Revisiting the Markov Property for Machine Translation

Cunxiao Du
Singapore Management University
80 Stamford Rd, Singapore 178902
cnsdunm@gmail.com

Hao Zhou
Insititute for AI Industry Research (AIR)
Tsinghua University
haozhou0806@gmail.com

Zhaopeng Tu
Tencent AI Lab
tuzhaopeng@gmail.com

Jing Jiang
Singapore Management University
jingjiang@smu.edu.sg

Abstract

In this paper, we re-examine the Markov property in the context of neural machine translation. We design a Markov Autoregressive Transformer (MAT) and undertake a comprehensive assessment of its performance across four WMT benchmarks. Our findings indicate that MAT with an order larger than 4 can generate translations with quality on par with that of conventional autoregressive transformers. In addition, counter-intuitively, we also find that the advantages of utilizing a higher-order MAT do not specifically contribute to the translation of longer sentences.

1 Introduction

Markov models are classic probabilistic graphical models based on the Markov property. The Markov property reduces computation complexity and thus makes Markov models highly appealing. Markov models have been extensively used in many NLP tasks such as part-of-speech tagging (Ma and Hovy, 2016; Shao et al., 2017) and dependency parsing (Zhang et al., 2020a,b). Statistical machine translation (SMT) has also employed Markov models, e.g., Lavergne et al. (2011).

However, with the rise of deep learning in machine translation, autoregressive models (Sutskever et al., 2014; Bahdanau et al.; Gehring et al., 2017), particularly autoregressive transformers (Vaswani et al., 2017), have gradually become mainstream. During decoding, autoregressive models rely on all the previous tokens. As a result, they can model long-range dependencies and are thus considered to have superior abilities to express token dependency than Markov models. The performance of recent advanced Markov models (Wang et al., 2018; Sun et al., 2019; Deng and Rush, 2020) in MT are also significantly lower than those of the autoregressive model.

The Markov property dictates that, during decoding, each token can only observe the previous k tokens. This characteristic is a considerable drawback for generation tasks that require long contexts, such as story generation. However, we believe that in translation, since the source sentence is fully visible, introducing the Markov property on the decoder side might not greatly affect translation performance.

To investigate this hypothesis, we introduce the Markov Autoregressive Transformer (MAT) and evaluate its performance on translation. MAT possesses two main features: 1) minimal modifications to autoregressive transformers, and 2) support for high-order Markov models. Specifically, the key idea of the kth-order Markov property is that the next output token by the model is only dependent on the previous k tokens. In this paper, we point out that this objective can be achieved with a simple modification to the causal mask in the decoder part. In contrast to previous Markov models, this simple modification ensures that our MAT has only marginal alterations compared to the autoregressive transformer. This allows us to effectively isolate and examine the effects of the Markov property in a manner akin to a controlled variable experiment. In addition to the aforementioned benefit, this straightforward modification also enables us to train MAT in parallel, like the vanilla transformer.

We evaluate MAT on several WMT benchmarks and make the following observations:

- The first-order Markov property significantly impairs model performance. For instance, on the WMT14 EN-DE task, there is a decline of approximately 3.4 BLEU points (§4.3).
- For the kth-order Markov property, as k increases, the performance of the model becomes increasingly comparable to that of an autoregressive model (e.g., when k=5) (§4.4).
- The benefits of a larger k are not necessarily specific to longer sentences (§4.4).

Findings of the Association for Computational Linguistics: EACL 2024, pages 582–588
March 17-22, 2024

In addition to the aforementioned findings, we also discover that MAT also enjoys the following advantages: 1) Linear complexity of attention. To generate a sentence with the length of n, the complexity of attention is only $O(kn)$ compared with $O(n^2)$ in vanilla autoregressive transformers. For a sample length of 25, the computation for decoder self-attention is reduced by approximately threefold. 2) Key-Value cache free inference. Because MAT only attends to the embeddings of the previous k tokens, it does not require caching any keys and values of the previous tokens during inference. This reduces the memory bandwidth required by the cache at the decoding stage. By limiting the dependence on a fixed number of preceding tokens, the Markov property can potentially simplify the translation model, thereby reducing complexity and computational requirements. This might lead to a balance where adequate performance can be achieved more efficiently.

