A Generalized Formalism of Auto-Regressive Decoding for Speech Processing

Julia Gachot **, *Philipp Allgeuer*, *Marie S. Bauer*, *Stefan Wermter*

Knowledge Technology, Department of Informatics, University of Hamburg, Germany

julia.gachot@uni-hamburg.de, philipp.allgeuer@uni-hamburg.de, marie.bauer@uni-hamburg.de, stefan.wermter@uni-hamburg.de

Abstract

In speech processing, most state-of-the-art sequence prediction models rely on auto-regressive (AR) strategies to generate output sequences based on the raw predictions of the model. Despite their crucial role in the inference process, a comprehensive overview of AR strategies as a unified field is lacking, due largely to implicit and multiple definitions of next-token decoding. This context complicates the choice, comparison, and evaluation of strategies, while creating inconsistencies in the characterization of approaches as auto-regressive or not. We begin by setting explicit inclusion criteria for the field of AR search in speech processing, and derive a generalized theoretical framework to categorize and report on search strategies for neural models. We show the capabilities of this formalism in simplifying the design of benchmarks centered around the decoding process, allowing for ablation studies that are focused on search strategies.

Index Terms: Auto-regressive (AR) models, Generation, Decoding, Deep Neural Networks (DNN)

1. Introduction

Across language modeling tasks, auto-regressive (AR) neural networks dominate the state-of-the-art for sequence prediction [1, 2, 3, 4]. A model's generation strategy refers to the local search process towards composing a sequence, through an iterative scoring and updating of partial candidates. These models are trained to estimate an auto-regressive score from an input and a prior, which represents the current decoding status. At inference time, sequences are iteratively constructed to approximate a local maximum of an objective function, using the model's raw predictions. The most commonly used objective function is Maximum a Posteriori (MAP). It is generally the one used for beam search [5], where the MAP estimates are combined to a breadth-first heuristic for the exploration of the search space. Other heuristics are present in literature, for example, that are tailored to improve performance on specific speech processing tasks. In this direction, early strategies proposed variants on the objective function side, such as sampling [6, 7, 8, 9], which introduces some randomness in the candidates. More recently, from the modeling side, models called Non-Auto-Regressive (NAR) often maintain a notion of objective function, but propose a parallelized decoding process [10]. The existence of these iterative sequence prediction methods, which deviates from the classic left-to-right next-token prediction, raises the question of where to draw the line between auto-regressive and other approaches. New search strategies have also appeared in research papers around major developments in sequence modeling architectures, such as LLMs. The focus of these papers is often a task, a model, or a training method, while the inference process is treated more as an implementation detail.

As some of these decoding strategies are reused across tasks, they begin to exist in multiple versions (e.g., speculative sampling, NAR sequence generation), creating the need for a consistent way to place and compare methods with existing literature. The lack of formalization of inference and training schemes as distinct aspects of modeling, given the drastically different questions and challenges they represent, has been identified in literature [11]. It is also reflected on the experimental side, where evaluating search strategies' behavior is difficult across speech processing tasks, and therefore rarely done [4, 12]. To simplify making these benchmarks, we propose a generalized formalism to conceptualize and implement AR generation algorithms beyond specific tasks and models. This framework aims to move the field away from the ubiquitous conceptualization of these approaches as 'model likelihood maximizers' at inference time. Instead, we define AR strategies for neural networks from the perspective of designing a recurrence relation that efficiently estimates sequence likelihood, updates candidates, and determines when and how to end the iterative process. From this generalized and modular definition of the steps intrinsic to AR generation strategies, explicit criteria for a decoding method to be considered auto-regressive can be derived. We examine cases that are traditionally viewed as being at the edge of the field, and show the capabilities of this framework in categorizing approaches in a systematic way, from the specificities of their structure. Using the modularity of the proposed formalism, we also demonstrate how to isolate the contribution of each step to the overall performance by proposing an ablation study methodology for sequence prediction search strategies.

