-2025-01: **Julian Chingoma**
On Proportionality in Complex Domains

ILLC DS-2025-02: **Dmitry Grinko**
Mixed Schur-Weyl duality in quantum information

ILLC DS-2025-03: **Rochelle Choenni**
Multilinguality and Multiculturalism: Towards more Effective and Inclusive Neural Language Models

ILLC DS-2025-04: **Aleksi Anttila**
Not Nothing: Nonemptiness in Team Semantics

ILLC DS-2025-05: **Niels M. P. Neumann**
Adaptive Quantum Computers: decoding and state preparation

ILLC DS-2025-06: **Alina Leidinger**
Towards Language Models that benefit us all: Studies on stereotypes, robustness, and values

ILLC DS-2025-07: **Zhi Zhang**
Advancing Vision and Language Models through Commonsense Knowledge, Efficient Adaptation and Transparency

ILLC DS-2025-08: **Sophie Klumper**
The Gap and the Gain: Improving the Approximate Mechanism Design Frontier in Constrained Environments

ILLC DS-2026-01: **Bryan Eikema**
A Sampling-Based Exploration of Neural Text Generation Models