GSD: GENERALIZED STOCHASTIC DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Although substantial progress has been made in various text generation tasks, there remains a vast gap between current generations and human languages. One reason is that virtually all decoding methods currently developed are pragmatic to address the text degeneration problem, which exists in both deterministic and stochastic decoding algorithms. So, why text generated from these algorithms are degenerated? What is the critical difference between these algorithms? Moreover, is it possible to design a generalized framework where existing decoding algorithms can be naturally connected, uniformly described, and mutually inspired?

In this paper, we try to explore answers to these intriguing questions. Correctly, we propose a generalized decoding framework that can be used to describe and connect existing popular decoding algorithms. Based on the framework, we propose a novel implementation with distinctive design from existing decoding algorithms. As far as we know, this is the first work trying to propose a generalized framework to bridge these decoding algorithms using formal theorems and concrete implementations. By setting up different conditions, our framework provides infinite space to develop new decoding algorithms. Experiments show that text produced by our method is closest to the characteristics of human languages. Source code and the generated text can be accessed from `https://github.com/ginoailab/gsd.git`.

1 INTRODUCTION

Benefiting from large-scale pre-trained language models (Radford et al., 2019; Lewis et al., 2020; Tseng et al., 2020; Sun & Yang, 2020), considerable advances have been observed in many natural language generation (NLG) tasks. Nevertheless, given these pre-trained models, only limited decoding strategies are available to use, such as Greedy, Sampling, Beam Search, Top-k (Fan et al., 2018; Holtzman et al., 2018), and the recently proposed nucleus sampling (i.e., top-p) (Holtzman et al., 2020). These decoding algorithms are essential because virtually all text generation tasks need them to transfer inferred predictions to successive text (Basu et al., 2021). However, few of these decoding algorithms can resist the risk of text degeneration. That is, text produced by these algorithms exhibits quite different characteristics from those used by humans, which contains many generic words, repeated loops, and irrelevant sentences.

This problem has been confirmed by many previous works (Holtzman et al., 2020; Welleck et al., 2020b). Some researchers deem that it is ineluctable because generators do empower high probabilities to the correct words, but locally, the highest probability can only be assigned to the text with generic, repetitive, or meaningless phrases. However, we suppose that this problem is triggered by the distinct manners between humans and the decoding algorithms about how text is produced.

Concretely, it is very odd for humans to be told to utter more diversely or more fluently. Diverse or not, it usually depends on personal habits of language usage. Similarly, fluency or not, it is generally determined by people's familiarity with that language (patients with language disorders are excluded). For most decoding algorithms that are currently popular and widely used since virtually all of them must be configured with at least one hyper-parameter before being available to use. Then, these hyper-parameters become the only exceptions in the decoding system that are not controlled by the algorithms, which brings many uncertainties to the decoding system.

Based on these observations, we propose a novel decoding algorithm with a hyper-parameter-free design. All behaviors of our decoding method are automatically controlled and dynamically adjusted

by the algorithm itself, which is distinctive from all popular decoding algorithms currently developed. We name this novel method as Intrinsic Decoding. Detailed comparisons between Intrinsic Decoding and existing algorithms will be described in the following sections.

More importantly, we also propose a generalized decoding framework (GSD) to connect these decoding algorithms in formal mathematical theorems, including both existing decoding methods and our proposed Intrinsic Decoding. The relationships between GSD and these algorithms can be vividly expressed in Figure 1.

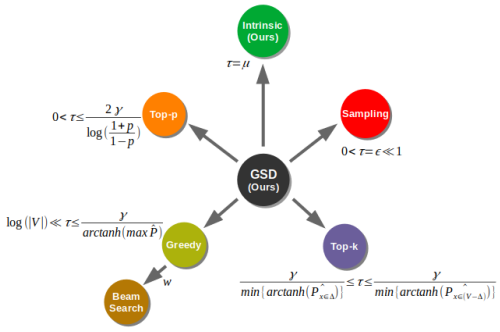

Figure 1: Illustration of the relationships between GSD and other decoding algorithms (including our proposed algorithm Intrinsic Decoding). Notations used in the figure will be described in Section 2 and 3.

In the following sections, we first introduce the problem statement of decoding for natural language generation tasks using formal mathematical notions, which will be uniformly used by all the equations and inequalities in this paper. Then, we briefly introduce related works of some existing popular decoding algorithms. These algorithms are famous and well-known in both academic and industrial areas. After that, in Section 3 we detail the proposed generalized decoding framework with formal mathematical theorems, connecting it with existing decoding algorithms. At the end of Section 3, we introduce our proposed method, Intrinsic Decoding, as an implementation of our GSD framework. In Section 4, empirical results of our proposed method are reported. Finally, in Section 5 we conclude our work.

Some theorem proofs and example generations are included in Appendix A.1, A.2 and A.3, which are helpful to understand our work.

2 Background

2.1 Problem Statement

Given a vocabulary V and a pre-trained generator $\mathcal{G}$, a general text generation task is to produce a sequence of words $\mathbf{x} = \{x_i\}, x_i \in V$ conditioned on some specific types of context $\mathbf{x}_\delta$, where δ represents a span containing $\mathbf{x}$. The context can be either text (e.g., in summarization (Shen et al., 2019b; Matsumaru et al., 2020), translation (Chen et al., 2020; Wang et al., 2020), story-telling (Ippolito et al., 2020) tasks), images (e.g., image caption (Alikhani et al., 2020)), or structured input (e.g., AMR-to-text (Liu et al., 2021), tabular-to-text (Li & Rush, 2020)). For open-domain generation tasks (Goldfarb-Tarrant et al., 2019; Prabhumoye et al., 2019), both $\mathbf{x}_\delta$ and x are text tokens. Specifically, $\mathcal{G}$ can be pre-trained using any properly configured neural models, such as GRU (Su et al., 2020b; Cho et al., 2014; Shi et al., 2020), Transformer Vaswani et al. (2017); Thapliyal & Soricut (2020), it accepts a span of context $\mathbf{x}_\delta$, performs inference, and finally gives a batch of predictions indicating the possibility for each word $x \in V$ that may appear after each individual $x \in \delta$. Formally, we represent such predictions by a distribution $\widehat{P}$ as defined in Equation 1.