2. Related Work

2.1. AR decoding as a combinatorial optimization strategy

Decoding strategies for neural sequence predictors constitute a subset of methods for discrete combinatorial optimization, in which heuristics are used to explore large structured solution spaces. Several taxonomies propose an overview of these heuristics, but rarely cover machine learning approaches [13], or only for model optimization [14]. In sequence prediction, taxonomies tend to focus on AR architectures [15], and frameworks exist to evaluate the compositional capabilities of models [16], but not their decoding strategies. Without a systematic comparison framework in this rapidly evolving field, very similar metaheuristic algorithms are proposed under different names [14]. This also happens with certain subgroups of neural generation methods, such as the parallelization of the decoding process to decrease the number of inference steps, which can be found under many different names, although speculative decoding is

** indicates the corresponding author.

the most commonly used [17, 18, 19, 20, 21]. One reason may be that presenting a 'one-step edit' is often met with concerns from reviewers regarding the technical novelty of the approach, even for papers that later have a high impact [7, 22]. Within the framework proposed in this paper, we compose a general structure of AR decoding as a modular set of steps to simplify locating and characterizing contributions within the structure in both existing and future algorithms.

Even beam search usually does not refer to the original algorithm anymore [5], but has been adapted with variations for neural networks, especially regarding their termination condition. Taking for granted the contents of searches creates a risk of insufficient reporting, which is documented even beyond the case of generation with neural models [23]. By moving away from presenting strategies as an indivisible block, our framework clarifies the core elements to document when presenting or using a strategy, detailed in Sec. 3.3. Regarding these steps, some formalisms in the literature describe beam search as several steps being repeated [24], but they do not detail how the estimation could be performed with a sequence prediction model. For neural models, the formalism of Best-First Beam Search is general enough to recover some beam search variants, but it is not designed to encompass stochastic methods [25]. Our formalism encompasses a wider diversity of approaches and is oriented specifically toward neural models.

2.2. Current categorizations of AR strategies

Without an existing taxonomy, the scope of surveys on AR strategies can be used as a proxy for how strategies are currently grouped in the literature, as well as for the open questions they reflect. AR decoding strategies are often surveyed in a task-specific manner, highlighting a strong task-method association for certain tasks. In automatic speech recognition (ASR), end-to-end speech-to-text translation, and grammatical error correction (GEC), beam search and its variants are the expected strategies [1, 3, 26]. In machine translation (MT), beam search also appears as the reference [27, 28], although some works suggest that stochastic methods could also be promising [29]. Text-To-Speech (TTS) is a task where NAR models have been omnipresent over the last five years, with less attention given to search strategies than in some NLP tasks [30, 2]. In the language model-based subset of TTS, sampling with temperature is classically used, like VoiceCraft [31]. For LLM-based generation task benchmarks, however, beam search is mostly used as a baseline and is often systematically evaluated against stochastic methods [32, 4, 33]. This has led to the widespread view of deterministic and stochastic strategies as two distinct classes of methods, due to the benefits of randomness in introducing diversity in generated outputs [34, 35, 22]. We argue that this intuition is no longer representative of the field. Some non-stochastic methods generate diverse outputs by penalizing repetitions [36], or by performing a sampling step that, while not involving randomness in the process, would exclude them from MAP strategies if interpreted strictly [24]. As a consequence, findings about AR strategies across tasks are rarely explored, and the papers that inspect it demonstrate vast performance differences even within text generation [4, 37, 38]. In this paper, by adopting a modular view of these approaches, we propose a formalism that categorizes these searches more reliably, keeping the same structure regardless of whether the search is deterministic or stochastic, or tested for a specific task.

3. Generalized Auto-Regressive Formalism

This framework is intended for tasks that can be formalized as a discrete compositional optimization problem, and more precisely a stochastic integer problem under probabilistic constraints (SIPC) [24]. In SIPC, both random variables and decision variables are integers, which is why outputs are often represented as an ordered one-dimensional sequence of integers or tokens.

3.1. Assumptions and inclusion criteria

We define a generation approach as a tuple $(\mathcal{M},\ g_{AR})$, where $\mathcal{M}$ is a neural model, and g_{AR} its inference algorithm. In the following, we call this tuple an auto-regressive predictor, provided it fits the following criteria.

Model assumptions. The model is trained to estimate conditional probabilities over $\mathcal{A}$, a finite alphabet or set of tokens. During inference, the notion of decoding status is leveraged to estimate the conditional probability through the updating of an internal state or a prior. The model's predictions belong to $\mathcal{A}^*$, the Kleene closure of its alphabet.