2 Preliminaries

Task Definition. Machine translation aims to translate an input sentence X in a source language into an output sentence Y in a target language. The detailed definition is provided in the Appendix A.1.

Markov Property The Markov property (Markov, 1954) is a stochastic property that states that the probability of a future state depends only on the current state and not on the sequence of states that preceded it. For MT, mathematically, given a source sentence X and a sequence of previously generated target tokens $y_1, y_2, \ldots, y_{n-1}$, and the k-order Markov properties allow for longer-distance dependencies, as described by the following:

$$P(y_n|X, y_1, y_2, \cdot, y_{n-1}) = P(y_n|X, y_{n-k}, \cdot, y_{n-1}).$$

3 Markov Autoregressive Transformer (MAT)

3.1 Overview

Our MAT consists of two parts: 1) an Encoder, and 2) a Markov Decoder. We keep the Encoder the same as in the vanilla transformer. For the Markov Decoder, the only difference lies in the attention mechanism, which is elaborated as follows.

3.2 Markov Attention Mechanism

To keep the Markov property in the decoder, we use a mechanism called transparent Markov attention.

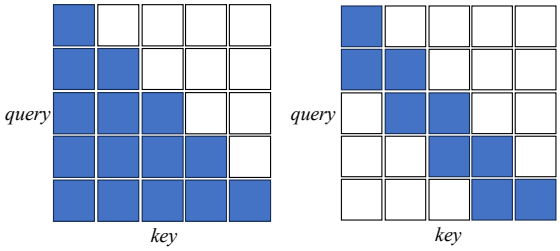

Figure 1: The illustration of the original casual attention mask (left) and $second$-Order Attention Mask (right).

To be specific, Markov attention has two characteristics:

- k-Order Attention Mask. To prevent the current token from accessing the information beyond what the Markov property allows, we may use a lower triangular matrix to only keep the attention weights within the window size k. However, it is worth noting that using this kind of mask alone does not guarantee that information will not leak (Chelba et al., 2020). This is because as the number of layers L increases, the current token will encompass information from the former tokens than k, violating the Markov property of only observing the previous k tokens. A clearer example is provided in the Appendix A.2.

- Transparent Attention. Inspired by the two-stream attention (Yang et al., 2019), we propose a simple method called Transparent Attention to fix the information leakage in the k-Order Attention Mask. With such attention, the keys and values of previous tokens are not updated, i.e., they are always set to be the static word embeddings of the corresponding tokens.

4 Experiments

4.1 Data

We conduct experiments on major benchmark MT datasets at different scales that are widely used in previous studies: WMT14 English⇔German (En⇔De, 4.5M pairs), and large-scale WMT17 English⇔Chinese (En⇔Zh, 20M pairs). For fair comparison, we report BLEU scores (Papineni et al., 2002) on En⇔De and Zh⇒En, and Sacre BLEU scores (Post, 2018) on En⇒Zh. The other details can be found in Appendix A.3.

Model	WMT14		WMT17	
	En-De	De-En	En-Zh	Zh-En
Autoregressive Transformer (Vaswani et al., 2017)	27.8	31.3	34.4	24.0
Autoregressive Transparent Transformer	27.3	31.2	33.9	23.3
Markov Models				
Bigram CRF (Sun et al., 2019)	23.4	27.2	-	-
Non-autoregressive Markov Transformer (Deng and Rush, 2020)	24.4	29.4	-	-
Autoregressive Markov Transformer (Ours, k=5)	27.5	31.0	33.9	23.3

Table 1: BLEU scores on two benchmarks.