$$\widehat{P}(x|\mathbf{x}_\delta) = softmax(\mathcal{G}_{\mathbf{x}_\delta}) \tag{1}$$

In the following sections, we refer to $\widehat{P}$ as the inference distribution, which is fully controlled by $\mathcal{G}$. Based on $\widehat{P}$, the decoding task aims to answer the question of how to get $\mathbf{x}$ from $\widehat{P}(x|\mathbf{x}_\delta)$? Intuitively, a naive solution is to argument-maximize $\widehat{P}$ iteratively until encountering an End-of-Sequence (EOS) token, which is known as Greedy Decoding. Simple as it is, Greedy Decoding is fast to run and straightforward to interpret. While just as implied by its name, Greedy Decoding shares the same weakness as other greedy-based algorithms. That is, locally optimized searching does not make any promise to the global best answer.

Indeed, both Greedy Decoding and Beam Search are approximations of the MAP decoding. To visualize the detailed decoding process of these deterministic algorithms, we plot the probabilities in each step to generate an example. As depicted in Figure 2.1, probabilities assigned by Greedy Decoding and Beam Search are pretty different from probabilities of the samples obtained from data generated by humans.

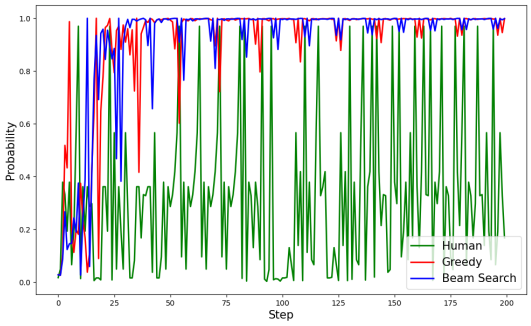

Figure 2: Decoding probabilities assigned by Greedy Decoding and Beam Search.
See Appendix A.3 for more figures about such kinds of comparisons.

One of the biggest challenges for deterministic decoding is that natural languages are so flexible that even for humans, we often give distinct utterances for the same prompt, which is hard to represent by deterministic logic.

This is straightforward to verify. For example, rather than mechanically recite, human beings prefer to express individually. Everyone has a preferred way to speak and has a unique subset dictionary to realize this personalization. This personalized sub-vocabulary is essential to make up distinct oral and writing styles of human communications. Most importantly, this sub-vocabulary is not deterministic. Instead, it varies. Previous work (Holtzman et al., 2020) has confirmed that deterministic strategy may not be the best answer to the question of how to get $\mathbf{x}$ from $\widehat{P}(x|\mathbf{x}_\delta)$.

Given these observations, a natural choice to get $\mathbf{x}$ from $\widehat{P}(x|\mathbf{x}_\delta)$ is to sample, instead of to search the solution space defined by V and $\mathcal{G}$. This strategy inspires a batch of new decoding algorithms developed for various NLG tasks. These algorithms are generally referred to as stochastic approaches, which will be detailed in the next section.

2.2 Related Work

In this section, we discuss the properties of some popular decoding algorithms that are widely used (Kang & Hashimoto, 2020; Su et al., 2020a)in both industry and research NLG domains.

For deterministic decoding, many recent works (Murray & Chiang, 2018; Stahlberg & Byrne, 2019; Welleck et al., 2020a) have been proposed to investigate how to better control the quality of generated text. For instance, Murray & Chiang (2018) draws an important conclusion that for beam search, it is not always true that a wider beam will help translation, sometimes it hurts. Stahlberg & Byrne (2019) shows constraining search with a minimum translation length is at the root of the problem of empty translations.

Recently, some works have been proposed to investigate how to explicitly control the property of the generated text by manipulating the decoding hyper-parameters under some specific principles. For example, Pang & He (2020) model generation as an offline reinforcement learning task with reference demonstrations. It aims to maximize quality given model-generated histories. Nadeem et al. (2020) propose to identify key properties that are shared among some decoding algorithms and investigate what will happen if meeting or violating the identified properties. (Basu et al., 2021) propose to design a feedback-based adaptive top-k decoding algorithm that generates text with a predetermined perplexity. However, none of them aim to design a generalized decoding framework that can be used to describe and connect existing popular decoding algorithms, which is just the focus of our paper.

For clarity, we denote the decoding probability assigned by a decoding algorithm by $P(x|\mathbf{x}_\delta)$, while keeping to use $\widehat{P}(x|\mathbf{x}_\delta)$ as in Section 2.1 to represent the generation probability inferred by a pre-trained NLG model $\mathcal{G}$.

Recall the sub-vocabulary observations about how human communicates introduced in Section 2.1. Surprisingly, we find that these stochastic decoding algorithms are also related to some sub-

vocabulary equivalents or subsets if formally described by mathematical language. To describe these different stochastic decoding methods uniformly, we use Equation 2 to abstract the relationship between $P(x|\mathbf{x}_\delta)$ and $\widehat{P}(x|\mathbf{x}_\delta)$. Different from V, Δ defined in Equation 2 is exactly the sub-vocabulary equivalent in mathematics containing relevantly important words for the decoding steps. It is also the key difference to discriminate between these stochastic decoding methods.

$$P(x|\mathbf{x}_\delta) = \begin{cases} \phi(\widehat{P}(x|\mathbf{x}_\delta)), x \in \Delta \\ 0, x \notin \Delta \end{cases} \tag{2}$$

We briefly introduce the theoretical backgrounds of Sampling Decoding, Top-k Decoding, and Nucleus Decoding (i.e., Top-p), especially how different Δ is constructed by these stochastic methods.