Decoding assumptions. The generation process g_{AR} is an iterative local maxima search, designed to be tight [39]. An objective function (e.g., MAP) is used at least once throughout the process to inform the updating of a finite set of candidates sampled from $\mathcal{A}^*$.

3.2. General formalism

Based on the assumptions in Section 3.1, we now define the core steps of an auto-regressive predictor's inference algorithm g_{AR}. In the general SIPC formalism, at each iteration t of the decoding process, a set of $B_t \in \mathbb{N}_+$ candidate sequences $\mathbf{Y}_t = \{\mathbf{y}^b \in \mathcal{A}^* | \ b \in 1...B_t\}$ is updated. The specificity of using SIPC strategies with neural models is that estimating the conditional probability requires a notion of a prior, denoted as a set $\mathbf{Z}_t$, to keep track of the decoding process using various sources of information. In the simplest cases, $\mathbf{Z}_t$ coincides with the set $\mathbf{Y}_t$ of candidates resulting from the $t-1$ inference step, but an alternative state variable is used by some architectures.

While the nature of the prior $\mathbf{Z}_t$ is often evident for a specific algorithm, proposing a general definition for it is one of the key levers of our formalism for enabling a systematic comparison between models and tasks. Additionally, examining if and how a prior is defined and updated throughout the generation process is a good proxy to distinguish auto-regressive (AR) strategies from other local search methods. In the same spirit, instead of defining an iteration around the evaluation of an objective function, we outline a modular structure for AR strategies that generalizes across speech processing tasks. We formalize an iteration t as a function $f^t_{(\mathcal{M},\ g_{AR})}(\mathbf{Y}_t, \mathbf{Z}_t)$ that returns an updated prior $\mathbf{Z}_{t+1}$ and set of candidate sequences $\mathbf{Y}_{t+1}$ from the past step output and prior.

$$(\mathbf{Y}_{t+1}, \mathbf{Z}_{t+1}) = f^t_{(\mathcal{M},\ g_{AR})}(\mathbf{Y}_t, \mathbf{Z}_t), \text{ with } \mathbf{Y}_{t+1} \neq \mathbf{Y}_t \quad (1)$$

Based on the assumptions of Sec. 3.1, once all of the iterations $f^t_{(\mathcal{M},\ g_{AR})}$ are complete, each of the following four steps has happened at least once:

① **Estimation.** For each candidate, the model $\mathcal{M}$ is assumed to estimate a probability mass function (PMF). The output of this step is P_t, a PMF conditioned on $\mathbf{Y}_t$ and $\mathbf{Z}_t$.

$$\begin{aligned} P_{t+1} : \mathcal{A}^* &\longrightarrow [0,1] \\ \mathbf{a} &\longmapsto p(\mathbf{a} \mid \mathbf{Y}_t, \mathbf{Z}_t) \end{aligned} \quad (2)$$

This is often implemented by performing the forward pass of an input $\mathbf{x}$, post-processing the resulting logits as necessary, and applying a softmax. In this classic case, the PMF is given by

$$P_{t+1} = \text{softmax}(\mathcal{M}(\mathbf{x}, \mathbf{Z}_t)). \tag{3}$$

② **Decision.** When a generation approach is auto-regressive, the model estimation should intervene in refining the candidate sequence sets. By evaluating an objective function f_{obj}, the model's PMF estimates from eq. 2 are aggregated and interpreted. The highest scoring output according to f_{obj} can then be used to construct $\mathbf{Y}_{t+1}$. If a sequence is generated from left-to-right, the decision step consists of evaluating the B_t highest scoring sequence continuations $\mathbf{a}_{t+1} \in \mathcal{A}^*$ to the elements of $\mathbf{Y}_t$. Based on this objective f_{obj}, the candidate sequences become

$$\mathbf{Y}_{t+1} = \underset{\mathbf{y} \in \mathcal{A}^*}{\text{B}_\text{t}\text{-argmax}}\, f_{obj}(\mathbf{y}). \tag{4}$$

The following is expressed with that idea in mind for clarity, but can easily be adapted to other cases, such as parallelized generation strategies that do not adopt that implementation. With a Maximum A Posteriori (MAP) objective, the general decision step from eq. 4 becomes

$$\mathbf{Y}_{t+1} = \underset{\mathbf{y} \in \mathcal{A}^*}{\text{B}_\text{t}\text{-argmax}} \sum_{\mathbf{a}_t \in \mathbf{y}} \log\left(P_t(\mathbf{a}_t)\right). \tag{5}$$