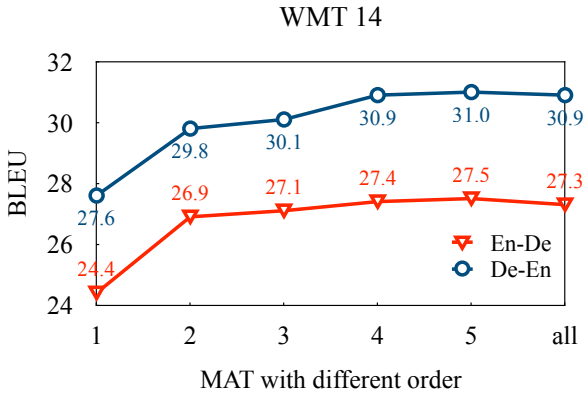

Figure 2: In the WMT14 EN-DE dataset, experimental results for MAT with varying values of k. It indicates that as k increases, the BLEU score for MAT exhibits an upward trend. However, the improvements plateau when k exceeds 3.

4.2 Baselines

To investigate the impact of the Markov property on model performance, we consider the following models as our baselines: 1) Standard Autoregressive Transformer, which attends to *all* previous tokens, 2) Transparent Attention Transformer, i.e., the transformer with transparent attention, which attends to the contextualized embeddings of the previous k tokens, and 3) two other Markov Translation Models as reference points. The details of these two models can be found at Appendix A.4.

4.3 Results

Comparison between our MAT model and the baselines is shown in Table 1. From the table, we observe the following:

- *Transparent Attention slightly decreases the BLEU score of the model.* Comparing Autoregressive Transformer and Autoregressive Transparent Transformer, it is evident that employing transparent attention leads to an average performance drop of approximately 0.3 on the WMT14 En⇔De benchmark and about 0.6 on the WMT17 En⇔Zh benchmark, which is not substantial.

- *MAT demonstrates significant improvement over previous Markov models.* Compared to previous Markov models for MT, i.e., Bigram CRF and Non-autoregressive Markov Transformer, we observe that on the WMT14 En⇔De dataset, MAT, with the same model size, achieves an improvement of 2-3 BLEU points. Notably, the order choice of MAT is 5, consistent with the Non-autoregressive Markov Transformer. This, in fact, suggests that the Markov property is not the primary reason for the relatively low performance of earlier Markov models. For the Bigram CRF model, we postulate that one primary limitation is its sole reliance on first-order Markov properties. Furthermore, modeling the relationship between tokens (i.e., the transition matrix) using a low-rank matrix might also contribute to its performance degradation. Regarding the Non-autoregressive (Gu et al., 2018; Du et al., 2021) Markov Transformer, we hypothesize that the main reason for its performance decline might be the pruning during inference through a lower-order Markov model, resulting in the absence of suitable candidates within the candidate set.

- *MAT achieves performance comparable to the standard Autoregressive Transformer, albeit slightly worse.* We observe that the performance of MAT slightly decreases compared to the standard Autoregressive Transformer. However, compared with the transparent autoregressive Transformer, MAT's performance remains almost the same. This suggests that