Among these methods, Sampling Decoding may be the simplest one because of its most flexible hypothesis of Δ: Δ is free to include any elements in V. In other words, Δ is V itself. Therefore, there is no algorithm-specific sub-vocabulary. All words in the entire dictionary space are candidates to sample from. This behavior can be formally described by Equation 3.

$$\phi(\widehat{P}) = \widehat{P}(x|\mathbf{x}_\delta), \Delta = V \tag{3}$$

Sampling Decoding extends the search space to V while maintaining high decoding efficiency compared to the deterministic methods. This is quite different from Beam Search, whose speed is very sensible to the search width parameter w. However, as remained by Holtzman et al. (2020), the problem with Sampling Decoding is that it has a long probability tail, which brings a lot of irrelevant samples in the decoding process.

To alleviate this problem, a Δ-limited decoding method, Top-k sampling, is proposed to decode text from pre-trained generators (Fan et al., 2018; Holtzman et al., 2018; Radford et al., 2019). Unlike Sampling Decoding, Top-k sampling has a limited subset of Δ, which consists of the top-k most promising candidates for the algorithm to consider. Equation 4 formally describes the conditions of how a proper Δ subset should be constructed.

$$S = \{\Omega | min\{\widehat{P}_{x\in\Omega}(x|\mathbf{x}_\delta)\} \geq max\{\widehat{P}_{x\in V-\Omega}(x|\mathbf{x}_\delta)\}, |\Omega| = k, x \in V\} \tag{4}$$

Given Equation 4, Top-k sampling can be precisely defined by Equation 5. Notably, S is defined as a super-set containing all valid Δ sets that can be used by Top-k. A benefit of defining Δ in such a way is that it takes into account all special and general cases of $\widehat{P}$. This definition holds even if there exist identical elements in $\widehat{P}$.

$$\phi(\widehat{P}) = \widehat{P}(x|\mathbf{x}_\delta) / \sum_{x\in\Delta} \widehat{P}(x|\mathbf{x}_\delta), \Delta \in S \tag{5}$$

Top-k sampling has a similar subset Δ to integrate the relevantly significant words for a given context. Promising as it is, the challenge for top-k is that the hyper-parameter of k is hard to specify (Holtzman et al., 2020): large k is problematic for flat $\widehat{P}$, while small k is precarious for peaked $\widehat{P}$.

Based on the observation, Holtzman et al. (2020) proposes a novel method to define Δ, the size of which is dynamically changed for different shapes of $\widehat{P}$. As described by Equation 6, the limitation for Δ is relaxed to the aggregated mass of the probabilities, instead of the size of Δ containing these probabilities. This method is known as Nucleus Sampling or Top-P Decoding, the definition of which is formally described by Equation 7.

$$S = \{\Omega | \sum_{x\in\Omega} \widehat{P}(x|\mathbf{x}_\delta) \geq p, x \in V\} \tag{6}$$

$$\phi(\widehat{P}) = \widehat{P}(x|\mathbf{x}_\delta) / \sum_{x\in\Delta} \widehat{P}(x|\mathbf{x}_\delta), \Delta \in S, |\Delta| = min\{|\Omega|, \forall\Omega \in S\} \tag{7}$$

Despite the relatively complex form of definition, Equation 7 can be efficiently executed by filtering the accumulated sum of $\widehat{P}$).

Given the lesson learned from Top-k, an akin question for Top-p sampling is how the hyper-parameter p is determined? General answer for this question is that small p values force the decoding algorithm to generate fluent expressions, while large p values encourage diverse generations. Then, what if p is small because we want fluent terms while $\widehat{P}$ is flat? Or what will happen if $\widehat{P}$ is peaked, but p has been set to a large value to favor our diverse demand?

Indeed, similar questions could be endless as long as there is a hyper-parameter affecting the performance of the decoding algorithm. Except for Sampling Decoding and Greedy Search, all decoding algorithms discussed so far involve at least one hyper-parameter managing specific properties of that method. Unfortunately, the two exceptions both have known issues that stop them from being considered as potential alternatives. These are the key observations inspiring our work.

3 METHOD

In this paper, we propose a novel decoding algorithm where no hyper-parameter is needed to manage the properties of the decoding performance. All behaviors are dynamically balanced and intrinsically adjusted, just like humans. Before introducing it, we will first introduce a more generalized decoding framework, referred to as GSD in this paper, to connect the decoding algorithms discussed above, including our proposed hyper-parameter-free decoding method that will be introduced later.

Since there is no oracle to consult, the only resource that can be referenced during a decoding process is the pre-trained generator $\mathcal{G}$. Therefore, we plan to hand over all hyper-parametric controlling tasks to $\mathcal{G}$, hoping it to automatically remind the decoding process of how flat or peaked the current inference is.

We first need a metric to quantify the flat/peaked degree of current inference. A natural choice is variance. While we prefer to use entropy since variance is relatively harder to control because of its unlimited value range. Entropy is an indicator to determine the uncertainty of a system. Formally, we use γ defined in Equation 8 to quantify the degree of uncertainty.

Then, we need an activation gate to control how γ will be engaged in the decoding system. More generic mappings, such as neural networks or linear combinations of basic functions, can also be used if efficiency is not a sensible factor in some NLG systems. In this paper, we use $tanh$ to manage γ, whose mathematical properties have been well studied. The gated result of γ is defined by Equation 9.

Finally, we can define the proposed GSD via Equation 10. S is a super-set of our focus Δ, each element Ω in S is a proper candidate for Δ. In practice, we prefer the ones with minimized size.