③ **Update of prior.** This last step is based on the assumption that a generation approach is considered auto-regressive if it includes a mechanism to keep track of the past inference steps. In the simplest (and most common) case, $\mathbf{Z}_t$ is a set of sequences from $\mathcal{A}^*$. At each decoding iteration t, we have that the prior passed for the estimation step is exactly the set of candidates $\mathbf{Y}_t$, meaning $\mathbf{Z}_t \leftarrow \mathbf{Y}_t$. Otherwise, the relation can have a more complex expression, for example, if the prior is based on the model's internal state.

④ **Termination test.** Finally, a boolean termination condition must ensure that, after a finite number of iterations, the decoding process terminates. At a given iteration t, that is tested using

$$f_{term} : (\mathcal{A}^*)^{B_t} \rightarrow \{0, 1\} \tag{6}$$

$$\mathbf{Y}_t \mapsto \begin{cases} 1 & \text{if termination condition} \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where f_{term} denotes the function that tests the candidate $\mathbf{Y}_t$ for decoding termination.

3.3. Reporting design choices

Within the framework defined in this paper, defining an auto-regressive decoding strategy g_{AR} consists of making a set of design choices governing a recurrence relation. This begins with specifying an **initial condition**, i.e., the contents of $\mathbf{Y}_0$ and $\mathbf{Z}_0$. Then the **recurrence relation** of eq. 1 can either be the same at each iteration, meaning defining $f^0_{(\mathcal{M},\, g_{AR})}$ is enough, or if the recurrence relation evolves throughout the decoding, each iteration expression is required. Reporting on an iteration t, means defining the **estimation**, **decision**, and/or **update of prior** steps within $f^t_{(\mathcal{M},\, g_{AR})}$, as well as if it includes a **termination test**.

3.4. Formalizing classic search strategies

We demonstrate the capability of our framework in encompassing both beam search and sampling by following the reporting guidelines of Sec. 3.3.

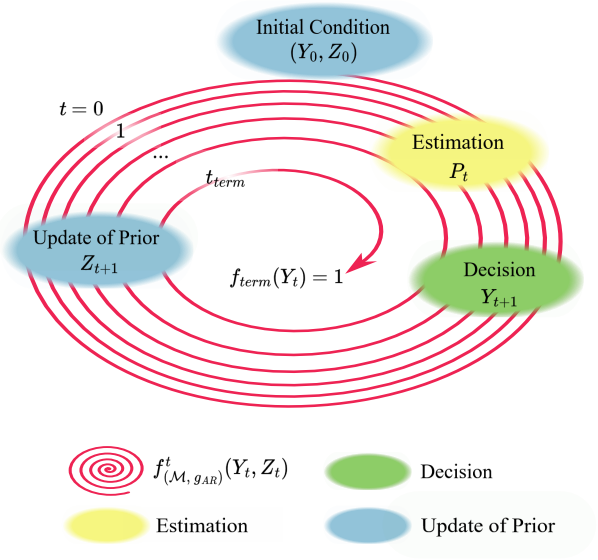

Figure 1: *Structure of the most common local search process for sequence generation with neural models. The formulation of the iteration $f^t_{(\mathcal{M},\, g_{AR})}(\mathbf{Y}_t, \mathbf{Z}_t)$ does not depend on the iteration step t, which is true for beam search and most of its variants. At each iteration, an estimation, a decision, and an update of the prior are performed until reaching a satisfying approximation at the center. The termination condition is defined to reflect what an acceptable output is for the algorithm at hand.*