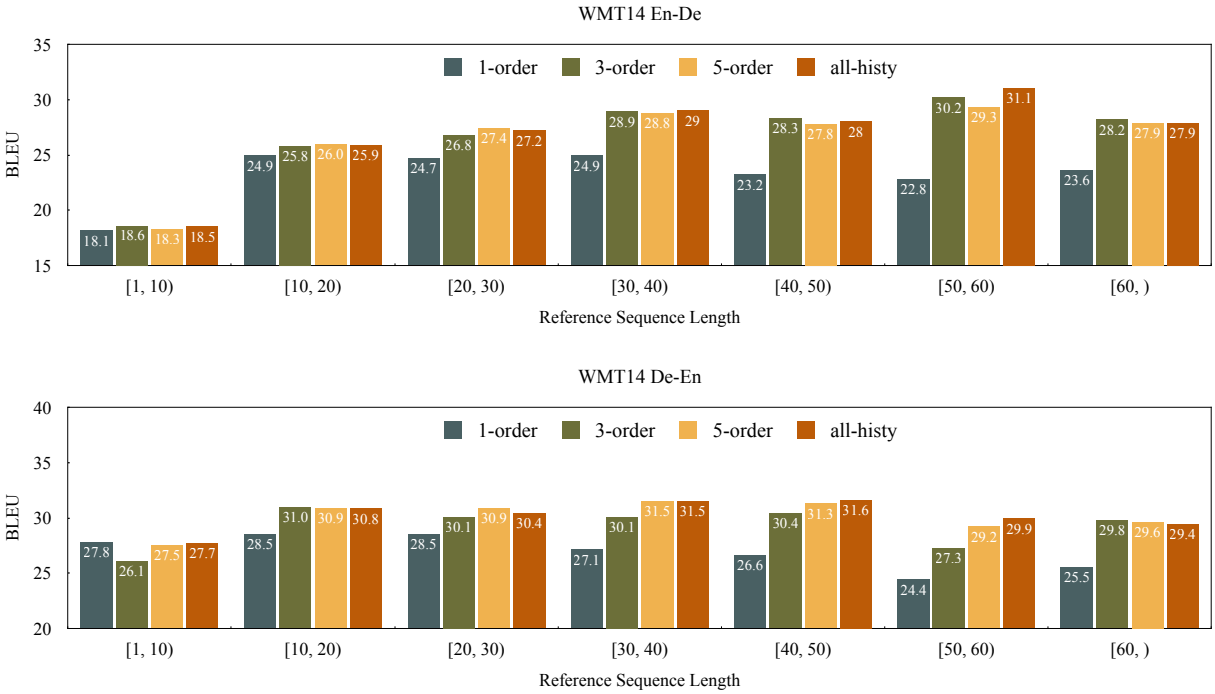

Figure 3: Performance of the generated translations with respect to the lengths of the reference sentences.

within the current MAT architecture, employing the 5-order Markov property does not compromise its translation capabilities.

4.4 Analysis

MAT with Different Order Recall that in our MAT model with kth-order Markov property, k indicates MAT's ability to process previous tokens. An intuitive hypothesis is that a larger k might yield better performance because it captures a longer context. However, we find that empirical results do not fully align with it. In Figure 2, we plot the performance with respect to different values of k. We find the following three observations: 1) At k=1, the model's performance sees a significant drop compared to a non-Markov model. One potential reason is that the complexity of the translation data far exceeds what a first-order Markov model can encapsulate, and another reason is the self-attention in the transformer decoder is no longer useful. Therefore, the decline may also be related to the architecture of the transformer. 2) When k is in the range of 2-4, increasing k provides noticeable gains. This phenomenon is evident across datasets from both directions. 3) For k values greater than 4, further increasing k does not result in significant performance improvements.

MAT for References of Different Lengths We further examine the impact of different reference lengths on MAT's performance in Figure 3.

For k=1, there is a noticeable degradation in performance across all sentence lengths. This observation is consistent with previous experiments.

Interestingly, the advantages of a higher-order MAT do not always become more pronounced in longer sentences. For instance, in the WMT14 en-de results, the 3rd-order MAT consistently outperforms the 5th-order MAT for sample buckets with sentence lengths over 40. This is counter-intuitive because as a sentence gets longer, a higher-order Markov model, with its ability to access a broader previous context, supposedly would be able to utilize more information and give better results.

This unexpected phenomenon might be attributed to particular linguistic characteristics of the target language. This theory gains traction when looking at the WMT14 de-en results, where the 3rd-order MAT is only better than the 5th-order MAT in buckets with sentence lengths beyond 60.

5 Conclusions

In this paper, we re-examine the Markov property in machine translation. We design an experimental Markov model based on the transformer architecture. We verify that higher-order Markov properties have a very slight impact on the model's translation quality. Moreover, we find that longer sentences do not necessarily require higher-order Markov models. In the future, we aim to design faster and more lightweight models to leverage the advantages of

the Markov property. And also extend this idea to large language model and other tasks needs real-time decoding like rumor detection (Zhang and Gao, 2023) and infodemic surveillance (Zhang and Gao, 2024).