One may question Equation 9 because of τ, which has not been described currently. It seems like a hyper-parameter. While this is not true. We design τ in Equation 9 for three intentions. Firstly, it is a bridge for GSD to connect other decoding algorithms. By applying different restrictions on τ, GSD can be concretized to different decoding algorithms. Secondly, it is the key to our hyper-parameter-free decoding algorithm that will be introduced later. Thirdly, the presence of τ provides infinite space to design new stochastic decoding algorithms.

$$\gamma(\widehat{P}) = -\sum \widehat{P} log(\widehat{P}) \tag{8}$$

$$\rho(\widehat{P}) = tanh(\gamma/\tau) = \frac{e^{\gamma/\tau} - e^{-\gamma/\tau}}{e^{\gamma/\tau} + e^{-\gamma/\tau}} \tag{9}$$

$$S = \{\Omega | \sum_{x \in \Omega} \widehat{P}(x|\mathbf{x}_\delta) \geq \rho, x \in V\} \tag{10}$$

For the first intention, we detail the restrictions by four theorems formally describing how exactly GSD concretizes to existing decoding algorithms by different restrictions applied on τ. Specifically,

Theorem 3.1 3.4 characterizes the conditions under which GSD will become Sampling Decoding, Greedy Decoding, Top-k Sampling, and Nucleus Sampling, respectively.

Theorem 3.1 *GSD has the same effect as Sampling Decoding if $\tau = \epsilon$ holds, where $1 \gg \epsilon > 0$ is a tiny value close to 0.*

Proof 3.1 *$\tau = \epsilon$ will force $\gamma/\tau \to \inf$. As a result, the value of $\rho(\widehat{P}) = tanh(\gamma/\tau)$ will approach 1. Since the maximized value of $\sum_{x\in\Omega} \widehat{P}(x|\mathbf{x}_\delta)$ is limited to 1, Ω in Equation 10 has to include as many elements as possible to exceeds τ. Extremely, it can only be equal to $|V|$ at most, that is, sampling from the whole vocabulary set. This is just the definition of Sampling Decoding (see Equation 3).*

Theorem 3.2 *GSD will degenerate to Greedy Decoding when the value range of τ is limited by Inequality 11, where $|V|$ is the vocabulary size.*

$$log(|V|) \ll \tau \leq \frac{\gamma}{arctanh(max\widehat{P})} \tag{11}$$

See Appendix A.1 for the proof of Theorem 3.2.

Theorem 3.3 *GSD becomes Top-k Decoding if τ is limited by Inequality 12, where $|\Delta| = k$.*

$$\frac{\gamma}{min\{arctanh(\widehat{P}_{x\in\Delta}(x|\mathbf{x}_\delta))\}} \leq \tau \leq \frac{\gamma}{max\{arctanh(\widehat{P}_{x\in(V-\Delta)}(x|\mathbf{x}_\delta))\}} \tag{12}$$

Proof 3.2 *Given the fact the both $tanh$ and $arctanh$ are monotone increasing functions, we can rewrite Inequality 12 by Inequality 13, which is the equivalent way to define S as in Equation 4.*

$$max\{\widehat{P}_{x\in(V-\Delta)}(x|\mathbf{x}_\delta)\} \leq tanh(\frac{\gamma}{\tau}) \leq min\{\widehat{P}_{x\in\Delta}(x|\mathbf{x}_\delta)\} \tag{13}$$

Theorem 3.4 *GSD is equivalent to Nucleus Sampling by limiting τ using Inequality 14, where p is the nucleus size.*

$$0 < \tau \leq \frac{2\gamma}{log(\frac{1+p}{1-p})} \tag{14}$$

Proof 3.3 $\tau \leq \frac{2\gamma}{log(\frac{1+p}{1-p})} \Rightarrow \rho \geq p$

Given these theorems and proofs, it can be clearly understood that these algorithms are indeed related and our proposed decoding framework does have the ability to describe and connect them.

For the second intention, ensuring hyper-parameters are intrinsically managed is an important property to imitate human language generation habits. This is reasonable because it is very odd for humans to be told to utter more diversely or more fluently. Diverse or not, it usually depends on personal habits of language usage. Similarly, fluency or not is generally determined by people's familiarity with that language. Neither diversity nor fluency is a pre-specified hyper-parameter for routine communications. Both are automatically controlled by the person himself who is producing the oral utterances or written text. Therefore, the decoding algorithm should automatically define the hyper-parameters it will use in the decoding process.

Based on the observation, we define τ by Equation 15, where $|B|$ is the batch size of each inference action. Equation 15 is reasonable because in all formulations we have introduced above, batch information has not yet been considered or modeled. Actually, besides the horizontal direction along which the vocabulary items are inferred, information in the vertical batches is also vital since different samples in the same batch may observe quite distinct contexts.

Equation 16 formally defines the border of how the Δ subset of our hyper-parameter-free decoding algorithm should be restricted. We refer to the decoding method defined by Equation 16 as Intrinsic Decoding. Like other decoding algorithms discussed in this paper, it is also a special case of our proposed decoding framework. While different from all the other methods, Intrinsic Decoding is hyper-parameter-free. Neither deterministic search nor unlimited sampling strategies are involved in Intrinsic Decoding, which is the critical difference between Intrinsic Decoding and other hyper-parameter-free methods (e.g., Greedy and Sampling).

$$\tau = \mu(\widehat{P}) = \frac{1}{|B|} \sum_{\mathbf{x}_\delta \in B} \gamma(\widehat{P}) \tag{15}$$

$$\rho(\widehat{P}) = \frac{e^{\gamma/\mu} - e^{-\gamma/\mu}}{e^{\gamma/\mu} + e^{-\gamma/\mu}} \tag{16}$$

Notably, the parameters ρ and μ are automatically controlled and dynamically adjusted. Thus, both ρ and μ are not hyper-parameters specified by users.