3.4.1. Beam search

A specificity of beam search is its beam size, the fixed number $B \in \mathbb{N}_+$ of candidate sequences in any $\mathbf{Y}_t$. The **initial condition** in beam search depends on the training strategy of $\mathcal{M}$, the chosen model, e.g., BOS or prompt tokens. Regarding the **recurrence relation**, in beam search, all iterations follow the same template, illustrated in Fig. 1, and contain an estimation and decision steps followed by a termination test. If the termination test is negative, the iteration ends with an update of the prior, or nothing if the test is positive. Each **estimation** consists of applying the model $\mathcal{M}$ forward pass to the input sample ($\mathbf{x}$) and prior at t ($\mathbf{Y}_t$) to generate logits. The softmax function is applied to the logits and yields the PMF for step $t+1$, as described in ①. The **decision** is based on the Maximum A Posteriori objective function of eq. 5. Each **update of the prior** consists of setting $\mathbf{Z}_{t+1} \leftarrow \mathbf{Y}_{t+1}$. The **termination** of beam search is also dependent on the model; for example, the first appearance of an End-Of-Sequence (EOS) token within the set of candidates can be a termination test. Once the test is positive, at t_end, the most likely sequence in $\mathbf{Y}_{t_{end}}$, is returned as a final output.

3.4.2. Temperature sampling

In sampling with temperature [6, 40], B candidates are updated at each iteration. Similarly to Sec. 3.4.1, the order and content of steps within an iteration remain identical throughout the decoding. For a given model, switching from beam search to sampling has no influence on the definition of the initial condition, termination and update of prior. Within our framework, the differences between these two methods are restricted to the estimation and decision steps. For the **estimation**, temperature rescaling introduces a slight difference in the estimation step's eq. 3. It becomes $P_{t+1} = \text{softmax}(f_T(\mathcal{M}(\mathbf{x}, \mathbf{Z}_t)))$ for f_T the temperature sampling function, applied to the raw logits. At the

decision step, the aggregated probability estimates are filtered to keep B^2 sequences ranked using MAP, like in eq. 5, except $\mathrm{B_t}$-argmax becomes B^2-argmax. Finally, B sequences from the resulting set are randomly sampled to compose $\mathbf{Y}_{t+1}$.

4. Discussion

4.1. Edge cases

Presented with sequence prediction methods that deviate from the classic left-to-right next-token prediction paradigm, we demonstrate how our proposed formalism draws the line between auto-regressive and other approaches. We look into ten methods over various speech processing tasks, published between 2018 and 2025. Among them, half propose a speculative decoding strategy [17, 18, 19, 20, 21], while the others present non-AR methods [41, 42, 43, 44, 45]. The principle for inclusion is to consider a generation approach to be auto-regressive if both the model and decoding assumptions defined in Sec. 3.1 hold. Among the ten papers, based on the **decoding assumptions**, only one approach is excluded for not formulating the task as an SIPC [43]. The **model assumptions** hold for all speculative decoding methods as well as, perhaps surprisingly, two approaches from the non-AR group [42, 45].

Based on these assumptions, models that do not estimate a conditional probability are excluded, even if a mechanism is used to constrain the coherence of sequence [41]. However, a diversity of approaches is included, even when using novel variables as prior for their conditional probability estimation step. These priors often differ from the usage of those employed in traditional next-token predictors, or their construction from past candidate sequence sets $\mathbf{Y}_t$, but the overall structure of their generation process remains analogous. As a result, our framework can be used to compare traditional methods with algorithms that do not adopt the default left-to-right decoding process [18, 42], or do not constrain how far ahead a prior can influence the conditional probability [17, 45]. In addition to relaxing the aforementioned *left-to-right* and *next-token* aspects, as long as the generation is conditional on a decoding status variable, even using past candidate sequence sets is not mandatory to update a prior [19, 44]. In addition to methods that can intuitively be identified as auto-regressive, our proposed framework includes search approaches that follow a similar structure, making them comparable to the rest of the field.

4.2. Categorization

As discussed in Sec. 3.2, the proposed framework is designed to describe and interpret generation strategies as local search by clarifying their recurrence relation mechanism, as described in eq. 1. Each of the design choices ① to ④ that constitute this relation can also be used to compare methods from the standpoint of their design choices and structure. Most search strategies adopt an iteration formulation that does not change over time as described in Fig. 1, meaning $f^0_{(\mathcal{M},\ g_{AR})}$ is reused at each iteration t. In that case, the order and individual definition of steps are enough to describe the generation process entirely, from which an immediate and systematic step-by-step comparison can be made to compare generation strategies. Moreover, most search strategies for neural networks share steps with beam search or other traditional methods, meaning the nature of the proposed changes can be located at specific design choices. Note that various steps propose novel estimation steps [20, 44, 6], decision steps [17, 45, 22, 36], or update the decoding prior novel ways [18, 19, 21]. These categories propose a new angle on how to group tasks, to complement the classic paradigms of restricting to a task, or to deterministic or stochastic strategies.