6 Limitations

In this article, we primarily explore the impact of the Markov property on model translation quality. We acknowledge that there are still several limitations of our study: 1) Compared to other Markov models, e.g., bigram CRF, our model cannot generate translations in parallel (i.e., in a non-autoregressive manner). Although our model can achieve acceleration compared to the standard autoregressive transformer, we have not fully explored the potential of Markov models in parallel generation. 2) Our current experiments are based on the transformer, neglecting other architectures, such as CNNs (Wu et al., 2019) or advanced RNNs (Sun et al., 2023). Markov models might perform better on RNN translation models. 3) Regarding the scaling laws (Ghorbani et al., 2021) for Markov models, due to our limited GPU resources, we are unable to further explore Markov models of different sizes. If more resources become available in the future, it might be meaningful to investigate the performance of scaling laws within Markov models.

Acknowledgement

We thank the anonymous reviewers for their helpful comments during the review of this paper. The first author wants to give special thanks to Songlin Yang from MIT, because she encouraged him to transform the idea into a paper. This work is partially supported by the Natural Science Foundation of China (62376133).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Ciprian Chelba, Mia Chen, Ankur Bapna, and Noam Shazeer. 2020. Faster transformer decoding: N-gram masked self-attention.

Xiang Deng, Ahmed Hassan Awadallah, Christopher Meek, Oleksandr Polozov, Huan Sun, and Matthew Richardson. 2021. Structure-grounded pretraining for text-to-SQL. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1337–1350, Online. Association for Computational Linguistics.

Yuntian Deng and Alexander M. Rush. 2020. Cascaded text generation with markov transformers. In *NeurIPS*.

Cunxiao Du, Zhaopeng Tu, and Jing Jiang. 2021. Order-agnostic cross entropy for non-autoregressive machine translation. In *Proc. of ICML*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*.

Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. 2021. Scaling laws for neural machine translation. In *ICLR*.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *ICLR*.

Thomas Lavergne, A. Allauzen, Josep Maria Crego, and François Yvon. 2011. From n-gram-based to crf-based translation models. In *WMT@EMNLP*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *ACL*.

A. A. Markov. 1954. *Theory of Algorithms*. Academy of Sciences of the USSR.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *WMT*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Yan Shao, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. 2017. Character-based joint segmentation and POS tagging for Chinese using bidirectional RNN-CRF. In *IJCNLP*.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. 2023. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*.

Zhiqing Sun, Zhuohan Li, Haoqing Wang, Zi Lin, Di He, and Zhi-Hong Deng. 2019. Fast structured decoding for sequence models. In *NeurIPS*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

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 *NeurIPS*.

Weiyue Wang, Derui Zhu, Tamer Alkhouli, Zixuan Gan, and Hermann Ney. 2018. Neural hidden Markov model for machine translation. In *ACL*.

Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *ICLR*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Xuan Zhang and Wei Gao. 2023. Towards llm-based fact verification on news claims with a hierarchical step-by-step prompting method. *arXiv preprint arXiv:2310.00305*.

Xuan Zhang and Wei Gao. 2024. Predicting viral rumors and vulnerable users with graph-based neural multi-task learning for infodemic surveillance. *Information Processing & Management*.

Yu Zhang, Zhenghua Li, and Min Zhang. 2020a. Efficient second-order treecrf for neural dependency parsing. In *ACL*.

Yu Zhang, Houquan Zhou, and Zhenghua Li. 2020b. Fast and accurate neural crf constituency parsing. *ArXiv*, abs/2008.03736.

A Appendix

A.1 Task Definition

Given a sentence X in a source language, machine translation aims to produce a sentence Y in a target language that has the same semantic meaning as X. Formally, an MT system attempts to output the best translation Y^*:

$$Y^* = \text{argmax}_Y P_\theta(Y|X),$$

where $P_\theta(Y|X)$ is the probability of translation Y given source X.