Finally, for the third intention, the presence of τ provides an extensive way to discover and design new decoding algorithms. By setting up different conditions, our framework provides infinite space for new decoding approaches.

To summarize, we first propose GSD, a generalized decoding framework that can be concretized to other decoding algorithms introduced in Section 2.1. Then, we detail the connections between GSD and these decoding algorithms using formal theorems. Finally, we propose Intrinsic Decoding, the first hyper-parameter-free stochastic decoding algorithm that is not designed under deterministic search or unlimited sampling strategies.

4 EVALUATION

In this section, we describe the setups of all experiments and report the corresponding results.

4.1 SETUPS

We use GPT-2 (Radford et al., 2019) as the pre-trained generator to assign inference probabilities for all the decoding algorithms. GPT-2 is a popular generator that has been used in many generation tasks (Lawrence et al., 2019). For the corpus, we use the same testing datasets as GPT-2, which is released by OpenAI at [1]. Each decoding algorithm is executed under the same contexts built from this testing corpus. Concretely, the default batch size is 50. All generations are limited to a maximized length 200. For Beam Search, we run it under different search width $w = (2, 4, 6, 8, 10)$. For top-k decoding, $k = (20, 40, 60, 80, 100)$ are used as the size of Δ containing the top probabilities. While for Nucleus Sampling (Top-p Decoding), we use $p = (0.15, 0.35, 0.55, 0.75, 0.95)$ as the nucleus size of Δ. Also, we compare the performance of the decoding algorithms under different temperatures $t = (0.1, 0.3, 0.5, 0.7, 0.9)$.

We use perplexity (PPL) (Li et al., 2020) as the metric to evaluate the fluency of the generations. Perplexity closest to human generations is considered as the best indicator of fluent generations. Self-BLEU (Shen et al., 2019a) and distinct ratio (Welleck et al., 2020b; Wu et al., 2020) are two popular metrics to measure the diversity aspect of the generated text. Similar to perplexity, the gap between human generations and the decoding algorithms should be minimized if text generated by that algorithm is considered the most diverse. Following Holtzman et al. (2020), we include the Zipf coefficient as another measurement to test the statistical behavior of these decoding algorithms.

4.2 RESULTS

First, we compare the fluency performance by perplexity for different decode algorithms. As depicted in Figure 3, our proposed Intrinsic Decoding is the most promising algorithm to generate

[1] https://github.com/openai/gpt-2-output-dataset

Table 1: Metrics of different decoding algorithms

Method	PPL	Self-BLEU	Zipf	Distinct Rate
Human	12.38	0.31	0.93	0.72
Greedy	1.368	0.41	0.90	0.04
Beam Search, w = 2	1.382	0.41	0.90	0.04
Beam Search, w = 10	1.284	0.34	0.69	0.06
Sampling	68.488	0.27	1.02	0.82
Top-k, k = 100	14.244	0.35	0.90	0.65
Top-k, k = 40	10.619	0.39	0.89	0.59
Top-p, p = 0.15	1.479	0.37	0.89	0.04
Top-p, p = 0.75	9.687	0.36	**0.95**	0.58
Top-p, p = 0.95	36.707	**0.32**	0.96	**0.76**
Intrinsic (Ours)	**13.782**	0.35	**0.91**	0.63

fluent text as compared to humans. A distinctive line of our method can be observed in all three sub-figures, which is closest to humans' perplexity line.

Concretely, the perplexity of Beam Search is relatively more minor than other algorithms. Therefore, we plot a separated figure to contain it and embed it in the upper space of the left sub-figure. For the other two figures, the lines are relatively clear to read. From the embedded figure of Beam Search, it can be observed that increasing the search width helps to obtain more fluent generations. However, this benefit is traded by a costed search of the magnified paths inside the entire solution space. Thus, it is not sustainable with large w values if efficiency is sensible in practical systems.

Also, it can be observed from Figure 3 that increasing the size of top-k probabilities (k) or nucleus set (p) has a negative impact on the perplexity metric. This is because that enlarged Δ set for these algorithms improves the possibility to sample diverse words. However, the long tail problem of Sampling Decoding also comes alongside the benefit. Our method addressed this problem by transferring control of all hyper-parameters to the decoding algorithm itself, such that the hyper-parameter selection process can be completely avoided, making the decoding process be an intrinsic system just like how human communicates: no oracle is needed to configure ourselves before we can speak or write.

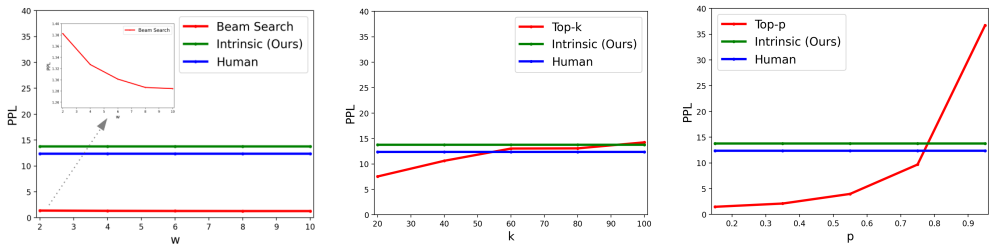

Figure 3: Comparisons of perplexity across different decoding algorithms.

In Table 1, we report results for all the generation metrics. From the table, it can be clearly observed that our proposed method is competitive among all deterministic and stochastic decoding algorithms with only a few exceptions. Even for the Distinct Rate metric, where our method is not the optimized solution, however, it is still the second-best alternative.