When constructing a benchmark of search strategies that are compatible with a given model, our framework groups methods that share a similar location or principle to optimize for speed, diversity, and so on. For instance, when maximizing diversity, a TTS strategy based on the MAP objective [36] and a stochastic decoding method for text generation [22] may not seem relevant to compare at first glance. However, within our framework, these strategies share most of their structure, and their contributions can both be located and understood as decision step variants. This categorization by steps holds beyond the traditional tasks and objective functions' scopes, which allows to compare and to draw new parallels between existing strategies.

4.3. Prospective ablations for AR decoding

When proposing a new model architecture, it is a common practice to perform ablation studies. For non-AR architectures, it often consists of evaluating the individual contribution of a score or architecture change to inference speed or output quality [41, 43, 45]. To evaluate generation strategies, similar experiments are sometimes performed by varying hyperparameter values to limit the effect of a step, such as the termination condition [19]. Within the formalism from Sec. 3.2, the added modularity in the structure can be used for assessing the respective contributions of novel estimation, decision, update of prior, or termination condition to the overall performance. Since these steps are structural components, instead of completely canceling a step entirely, the ablation is approximated by switching one or several steps with a baseline equivalent (e.g., the same step from beam search). The more the results are impacted by the replacement of a step, the more it contributes to overall performance, similarly to classic ablation studies.

This can help create links with non-AR models ablation studies, where new scores or architecture changes are often created as a way for the decoding process to be parallelized within the model. This modular view of searches can also open new research directions, having identified which elements of a search make the most effect. It creates the possibility to take advantage of several contributions of existing searches in certain cases, optimizing for more than one goal, including speed, diversity, or accuracy. For example, a search strategy that proposes a score at step ③ to increase diversity could be combined with an estimation method ① that speeds up the decoding. This addresses the need for more efficient and powerful searches, while keeping track of the provenance of each block, rather than by presenting each search as an entirely new and indivisible algorithm.

5. Conclusion

In this paper, we propose a framework that sets the recurrence relation at the core of reporting on generation strategies, including systematic inclusion criteria for a search to be auto-regressive or not (Sec. 4.1). It offers a new way to compare searches from the location and mechanism of their contributions within the search process (Sec. 4.2). Moreover, by shifting the focus from the comparison with searches that make similar claims or are used for the same task, our framework offers a new angle to include comparable auto-regressive predictors and compose high quality benchmarks. Finally, we transpose the notion of ablation studies to decoding strategies, and highlight possible research directions when designing new generation algorithms by optimizing for several aspects at the same time (Sec. 4.3).

6. Acknowledgments

The authors gratefully acknowledge funding from Horizon Europe, under the MSCA grant agreements No 101072488 (TRAIL), No 101168792 (SWEET) and No 101226624 (GREET), as well as from the German Research Foundation (DFG), project number 551629603.

7. Generative AI Use Disclosure

No generative model was used in the writing of this article.

8. References

[1] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter, and S. Watanabe, "End-to-end speech recognition: A survey," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 325–351, 2024.

[2] T. Xie, Y. Rong, P. Zhang, W. Wang, and L. Liu, "Towards controllable speech synthesis in the era of large language models: A systematic survey," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2025, pp. 764–791.

[3] N. Sethiya and C. K. Maurya, "End-to-end speech-to-text translation: A survey," *Computer Speech Language*, vol. 90, p. 101751, 2025.

[4] G. Wiher, C. Meister, and R. Cotterell, "On decoding strategies for neural text generators," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 997–1012, 09 2022.

[5] B. T. Lowerre, "The harpy speech recognition system." *Ph.D. thesis, Carnegie-Mellon University, U.S.A*, 1976.

[6] A. Holtzman, J. Buys, M. Forbes, A. Bosselut, D. Golub, and Y. Choi, "Learning to write with cooperative discriminators," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2018, pp. 1638–1649.

[7] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," in *International Conference on Learning Representations*, 2020.