Autoregressive neural machine translation (NMT) decomposes $P(Y|X)$ by predicting one token (e.g., a subword) of the target sequence at one time, conditioned on the entire source sequence and all previously predicted tokens in the target sequence.

Formally, given a source sequence $X = [x_1, x_2, ..., x_m]$ and a target sequence $Y = [y_1, y_2, ..., y_n]$, the model is trained to maximize the conditional probability:

$$P(Y|X) = \prod_{i=1}^{n} P(y_i|X, y_1, ..., y_{i-1}).$$

A.2 Information Leakage in k-Order Attention Mask

A second-order Markov property requires that only the two previous tokens, i.e., **all** & **you**, be visible when predicting **need**. However, as the number of layers progresses, tokens like **Attention** are visible to **need**, breaking the Markov property.

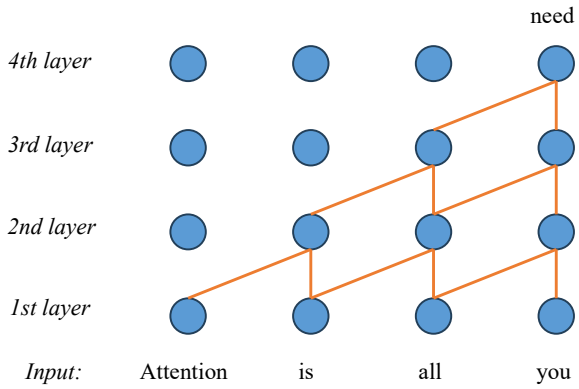

Figure 4: A second-order attention mask, where the orange lines indicate attention. The input token sequence is [**Attention**, **is**, **all**, **you**], and the token to be predicted is **need**.

A.3 Training Details

Loss Function The conventional Markov models require global normalization to tackle the label bias problem. However, here we cannot perform such normalization because the transition matrix is modeled by a parametric deep neural network which needs traversal of all the possible previous k tokens combination. After considering the trade-off, we decide to use local normalization as what the vanilla autoregressive transformer does Thus the loss function is as follows:

$$\begin{aligned}\mathcal{L} &= -\log P(y_1, y_2, \ldots, y_n|X) \\ &= -\sum_{i=1}^{n} \log P(y_i|X, y_{i-k}, \cdots, y_{i-1}). \quad (1)\end{aligned}$$

Here k is the order of the Markov decoder.

Data Processing We learned a BPE model with 32K merge operations for the dataset. We preprocessed the datasets with a joint BPE (Sennrich et al., 2016) with 32K merge operations for En⇔De, and 32K bpe for En⇔Zh.

Hyperparameters For our model and the baselines in our paper, we adopt the Transformer BASE

architecture, consisting of 6 encoder layers, 6 decoder layers, 8 attention heads, 512 model dimensions, and 2048 hidden dimensions. We use the AdamW optimizer for optimization. To prevent over-fitting, we adopt dropout equals to 0.2. All experiments are conducted on 8 NVIDIA 3090 GPU cards.

A.4 Previous Markov Models

Bigram CRF (Sun et al., 2019). The Bigram CRF employs the Linear-CRF as its decoder while leveraging the standard Transformer Encoder as the encoder part. More specifically, Bigram CRF utilizes a non-autoregressive Transformer decoder to model $P(y_i|x, pos_i)$. Subsequently, it deploys a low-rank matrix $M \in |V|^2$ to represent the transition probabilities between adjacent tokens, thereby achieving first-order Markov property.

Non-Autoregressive Markov Transformer (Deng et al., 2021). This paper utilizes the idea of cascade decoding, beginning with a non-autoregressive model (i.e., zero-order Markov model), and progressively incorporates higher-order Markov dependencies. To accelerate the generation process, it prunes the candidates of the lower-order Markov and also adopts parallel decoding at different positions.