5 Conclusion

In this paper, we proposed a generalized stochastic decoding framework. By proofing four mathematical theorems, we demonstrated that Greedy Search, Sampling, Top-k Decoding, and Top-p Sampling are special cases of our proposed framework. We also designed a novel decoding algorithm, Intrinsic Decoding, as an implementation of this framework. All parameters in Intrinsic Decoding

are automatically controlled and dynamically adjusted. Thus, there is no need to be configured with hard-specified hyper-parameters. To the best of our knowledge, this is the first work developing a unified method to bridge existing decoding algorithms using formal theorems and concrete implementations. By setting up different conditions, our framework is capable to provide infinite space for new decoding approaches.

REFERENCES

Malihe Alikhani, Piyush Sharma, Shengjie Li, Radu Soricut, and Matthew Stone. Cross-modal coherence modeling for caption generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6525–6535. Association for Computational Linguistics, 2020.

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. Mirostat: A Perplexity-Controlled Neural Text Decoding Algorithm. In *International Conference on Learning Representations*, 2021.

Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jingjing Liu. Distilling knowledge learned in BERT for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7893–7905. Association for Computational Linguistics, 2020.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734. Association for Computational Linguistics, 2014.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898. Association for Computational Linguistics, 2018.

Seraphina Goldfarb-Tarrant, Haining Feng, and Nanyun Peng. Plan, write, and revise: an interactive system for open-domain story generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 89–97. Association for Computational Linguistics, 2019.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. Learning to Write with Cooperative Discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1638–1649. Association for Computational Linguistics, 2018.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 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, 2020.

Daphne Ippolito, David Grangier, Douglas Eck, and Chris Callison-Burch. Toward better storylines with sentence-level language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7472–7478. Association for Computational Linguistics, 2020.

Daniel Kang and Tatsunori Hashimoto. Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 718–731. Association for Computational Linguistics, 2020.

Carolin Lawrence, Bhushan Kotnis, and Mathias Niepert. Attending to future tokens for bidirectional sequence generation. 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)*, pp. 1–10. Association for Computational Linguistics, 2019.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the*

58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880. Association for Computational Linguistics, 2020.

Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi. Rigid formats controlled text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 742–751. Association for Computational Linguistics, 2020.

Xiang Lisa Li and Alexander Rush. Posterior control of blackbox generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2731–2743. Association for Computational Linguistics, 2020.

Yixian Liu, Liwen Zhang, Xinyu Zhang, Yong Jiang, Yue Zhang, and Kewei Tu. Generalized supervised attention for text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pp. 4991–5003. Association for Computational Linguistics, 2021.

Kazuki Matsumaru, Sho Takase, and Naoaki Okazaki. Improving truthfulness of headline generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1335–1346. Association for Computational Linguistics, 2020.

Kenton Murray and David Chiang. Correcting Length Bias in Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 212–223. Association for Computational Linguistics, 2018.

Moin Nadeem, Tianxing He, Kyunghyun Cho, and James Glass. A Systematic Characterization of Sampling Algorithms for Open-ended Language Generation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pp. 334–346. Association for Computational Linguistics, 2020.

Richard Yuanzhe Pang and He He. Text Generation by Learning from Demonstrations. In *International Conference on Learning Representations*, 2020.

Shrimai Prabhumoye, Chris Quirk, and Michel Galley. Towards content transfer through grounded text generation. 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)*, pp. 2622–2632. Association for Computational Linguistics, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. *OpenAI Report*, pp. 24, 2019.

Dinghan Shen, Asli Celikyilmaz, Yizhe Zhang, Liqun Chen, Xin Wang, Jianfeng Gao, and Lawrence Carin. Towards generating long and coherent text with multi-level latent variable models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2079–2089. Association for Computational Linguistics, 2019a.

Sheng Shen, Daniel Fried, Jacob Andreas, and Dan Klein. Pragmatically informative text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4060–4067. Association for Computational Linguistics, 2019b.

Wenxian Shi, Hao Zhou, Ning Miao, and Lei Li. Dispersed exponential family mixture VAEs for interpretable text generation. In *International Conference on Machine Learning*, pp. 8840–8851. PMLR, 2020.

Felix Stahlberg and Bill Byrne. 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)*, pp. 3356–3362. Association for Computational Linguistics, 2019.

Hui Su, Xiaoyu Shen, Sanqiang Zhao, Zhou Xiao, Pengwei Hu, Randy Zhong, Cheng Niu, and Jie Zhou. Diversifying dialogue generation with non-conversational text. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7087–7097. Association for Computational Linguistics, 2020a.

Shang-Yu Su, Chao-Wei Huang, and Yun-Nung Chen. Towards unsupervised language understanding and generation by joint dual learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 671–680. Association for Computational Linguistics, 2020b.

Zhiqing Sun and Yiming Yang. An EM approach to non-autoregressive conditional sequence generation. In *International Conference on Machine Learning*, pp. 9249–9258. PMLR, 2020.

Ashish V. Thapliyal and Radu Soricut. Cross-modal language generation using pivot stabilization for web-scale language coverage. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 160–170. Association for Computational Linguistics, 2020.

Bo-Hsiang Tseng, Jianpeng Cheng, Yimai Fang, and David Vandyke. A generative model for joint natural language understanding and generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1795–1807. Association for Computational Linguistics, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. Improving neural language generation with spectrum control. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. ICLR, 2020.

Sean Welleck, Ilia Kulikov, Jaedeok Kim, Richard Yuanzhe Pang, and Kyunghyun Cho. Consistency of a Recurrent Language Model With Respect to Incomplete Decoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5553–5568. Association for Computational Linguistics, 2020a.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. ICLR, 2020b.

Sixing Wu, Ying Li, Dawei Zhang, Yang Zhou, and Zhonghai Wu. Diverse and informative dialogue generation with context-specific commonsense knowledge awareness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5811–5820. Association for Computational Linguistics, 2020.