[8] J. Hewitt, C. Manning, and P. Liang, "Truncation sampling as language model desmoothing," in *Findings of the Association for Computational Linguistics: EMNLP 2022*. Association for Computational Linguistics, 2022, pp. 3414–3427.

[9] A. DeLucia, A. Mueller, X. L. Li, and J. Sedoc, "Decoding methods for neural narrative generation," in *Proceedings of the First Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, A. Bosselut, E. Durmus, V. P. Gangal, S. Gehrmann, Y. Jernite, L. Perez-Beltrachini, S. Shaikh, and W. Xu, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 166–185.

[10] Y. Xiao, L. Wu, J. Guo, J. Li, M. Zhang, T. Qin, and T.-Y. Liu, "A survey on non-autoregressive generation for neural machine translation and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 10, pp. 11 407–11 427, 2023.

[11] G. Bachmann and V. Nagarajan, "The pitfalls of next-token prediction," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 2024, pp. 2296–2318.

[12] C. Wang and R. Sennrich, "On exposure bias, hallucination and domain shift in neural machine translation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020, pp. 3544–3552.

[13] J. Stork, A. E. Eiben, and T. Bartz-Beielstein, "A new taxonomy of global optimization algorithms," *Natural Computing*, vol. 21, pp. 219–242, 2022.

[14] K. Rajwar, K. Deep, and S. Das, "An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges," *Artificial Intelligence Review*, vol. 56, pp. 13 187–13 257, 2023.

[15] S. Tang, P. Feng, S. Yu, Y. Dong, and S. J. Qin, "A hierarchical taxonomy for deep state space models," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[16] N. Dziri, X. Lu, M. Sclar, X. L. Li, L. Jiang, B. Y. Lin, P. West, C. Bhagavatula, R. Le Bras, J. D. Hwang, S. Sanyal, S. Welleck, X. Ren, A. Ettinger, Z. Harchaoui, and Y. Choi, "Faith and fate: limits of transformers on compositionality," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23. Curran Associates Inc., 2023.

[17] M. Stern, N. Shazeer, and J. Uszkoreit, "Blockwise parallel decoding for deep autoregressive models," in *Neural Information Processing Systems*, 2018.

[18] M. Ghazvininejad, O. Levy, Y. Liu, and L. Zettlemoyer, "Mask-predict: Parallel decoding of conditional masked language models," in *Conference on Empirical Methods in Natural Language Processing*, 2019.

[19] X. Sun, T. Ge, F. Wei, and H. Wang, "Instantaneous grammatical error correction with shallow aggressive decoding," 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)*. Association for Computational Linguistics, 2021, pp. 5937–5947.

[20] Y. Leviathan, M. Kalman, and Y. Matias, "Fast inference from transformers via speculative decoding," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202, 2023, pp. 19 274–19 286.

[21] A. Santilli, S. Severino, E. Postolache, V. Maiorca, M. Mancusi, R. Marin, and E. Rodola, "Accelerating transformer inference for translation via parallel decoding," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2023, pp. 12 336–12 355.

[22] A. Vijayakumar, M. Cogswell, R. Selvaraju, Q. Sun, S. Lee, D. Crandall, and D. Batra, "Diverse beam search for improved description of complex scenes," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[23] M. Barley, N. de Kriek, S. Franco, A. Garcia-Olaya, T. Hartill, C. Triggs, H. Zwart, V. Alcázar, and P. Riddle, "A problem with the current methodology for comparing search algorithms and a proposed solution," *Proceedings of the International Symposium on Combinatorial Search*, vol. 18, no. 1, pp. 29–37, 2025.

[24] P. Beraldi and A. Ruszczyński, "Beam search heuristic to solve stochastic integer problems under probabilistic constraints," *European Journal of Operational Research*, vol. 167, no. 1, pp. 35–47, 2005.

[25] C. Meister, T. Vieira, and R. Cotterell, "Best-first beam search," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 795–809, 2020.

[26] C. Bryant, Z. Yuan, M. R. Qorib, H. Cao, H. T. Ng, and T. Briscoe, "Grammatical error correction: A survey of the state of the art," *Computational Linguistics*, vol. 49, no. 3, pp. 643–701, 2023.