A Appendix

A.1 Proofs of Theorems

Proof A.1 (For Theorem 3.2) *We first determine the value range of $\gamma(\widehat{P})$ defined by Equation 8.*

1) For the maximized value of γ, the problem is a standard convex optimization process that can be defined by Equation 17:

$$\begin{aligned} &min f(x) = \sum_{i=1}^{|V|} x_i log(x_i), x_i = \widehat{P}_i \\ &s.t. \sum_{i=1}^{|V|} x_i = 1 \end{aligned} \tag{17}$$

The Lagrangian dual form of Equation 17 can be represented by Equation 18:

$$L(f(x), \lambda) = \sum_{i=1}^{|V|} x_i log(x_i) + \lambda(\sum_{i=1}^{|V|} x_i - 1) \tag{18}$$

Take the first-order partial derivative of L to any element of x. We can get

$$\frac{\partial L(f(x), \lambda)}{\partial x_i} = \frac{\partial}{\partial x_i} \left[\sum_{i=1}^{|V|} x_i log(x_i) + \lambda(\sum_{i=1}^{|V|} x_i - 1) \right] = 0 \tag{19}$$

The solution of Equation 19 is $\lambda = -log(x_i) - \frac{1}{ln2}$.
Therefore, $x_1 = x_2 = ... = x_i, = x_{|V}$. By $\sum_{i=1}^{|V|} x_i = 1$, we can get $x_i = \frac{1}{|V}$.

Given the solutions of x_i, we can finally get the maximized value of $\gamma(\widehat{P}) = minf(x_i) = log(|V)$

2) For the minimized value of γ, the problem can be directly solved by observing derivative function figures of Equation 20:

$$f(y) = -ylog(y) \tag{20}$$

By observing the figures of $f(y)$'s first and second order derivative functions, the minimized value can be obtained: $minf(y) = f(1) = 0$.

Given 1) and 2), the range of γ can be determined: $\gamma \in [0, log(|V|)]$.

However, for a general NLG model $\mathcal{G}$, it rarely assigns full 0 or 1 probabilities for a batch of input contexts.

Therefore, the actual value range for γ is $\gamma \in (0, log(|V|)] \subset [0, log(|V|)]$.

Let $\tau \gg log(|V|)$, it will force $\gamma/\tau \to 0$. As a result, the value of $\rho(\widehat{P}) = tanh(\gamma/\tau)$ will approach 0. However, it will never be 0, which enables much more flexibility for Ω than in Theorem 3.1 because it only needs to include a tiny set of elements.

Recall the condition for Theorem 3.2: $\tau \leq \frac{\gamma(\widehat{P})}{arctanh(max\widehat{P})}$, it ensures $\sum_{\Delta} \widehat{P} \geq max\widehat{P}$. This condition forces the tiny set to include the top probability element in V, which is equal to Greedy Decoding.

A.2 Example Generations

Shared Context
Tokenized Ids = [9060, 8305, 1644, 1043, 5445, 4324, 290, 5445, 5405, 319, 300, 1192, 1603, 2975]
Text = *Image caption Police found broken window and broken glass on Lancaster Road*

Human
There have been raids on homes in Blackburn homes after a Wexford-based developer a llegedly forged a fraudulent loan and offered thousands of dollars in new loans to people without any experience in the business. According to police reports the owners of some homes were encouraged to move in in order to avoid debt risk. ...

Greedy Decoding
A man was arrested on suspicion of burglary and possession of a stolen vehicle. The man was taken

to hospital with serious injuries. He was taken to hospital with serious injuries. The man was taken to hospital with serious injuries. The man was taken to hospital with serious injuries. The man was taken to hospital with serious injuries. The man was taken to hospital with serious injuries. ...

Beam Search with width = 2
A police spokesman said: Ẅe are investigating the incident.Ẅe are appealing to anyone who witnessed the incident to contact us.T̈he lancaster road is a popular route for cyclists and pedestrians on the north-west side of the city."

Beam Search with width = 10
Police said they were called to the scene at around 11.30pm on Sunday after reports of a disturbance. They said they were called to the scene at around 11.30pm on Sunday after reports of a disturbance. They said they were called to the scene at around 11.30pm on Sunday after reports of a disturbance.

Sampling without temperature
A bystander suffered non-life threatening injuries after workers and emergency workers had to make an emergency stop while conducting routine traffic stop on lancaster road outside Athill in Cork last night. Lyte Snelling was up and about as the cops stopped, before continuing with their routine traffic stop. ...

Sampling with temperature = 0.9
A bystander suffered non-life threatening injuries after workers and emergency workers had to make an emergency stop while conducting routine traffic stop on lancaster road. The cause of the incident in Scarborough came as a separate report by the MCHA said a lancaster road basal had been broken in Scarborough. The report said a light on lancaster side side c̈aught fire and required immediate emergency crews to get it to safety.̈ Baker said as more details came in it was completely öut of platformäs to what was happening to the lancaster. Mr Baker said: T̈he lancaster is now getting into complete crisis mode.N̈o one's money is being wasted trying to stop our project from proceeding properly."̈

Top-k with k = 40, no temperature applied
A motorist suffered extensive bruises after he crashed his car into the lancaster on the Lilliana Drive, an isolated intersection, just after 5.30pm this week. Inspector George Cocker was called by a motorist, who said the driver's car had broken into the road, possibly because it had been parked on neighbouring roads. ...

Top-k with k = 40, temperature = 0.9
A motorist suffered extensive injuries after he crashed his car into a lancaster on the Lusitania Highway. No-one was arrested and no injuries were reported. In a statement issued by the Lusitania Police, officials said the driver was carrying a vehicle with an estimated weight of over 500kg weighing over 100kg. ...

Top-k with k = 100, no temperature applied
A motorist suffered extensive bruises after he crashed his car into the lancaster on the Lulles Street side of the intersection, just after 5.30pm this week. Dr Neil Wilson with CTA was treating the driver as if he had been beaten with a hammer and after being treated for cuts later, told the Liverpool Echo his car had been used to "cock it up". ...