[27] R. Leblond, J.-B. Alayrac, L. Sifre, M. Pislar, L. Jean-Baptiste, I. Antonoglou, K. Simonyan, and O. Vinyals, "Machine translation decoding beyond beam search," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021, pp. 8410–8434.

[28] M. Freitag and Y. Al-Onaizan, "Beam search strategies for neural machine translation," in *Proceedings of the First Workshop on Neural Machine Translation*. Association for Computational Linguistics, 2017, pp. 56–60.

[29] B. Eikema and W. Aziz, "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*. Barcelona, Spain (Online): International Committee on Computational Linguistics, 2020, pp. 4506–4520.

[30] N. Kaur and P. Singh, "Conventional and contemporary approaches used in text to speech synthesis: a review," *Artificial Intelligence Review*, vol. 56, pp. 5837–5880, 2023.

[31] P. Peng, P.-Y. Huang, S.-W. Li, A. Mohamed, and D. Harwath, "VoiceCraft: Zero-shot speech editing and text-to-speech in the wild," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 2024, pp. 12 442–12 462.

[32] C. Shi, H. Yang, D. Cai, Z. Zhang, Y. Wang, Y. Yang, and W. Lam, "A thorough examination of decoding methods in the era of LLMs," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2024, pp. 8601–8629.

[33] S. Welleck, A. Bertsch, M. Finlayson, H. Schoelkopf, A. Xie, G. Neubig, I. Kulikov, and Z. Harchaoui, "From decoding to meta-generation: Inference-time algorithms for large language models," *Transactions on Machine Learning Research*, 2024, survey Certification.

[34] C. Meister, T. Pimentel, G. Wiher, and R. Cotterell, "Locally typical sampling," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 102–121, 2023.

[35] K. Arora, L. El Asri, H. Bahuleyan, and J. Cheung, "Why exposure bias matters: An imitation learning perspective of error accumulation in language generation," in *Findings of the Association for Computational Linguistics: ACL*. Association for Computational Linguistics, 2022, pp. 700–710.

[36] Z. Tu, G. Zhang, Y. Lu, A. Adigwe, S. King, and Y. Guo, "Enabling beam search for language model-based text-to-speech synthesis," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[37] M. Josifoski, M. Peyrard, F. Rajič, J. Wei, D. Paul, V. Hartmann, B. Patra, V. Chaudhary, E. Kiciman, and B. Faltings, "Language model decoding as likelihood–utility alignment," in *Findings of the Association for Computational Linguistics: EACL 2023*. Association for Computational Linguistics, 2023, pp. 1455–1470.

[38] D. Ippolito, R. Kriz, J. Sedoc, M. Kustikova, and C. Callison-Burch, "Comparison of diverse decoding methods from conditional language models," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019, pp. 3752–3762.

[39] L. Du, H. Lee, J. Eisner, and R. Cotterell, "When is a language process a language model?" in *Findings of the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics, 2024, pp. 11 083–11 094.

[40] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive Science*, vol. 9, no. 1, pp. 147–169, 1985.

[41] J. Gu, J. Bradbury, C. Xiong, V. O. Li, and R. Socher, "Non-autoregressive neural machine translation," in *International Conference on Learning Representations*, 2018.

[42] N. Chen, S. Watanabe, J. Villalba, P. Żelasko, and N. Dehak, "Non-autoregressive transformer for speech recognition," *IEEE Signal Processing Letters*, vol. 28, pp. 121–125, 2021.

[43] K. Peng, W. Ping, Z. Song, and K. Zhao, "Non-autoregressive neural text-to-speech," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119, 13–18 Jul 2020, pp. 7586–7598.

[44] C. Xu, X. Liu, X. Liu, Q. Sun, Y. Zhang, M. Yang, Q. Dong, T. Ko, M. Wang, T. Xiao, A. Ma, and J. Zhu, "CTC-based non-autoregressive speech translation," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 13 321–13 339.

[45] Y. Yang, S. Liu, J. Li, Y. Hu, H. Wu, H. Wang, J. Yu, L. Meng, H. Sun, Y. Liu, Y. Lu, K. Yu, and X. Chen, "Pseudo-autoregressive neural codec language models for efficient zero-shot text-to-speech synthesis," in *Proceedings of the 33rd ACM International Conference on Multimedia*, ser. MM '25, 2025, p. 9316–9325.