Top-p with p = 0.15, no temperature applied
A man was arrested on suspicion of burglary and attempted burglary after he was found in a lancaster road. The man was taken to hospital with serious injuries. Police said the man was taken to hospital with serious injuries. Image copyright PA Image caption Police said the man was taken to hospital with serious injuries Image copyright PA Image caption Police said the man was taken to hospital with serious injuries Image copyright PA Image caption Police said the man was taken to hospital with serious injuries ...

Top-p with p = 0.95, no temperature applied
Alex Jones and five other co-defendants spoke on Monday about the discovery. They said they had been going by the name "Mike Jackson" in cars and a recording had been made, and that the investigation was finished when police finally announced they had found "evidence." ...

Top-p with p = 0.95, temperature = 0.3
A man who was driving a lancaster on the road in the city's north-east was arrested on suspicion of

driving under the influence of alcohol. The man, who was not named, was arrested at about 10:30 GMT on suspicion of driving under the influence of alcohol. He was taken to hospital with serious injuries. ...

Intrinsic Decoding without temperature (Ours)
A spokesman for Norwich City said: "The case is being investigated by the council's Financial Conduct Authority and the public has been informed." No further details have been given at this stage." Tops said the window had been returned to the club, but no break was found. Despite the police being called to the scene, Norwich City's President Dave O'Dwyer said he could not discuss the investigation. ...

Intrinsic Decoding with temperature = 0.9 (Ours)
A spokesman for Northern Ireland's National Police Chiefs' Council said: "We have located broken windows and broken glass in the areas of Carmarthen and Kilbride. "We are continuing to investigate and are working with the police to identify the perpetrators." Police in Carmarthen, Cumbria, said at least seven people had been taken to hospital with non-life threatening injuries. ...

More generation examples can be downloaded from [2]

A.3 Illustration of Decoding Probability

[2] https://github.com/ginoailab/gsd/blob/master/dumps/result.zip

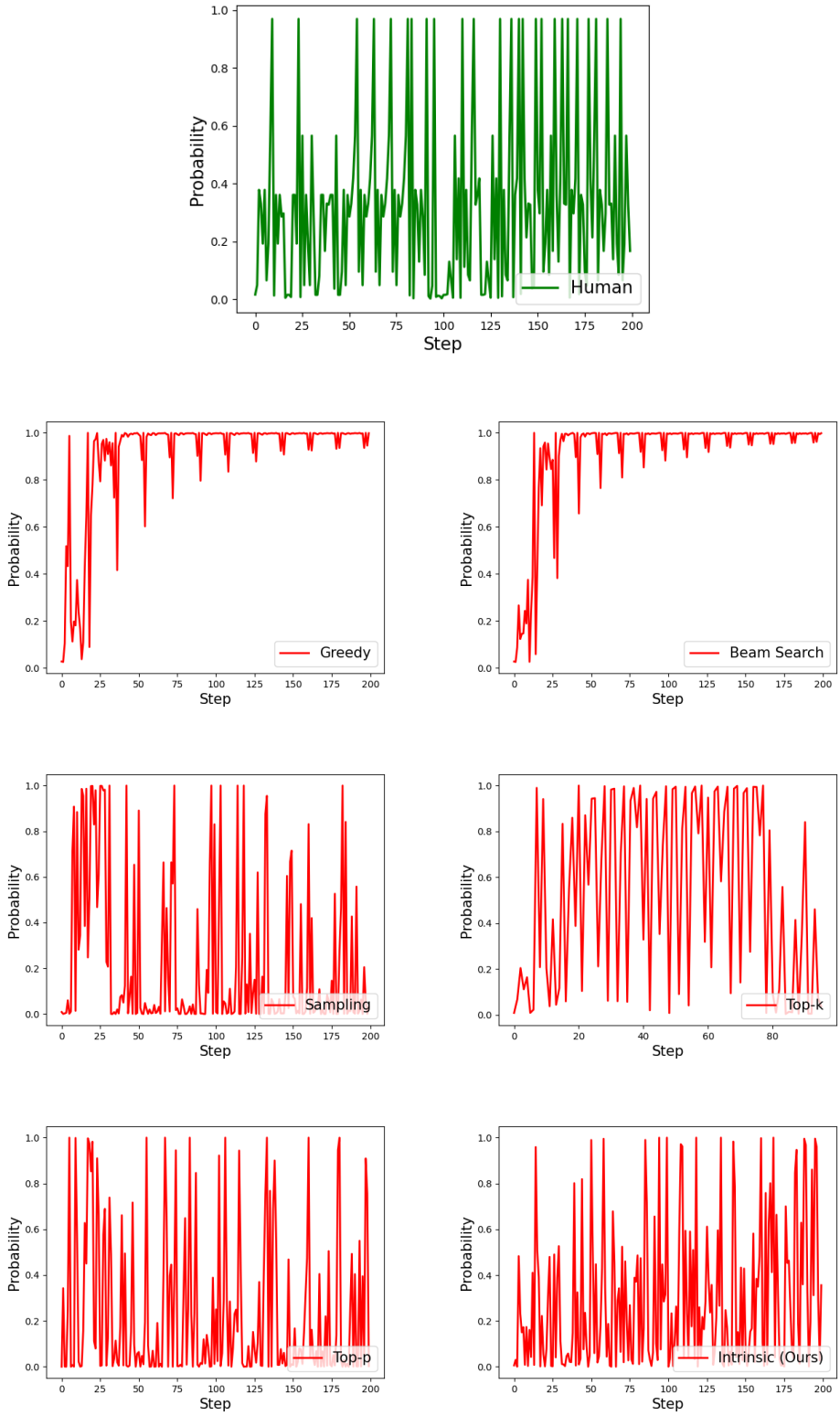

Figure 4: Decoding probabilities assigned by different decoding algorithms.